Knowledge Graph Representation Learning as Groupoid: Unifying TransE, RotatE, QuatE, ComplEx

Han Yang*
Peking University
Beijing, China
hyang001@pku.edu.cn

Junfei Liu
Peking University
Beijing, China
liujunfei@pku.edu.cn

ABSTRACT

Knowledge graph (KG) representation learning which aims to encode entities and relations into low-dimensional spaces, has been widely used in KG completion and link prediction. Although existing KG representation learning models have shown promising performance, the theoretical mechanism behind existing models is much less well-understood. It is challenging to accurately portray the internal connections between models and build a competitive model systematically. To overcome this problem, a unified KG representation learning framework, called GrpKG, is proposed in this paper to model the KG representation learning from a generic groupoid perspective. We discover that many existing models are essentially the same in the sense of groupoid isomorphism and further provide transformation methods between different models. Moreover, we explore the applications of GrpKG in the model classification as well as other processes. The experiments on several benchmark data sets validate the effectiveness and superiority of our framework by comparing two proposed models (GrpQ8 and GrpM2) with the state-of-the-art models.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**.

KEYWORDS

Groupoid, Knowledge graph, Representation learning

ACM Reference Format:
Han Yang and Junfei Liu. 2021. Knowledge Graph Representation Learning as Groupoid: Unifying TransE, RotatE, QuatE, ComplEx. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482442

1 INTRODUCTION

Knowledge graphs (KGs), e.g., Wikidata [18], Freebase [2], and DBpedia [1], contain relational facts in the form of $\text{head} \xrightarrow{\text{relation}} \text{tail}$, which have been widely used in various kinds of tasks, such as question answering [8] and recommendation system [7]. KG representation learning [19] aims to encode entities and relations into a low-dimensional, continuous vector space. The learned dense vector representations, a.k.a. embeddings, of entities and relations that mathematically support various machine learning models to perform KG completion and link prediction.

Formally, KG representation learning is usually formalized into three steps: The first is doing a relation based mapping from the head entity to its character, i.e., $\text{h} \xrightarrow{mapping(r)} \text{c}$. The second is to calculate the distance d between character c and the tail entity t under a specific metric. The third step is to minimize the loss function $F(d)$ and get the representations of entities and relations. For example, TransE [3] believes that triples (h, r, t) should satisfy $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \approx 0$ in which the $mapping(r)$ is simply a vector summation and the distance d is the Euclidean distance. In general, the mapping could be either linear or nonlinear, and could be defined in different representation spaces such as Euclidean space, complex space, Gaussian space, Lie group, etc. The distance metric is usually one of Manhattan, Euclidean, and cosine. The loss function tends to be one of the Pointwise [13], Pairwise [13], Multi-class loss [13] and so on. Various combinations of mappings, distances, and loss functions have produced different models. Although existing KG representation learning models have shown promising performance, the theoretical mechanism behind existing models is much less well-understood. Because of the heterogeneity between models, there is no unified abstract representation to help us do deeper analysis. Moreover, the performance of KG representation learning is not only affected by the model but also by the regularization, negative sampling methods, framework implementation, and so on. This makes it challenging for us to independently analyze the multiple factors that affect the performance of the model.

To address the aforementioned issues, in this work, 1) We represent popular KG representation learning models from a generic perspective: the groupoid with a metric space on it, as illustrated in Figure 1. To be more specific, we represent the entities, relations, and characters in the groupoid $(G, *)$. The relation-based mapping and the composition of the relations are considered as the groupoid operation $*$. The definition of groupoid coincides with the intrinsic structure of relations and entities without being too strict. 2) We propose a general KG representation learning framework, GrpKG for short. Most of the popular KG representation learning models can be considered a special case of GrpKG. Our GrpKG framework can adjust itself according to the characteristics of different KG data sets. In addition, we extend GrpKG to discrete groupoids with the help of the construction of algebra groupoids. 3) We reveal the intrinsic correlations between different models by exploring the

*Corresponding author.

CIKM '21, November 1–5,2021, VirtualEvent, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482442

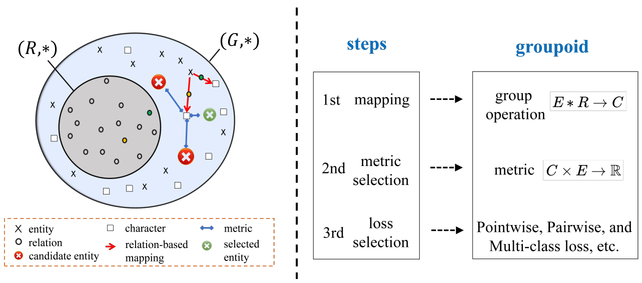


Figure 1: The overall architecutre of KG groupoid representation learning models.

isomorphism between groupoids. We discover that many models are essentially the same in the sense of groupoid isomorphism. Furthermore, we accurately portray the internal connections and provide the transformation methods between different models. The introduction of groupoids can bring many new research directions for KG representation learning. For example, we put forward a new classification standard for KG representation learning models. We classify different models according to the degree of freedom of the groupoid, and find that models with the same degree of freedom often have similar expressibility. It also shows that the groupoid can reflect some law of models, rather than just dividing models into several different categories. Ultimately, we build two models, GrpQ8 and GrpM2 for short, to verify the effectiveness of the proposed unified framework GrpKG.

To sum up, the main contributions of our work are as follows:

- We formalize the KG representation into three steps and reveal the groupoid structure hidden in the KG representation learning model. We also show that the groupoid is a powerful tool for KG representation learning.
- To the best of our knowledge, we are the first to explore the isomorphism between the KG representation models. We provide theoretical correlation analysis and transformation methods between current models.
- We propose two models, i.e., GrpMK and GrpQ8, which achieve state-of-the-art results. Extensive experiments on real-world data sets demonstrate the superiority of groupoid-guided models.

2 KG REPRESENTATION LEARNING MODELS AS GROUPOID

In this section, we present the detailed theoretical analysis of the groupoid structures hidden in popular KG representation learning models.

2.1 Preliminaries

Binary operation: a binary operation on a set G is a mapping of the elements of the Cartesian product $G \times G$ to G:

$$f : G \times G \rightarrow G. \tag{1}$$

The operation has the property of closure.

Groupoid: A groupoid $(G, *)$ is an algebraic structure on a set G with a binary operator $*$. The only restriction on the operator is closure. Associativity, commutativity, etc., are not required.

