Knowledge Graph Embedding by Normalizing Flows

Changyi Xiao[1], **Xiangnan He**[1*], **Yixin Cao**[2]
[1]School of Data Science, University of Science and Technology of China
[2]School of Computing and Information System, Singapore Management University
changyi@mail.ustc.edu.cn, {xiangnanhe, caoyixin2011}@gmail.com

Abstract

A key to knowledge graph embedding (KGE) is to choose a proper representation space, e.g., point-wise Euclidean space and complex vector space. In this paper, we propose a unified perspective of embedding and introduce uncertainty into KGE from the view of group theory. Our model can incorporate existing models (i.e., generality), ensure the computation is tractable (i.e., efficiency) and enjoy the expressive power of complex random variables (i.e., expressiveness). The core idea is that we embed entities/relations as elements of a symmetric group, i.e., permutations of a set. Permutations of different sets can reflect different properties of embedding. And the group operation of symmetric groups is easy to compute. In specific, we show that the embedding of many existing models, point vectors, can be seen as elements of a symmetric group. To reflect uncertainty, we first embed entities/relations as permutations of a set of random variables. A permutation can transform a simple random variable into a complex random variable for greater expressiveness, called a normalizing flow. We then define scoring functions by measuring the similarity of two normalizing flows, namely NFE. We construct several instantiating models and prove that they are able to learn logical rules. Experimental results demonstrate the effectiveness of introducing uncertainty and our model. The code is available at https://github.com/changyi7231/NFE.

Introduction

A knowledge graph (KG) contains a large number of triplets with form (h, r, t), where h is a head entity, r is a relation and t is a tail entity. Existing KGs often suffer from the incompleteness problem. To complement the KGs, knowledge graph embedding (KGE) models map entities and relations into low-dimensional distributed representations and define scoring functions to measure the likelihood of the triplets.

A key to KGE is to choose a proper representation space such that the embedding can make good use of the properties of the space (Ji et al. 2021). For example, point-wise Euclidean space is widely applied for representing entities/relations due to its simplicity yet effectiveness(Yang et al. 2014). Complex vector space is then proposed to learn sophisticated logical rules (Sun et al. 2018; Toutanova et al. 2015). Moreover, the usage of quaternion space brings more degree of freedom of rotation than complex space (Zhang et al. 2019). Besides, hyperbolic space is used to capture hierarchical logical rules (Chami et al. 2020). Motivated by the group theory, non-Abelian group is exploited to find the hidden group algebraic structure of relations (Yang, Sha, and Hong 2020). However, the above works hardly consider the uncertainty information of embedding, which limit the expressiveness.

To reflect uncertainty, a common method is to embed entities/relations as random variables (He et al. 2015). However, it has two drawbacks. First, it is difficult to parameterize a complex random variable directly due to the difficulty of computing the partition function (Goodfellow, Bengio, and Courville 2016). Second, it is intractable to compute the probability density function (PDF) of the sum of two random variables due to the difficulty of computing the convolution of two PDFs (Bertsekas and Tsitsiklis 2008).

In this paper, we propose a unified perspective of embedding and introduce uncertainty into KGE from the view of group theory. Our model makes the embedding more general and ensures the computation is tractable. We embed entities/relations as elements of a symmetric group, i.e., permutations of a set. (A permutation of a set X is an invertible function from X to X.) On one hand, since point vectors can be seen as elements of a symmetric group, our proposed embedding is a generalization of point vectors. Moreover, Cayley's theorem (Hungerford 2012), every group is isomorphic to a subgroup of a symmetric group, also shows the generality of our proposed embedding. On the other hand, the group operation of symmetric groups, the composition of functions, is easy to compute. We can embed entities/relations as permutations of different sets to reflect different properties of embedding. For example, if the set is a vector space (or a random variable space), then the permutations of that set reflect the certainty (or uncertainty) property.

In specific, to reflect uncertainty, we embed entities/relations as permutations of a set of random variables. The permutation or invertible function can transform a simple random variable into a complex random variable for greater expressiveness, which is called a normalizing flow (Papamakarios et al. 2021; Kobyzev, Prince, and Brubaker 2020). We can easily parameterize a complex invertible function (instead of a complex random variable), and compute the PDF of a random variable obtained by an invertible function acting on a simple random variable (Papamakarios et al.

[*]Corresponding author.

2021; Kobyzev, Prince, and Brubaker 2020). Our method can be seen as a generalized reparameterization method (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014). We then define the scoring function by measuring the similarity of two normalizing flows, namely Normalizing Flows Embedding (NFE). Finally, we instantiate several NFE models by choosing different invertible functions. We further prove that NFE is able to learn logical rules.

The main contributions of this paper are listed below:

1. We propose a unified perspective of embedding from the view of group theory, which offers a rigorous theoretical understanding of KGE.
2. We proposed NFE to introduce uncertainty into KGE, which enjoys generality, efficiency and expressiveness.
3. Experimental results demonstrate the effectiveness of introducing uncertainty into KGE and our model.

Background

In this section, we introduce the related background of our model, knowledge graph embedding, normalizing flows and group theory.

Knowledge Graph Embedding

Let $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. A KG contains a set of triplets $\mathcal{F} = \{(h,r,t)\} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. KGE aims at learning the distributed representations or embeddings for entities and relations. It defines a scoring function $f(h,r,t)$ to measure the likelihood of a triplet (h,r,t) based on the embeddings $(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$. Existing KGE models include translation-based models, multiplicative models and so on (Zhang et al. 2021).

Translation-based models learn embeddings by translating a head entity to a tail entity through a relation. TransE (Bordes et al. 2013), a representative model of translation-based models, defines the scoring function as the negative distance between $\boldsymbol{h}+\boldsymbol{r}$ and $\boldsymbol{t}$, i.e.,

$$f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = -\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|$$

where $\boldsymbol{h},\boldsymbol{r},\boldsymbol{t} \in \mathbb{R}^n$ and $\|\cdot\|$ is a norm of a vector.

