# RelWalk – A Latent Variable Model Approach to Knowledge Graph Embedding

**Danushka Bollegala**[1]*, **Huda Hakami**[2], **Yuichi Yoshida**[3] and **Ken-ichi Kawarabayashi**[3]
[1]University of Liverpool, Amazon. danushka@liverpool.ac.uk
[2]Taif University hahakami@tu.edu.sa
[3]National Institute of Informatics {yyoshida,k_keniti}@nii.ac.jp

## Abstract

Embedding entities and relations of a knowledge graph in a low-dimensional space has shown impressive performance in predicting missing links between entities. Although progresses have been achieved, existing methods are heuristically motivated and theoretical understanding of such embeddings is comparatively underdeveloped. This paper extends the random walk model (Arora et al., 2016a) of word embeddings to Knowledge Graph Embeddings (KGEs) to derive a scoring function that evaluates the strength of a relation $R$ between two entities $h$ (head) and $t$ (tail). Moreover, we show that marginal loss minimisation, a popular objective used in much prior work in KGE, follows naturally from the log-likelihood ratio maximisation under the probabilities estimated from the KGEs according to our theoretical relationship. We propose a learning objective motivated by the theoretical analysis to learn KGEs from a given knowledge graph. Using the derived objective, accurate KGEs are learnt from FB15K237 and WN18RR benchmark datasets, providing empirical evidence in support of the theory.

## 1 Introduction

Knowedge graphs (KGs) such as Freebase (Bollacker et al., 2008) organise information in the form of graphs, where entities are represented by the vertices and the relations between two entities are represented by the edges that connect the corresponding vertices. Despite the best efforts to create complete and large-scale KGs, most KGs remain incomplete and do not represent all the relations that exist between entities (Min et al., 2013). In particular, new entities are constantly being generated, and new relations are formed between new as well as existing entities. Therefore, it is unrealistic to assume that a real-world KG would be complete at any given time point. Developing approaches for KG completion is an important research field associated with KGs.

KG components can be embedded into numerical formats by learning representations (a.k.a embeddings) for the entities and relations in a given KG. The learnt KGEs can be used for *link prediction*, which is the task of predicting whether a particular relation exists between two given entities in the KG. Specifically, given KGEs for entities and relations, in link prediction, we predict $R$ that is most likely to exist between $h$ and $t$ according to some scoring formula. Thus, by embedding entities and relations that exist in a KG in some (possibly lower-dimensional and latent) space, we can infer previously unseen relations between entities, thereby expanding a given KG.

KGE can be seen as a two-step process. Given a KG represented by a set of relational triples $(h, R, t)$, where a semantic relation $R$ holds between a head entity $h$ and a tail entity $t$, first a scoring function is defined that measures the *relational strength* of a triple $(h, R, t)$. Second,the entity and relation embeddings that optimise the defined scoring function are learnt using some optimisation method. Despite the wide applications of entity and relation embeddings created via KGE methods, the existing scoring functions are heuristically motivated to capture some geometric requirements of the embedding space. For example, TransE (Bordes et al., 2011) assumes that the entity and relation embeddings co-exist in the same (possibly lower dimensional) vector space and translating (shifting) the head entity embedding by the relation embedding must make it closer to the tail entity embedding, whereas ComplEx (Trouillon et al., 2016) models the asymmetry in relations using the component-wise multi-linear inner-product among

*Danushka Bollegala holds concurrent appointments as a Professor at University of Liverpool and as an Amazon Scholar. This paper describes work performed at the University of Liverpool and is not associated with Amazon.

entity and relation embeddings.

Theoretical understanding of KGE methods is under developed. For example, it is not clear how the heuristically defined KGE objectives relate to the generative process of a KG. Providing such a theoretical understanding of the KGE process will enable us to develop KGE methods that address the weaknesses in the existing KGE methods. For this purpose, we propose Relational Walk (**RelWalk**), a theoretically motivated generative approach for learning KGEs. We are particularly interested in the semantic relationships that exist between entities such as the is-CEO-of relation between a person such as **Jeff Bezos** and a company such as **Amazon Inc.**

We model KGE as a random walk over the KG. Specifically, a random walker at the vertex corresponding to the (head) entity $h$ will uniformly at random select one of the outgoing edges corresponding to the semantic relation $R$, which will lead it to the vertex corresponding to the (tail) entity $t$. Continuing this random walk will result in a traversal over a path in the KG. Based on this random walk model we derive a relationship between the probability of $R$ holding between $h$ and $t$, $p(h, t \mid R)$, and their KGEs $\mathbf{R}$, $\boldsymbol{h}$ and $\boldsymbol{t}$. Interestingly, the derived relationship is not covered by any of the previously proposed heuristically-motivated scoring functions, providing the first-ever KGE method with a provable generative explanation.