Groupoid Isomorphism: An isomorphism is a bijective morphism. Given two groupoids $(G, *)$ and $(H, \odot)$, a groupoid isomorphism from $(G, *)$ to $(H, \odot)$ is a bijective groupoid homomorphism from G to H. This means that a groupoid isomorphism is a bijective function $f : G \longrightarrow H$ such that for all u and v in G it holds that

$$f(u * v) = f(u) \odot f(v). \tag{2}$$

The two groupoids $(G, *)$ and $(H, \odot)$ are isomorphic if there exists an isomorphism from one to the other. This is written as

$$(G, *) \cong (H, \odot). \tag{3}$$

Often shorter notations can be used:

$$G \cong H. \tag{4}$$

Metric Space: A metric space is an ordered pair (G, d) where G is a set and d is a metric on G, i.e., a function $d : G \times G \longrightarrow \mathbb{R}$ such that for any $x, y, z \in G$, the following holds:

1. $d(x, y) = 0 \iff x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

2.2 Groupoid Structure with Metric

Yang[21] once represents models as groups (a special case of the groupoid). However, we demonstrate that not all models satisfy the strict requirements of the group. For example, the RASCAL [14] model does not have an inverse element. Besides, the associativity of relations is not always dependable. One example is the watercolor mixing: $r_1 = addWhite, r2 = addRed, r3 = addSkyBlue$. Practically, $(r_1 * r_2) * r_3 = addSeniorAsh$. However, $r_1 * (r_2 * r_3) = AddPurple$. In order to accommodate all circumstances, we employ a comparatively relaxed algebraic structure, i.e., groupoid.

We argue that KG representation learning models can be abstracted into groupoids $(G, *)$ with defined metrics, shown in Figure 1. The set G is made up of entities, relations, and characters of entities, i.e., $G = E \cup R \cup C$, where E is the set of entities, R is the set of relations, and the C is the set of characters. As described in the introduction, KG representation learning models are often formalized into three steps. The relation-based mapping can be considered as a groupoid operation $*$, mapping entities and relations to characters. Notably, the relations can be represented as a subgroup of $(G, *)$, and the groupoid operation between relations can be viewed as the composition of relations. Additionally, there is a metric d (not always rigorous) defined on the set G to calculate the distances between entities and characters. After calculating the distances between the characters and the entities, a loss function is often applied. By minimizing the loss function, we finally get the embedding representation of entities and relations.

The metric space defined on G enables us to estimate the distance between the characters and the entities. The widely used metrics are Euclidean distance, inner product of vectors, cosine distance, etc. In fact, inner product and cosine distance are not rigorous metrics because they don't satisfy the triangle inequality. Different metric spaces can convert to each other under certain circumstances. When the two vectors are unit vectors, Euclidean distance is equivalent to cosine distance and inner product.

2.3 Groupoids in Current Models

Although many models are not described from the perspective of the groupoid in the original paper, this does not prevent us from reproducing them in the groupoid. One example is that RotatE[15] regards the relation as an element-wise rotation from the head entity to the tail entity. The process can be considered to be a complex multiplication in $\mathbb{C}^{\times}$.

We summarize the groupoid structures of some models in Table 1. We notice that each model has at least a fundamental groupoid. For instance, TransE's fundamental groupoid is the translation group. In most cases, entities and relations have the same or similar groupoid structures. For example, one is the subgroup of another. But sometimes, there are different groupoids between the two, such as SO2E[21]. At this time, to unify the model, we need to transform the vector and matrix, denoted as $\wedge$ and $\vee$. Besides, although most models can be seen as groups (a special case of the groupoid), some models are not groups (only are groupoids). For example, RESCAL[14] does not satisfy the definition of a group. It does not have an inverse element.

2.4 Requirements for KG Groupoid Representation Learning

In this section, we discuss the necessary, optional, negative requirements for KG groupoid representation. More explicitly, given a KG, to implement a groupoid structure $(G, *)$ in representation, one must satisfy the necessary requirements and try to avoid negative requirements.

Necessary requirements: We consider a component or an attribute is necessary for KG groupoid representation if and only if:

(1) If not, the model will not work.
(2) All models have, not individual models.
(3) Satisfying all the requirements is enough to form a basic representation model.

Based on the above conditions, we believe that the following components and attributes are required by KG groupoid representation:

(1) Binary operation: to simulate relation-based mapping.
(2) Closure: to ensure that entities, relations and characters are in the same representation space.
(3) Metric space: to be used to calculate the distance between the characters and the entities.
(4) Loss function: to help the model converge better and faster. The loss function in this paper refers to an optimization function that takes the distance between relations and characters as the dependent variable.

Optional requirements: We consider a requirement to be optional if and only if:

(1) Some existing models have.
(2) Adding this requirement will not have a negative impact on the KG presentation learning model.

Based on the above conditions, we believe that the following components and attributes are optional by KG groupoid representation:

(1) Identity element: an element e in G such that, for every a in G, one has $e \cdot a = a$ and $a \cdot e = a$. For example, the identity element of TransE is $\mathbf{0}^n$.
(2) Inverse element: For each a in G, there exists an element b in G such that $a \cdot b = e$ and $b \cdot a = e$, where e is the identity element. The inverse relation can help deduce the possible inverse element, the inverse element obtained by the optimization algorithm may be close to the deduced inverse element. Take TransE as an example, if the representation of the relation *hypernym* is $\mathbf{r}_1$, we can deduce the representation of its inverse element, i.e., $\overline{\mathbf{r}_1} = -\mathbf{r}_1$. And the representation of the relation *hyponym*, denoted as $\mathbf{r}_2$, may be close to the representation of the inverse element, i.e., $\mathbf{r}_2 \approx \overline{\mathbf{r}_1}$.

One may argue that the inverse element is a necessary requirement because the relation in KG often has the inverse relation. However, it is worth noting that, given a relation r_1, the representation of its inverse relation r_2 is obtained by the optimization algorithm rather than deduced by r_1. Even if there is no inverse element, the model can work well. For example, the matrix in the RESCAL[14] model does not always have an inverse element.

Negative requirements: We consider an attribute is negative if and only if an attribute will limit the representation capabilities of the model. We believe that the following attributes are negative:

(1) Commutativity: For all a, b in G, $a \cdot b = b \cdot a$. Commutativity often exists in the composition of relations. Generally, the relations are not always commutative in the data set, i.e., $r_1 * r_2 \not\equiv r_2 * r_1$. To put it in practical terms, $r_1 = isBrotherOf$ and $r2 = isWifeOf$. Obviously, $r_1 * r_2 = isBrotherInLawOf$ and $r_2 * r_1 = isSisterInLawOf$. Therefore, if a model is commutative, it forces all relations to be commutative, which is not realistic. We just need $r_1 * r_2 = r_2 * r_1$ to be true in some cases, not always true.
(2) Associativity: For all a, b, c in G, one has $(a \cdot b) \cdot c = a \cdot (b \cdot c)$. Associativity often exists in the multi-hop relations. Just like commutativity, $(r_1 * r_2) * r_3 \not\equiv r_1 * (r_2 * r_3)$. One example is the watercolor mixing, which we have shown in Section 2.2.