Multiplicative models measure the likelihood of a triplet by product-based similarity of entities and relations. DistMult (Yang et al. 2014), a representative model of multiplicative models, defines the scoring function as the inner product of $\boldsymbol{h}$, $\boldsymbol{r}$ and $\boldsymbol{t}$, i.e.,

$$f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = \langle \boldsymbol{h},\boldsymbol{r},\boldsymbol{t} \rangle := \sum_{i=1}^{n} \boldsymbol{h}_i\boldsymbol{r}_i\boldsymbol{t}_i$$

where $\boldsymbol{h},\boldsymbol{r},\boldsymbol{t} \in \mathbb{R}^n$ and $\langle \cdot,\cdot,\cdot \rangle$ is the inner product of three vectors.

Normalizing Flows

A normalizing flow is a sequence of invertible and differentiable functions, which transforms a simple probability distribution ($e.g.$, a standard normal) into a more complex distribution (Papamakarios et al. 2021; Kobyzev, Prince, and Brubaker 2020). Let $\boldsymbol{Z}$ be a random variable in $\mathbb{R}^n$ with a known and tractable PDF $p_{\boldsymbol{Z}}(\boldsymbol{z})$ and $\boldsymbol{X} = g(\boldsymbol{Z})$ be an invertible function which transforms $\boldsymbol{Z}$ into $\boldsymbol{X}$. The PDF of the random variable $\boldsymbol{X}$ follows

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{Z}}(g^{-1}(\boldsymbol{x}))|\det(\frac{\partial g^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}})| \qquad (1)$$

where g^{-1} is the inverse of g and $\det(\frac{\partial g^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}})$ is the determinant of the Jacobian of g^{-1} evaluated at $\boldsymbol{x}$. We next show an example of normalizing flows.

Example 1 (Pushing uniform to normal)**.** Let $z \sim U[0,1]$ be uniformly distributed and $x \sim \mathcal{N}(\mu,\sigma^2)$ be normally distributed. The invertible function

$$x = S(z) = \mu + \sqrt{2}\sigma \cdot \mathrm{erf}^{-1}(2z-1)$$

pushes z into x, where $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-s^2}\mathrm{d}s$ is the error function.

Group Theory

A group is a set G together with a group operation on G, here denoted $\cdot$, that combines any two elements a and b to form an element of G, denoted $a \cdot b$, such that the following three requirements are satisfied:

1. **Associativity** $\forall a,b,c \in G, (a \cdot b)\cdot c = a \cdot (b \cdot c)$.
2. **Identity element** $\exists e \in G$ such that $\forall a \in G, e \cdot a = a$ and $a \cdot e = a$.
3. **Inverse element** $\forall a \in G, \exists b \in G$ such that $a \cdot b = e$ and $b \cdot a = e$.

A permutation group is a group G whose elements are permutations of a given set X and whose group operation is the composition of functions in G, where permutations are invertible functions from the set X to itself. The group of all permutations of a set X is the symmetric group of X, denoted as $\mathrm{Sym}(X)$. The term permutation group thus means a subgroup of the symmetric group. By Cayley's theorem (Hungerford 2012), every group is isomorphic to a permutation group, which characterizes group structure as the structure of a set of invertible functions.

Normalizing Flows Embedding

In this section, we first propose a unified perspective of embedding and extend to the general form of normalizing flows embedding (NFE). We then define the concrete forms of NFE and prove that they are able to learn logical rules. Finally, we show the loss function.

Unified Perspective of Embedding

Most KGE models embed entities/relations as point vectors in a low-dimensional space. We propose a unified perspective of embedding from the view of symmetric groups and show that point vectors can be seen as elements of a symmetric group. We take the representative KGE models, TransE and DistMult, as examples to illustrate. We show the results for other models in Appendix B.

TransE Let $G = X = \mathbb{R}^n$, TransE embeds every entity/relation as a vector in G. We show that every vector can correspond to a permutation of X, i.e., an element of the symmetric group $\mathrm{Sym}(X)$. We define a map α from G to $\mathrm{Sym}(X)$:

$$\begin{aligned}&\alpha : G \to \mathrm{Sym}(X)\\&\alpha(\boldsymbol{g}) = f_{\boldsymbol{g}}\end{aligned}$$

where $\boldsymbol{g} \in G$ and the definition of $f_{\boldsymbol{g}}$ is as follows:

$$\begin{aligned}&f_{\boldsymbol{g}} : X \to X\\&f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g} + \boldsymbol{x}\end{aligned}$$

where $\boldsymbol{x} \in X$. The image of α is a permutation group, i.e., a subgroup of $\mathrm{Sym}(X)$. For every triplet $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we have:

$$f_{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{h} + \boldsymbol{x}, f_{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{r} + \boldsymbol{x}, f_{\boldsymbol{t}}(\boldsymbol{x}) = \boldsymbol{t} + \boldsymbol{x}$$

then TransE can be represented as:

$$\begin{aligned}f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) &= -\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|\\&= -\|\boldsymbol{r} + (\boldsymbol{h} + \boldsymbol{0}) - (\boldsymbol{t} + \boldsymbol{0})\|\\&= D(f_{\boldsymbol{r}}(f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}})), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}}))\\&= D(f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}}), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}}))\end{aligned}$$

where $\boldsymbol{x}_0 = \boldsymbol{0} \in X$, $\circ$ denotes functional composition and $D(\boldsymbol{a}, \boldsymbol{b}) = -\|\boldsymbol{a} - \boldsymbol{b}\|, \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$. $D(\cdot, \cdot)$ is a similarity metric, which outputs a real value.