We show that the *margin loss*, a popular training objective in prior work on KGE, naturally emerges as the log-likelihood ratio computed from the derived $p(h, t \mid R)$. Based on this result, we derive a training objective that is optimised for learning KGEs that satisfy our theoretical relationship. This enables us to empirically verify the theoretical relationships that we derived from the proposed random walk process.

Using FB15K237 and WN18RR benchmarks, we evaluate the learnt KGEs on link prediction and triple classification. Although we do not obtain state-of-the-art (SoTA) performance on these benchmark datasets, KGEs learnt using RelWalk perform consistently well on both tasks, providing empirical support to the theoretical analysis conducted in this paper. We re-emphasise that our main objective in this paper is to study KGEs from an interpretable theoretical perspective and not necessarily improving SoTA. To this end, we study the relationship between the concentration of the partition function as predicted by our theoretical analysis and the performance of the learnt KGEs. We observe that when the partition function is narrowly distributed, we are able to learn accurate KGEs. Moreover, we empirically verify that the learnt relation embedding matrices satisfy the orthogonality property as expected by the theoretical analysis.

## 2 Related Work

At a high-level of abstraction, KGE methods can be seen as differing in their design choices for the following two main problems: (a) how to represent entities and relations, and (b) how to model the interaction between two entities and a relation that holds between them. Next, we briefly discuss prior proposals to those two problems (refer to Wang et al. (2017); Nguyen (2017); Nickel et al. (2015) for an extended survey on KGE).

A popular choice for representing entities is to use vectors (Bordes et al., 2013; Ji et al., 2015; Yang et al., 2015), whereas relations have been represented by vectors, matrices (Bordes et al., 2011; Nguyen et al., 2016; Nickel et al., 2011) or tensors (Socher et al., 2013). ComplEx (Trouillon et al., 2016) introduced complex vectors for KGEs to capture the asymmetry in semantic relations. Ding et al. (2018) further improved CompIEx by imposing non-negativity and entailment constraints to ComplEx.

Given entity and relation embeddings, a scoring function evaluates the strength of a triple $(h, R, t)$. Scoring functions that encode various intuitions have been proposed such as the $\ell_1$ or $\ell_2$ norms of the vector formed by a translation of the head entity embedding by the relation embedding over the target embedding, or by first performing a projection from the entity embedding space to the relation embedding space (Yoon et al., 2016). As an alternative to using vector norms as scoring functions, DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) use the component-wise multi-linear dot product. Lacroix et al. (2018) proposed the use of nuclear 3-norm regularisers instead of the popular Frobenius norm for canonical tensor decomposition. Table 1 shows the scoring functions along with algebraic structures for entities and relations proposed in selected prior work in KGE learning. Given a scoring function, KGEs are learnt that assign higher scores to relational triples in existing KGs over triples where the relation does not

hold (negative triples) by minimising a loss function such as the logistic loss (RESCAL, DistMult, ComplEx) or marginal loss (TransE).

Alternatively to directly learning embeddings from a graph, several methods (Grover and Leskovec, 2016; Perozzi et al., 2014; Ristoski et al., 2018) have considered the vertices visited during truncated random walks over the graph as *pseudo sentences*, and have applied popular word embedding learning algorithms such as continuous bag-of-words model (Mikolov et al., 2013) to learn vertex embeddings. However, pseudo sentences generated in this manner are syntactically very different from sentences in natural languages.

On the other hand, our work extends the random walk analysis by Arora et al. (2016a) that derives a useful connection between the joint co-occurrence probability of two words and the $\ell_2$ norm of the sum of the corresponding word embeddings. Specifically, they proposed a latent variable model where the words in a corpus are generated by a probabilistic model parametrised by a time-dependent discourse vector that performs a random walk. In contrast to Arora's model that uses co-occurrences as a generic relation, in our work we include relations as labels for the edges in the graph. Bollegala et al. (2018) extended the model proposed by Arora et al. (2016a) to capture co-occurrences involving more than two words. Specifically, they defined the co-occurrence of $k$ unique words in a given context as a $k$-way co-occurrence, where Arora et al. (2016a) result could be seen as a special case corresponding to $k = 2$. Moreover, it has been shown that it is possible to learn word embeddings that capture some types of semantic relations such as antonymy and collocation using 3-way co-occurrences more accurately than using 2-way co-occurrences. However, that model does not explicitly consider the relations between words/entities and uses only a corpus for learning the word embeddings.