Negative requirements are not immutable. When there is no non-commutative or non-associative relation in the data set, negative requirements can be turned into optional requirements.

2.5 Isomorphism between Models

In this section, we use the isomorphism between groupoids to explore the equivalences and differences between models.

TorusE and RotatE

TorusE[5] is defined on a quotient space, $\mathbb{R}^n/\sim = \{[\boldsymbol{x}] \mid \boldsymbol{x} \in \mathbb{R}^n\} = \{\{\boldsymbol{y} \in \mathbb{R}^n \mid \boldsymbol{y} \sim \boldsymbol{x}\} \mid \boldsymbol{x} \in \mathbb{R}^n\}$, where $\sim$ is an equivalence relation and $y \sim x$ if and only if $y - x \in \mathbb{Z}^n$. The groupoid operation μ is also derived from the original vector space: $\mu([\boldsymbol{x}], [\boldsymbol{y}]) = [\boldsymbol{x}] + [\boldsymbol{y}] \triangleq [\boldsymbol{x} + \boldsymbol{y}]$. From the definition, we know the fundamental groupoid of TorusE[5] is the quotient group, i.e., $\mathbb{R}/\mathbb{Z}$.

RotatE[15] maps the head and tail entities to the complex embeddings, i.e., $\mathbf{h}, \mathbf{t} \in \mathbb{C}^n$ and define the mapping induced by the relation as an element-wise rotation from the representation of head entity $\mathbf{h}$ to the representation of tail entity $\mathbf{t}$, i.e., $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$, where

Table 1: Summary of the groupoid structures in current models.

Model	F-groupoid[1]	Space[1]	BO[2]	Com[2]	Metric[3]:d	Freedom[1]	Loss Function
TransE[3]	T	$\mathbb{R}^n$	VA	Y	MD: $\lVert \mathbf{h}+\mathbf{r}-\mathbf{t} \rVert_{1/2}$	n	$\sum_{d\in D^+}\sum_{d'\in D^-}(\lambda+d'-d)_+$
DisMult[20]	$\mathbb{R}^\times$	$\mathbb{R}^n$	EP	Y	IP:$\mathbf{h}^\top \operatorname{diag}(\mathbf{r})\mathbf{t}$	n	$\sum_{d\in D^+}\sum_{d'\in D^-}(\lambda+d'-d)_+$
ComplEx[17]	$\mathbb{C}^\times$	$\mathbb{C}^n$	CP	Y	IP:Re $\left(\mathbf{h}^\top \operatorname{diag}(\mathbf{r})\bar{\mathbf{t}}\right)$	$2n$	$\sum_{d\in D}\log(1+\exp(-l(d)\cdot d))$
RotatE[15]	$U(1)\leqslant\mathbb{C}^\times$	$\mathbb{T}^n\in\mathbb{C}^n$	CP	Y	MD:$\lVert \mathbf{h}\circ\mathbf{r}-\mathbf{t} \rVert$	$n<2n$	$-\log\sigma(\gamma-d)-\sum_{i=1}^{n}\frac{1}{k}\log\sigma(d'-\gamma)$
TorusE[5]	$(\mathbb{R}/\mathbb{Z})^n$	$\{[\boldsymbol{x}] \mid \boldsymbol{x}\in\mathbb{R}^n\}$	TA	Y	CS:$\lVert g([\mathbf{h}+\mathbf{r}])-g([\mathbf{t}]) \rVert_2$	n	$\sum_{d\in D^+}\sum_{d'\in D^-}(\lambda+d'-d)_+$
RESCAL[14]	-	$\mathbb{R}^{n\times n}$	MPC	N	IP[4]:$(\mathbf{h}^\wedge\mathbf{M}_r\mathbf{t}^\wedge)^\vee$	n^2	$\frac{1}{2}\sum_{d\in D}(d-l(d))^2$
SU2E[21]	$SU(2)$	$\mathbb{C}^{2\times2\times n}$	MPC	N	MD[4]: $\left\lVert(\mathbf{M_r}\cdot\mathbf{h}^\wedge-\mathbf{t}^\wedge)^\vee\right\rVert$	$3n$	$\sum_{d\in D^+}\sum_{d'\in D^-}(\lambda+d'-d)_+$
SO3E[21]	$SO(3)$	$\mathbb{R}^{3\times3\times n}$	MP	N	MD[4]: $\left\lVert(\mathbf{M_r}\cdot\mathbf{h}^\wedge-\mathbf{t}^\wedge)^\vee\right\rVert$	3n	$\sum_{d\in D^+}\sum_{d'\in D^-}(\lambda+d'-d)_+$
QuatE[22]	$HU(1)\leqslant H$	$\mathbb{H}_u^n\in\mathbb{H}^n$	HP	N	IP:$<\mathbf{h}\bigotimes\mathbf{r},\mathbf{t}>$	3n<4n	$\sum_{d\in D}\log(1+\exp(-l(d)\cdot d))+\lambda_1\lVert Q\rVert_2^2$
Rotate3D[6]	$HU(1)$	$\mathbb{H}_u^n$	rhr^{-1}	N	MD:$\sum_{i=1}^{n}\lVert \mathbf{h}_i\odot\mathbf{r}_i-\mathbf{t}_i \rVert$	3n	$-\log\sigma(\gamma-d)-\sum_{i=1}^{n}p\log\sigma(d'-\gamma)$
GrpMK	-	$\mathbb{R}^{k\times k\times n}$	MP	N	FM: $\lVert log((h*r)^{-1}*t) \rVert$	k^2n	$\sum_{d_{hrt}\in D}\left(d_{hrt}+\log\left(\sum_{d\in D_{hrt}}\exp(-d)\right)\right)+\lambda\lVert G\rVert$
GrpQ8	H	$\mathbb{H}^n$	HP	N	AM:$\lVert \mathbf{h}*\mathbf{r}-\mathbf{t} \rVert$	$4n$	$\sum_{d_{hrt}\in D}\left(d_{hrt}+\log\left(\sum_{d\in D_{hrt}}\exp(-d)\right)\right)+\lambda\lVert G\rVert$

[1] Some models have additional restrictions on the relations, so the subgroups constituted by the relations have different fundamental groupoids (F-groups), representation spaces (Space), and degree of freedom (Freedom) with entities’. We distinguish them by $\leqslant$, $\in$, and $<$, respectively, i.e., $R\leqslant G$, $R\in G$, and $R<G$.
[2] BO is the abbreviation for the binary operation, VA refers to vector addition, EP refers to element-wise product, CP refers to complex multiplication, TA refers to the addition on torus quotient group, MPC refers to complex matrix product, MP refers to real matrix product, HP stands for Hamilton product. Com is the abbreviation for commutativity. Y means BO is commutative, N means BO is not commutative.
[3] We divide the metric of the models into several classes, MD refers to the Minkowski distance, and IP refers to the inner product. CS refers to cosine similarity, FM refers to Freobeniu metric, AM refers to Algebra metric. FM and AM are defined in Section 3.1.
[4] To unite the representation space of entities and relations, we transform the representation between vectors and matrices to satisfy the multiplication of vectors and matrices. $\wedge$ means to copy the vector in rows to form a square matrix. $\vee$ means to average the matrix and compress it into vectors or scalars.