DistMult Let $G = (\mathbb{R}\backslash\{0\})^n$ and $X = \mathbb{R}^n$, DistMult embeds every entity/relation as a vector in G. We can similarly define a map α from G to $\mathrm{Sym}(X)$:

$$\begin{aligned}&\alpha : G \to \mathrm{Sym}(X)\\&\alpha(\boldsymbol{g}) = f_{\boldsymbol{g}}\end{aligned}$$

where $\boldsymbol{g} \in G$ and the definition of $f_{\boldsymbol{g}}$ is as follows:

$$\begin{aligned}&f_{\boldsymbol{g}} : X \to X\\&f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g} \odot \boldsymbol{x}\end{aligned}$$

where $\boldsymbol{x} \in X$ and $\odot$ is Hadamard product. The image of α is another subgroup of $\mathrm{Sym}(X)$. For every triplet $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we have:

$$f_{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{h} \odot \boldsymbol{x}, f_{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{r} \odot \boldsymbol{x}, f_{\boldsymbol{t}}(\boldsymbol{x}) = \boldsymbol{t} \odot \boldsymbol{x}$$

then DistMult can be represented as:

$$\begin{aligned}f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) &= \langle \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \rangle\\&= \langle \boldsymbol{r} \odot (\boldsymbol{h} \odot \boldsymbol{1}), \boldsymbol{t} \odot \boldsymbol{1} \rangle\\&= D(f_{\boldsymbol{r}}(f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}})), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}}))\\&= D(f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}}), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}}))\end{aligned}$$

where $\boldsymbol{x}_0 = \boldsymbol{1} \in X$ and $D(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^T\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$.

Unified Representation Therefore, we get a unified representation of TransE and DistMult from the view of symmetric groups:

$$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = D(f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}}), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}})) \qquad (2)$$

The unified representation is defined by measuring the similarity of $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}$ evaluated at $\boldsymbol{x}_{\boldsymbol{0}}$ and $f_{\boldsymbol{t}}$ evaluated at $\boldsymbol{x}_{\boldsymbol{0}}$. It is composed of three parts, the initial object $\boldsymbol{x}_0 \in X$, the permutations $\{f_{\boldsymbol{h}}, f_{\boldsymbol{r}}, f_{\boldsymbol{t}}\}$ and the similarity metric $D(\cdot, \cdot)$. Thus, we can define scoring functions in terms of permutations of a set.

TransE and DistMult embed entities/relations into different subgroup of the same symmetric group $\mathrm{Sym}(\mathbb{R}^n)$ to get different scoring functions. Since every point vector corresponds to an element of $\mathrm{Sym}(\mathbb{R}^n)$, the elements of $\mathrm{Sym}(\mathbb{R}^n)$ can reflect the certainty property of point vectors. Thus, we can reflect different properties of embedding and define different scoring functions by choosing different symmetric groups. Next, we introduce the uncertainty of embedding by choosing a suitable symmetric group.

General Form of NFE

To introduce uncertainty, we let X be the set of random variables on $\mathbb{R}^n$ and embed entities/relations as elements of $\mathrm{Sym}(X)$. For every triplet $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we have the corresponding permutations or invertible functions $\{f_{\boldsymbol{h}}, f_{\boldsymbol{r}}, f_{\boldsymbol{t}}\}$. Our NFE is defined in the form of Eq.(2), where $\boldsymbol{x}_0 \in X$ is a random variable and $D(\cdot, \cdot)$ is a similarity metric between two random variables or two PDFs. $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}})$ or $f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}})$ is a normalizing flow, which transforms a random variable $\boldsymbol{x}_0$ into another random variable, this reflects uncertainty information of $\{f_{\boldsymbol{h}}, f_{\boldsymbol{r}}, f_{\boldsymbol{t}}\}$. Thus, NFE is to measure the similarity of two normalizing flows. We can easily compute the PDFs of $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}})$ and $f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}})$ by Eq.(1), then the general form of NFE is represented as

$$\begin{aligned}&f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = D(f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_{\boldsymbol{0}}), f_{\boldsymbol{t}}(\boldsymbol{x}_{\boldsymbol{0}}))\\&\qquad = D(p_{\boldsymbol{x}_0}(f_{\boldsymbol{h}}^{-1} \circ f_{\boldsymbol{r}}^{-1}(\boldsymbol{x}))|\det(\frac{\partial f_{\boldsymbol{h}}^{-1} \circ f_{\boldsymbol{r}}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}})|,\\&\qquad\qquad p_{\boldsymbol{x}_0}(f_{\boldsymbol{t}}^{-1}(\boldsymbol{x}))|\det(\frac{\partial f_{\boldsymbol{t}}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}})|) \qquad (3)\end{aligned}$$

In next section, we define concrete $\boldsymbol{x}_0$, $\{f_{\boldsymbol{h}}, f_{\boldsymbol{r}}, f_{\boldsymbol{t}}\}$ and $D(\cdot, \cdot)$ to get the concrete forms of NFE.

Comparison of KG2E and NFE KG2E (He et al. 2015) embeds entities/relations as random variables and defines the scoring function as $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = D(\boldsymbol{h} - \boldsymbol{t}, \boldsymbol{r})$, where $\boldsymbol{h}, \boldsymbol{r}$ and $\boldsymbol{t}$ are random variables with normal distributions. However, it has two drawbacks to represent entities/relations as random variables to model uncertainty. First, it is difficult to parameterize a complex random variable directly due to the difficulty of computing the partition function (Goodfellow, Bengio, and Courville 2016). We get a complex random variable by using a complex invertible function to act on a simple random variable. It is easy to parameterize a complex invertible function and compute the PDF of a random variable obtained by an invertible function acting on a simple random variable by Eq.(1) (Papamakarios et al. 2021; Kobyzev, Prince, and Brubaker 2020). Thus, our method can be seen as a generalized reparameterization method (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014). Second, KG2E involves computing the PDF of the sum/difference of two random variables, i.e., the PDF of $\boldsymbol{h} - \boldsymbol{t}$, which is intractable in most cases due to the difficulty of computing the convolution of two PDFs (Bertsekas and Tsitsiklis 2008). For example, if we embed a head entity $\boldsymbol{h}$ and a relation $\boldsymbol{r}$ as random variables with Beta distribution, then it is hard to compute the PDF of $\boldsymbol{h} - \boldsymbol{t}$. In contrast, Eq.(3) requires computing the PDF of $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_0)$, which is still a normalizing flow because a composition of invertible functions remains invertible. It is easy to compute the composition of $f_{\boldsymbol{h}}$ and $f_{\boldsymbol{r}}$ and the PDF of $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_0)$ by Eq.(1). In conclusion, NFE is more general and computationally simple.

Concrete Form of NFE

Based on the general form of NFE, Eq.(3), we define the concrete initial random variables $\boldsymbol{x}_0$, invertible functions $\{f_{\boldsymbol{h}}, f_{\boldsymbol{r}}, f_{\boldsymbol{t}}\}$ and similarity metrics $D(\cdot, \cdot)$ to get the concrete forms of NFE.

For the initial random variable $\boldsymbol{x}_0$, we can choose it to be a simple random variable for the convenience of computing Eq.(3), such as a random variable with uniform distribution $U[-\sqrt{3}, \sqrt{3}]^n$ or a random variable with standard normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

The invertible functions $f_{\boldsymbol{g}}$ can be chosen as $f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g} + \boldsymbol{x}$ as in TransE or $f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g} \odot \boldsymbol{x}$ as in DistMult. We use the composition of the two functions:

$$f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g}_\sigma \odot \boldsymbol{x} + \boldsymbol{g}_\mu \tag{4}$$

where $\boldsymbol{g}_\sigma \in \mathbb{R}^n$, the entries of $\boldsymbol{g}_\sigma$ are non-zero, and $\boldsymbol{g}_\mu \in \mathbb{R}^n$. Thus, for every triplet $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we have that

$$\begin{aligned} & f_{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{h}_\sigma \odot \boldsymbol{x} + \boldsymbol{h}_\mu, f_{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{r}_\sigma \odot \boldsymbol{x} + \boldsymbol{r}_\mu \\ & f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{r}_\sigma \odot \boldsymbol{h}_\sigma \odot \boldsymbol{x} + \boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu \\ & f_{\boldsymbol{t}}(\boldsymbol{x}) = \boldsymbol{t}_\sigma \odot \boldsymbol{x} + \boldsymbol{t}_\mu \end{aligned} \tag{5}$$

We denote the PDF of $f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}_0)$ as $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and the PDF of $f_{\boldsymbol{t}}(\boldsymbol{x}_0)$ as $q_{\boldsymbol{t}}(\boldsymbol{x}_0)$. If $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $f_{\boldsymbol{g}}$ is a linear (affine) function $f_{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g}_\sigma \odot \boldsymbol{x} + \boldsymbol{g}_\mu$, then $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $q_{\boldsymbol{t}}(\boldsymbol{x}_0)$ are still normal distributions. A more expressive choice of $f_{\boldsymbol{g}}$ than Eq.(4) can be piecewise linear functions:

$$f_{\boldsymbol{g}}(\boldsymbol{x})_i = \begin{cases} \boldsymbol{g}_{\sigma_1 i} \odot \boldsymbol{x}_i + \boldsymbol{g}_{\mu i} & \text{if } \boldsymbol{x}_i \leq 0 \\ \boldsymbol{g}_{\sigma_2 i} \odot \boldsymbol{x}_i + \boldsymbol{g}_{\mu i} & \text{if } \boldsymbol{x}_i > 0 \end{cases} \tag{6}$$

where $\boldsymbol{g}_{\sigma_1}, \boldsymbol{g}_{\sigma_2}, \boldsymbol{g}_\mu \in \mathbb{R}^n$. For simplicity, we denote Eq.(6) as

$$f_{\boldsymbol{g}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{g}_{\sigma_1} \odot \boldsymbol{x} + \boldsymbol{g}_\mu & \text{if } \boldsymbol{x} \leq \boldsymbol{0} \\ \boldsymbol{g}_{\sigma_2} \odot \boldsymbol{x} + \boldsymbol{g}_\mu & \text{if } \boldsymbol{x} > \boldsymbol{0} \end{cases}$$

To ensure Eq.(6) is invertible, we need to constraint $\boldsymbol{g}_{\sigma_1} \odot \boldsymbol{g}_{\sigma_2} > \boldsymbol{0}$. Since the composition of two piecewise linear functions with two pieces is a piecewise linear function with three pieces, we still implement $f_{\boldsymbol{r}}$ as a linear function. This ensures that for any $f_{\boldsymbol{h}}(\boldsymbol{x})$ and $f_{\boldsymbol{r}}(\boldsymbol{x})$, there exists a $f_{\boldsymbol{t}}(\boldsymbol{x})$ such that $f_{\boldsymbol{t}}(\boldsymbol{x}) = f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x})$ for any $\boldsymbol{x}$. Thus, for every triplet $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$, we have that

$$\begin{aligned} & f_{\boldsymbol{h}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{h}_{\sigma_1} \odot \boldsymbol{x} + \boldsymbol{h}_\mu & \text{if } \boldsymbol{x} \leq \boldsymbol{0} \\ \boldsymbol{h}_{\sigma_2} \odot \boldsymbol{x} + \boldsymbol{h}_\mu & \text{if } \boldsymbol{x} > \boldsymbol{0} \end{cases}, \\ & f_{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{r}_\sigma \odot \boldsymbol{x}_i + \boldsymbol{r}_\mu, \\ & f_{\boldsymbol{r}} \circ f_{\boldsymbol{h}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{r}_\sigma \odot (\boldsymbol{h}_{\sigma_1} \odot \boldsymbol{x} + \odot \boldsymbol{h}_\mu) + \boldsymbol{r}_\mu & \text{if } \boldsymbol{x} \leq \boldsymbol{0} \\ \boldsymbol{r}_\sigma \odot (\boldsymbol{h}_{\sigma_2} \odot \boldsymbol{x} + \odot \boldsymbol{h}_\mu) + \boldsymbol{r}_\mu & \text{if } \boldsymbol{x} > \boldsymbol{0} \end{cases}, \\ & f_{\boldsymbol{t}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{t}_{\sigma_1} \odot \boldsymbol{x} + \boldsymbol{t}_\mu & \text{if } \boldsymbol{x} \leq \boldsymbol{0} \\ \boldsymbol{t}_{\sigma_2} \odot \boldsymbol{x} + \boldsymbol{t}_\mu & \text{if } \boldsymbol{x} > \boldsymbol{0} \end{cases} \end{aligned} \tag{7}$$