$|\mathbf{r}_i|=1$. From the definition, we know the fundamental groupoid of relations of RotatE[15] is

$$U(1)=\left\{U \mid U\in \mathrm{GL}(1,\mathbb{C}), U^\dagger U=1\right\}=\{z\in\mathbb{C}:|z|=1\}. \tag{5}$$

Let

$$\varphi:\mathbb{R}\mapsto U(1), \quad \varphi(x)=e^{2\pi i x}, \tag{6}$$

φ is a surjective group homomorphism, and that $\ker\varphi=\mathbb{Z}$. Then, we can invoke the first isomorphism theorem and we have $\mathbb{R}/\mathbb{Z}$ is isomorphic to $U(1)$, i.e.,

$$\mathbb{R}/\mathbb{Z}\cong U(1). \tag{7}$$

In other words, TorusE $\cong$ RotatE.

TransE and DistMult

It is easy to know TransE[3] is a translation group T=$(\mathbb{R},+)$ and the fundamental groupoid of DistMult[20] is $(\mathbb{R}^\times,\times)$, the multiplicative group of all nonzero real numbers.

Let

$$\varphi_1:\mathbb{R}\mapsto\mathbb{R}_{>0}, \quad \varphi(x)=e^x, \tag{8}$$

we have that the group of all real numbers with addition, $(\mathbb{R},+)$, is isomorphic to the group of positive real numbers with multiplication $(\mathbb{R}_{>0},\times)$, i.e.,

$$(\mathbb{R}_{>0},\times)\cong(\mathbb{R},+). \tag{9}$$

Let

$$\varphi_2:\mathbb{R}^\times\mapsto\mathbb{R}_{>0}, \quad \varphi(x)=|x|, \tag{10}$$

φ_2 is surjective group homomorphism, and that $\ker\varphi=\mathbb{Z}_2$, Then we have

$$\mathbb{R}^\times/\mathbb{Z}_2\cong\mathbb{R}_{>0}. \tag{11}$$

Since (9) and (11), we have

$$\mathbb{R}^\times/\mathbb{Z}_2\cong\mathbb{R}_{>0}\cong(\mathbb{R},+). \tag{12}$$

In other words, TransE $\cong$ DistMult$/\mathbb{Z}_2$. DistMult[20] can be degenerated into TransE[3], if the representation space of DistMult[20] is restricted to positive numbers.

RotatE and ComplEx

ComplEx[17] is defined on the complex space. The score function is $\phi(r,h,t;\Theta)=\mathrm{Re}\left(\mathbf{h}^\top\operatorname{diag}(\mathbf{r})\bar{\mathbf{t}}\right)=\mathrm{Re}\left(\sum_{k=1}^{n}\mathbf{h}_k\mathbf{r}_k\bar{\mathbf{t}}_k\right)$. We divide the model into three steps for better analysis. The first step is calculating character through groupoid operation, $\mathbf{c}=\mathbf{h}\circ\mathbf{r}$ The second step is calculating the distance between the character c and the tail entity t. The third step is to minimize the loss function $L(d)$.

We focus on the first step rather than the second step or the third step. The fundamental groupoid of the ComplEx[17] is $\mathbb{C}^\times$. Clearly, The fundamental groupoid of the RotatE[15] is a subgroup of the ComplEx’s, i.e. $U(1)=\{z\in\mathbb{C}:|z|=1\}\leqslant\mathbb{C}^\times$. In other words, RotatE[15] can be extended to ComplEx[17] by removing the restriction on the modulus of relation, i.e. RotatE$_{r\in C^n}$ $\cong$ ComplEx. In fact, the relation-based mapping in ComplEx[17] can be considered as the combination of the rotation transformation and scaling transformation.

TransE and RotatE

We have proved that $\mathbb{R}/\mathbb{Z}$ is isomorphic to $U(1)$ in (9). As a result, we have $R_{[0,1)}$ is isomorphic to $U(1)$.

Let

$$\varphi:\mathbb{R}_{[-0.5,0.5)}\mapsto\mathbb{R}_{[0,1)}, \quad \varphi(x)=x+0.5, \tag{13}$$

Clearly, φ is a bijective function. Then, we have $\mathbb{R}_{[-0.5,0.5)}\cong\mathbb{R}_{[0,1)}\cong U(1)$. In other words, If $E,R,C\subset[-0.5,0.5)^n$, TransE$_{[-0.5,0.5)}$ $\cong$RotatE. Generally, the condition, $E,R,C\subset[-0.5,0.5)^n$ is not difficult to achieve. It is because that entities and relations are usually initialized in a small range, such as $[-10^{-3},10^{-3})$. Thus, almost all the representation of entities and relations are still in the range $[-0.5, 0.5)$ after limited optimization iteration.

Rotate3D and SU2E

Rotate3D[6] is defined on unit quaternions and its fundamental groupoid is $HU(1) = \{q \mid q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}, a^2 + b^2 + c^2 + d^2 = 1\}$, where a, b, c, and d are real numbers; and $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the basic quaternions. Let $\mathbf{u} = b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, The groupoid operation $*$ is:

$$q_1 q_2 = [a_1 a_2 - \mathbf{u}_1 \cdot \mathbf{u}_2, a_1 \mathbf{u}_2 + a_2 \mathbf{u}_1 + \mathbf{u}_1 \times \mathbf{u}_2]\,. \tag{14}$$

The fundamental groupoid of SU2E[21] is $SU(2) = \{U \mid U \in \mathrm{GL}(2, \mathbb{C}), U^\dagger U = 1_{2\times 2}, |U| = 1\}$.