The similarity metric $D(\cdot, \cdot)$ can be chosen as the negative Kullback–Leibler (KL) divergence between the PDFs, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$,

$$D(p, q) = -KL(p, q) = -\int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \tag{8}$$

or negative Wasserstein distance between the PDFs (Peyré, Cuturi et al. 2019), $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$,

$$\begin{aligned} D(p, q) &= -W(p, q) \\ &= -\inf_{\gamma \in \Gamma(p,q)} \iint \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \gamma(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} \end{aligned} \tag{9}$$

where Γ denotes the set of all joint distributions of $(\boldsymbol{x}, \boldsymbol{y})$. Wasserstein distance is still valid if the support sets of $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$ are not overlapped, while the KL divergence is not valid. For example, $W(p(\boldsymbol{x}), q(\boldsymbol{y})) = 4$ and $KL(p(\boldsymbol{x}), q(\boldsymbol{y})) = \infty$ if $\boldsymbol{x} \sim U[0, 1]$ and $\boldsymbol{y} \sim U[2, 3]$. Therefore, we use Wasserstein distance instead of KL divergence. However, Wasserstein distance is difficult to compute in most cases (Peyré, Cuturi et al. 2019). It has a tractable solution when $n = 1$:

$$W(p, q) = \int_0^1 (F^{-1}(z) - G^{-1}(z))^2 \mathrm{d}z$$

where $F^{-1}(z)$ and $G^{-1}(z)$ are the inverse cumulative distribution function of $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$, respectively. Our idea is to decompose the n-dimensional Wasserstein distance into a sum of 1-dimensional Wasserstein distances. We have the following proposition to realize it.

Proposition 1. *For two PDFs, $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$, $W(p, q) = \sum_{i=1}^n W(p_i, q_i)$ iff $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$ share the same copula, where p_i and q_i are the marginal distributions of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, respectively (Cuestaalbertos, Ruschendorf, and Tuerodiaz 1993).*

See Appendix C for the proofs. Thus, we can get the following corollary.

Corollary 1. *For two PDFs, $p(\boldsymbol{x})$ and $q(\boldsymbol{y})$, $W(p, q) = \sum_{i=1}^n W(p(\boldsymbol{x}_i), q(\boldsymbol{y}_i))$ if $p(\boldsymbol{x}) = \prod_{i=1}^n p(\boldsymbol{x}_i)$ and $q(\boldsymbol{y}) = \prod_{i=1}^n q(\boldsymbol{y}_i)$.*

We design proper scoring functions such that $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $q_{\boldsymbol{t}}(\boldsymbol{x}_0)$ satisfy the conditions in Corollary 1. In summary, we can define the concrete forms of NFE. We have the following propositions.

Proposition 2. *If $\boldsymbol{x}_0 \sim U[-\sqrt{3}, \sqrt{3}]^n$ or $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the invertible functions are Eq.(5) and similarity metric is Eq.(9), then the scoring function is*

$$\begin{aligned} f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = &- \|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2 \\ &- \||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\sigma| - |\boldsymbol{t}_\sigma|\|_2^2 \end{aligned} \tag{10}$$

Proposition 3. *If $\boldsymbol{x}_0 \sim U[-\sqrt{3}, \sqrt{3}]^n$, the invertible functions are Eq.(7) and similarity metric is Eq.(9), then the* *scoring function is*

$$
\begin{aligned}
& f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = -\|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2 \\
& - \frac{1}{2}\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_1}| - |\boldsymbol{t}_{\sigma_1}|\|_2^2 \\
& - \frac{1}{2}\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_2}| - |\boldsymbol{t}_{\sigma_2}|\|_2^2 - \sqrt{\frac{3}{4}}(\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu)^T \\
& (|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_2}| - |\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_1}| + |\boldsymbol{t}_{\sigma_1}| - |\boldsymbol{t}_{\sigma_2}|) \qquad (11)
\end{aligned}
$$

Proposition 4. *If $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, the invertible functions are Eq.(7) and similarity metric is Eq.(9), then the scoring function is*

$$
\begin{aligned}
& f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = -\|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2 \\
& - \frac{1}{2}\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_1}| - |\boldsymbol{t}_{\sigma_1}|\|_2^2 \\
& - \frac{1}{2}\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_2}| - |\boldsymbol{t}_{\sigma_2}|\|_2^2 - \sqrt{\frac{2}{\pi}}(\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu)^T \\
& (|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_2}| - |\boldsymbol{r}_\sigma \odot \boldsymbol{h}_{\sigma_1}| + |\boldsymbol{t}_{\sigma_1}| - |\boldsymbol{t}_{\sigma_2}|) \qquad (12)
\end{aligned}
$$

The first term of Eq.(10) is to measure the difference of the mean of $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $p_{\boldsymbol{t}}(\boldsymbol{x}_0)$, the second term is to measure the difference of the standard deviation of $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $p_{\boldsymbol{t}}(\boldsymbol{x}_0)$. If $\boldsymbol{h}_\sigma = \boldsymbol{r}_\sigma = \boldsymbol{t}_\sigma = \boldsymbol{1}$, Eq.(10) reduces to $f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = -\|\boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2$, which is the same as TransE, and the second term of Eq.(10) is equal to 0, i.e., the standard deviations of $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $q_{\boldsymbol{t}}(\boldsymbol{x}_0)$ are the same. If $\boldsymbol{h}_\mu = \boldsymbol{r}_\mu = \boldsymbol{t}_\mu = \boldsymbol{0}$, Eq.(10) reduces to $f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) = -\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\sigma| - |\boldsymbol{t}_\sigma|\|_2^2$. Eq.(11) and Eq.(12) are similar, the only difference is the coefficient of the fourth terms, one is $-\sqrt{\frac{3}{4}}$, the other is $-\sqrt{\frac{2}{\pi}}$. If $\boldsymbol{h}_{\sigma_1} = \boldsymbol{h}_{\sigma_2}$ and $\boldsymbol{t}_{\sigma_1} = \boldsymbol{t}_{\sigma_2}$, Eq.(11) or Eq.(12) reduces to Eq.(10).