Let

$$\varphi : a + bi + cj + dk \mapsto \begin{pmatrix} a + bi & c + di \\ -c + di & a - bi \end{pmatrix}, \tag{15}$$

Since $\left|\begin{pmatrix} a + bi & c + di \\ -c + di & a - bi \end{pmatrix}\right| = a^2 + b^2 + c^2 + d^2 = 1$, the unit quaternions correspond to elements in $SU(2)$. In fact, φ is a bijective function and preserves the groupoid operation. So we have $HU(1)$ is isomorphic to $SU(2)$:

$$HU(1) \cong SU(2). \tag{16}$$

It is worth noting that the entities of SU2E[21] are not in $SU(2)$. They are elements in $\mathbb{C}^2$. For unified representation, we need to repeat them in rows to form into 2×2 matrices. The SO3E[21] is in a similar way. Besides, the entities of Rotate3D are not in $HU(1)$. We need to consider them in H_u, whose elements correspond to the points in R^3 one by one. All in all, Rotate3D[6] and SU2E[21] can be considered as element-wise rotations from the head entity h to the tail entity t in 3D space.

SU2E and SO3E

The fundamental groupoid of SO3E[21] is

$SO(3) = \{Q|Q \in \mathrm{GL}(3, \mathbb{R}), Q^{\mathrm{T}} Q = 1_{3\times 3}, |Q\ | = 1\}$. If U is an element of $SU(2)$ and $x \in su(2) \cong \mathbb{R}^3$, we have that

$$\mathrm{Tr}\left(UxU^{-1}\right) = 0, \quad \left(UxU^{-1}\right)^\dagger = \left(U^{-1}\right)^\dagger xU^\dagger = UxU^{-1}, \tag{17}$$

and we conclude that U induces a special orthogonal transformation $\mathbb{R}^3 \to \mathbb{R}^3$.

Let

$$\varphi : x \mapsto UxU^{-1}, \tag{18}$$

φ maps the element in $SU(2)$ to the element in $SO(3)$.

Since $U_1\left(U_2 x U_2^{-1}\right) U_1^{-1} = (U_1 U_2)\, x\, (U_1 U_2)^{-1}$, φ is a group homomorphism. Actually, φ is surjective and the kernel is $\mathbb{Z}_2$. Then, we can invoke the first isomorphism theorem and we have shown:

$$SU(2))/\mathbb{Z}_2 \cong SO(3). \tag{19}$$

In other words, SU2E/$\mathbb{Z}_2$ ≅ SU2E. SU2E[21] can be degenerated into SO3E[21].

QuatE and Rotate3D

The fundamental groupoid of QuatE[22] is the same as Rotate3D[6], i.e., $HU(1)$. But they have different relation-based mappings. The mapping of QuatE[22] is Hamilton product and the mapping of Rotate3D[6] is:

$$q * v = qvq^{-1} = \left[0, \mathbf{v}_\parallel + \cos 2\theta \mathbf{v}_\perp + \sin 2\theta \mathbf{w}\right], \tag{20}$$

where $\mathbf{v}_\parallel$ is the component of v along $\mathbf{r}$, $\mathbf{v}_\perp$ is the component of v orthogonal to $\mathbf{r}$, v is pure quaternions, i.e., $v = [0, \mathbf{v}]$.

Generally, entities and relations in different representation spaces will lead to the divergence of entities and relations. However, the divergence may decrease as the density of the data set becomes larger. Because if the data set is dense enough, each entity can be got from a common source entity after several relation-based mapping.

3 WHAT CAN WE BENEFIT FROM GROUPOID?

3.1 A General KG Representation Learning Method

In Section 2.3, we have revealed that many models can be represented by the groupoid structure. In this section, we will put forward a general method of learning the KG representation as the groupoid, named GrpKG.

The basic steps: As mentioned in the previous section, to create a KG groupoid representation, we only need to meet the four necessary conditions. We can follow the steps to construct a KG groupoid representation model:

(1) Create a fundamental groupoid or select a common fundamental groupoid (F, Δ). For example, the translation group T, the unitary group $U(1)$, etc. Once the groupoid is selected, the representation space and the binary operation will be determined accordingly.
(2) Conduct the direct product of the fundamental groupoid with itself. Given groups (F, Δ), we can get the final groupoid $(G, *)$ by conducting the direct product of F.

$$G = \underbrace{F \times F \times F \times \cdots \times F}_{n}, \tag{21}$$

and the operation binary $*$ is defined component-wise:

$$(x_1, x_2, \cdots, x_n) * (y_1, y_2, \cdots, y_n) = (x_1 \Delta y_1, \cdots, x_n \Delta y_n), \tag{22}$$

where $x, y \in G$.
(3) Create a metric space or select a common metric space, which is not always strict here. For example, the inner product is not strictly a metric space.
(4) Determine the loss function. We need to pay attention to the mutual restriction between the loss function and the groupoid.

Challenges and optimization: We can create a KG groupoid representation according to the above four steps. However, to make the model more universal, there are still the following challenges:

(1) We argue that the groupoid cannot be chosen arbitrarily. The general method must be able to change according to the characteristics of the data sets. For example, whether the data set contains non-exchangeable relations, whether the data set contains commutative data, and the density of the data set.
(2) Practically, continuous infinite groups are required for the need of optimization. However, the number of common infinite groupoids is limited. A large number of discrete groups and finite groups cannot be utilized.
(3) The properties of common infinite groupoids are not always to meet our requirements. For example, All the groups (the

most common groupoid) are associative. If there are relations that do not satisfy the associativity in the data set, we are required to create the new groupoid.

(4) The groupoid itself does not define a metric, and current models usually use Euclidean distance or cosine distance. But if the model is not in Euclidean space, it cannot be carried out.

To solve these challenges, we first design a method of creating a groupoid that is able to make corresponding adjustments according to the characteristics of the data set and then proposed more universal metrics for KG groupoid models.

Learning KG representation from small discrete groupoids: We utilize small discrete groups to construct the algebra groupoid. Firstly, we define the multiplication and the addition on the discrete group. And then, we define the norm of numbers and get the algebra groupoid. Finally, we use the algebra groupoid to create the KG groupoid representation learning model. Take Z_3 as example, the number Z_3 algebra groupoid in the form:

$$z = a + b\mathbf{i} + c\mathbf{j}, \tag{23}$$

where a, b, and c are real numbers; and i and j are the basic number. the component-wise addition is:

$$\begin{aligned}&(a_1 + b_1\mathbf{i} + c_1\mathbf{j}) + (a_2 + b_2\mathbf{i} + c_2\mathbf{j})\\&= (a_1 + a_2) + (b_1 + b_2)\,\mathbf{i} + (c_1 + c_2)\,\mathbf{j},\end{aligned} \tag{24}$$

and the component-wise scalar multiplication is:

$$\begin{aligned}&\lambda(a + b\mathbf{i} + c\mathbf{j})\\&= \lambda a + (\lambda b)\mathbf{i} + (\lambda c)\mathbf{j},\end{aligned} \tag{25}$$

The multiplication table of Z_3 algebra groupoid is presented in Table 2. The blue area is the original multiplication table of Z_3, we expand it to the orange part. The norm of $|z| = \sqrt{a^2 + b^2 + c^2}$.

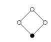
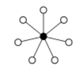
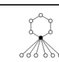
Table 2: Multiplication table of Z_3 algebra groupoid.

	1	i	j	-1	$-i$	$-j$
1	1	i	j	-1	$-i$	$-j$
i	i	j	1	$-i$	$-j$	-1
j	j	1	i	$-j$	-1	$-i$
-1	-1	$-i$	$-j$	1	i	j
$-i$	$-i$	$-j$	-1	i	j	1
$-j$	$-j$	-1	$-i$	j	1	i

Since (24), (25), and Table 2, the groupoid operation $*$ is automatically defined. Now, we have the $Z3$ algebra groupoid $(Z3A, *)$. The algebra here is not strict algebra in mathematics, but it is enough for KG groupoid representation learning. Other examples are shown in Table 3. In particular, O is non-associative and can represent KG with non-associative relations.

Actually, we can select the original groupoid according to the characteristics of the data sets. In section 2.4, we have mentioned that commutativity and associativity are negative requirements. If the data set has non-commutative or non-associative relations, it is better to choose a non-commutative or non-associative groupoid.

Table 3: Algebra derived from small groupoids[1].

Order	Groupoid	Algebra	ACG[2]	Com[2]	Ass[2]
1	Z_1	R		Y	Y
2	Z_2	C		Y	Y
4	K_4	$K4A$		Y	Y
6	Dih_3	$DIH3A$		N	Y
8	Q_8	H		N	Y
16	O^3	OA	-	N	N

[1] Small groupoids refer to the finite groupoids of small order.
[2] ACG refers to the cycle graph of the algebra groupoid. Com refers to the commutativity. Ass refers to the associativity.
[3] O refers to Octonion. Octonion is a groupoid but it is not a group.

As long as it is a non-commutative or non-associative discrete groupoid, the algebra groupoid built from the discrete groupoid must be non-commutative or non-associative. Therefore, we only need to study the properties of discrete groupoids and then select those discrete groupoids that meet the conditions.

Metric of groupoid: For algebra groupoids, we can derive metrics directly from the norm, named algebra metric. Given a triple (h, r, t) in KG, the fundamental metric of the character $h * r$ and the tail entity t can be defined as:

$$|h * r - t|, \tag{26}$$

where $|\cdot|$ is the norm of the algebra groupoids.

Empirically, many other widely used groupoids are Lie Groups. All models in Table 1 are Lie groups except RESCAL[14]and GrpMK. For the Lie group $\mathcal{G}$, we can define a new metric, named Frobenius metric. Given a triple (h, r, t) in KG, the metric of character $h * r$ and tail entity t can be defined as:

$$\|log((h * r)^{-1}t)\|, \tag{27}$$

where $\|\cdot\|$ is Frobenius norm. In fact, it is geodesic distance between the character and tail entity.

3.2 Theoretical Correlations between Current Models

In recent years, KG representation learning has attracted a lot of attention, and much work has thus far been demonstrated quite effectively. However, the theoretical mechanism behind them is much less well-understood. The isomorphism of groupoids lays the theoretical foundation for intrinsic correlations between models, leading to a better understanding of KG representation learning.

In section 2.5, we have analyzed the isomorphism of groupoids in Table 1. In mathematics, isomorphism is a structure-preserving mapping between two structures of the same type that can be reversed by an inverse mapping. Two isomorphic models mean that the models have similar relational composition, and the entities and relations of the two models have identical intrinsic connections. One

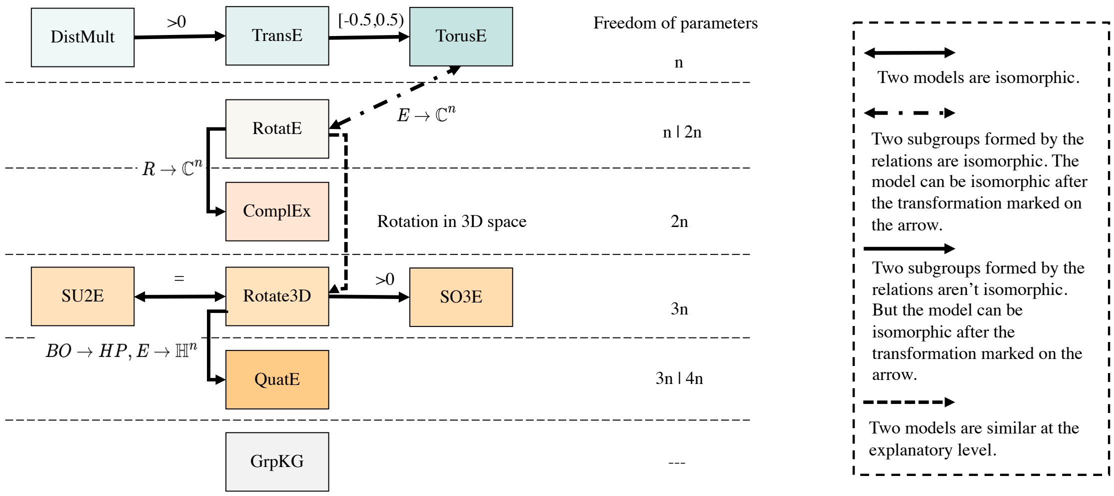


Figure 2: Intrinsic correlations between models.

example is that TorusE[5] is defined on quotient space. The relation-based mapping and the composition of relations are considered as the groupoid operation. In general, such an abstract definition is not straightforward to understand. With the help of isomorphism, we discover that TorusE[5] is equal to the rotation in the complex space, which is more apparent to understand. Compared with RotatE[15], we can perceive that they are highly similar.

We mainly examine the relation-based mapping of the models and the composition of the relations. It is because these two processes are the core of a model. The metrics or the loss functions can be empirically replaced with others, such as replacing Euclidean distance with cosine distance. In fact, they are equivalent under certain circumstances. We can summarize the intrinsic correlations of the models in Figure 2.