If we choose $\boldsymbol{x}_0$ to be a random variable with Dirac delta distribution (i.e. a point vector), then Eq.(3) do not model uncertainty. Thus, NFE can be seen as a generalization of conventional models. We have the following proposition to illustrate this:

Proposition 5. *Let $k > 0$, if $\boldsymbol{x}_k \sim U[-\frac{\sqrt{3}}{k}, \frac{\sqrt{3}}{k}]^n$ or $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{0}, \frac{\boldsymbol{I}}{k^2})$, the invertible functions are Eq.(5) and similarity metric is Eq.(9), denote the scoring function as $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$, then $\boldsymbol{x}_k$ tends to a random variable with Dirac distribution as k tends to infinity and*

$$
\begin{aligned}
& \lim_{k\to\infty} f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}) \\
= & \lim_{k\to\infty} -\|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2 - \frac{1}{k^2}\||\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\sigma| - |\boldsymbol{t}_\sigma|\|_2^2 \\
= & - \|\boldsymbol{r}_\sigma \odot \boldsymbol{h}_\mu + \boldsymbol{r}_\mu - \boldsymbol{t}_\mu\|_2^2 \qquad (13)
\end{aligned}
$$

Proposition 5 shows that $1/k^2$ in $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$ reflects the degree to which $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$ focuses on uncertainty. Higher value of $1/k^2$ indicates $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$ focusing more on uncertainty. If $1/k^2 = 0$, the second term of $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$ is dropped. Thus, the second term of $f_k(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})$ or the second term of Eq.(10) is to model uncertainty. Eq.(11) or Eq.(12) has the similar result as Eq.(10). In conclusion, our proposed model is more general than conventional models.

Logical Rules

The inductive ability of a scoring function is reflected in its ability to learn logical rules (Zhang et al. 2021). The symmetry, antisymmetry, inverse and composition rules are defined as follows:

Symmetry Rules: A relation r is symmetric if $\forall h, t, (h, r, t) \in \mathcal{F} \to (t, r, h) \in \mathcal{F}$.

Antisymmetry Rules: A relation r is antisymmetric if $\forall h, t, (h, r, t) \in \mathcal{F} \to (t, r, h) \notin \mathcal{F}$.

Inverse Rules: A relation r_1 is inverse to a relation r_2 if $\forall h, t, (h, r_1, t) \in \mathcal{F} \to (t, r_2, h) \in \mathcal{F}$.

Composition Rules: A relation r_3 is the composition of a relation r_1 and a relation r_2 if $\forall h, \widetilde{t}, t, (h, r_1, \widetilde{t}) \in \mathcal{F} \wedge (\widetilde{t}, r_2, t) \in \mathcal{F} \to (h, r_3, t) \in \mathcal{F}$.

We have the following proposition about our proposed scoring functions and logical rules.

Proposition 6. *Scoring functions Eq.(10), Eq.(11) and Eq.(12) can learn symmetry, antisymmetry, inverse and composition rules.*

Loss Function

We use the same loss function, binary classification loss function with reciprocal learning, as in (Dettmers et al. 2018). For every triplet (h, r, t), our loss function is

$$\ell(h, r, t) = \sum_{t' \in \mathcal{E}} \log(1 + \exp(-y_{t'}(\gamma - f(h, r, t'))))$$

where γ is a fixed margin and $y_{t'} = 1$ if $t' = t$, otherwise $y_{t'} = -1$.

Discussion

Normalizing Flows We implement the invertible functions as linear functions or piecewise linear functions, which are spline functions. To construct more expressive invertible functions for better performance, piecewise-quadratic functions (Durkan et al. 2019) or cubic splines (Durkan et al. 2019) or even invertible neural networks (Huang et al. 2018) can be an option. In addition to designing invertible functions, Dinh et al. (2019) generalize Eq.(1) to piecewise invertible functions and Grathwohl et al. (2018) propose a continuous version of normalizing flows. This is also a potential research direction.

Normalizing Flows Embedding The similarity of two random variables can be measured by f-divergence (Gibbs and Su 2002) or Wasserstein distance. However, these metrics all involve computing a definite integral, which may have no closed form. This may limit the introduction of uncertainty. Since the 1-dimensional Wasserstein distance of two piecewise linear distributions always has a closed form, one solution is to approximate any distribution with a piecewise linear distribution (Petersen and Voigtlaender 2018). For example, we can approximate a normal distribution with a triangular distribution. In order to compute the Wasserstein distance efficiently, we decompose n-dimensional Wasserstein distance into a sum of 1-dimensional Wasserstein distance by Corollary 1, a sufficient condition of Proposition 1.

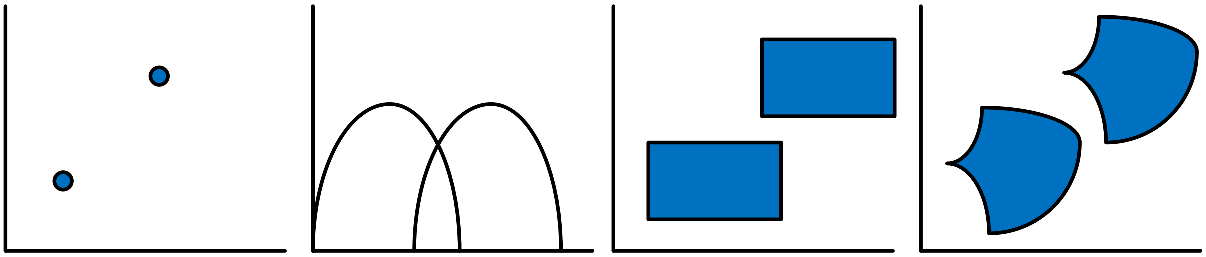

Figure 1: We define permutations α of different sets X with the same form, $\alpha(\boldsymbol{g})(\boldsymbol{x}) = \boldsymbol{g} + \boldsymbol{x}$, which translates one object to another. From left to right, the first corresponds to the translation of points, the second corresponds to the translation of random variables, the third corresponds to the translation of hyper-rectangles, and the fourth corresponds to the translation of manifolds.

We can design other invertible functions such that $p_{\boldsymbol{rh}}(\boldsymbol{x}_0)$ and $p_{\boldsymbol{t}}(\boldsymbol{x}_0)$ share the same copula. We choose the initial random variable $\boldsymbol{x}_0$ as a continuous random variable, $\boldsymbol{x}_0$ can also be a discrete distribution. How to choose a suitable $\boldsymbol{x}_0$ is also worth exploring.