(1) Rotate3D[6] and SU2E[21] can be considered as the same model, although they are in different representation spaces.
(2) SO3E[21] can be considered as a special case of Rotate3D[6] if we restrict the real part of the quaternions greater than zero. Similarly, SO3E[21] is a specical case of SU2E[21] since Rotate3D[6] is isomorphic to SU2E[21].
(3) TorusE[5] has the same relation-based mapping as RotatE[15]. And if TorusE[5] removes the restriction on entities, i.e., $|E| = 1$. TorusE can extend to RotatE[15].
(4) RotatE[15] can extends to ComplEx[17] if TorusE[5] remove the restriction on relations, i.e., $|R| = 1$.
(5) TransE[3] can be considered as a specical case of TorusE[3] if the representation space $E, R, C \subset [-0.5, 0.5)^n$.
(6) DistMult[20] can be degenerated into TransE[3] if the representation space of DistMult[20] is restricted to positive numbers.
(7) Rotate3D[6] can extends to QuatE[22] if Rotate3D[6] represents the entities in $\mathbb{H}^n$ and changes the binary operation to Hamilton product.

3.3 A New Way to Study KG Representation Learning Models

The introduction of groupoids allows us to have a powerful tool to investigate KG representation learning, which leads to new fields as follows:

New classification standards: With the help of the groupoid, we put forward a novel classification standard in the true sense of the word, named degree of freedom. The degree of freedom can reveal some law of models, rather than just dividing models into several different categories.

Current classification standards[10, 19] are often based on the implementation structure of the models and divide KG representation learning models into translational models, semantic matching models, neural network models, etc. Overlaps cannot be avoided between different categories and the performances of models in the same category are challenging to show consistent commonality.

The degree of freedom is the number of free parameters in the representation of a entity or relation. We have presented the degree of freedom of the models in Table 1. If the entities and relations are not in the same representation space, this model has two degrees of freedom. The degree of freedom is a good indicator to evaluate the expressivity of models. We will give specific instructions during the experiments.

Other new fields: We can utilize the geodesic distance on the Lie group to give all models a uniform form of metric, shown in Section 3.1. Besides, We can use the matrix representation of the groupoid to study the characteristics of the relation, such as whether a dense matrix means a dense relation. There are similar studies in [9], but it is limited to the bilinear models. Now we can extend the analysis to more models. What's more, we can study the characteristics of the entities and relations from the perspective of the eigenvalues of the matrix.

3.4 Isomorphism Promotes the Analysis of Key Factors

The isomorphism of groupoids provides a basis for us to compare two different models. We can choose two isomorphic models and then analyze the reasons why they produce different experimental results. These factors may be metrics, negative sampling, regularization, and so on.

For example, we find that RotatE's experimental results are significantly better than TorusE[5]. However, the two models are isomorphic which means that the two models should have similar results. This anomaly prompts us to analyze other factors, such as negative sampling, regularization, and so on. And we finally find that a proper negative sample is very beneficial. Further analysis is in the experiments.

4 GRPMK AND GRPQ8

To verify KG groupoid representation learning models, we utilize GrpKG to construct two models, GrpMK and GrpQ8. The specific parameters are shown in Table 4.

Table 4: The parameters of GrpMK and GrpQ8.

Models	F-groupoid	degree of freedom	Metric
GrpMK	$(\mathbb{R}_{k\times k}, *)$	$k^2 * n$	Frobenius metric
GrpQ8	Q_8 Algebra	$4n$	Algebra metric

GrpMK selects the $k \times k$ matrix as the fundamental groupoid (not a group). The groupoid operation is matrix multiplication. And GrpQ8 selects the Q_8 algebra groupoid as the fundamental groupoid. Actually, GrpQ8 is isomorphic to H.

Loss function: We use cross entropy as the loss function:

$$\begin{aligned} L &= \sum_{d_{hrt}\in D} CrossEntropyLoss(-d_{hrt}) + \lambda\|G\| \\ &= \sum_{d_{hrt}\in D} -\log\left(\frac{\exp(-d_{hrt})}{\sum_{d\in D_{hrt}}\exp(-d)}\right) + \lambda\|G\| \\ &= \sum_{d_{hrt}\in D}\left(d_{hrt} + \log\left(\sum_{d\in D_{hrt}}\exp(-d)\right)\right) + \lambda\|G\|, \end{aligned} \tag{28}$$

where d_{hrt} is the metric between the character c ($c = h * r$) and the tail entity t. D is the set of the metrics of the training triples and $D_{h,r,t}$ is the set of metrics of the triples, which is created by replacing the tail entity with all the entities in the data set. $\|G\|$ is the regulation used to prevent over-fitting.

5 RELATED WORK

Recently, KG embedding has attracted significant research interest. TransE[3] is a translation based model and it believes that triples (h, r, t) should satisfy $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \approx 0$. TorusE[5] solves the regularization problem of TransE[3] by changing the representation space from the vector space to a torus. RotatE[15] firstly defines each relation as a rotation from the head entity to the tail entity in the complex vector. Rotate3D[6] extends the RotatE[15] and maps entities to the three-dimensional space and defines relations as rotations from head entities to tail entities. QuatE[22] introduce more expressive hypercomplex representations to model entities and relations.

Besides above models, there are also some semantic matching models. RESCAL[20] represents the entities as vectors and the relations as matrixes and the score function are defined by a bilinear function. DistMult[20] simplifies RESCAL[14] by restricting M_r to diagonal matrices. However, the DistMult[20] algorithm cannot handle asymmetric relations. ComplEx[17] entends DistMult[20] to the complex field so as to better model asymmetric relations.

Yang[21] demonstrates a group-based embedding framework is essential for designing embedding models and proposes SU2E[21]and SO3[21]. However, Yang[21] fails to explore the intrinsic correlations between the models.

6 EXPERIMENTS

6.1 Datasets

WN18[3] is a subset of WordNet[12] and **FB15k**[3] is a subset extracted from a Freebase[2], a large collaborative knowledge base consisting of data composed mainly by its community members. **FB15k-237**[16] and **WN18RR**[4] are derived from **FB15k**[3] and **WN18**[3], respectively, with inverse relations removed.

6.2 Link Prediction

The task of link prediction aims to complete a triple when one entity of (h, r, t) is missing by minimizing the loss function. The following measures are used as our evaluation metrics: (1) MRR: mean reciprocal rank of correct entities; (2) H@: proportion of valid entities ranked in top 10.

Experimental setup: We conduct link prediction experiments in the same way as other knowledge graph embedding models[15].

Result: Table 5 provides the experimental data on the link prediction. It can be seen that:

(1) GrpQ8 and GrpM2 generally outperform other models, which shows that groupoid-based models are effective.
(2) GrpM2 is not a group but still has good performance, which shows that it is too strict to use the groups as the structure of the KG representation learning models. Therefore, the groupoid is more suitable.
(3) The introduction of groupoids provides an effective tool for model research. For example, we find that the link prediction results of models with the same degree of freedom are similar. On the whole, models with higher degrees of freedom have more expressivity.
(4) ComplEx[17] and RotatE's better results than TorusE[5] shows that removing the restriction on the norm of entities or relations is beneficial to improve the model's performance when the degree of freedom is small. However, the results of GrpQ8 and QuatE[22] are overall similar. We guess the benefit will become weaker as the degree of freedom increases.