Symmetric Groups The view of symmetric groups gives us a unified perspective of embedding. We unify embedding as permutations of a set. On one hand, we can easily parameterize a complex permutation and obtain a complex object by leveraging a permutation to act on a simple object. On the other hand, the group operation of symmetric groups, the composition of functions, is easy to compute. To introduce different properties, we can choose symmetric groups of different sets. For example, the set of points, random variables, hyper-rectangles (Abboud et al. 2020), manifolds (Xiao, Huang, and Zhu 2016a) and groups (Ebisu and Ichise 2018). Figure 1 shows an example.

Related Work

Group Embedding TorusE (Ebisu and Ichise 2018) defines embedding in an n-dimensional torus space which is a compact Lie group. MobiusE (Chen et al. 2021) embeds entities/relations to the surface of a Mobius ring. Cai et al. (2019) show that logical rules have natural correspondence to group representation theory. DihEdral (Xu and Li 2019) models the relations with the representation of dihedral group. NagE (Yang, Sha, and Hong 2020) finds the hidden group algebraic structure of relations and embeds relations as group elements. NFE embeds entities/relations as elements of a permutation group.

Uncertainty To model uncertainty in KGs, KG2E (He et al. 2015) represents entities/relations as random variables with multivariate normal distributions. TransG (Xiao, Huang, and Zhu 2016b) embeds entities as random variables with normal distributions and draws a mixture of normal distribution for relation embedding to handle multiple semantic issue. NFE introduces uncertainty by embedding entities/relations as permutations of a set of random variables.

Normalizing Flows Normalizing Flows should satisfy two conditions in order to be practical, the invertible function has tractable inverse and the determinant of Jacobian is easy to compute. A basic form of normalizing flows can be element-wise invertible functions. NICE (Dinh, Krueger, and Bengio 2014) and RealNVP (Dinh, Sohl-Dickstein, and Bengio 2016) utilize affine functions to construct coupling layers. Müller et al. (2019) propose a powerful generalization of affine functions, based on monotonic piecewise polynomials. Ziegler and Rush (2019) introduce an invertible non-linear squared function. Durkan et al. (2019) model the coupling function as a monotone rational-quadratic spline. Jaini, Selby, and Yu (2019) propose a strictly increasing polynomial and prove such polynomials can approximate any strictly monotonic univariate continuous function.

Experiments

In this section, we first introduce the experimental settings and compare NFE with existing models. We then show the effectiveness of introducing uncertainty. Finally, we conduct ablation studies. Please see Appendix D for more experimental details.

Experimental Settings

Datasets We evaluate our model on three popular knowledge graph completion datasets, WN18RR (Dettmers et al. 2018), FB15k-237 (Toutanova et al. 2015) and YAGO3-10 (Dettmers et al. 2018).

Evaluation Metric We use the filtered MR, MRR and Hits@N (H@N) (Bordes et al. 2013) as evaluation metrics and choose the hyper-parameters with the best filtered MRR on the validation set.

Baselines We compare the performance of NFE with several translational models, including TransE (Bordes et al. 2013), RotatE (Sun et al. 2018), bilinear models, including DistMult (Yang et al. 2014), ComplEx (Toutanova et al. 2015), QuatE (Zhang et al. 2019), TuckER (Balažević, Allen, and Hospedales 2019a), neural networks models, including ConvE (Dettmers et al. 2018), HypER (Balažević, Allen, and Hospedales 2019b), and group embedding models, including DihEdral (Xu and Li 2019), NagE (Yang, Sha, and Hong 2020).

	WN18RR			FB15k-237			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE	0.218	0.036	0.506	0.335	0.240	0.526	0.539	0.455	0.691
RotatE	0.476	0.428	0.571	0.338	0.241	0.533	0.495	0.402	0.670
DistMult	0.396	0.379	0.427	0.289	0.206	0.452	0.536	0.471	0.652
ComplEx	0.428	0.440	0.410	0.247	0.158	0.510	0.360	0.260	0.550
QuatE	0.481	0.436	0.564	0.311	0.221	0.495	—	—	—
TuckER	0.470	0.443	0.526	**0.358**	**0.266**	**0.544**	—	—	—
ConvE	0.430	0.440	0.520	0.325	0.237	0.501	0.440	0.350	0.620
HypER	0.465	0.436	0.522	0.341	0.252	0.520	0.533	0.455	0.678
DihEdral	0.480	**0.452**	0.536	0.325	0.237	0.501	0.440	0.350	0.620
NagE	0.477	0.432	0.574	0.340	0.244	0.530	—	—	—
NFE-2	0.476	0.431	0.569	0.352	0.256	0.542	0.563	0.489	**0.699**
NFE-1	**0.483**	0.438	**0.576**	0.355	0.261	0.543	**0.570**	**0.498**	0.697
NFE-w/o-uncertainty	0.475	0.430	0.568	0.352	0.257	0.542	0.563	0.490	0.693
NFE-1-μ	0.218	0.036	0.506	0.335	0.240	0.526	0.539	0.455	0.691
NFE-1-σ	0.286	0.126	0.522	0.342	0.247	0.531	0.519	0.429	0.676

Table 1: Knowledge graph completion results on WN18RR, FB15k-237 and YGAO3-10 datasets.

$1/k^2$	0	1/16	1/8	1/4	1/2	1	2	4	8	16
MRR	0.475	0.478	0.478	0.478	0.479	0.483	0.483	0.483	0.482	0.482
H@1	0.430	0.433	0.433	0.433	0.434	0.438	0.438	0.436	0.436	0.436
H@10	0.568	0.572	0.573	0.571	0.573	0.576	0.575	0.575	0.572	0.572

Table 2: The performance of NFE-1 with different $1/k^2$ on WN18RR dataset. Higher value of $1/k^2$ indicates the model focusing more on uncertainty.