6.3 Extensions of TransE

In this experiment, we try to extend TransE[3] according to the conditions in Figure 2 and compare it with corresponding models, i.e.

Table 5: Link prediction results on benchmark data sets.

Models	FB15k-237		WN18RR		FB15k		WN18		DF
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	
TransE^{1} [3]	0.288	0.190	0.219	0.015	0.614	0.459	0.545	0.211	n
DistMult^{2} [20]	0.295	0.204	0.449	0.419	0.798	-	0.797	-	n
TorusE[5]	-	-	-	-	0.733	0.674	0.947	0.943	n
RotatE[15]	0.338	0.241	0.476	0.428	0.797	0.746	0.947	0.942	n|2n
ComplEx[17]	0.35	-	0.47	-	0.80	-	0.95	-	2n
Rotat3D[6]	0.347	0.250	0.489	0.442	0.789	0.728	**0.951**	**0.945**	3n
SU2E[21]	0.340	0.243	0.476	0.429	0.791	0.734	0.950	0.944	3n
SO3E[21]	0.340	0.244	**0.477**	0.432	0.794	0.740	0.950	0.944	3n
QuatE[22]	0.348	0.248	0.488	**0.438**	0.833	**0.800**	0.950	0.944	3n|4n
GrpQ8	**0.355**	**0.262**	0.474	0.435	0.832	0.786	0.950	**0.945**	4n
GrpM2	0.350	0.257	0.459	0.414	**0.842**	**0.800**	0.950	**0.945**	4n

Results of ^{1} are taken from [23]. Results of ^{2} is taken from [11]. Others are taken from original paper.

$$\boxed{\text{TransE}} \xrightarrow{[-0.5,0.5)} \boxed{\text{TorusE}} \xrightarrow{E \to \mathbb{C}^n} \boxed{\text{RotatE}} \xrightarrow{R \to \mathbb{C}^n} \boxed{\text{ComplEx}}$$

TransE extends to TransE^1 with the entities in $[-0.5, 0, 5)$, and the metric changed to cosine distance. TransE^2 removes the restriction on the norm of entities from TransE^1. TransE^3 eliminates the limitation on the norm of relations from TransE^2.

Table 6: Link prediction results of TransE and its extensions.

Models	FB15k				
	Changes	MRR	H@1	H@3	H@10
TorusE[5]	-	0.733	0.674	0.771	0.832
RotatE[15]	-	0.797	0.746	0.830	0.884
ComplEx[17]	-	0.80	-	-	0.89
TransE[3]	-	0.594	0.506	0.648	0.750
TransE^1	$[-0.5, 0.5)$	0.731	0.664	0.774	0.848
TransE^2	$E \to \mathbb{C}^n$	0.754	0.688	0.799	0.870
TransE^3	$R \to \mathbb{C}^n$	**0.810**	**0.762**	**0.846**	**0.896**

We can observe from Table 6 that the link prediction results of TransE^1, TransE^2, and TransE^3 are gradually improving, which corresponds to the trend of ToursE, RotatE, and ComplEx. The results demonstrate that TransE and the other three models do have the relationship shown in Figure 2.

6.4 The Analysis of Key Factors

In Section 3.4, we argue that isomorphism could assist us in analyzing two different models. We observe that the link prediction results of RotatE are significantly ahead of TorusE. With the help of the isomorphism of the groupoid, we can exclude the impact of relation-based mapping. Further analysis shows that among metrics, negative sampling, and regularization, negative sampling is the critical factor. Table 7 presents the results.

We take MRR as the measure. When other variables remain unchanged, we find that MRR has been significantly improved with the number of negative samples increasing. It shows that the number of negative samples has a direct effect on link prediction.

Table 7: The impact of negative sampling on link prediction.

Num	1	3	5	10
MRR/H@1	0.633/0.510	0.696/0.581	0.721/0.611	0.752/0.654
Num	50	150	200	256
MRR/H@1	0.788/0.723	0.795/0.741	0.791/0.738	0.791/0.739

It is worth noting that the number of negative samples is also limited by GPU memory. If the number of negative samples is too large, it will fill up the GPU memory and hinder the training. In fact, there is an interactive relationship between the metric and negative sampling. Using Euclidean distance as a metric requires extra space to store intermediate variables, which will limit the number of negative samples to several hundred. If the inner product of the vector is used as the metric, we can use matrix multiplication to combine the inner product and the sum process. As a result, there is no need to store intermediate variables, and the number of negative samples can easily exceed thousands (taking FB15k as an example).

7 CONCLUSION

In this work, we attempt to utilize groupoids to represent KG representation learning models. Specifically, we propose a general KG representation learning framework, named GrpKG. Many models can be seen as special cases of the GrpKG. We also prove that models can be converted to each other in the sense of groupoid isomorphism. In addition, we argue that the groupoid is a new way to study KG representation learning models. In future work, we expect to utilize the properties of the groupoid to analyze the characteristics of the models.

REFERENCES

[1] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia-A crystallization

point for the Web of Data. *Journal of web semantics* 7, 3 (2009), 154–165.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *2008 ACM SIGMOD International Conference on Management of Data*. 1247–1250.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*. 1–9.

[4] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[5] Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *32rd the AAAI Conference on Artificial Intelligence*, Vol. 32.

[6] Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. 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*. 385–394.

[7] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[8] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *12th ACM International Conference on Web Search and Data Mining*. 105–113.

[9] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 985–991.

[10] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[11] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744* (2017).

[12] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[13] Sameh K Mohamed, Vít Novácek, Pierre-Yves Vandenbussche, and Emir Muñoz. 2019. Loss Functions in Knowledge Graph Embedding Models.. In *DL4KG@ ESWC*. 1–10.

[14] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data.. In *International Conference on Machine Learning*, Vol. 11. 809–816.

[15] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*. 1–18.

[16] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*. 57–66.

[17] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*. 2071–2080.

[18] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[19] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[20] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*.

[21] Tong Yang, Long Sha, and Pengyu Hong. 2020. NagE: Non-abelian group embedding for knowledge graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1735–1742.

[22] SHUAI ZHANG, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. *Advances in Neural Information Processing Systems* 32 (2019), 2735–2745.

[23] Zhaocheng Zhu, Shizhen Xu, Jian Tang, and Meng Qu. 2019. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *The World Wide Web Conference*. 2494–2504.