Results

Due to the great generality, our proposed NFE is able to have different instantiations, Eq.(10) and Eq.(11) and Eq.(12). We denote Eq.(10) as NFE-1. Eq.(11) and Eq.(12) achieve almost same result, we denote them as NFE-2. We compare the performance of NFE-1 and NFE-2 with baseline models. See Table 1 for the results. NFE-1 and NFE-2 achieve state-of-the-art performance on three datasets, especially on YAGO3-10, which is the largest dataset. NFE-2 is slightly inferior to NFE-1. Although NFE-2 is more expressive than NFE-1, NFE-2 may be more difficult to optimize. NFE-1 are derived from TransE and DistMult, but NFE-1 outperforms TransE and DistMult significantly on all metrics on three datasets. NFE-1 is better than neural networks models, ConvE and HypER, and group embedding models, DihEdral and NagE. The results show the effectiveness of our model.

Uncertainty

Here we focus on the NFE-w/o-uncertainty in Table 1 and Table 2. Proposition 5 shows that NFE-1 can be seen as a generalization of existing models to model uncertainty. NFE-1 can reduces to Eq.(13), which do not model uncertainty. We denote Eq.(13) as NFE -w/o-uncertainty. Table 1 shows that NFE-1 achieves better performance than NFE-w/o-uncertainty on all metrics on three datasets. Thus, the results show the effectiveness of introducing uncertainty.

Proposition 5 shows that $1/k^2$ in Eq.(13) reflects the degree to which Eq.(13) focuses on uncertainty. Higher value of $1/k^2$ indicates the model focusing more on uncertainty. Table 2 shows the performance of Eq.(13) with different $1/k^2$ on WN18RR dataset. The results show that the performance of Eq.(13) generally gets worse as $1/k^2$ gets smaller. This also shows the effectiveness of introducing uncertainty.

Ablation Study

The invertible functions of NFE-1 is Eq.(4), the composition of function of TransE and function of DistMult. We conduct ablation studies to analyze the performance of NFE-1 only using one of the functions. We denote Eq.(10) using function of TransE as NFE-1-μ, Eq.(10) using function of DistMult as NFE-1-σ. Table 1 shows that the performance of NFE-1 is better than NFE-1-μ and NFE-1-σ. The reason is that NFE-1 is more expressive than NFE-1-μ and NFE-1-σ.

Conclusion

We propose a unified perspective of embedding and introduce uncertainty into KGE from the view of group theory. We embed entities/relations as elements of a symmetric group to introduce uncertainty. Based on the embedding, NFE is defined by measuring the similarity of two normalizing flows. Experiments verify the effectiveness of NFE.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2020AAA0106000), the National Natural Science Foundation of China (U19A2079), and the CCCD Key Lab of Ministry of Culture and Tourism.

References

Abboud, R.; Ceylan, I.; Lukasiewicz, T.; and Salvatori, T. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33: 9649–9661.

Balažević, I.; Allen, C.; and Hospedales, T. 2019a. 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)*, 5185–5194.

Balažević, I.; Allen, C.; and Hospedales, T. M. 2019b. Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*, 553–565. Springer.

Bertsekas, D. P.; and Tsitsiklis, J. N. 2008. *Introduction to probability*, volume 1. Athena Scientific.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Cai, C.; Cai, Y.; Sun, M.; and Xu, Z. 2019. Group representation theory for knowledge graph embedding. *arXiv preprint arXiv:1909.05100*.

Chami, I.; Wolf, A.; Juan, D.-C.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6901–6914.

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

Cuestaalbertos, J. A.; Ruschendorf, L.; and Tuerodiaz, A. 1993. Optimal coupling of multivariate distributions and stochastic processes. *Journal of Multivariate Analysis*, 46(2): 335–361.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Dinh, L.; Sohl-Dickstein, J.; Larochelle, H.; and Pascanu, R. 2019. A RAD approach to deep mixture models. *arXiv preprint arXiv:1903.07714*.

Durkan, C.; Bekasov, A.; Murray, I.; and Papamakarios, G. 2019. Neural spline flows. *Advances in neural information processing systems*, 32.

Ebisu, T.; and Ichise, R. 2018. Toruse: Knowledge graph embedding on a lie group. In *Thirty-second AAAI conference on artificial intelligence*.

Gibbs, A. L.; and Su, F. E. 2002. On choosing and bounding probability metrics. *International statistical review*, 70(3): 419–435.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press.

Grathwohl, W.; Chen, R. T.; Bettencourt, J.; Sutskever, I.; and Duvenaud, D. 2018. FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models. In *International Conference on Learning Representations*.

He, S.; Liu, K.; Ji, G.; and Zhao, J. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, 623–632.

Huang, C.-W.; Krueger, D.; Lacoste, A.; and Courville, A. 2018. Neural autoregressive flows. In *International Conference on Machine Learning*, 2078–2087. PMLR.

Hungerford, T. W. 2012. *Abstract algebra: an introduction*. Cengage Learning.

Jaini, P.; Selby, K. A.; and Yu, Y. 2019. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, 3009–3018. PMLR.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kobyzev, I.; Prince, S. J.; and Brubaker, M. A. 2020. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3964–3979.

Müller, T.; McWilliams, B.; Rousselle, F.; Gross, M.; and Novák, J. 2019. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5): 1–19.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57): 1–64.

Petersen, P.; and Voigtlaender, F. 2018. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108: 296–330.

Peyré, G.; Cuturi, M.; et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, 1278–1286. PMLR.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2018. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; and Gamon, M. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1499–1509.

Xiao, H.; Huang, M.; and Zhu, X. 2016a. From one point to a manifold: knowledge graph embedding for precise link prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 1315–1321.

Xiao, H.; Huang, M.; and Zhu, X. 2016b. TransG: A Generative Model for Knowledge Graph Embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2316–2325.

Xu, C.; and Li, R. 2019. Relation Embedding with Dihedral Group in Knowledge Graph. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 263–272.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv e-prints*, arXiv–1412.

Yang, T.; Sha, L.; and Hong, P. 2020. NagE: Non-abelian group embedding for knowledge graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1735–1742.

Zhang, J.; Chen, B.; Zhang, L.; Ke, X.; and Ding, H. 2021. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2: 14–35.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2735–2745.

Ziegler, Z.; and Rush, A. 2019. Latent normalizing flows for discrete sequences. In *International Conference on Machine Learning*, 7673–7682. PMLR.