## 3 Relational Walk

Let us consider a KG, $\mathcal{D}$, where the *knowledge* is represented by relational triples $(h, R, t) \in \mathcal{D}$. Here, $R$ is a relational predicate with two arguments, where $h$ (*head*) and $t$ (*tail*) entities respectively filling the first and second arguments. In this work, we assume relations to be asymmetric in general (if $(h, R, t) \in \mathcal{D}$ then it does not necessarily follow that $(t, R, h) \in \mathcal{D}$). The goal of KGE is to learn embeddings for the relations and entities in the KG such that the entities that participate in similar relations are embedded closely to each other in the entity embedding space, while at the same time relations that hold between similar entities are embedded closely to each other in the relational embedding space. We call the learnt entity and relation embeddings collectively as KGEs. We assume that entities and relations are embedded in the same vector space, allowing us to perform linear algebraic operations using the embeddings in the same vector space.

| KGE method | Score function $f(h, R, t)$ | Relation parameters |
|---|---|---|
| Unstructured (Bordes et al., 2012) | $\|\|\boldsymbol{h} - \boldsymbol{t}\|\|_{\ell_{1/2}}$ | none |
| Structured (Bordes et al., 2011) | $\|\|\mathbf{R}_1\boldsymbol{h} - \mathbf{R}_2\boldsymbol{t}\|\|_{\ell_{1,2}}$ | $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{d\times d}$ |
| TransE (Bordes et al., 2013) | $\|\|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|\|_{\ell_{1/2}}$ | $\boldsymbol{r} \in \mathbb{R}^d$ |
| DistMult (Yang et al., 2015) | $\langle \boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \rangle$ | $\boldsymbol{r} \in \mathbb{R}^d$ |
| RESCAL (Nickel et al., 2011) | $\boldsymbol{h}^\top \mathbf{R} \boldsymbol{t}$ | $\mathbf{R} \in \mathbb{R}^{d\times d}$ |
| ComplEx (Trouillon et al., 2016) | $\langle \boldsymbol{h}, \boldsymbol{r}, \bar{\boldsymbol{t}} \rangle$ | $\boldsymbol{r} \in \mathbb{C}^d$ |

Table 1: Score functions proposed in selected prior work on KGEs. Entity embeddings $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$ are vectors in all models, except in ComplEx where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{C}^d$. Here, $\ell_{1/2}$ denotes either $\ell_1$ or $\ell_2$ norm of a vector. In ComplEx, $\bar{\boldsymbol{t}}$ is the element-wise complex conjugate.

Following our aforementioned modelling of a knowledge base as a graph, let us consider a random walker who is at a vertex corresponding to some entity $h$. This entity will have one or more semantic relations with other entities in the KG. The random walker will uniformly at random pick one of the outgoing edges corresponding to a particular semantic relation $R$, and follow it to land on the entity $t$. This one-step of the random walk thus *generates* a tuple $(h, R, t)$ in the KG. The random walker proceeds by using $t$ as the new starting point. Multiple steps of this random walk trace a single *path* in the KG.

To illustrate a random walk over a KG, let us assume that we are currently at the vertex corresponding to the company entity **Amazon Inc.** Possible outgoing edges at **Amazon Inc.** would correspond to semantic relations such as has-ceo, is-headquarted-at, founded-in etc., where **Amazon Inc.** is the head entity. If there are only three such outgoing relations at **Amazon Inc.**, then the random walker will pick any one of those relations with a probability $1/3$. For example, by selecting has-ceo, is-headquarted-at or founded-in the random walker would arrive at entities respectively **Jeff Bezos**, **Seattle** or **1994**. Let us assume that the

random walker selected the has-ceo relation and landed at **Jeff Bezos**. The random walker might subsequently continue its random walk from **Jeff Bezos** following the relation born-in and transiting to **New Mexico, US**. Prior work studying inferences in KGs have successfully used random walk models similar to what we describe here (Gardner et al., 2013; Lao et al., 2012, 2011; Lao and Cohen, 2010).

Let us consider a random walk characterised by a time-dependent *knowledge vector* $\boldsymbol{c}_k$, where $k$ is the current time step. The knowledge vector represents the knowledge we have about a particular group of entities and relations that express some facts about the world. For example, when we are talking about **Amazon Inc.**, we will use the knowledge associated with **Amazon Inc.** such as its CEO, location of the headquarters, when it was founded etc. Therefore, it is intuitive to assume that the entities associated with **Amazon Inc.** with some set of semantic relations can be generated from this knowledge vector. Each entity and relation has time-independent latent representations that capture their correlations with $\boldsymbol{c}_k$. For entities $h$ and $t$, we denote their representations by $d$-dimensional vectors respectively $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$.

We assume the task of generating a relational triple $(h, R, t)$ in a given KG to be a two-step process as described next. First, given the current knowledge vector at time $k$, $\boldsymbol{c} = \boldsymbol{c}_k$ and the relation $R$, we assume that the probability of an entity $h$ satisfying the first argument of $R$ to be given by the loglinear entity production model in (1).

$$p(h \mid R, \boldsymbol{c}) = \frac{1}{Z_c} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right). \tag{1}$$

Here, $\mathbf{R}_1 \in \mathbb{R}^{d \times d}$ is a relation-specific orthogonal matrix that evaluates the appropriateness of $h$ for the first argument of $R$. For example, if $R$ is the is-ceo-of relation, we would require a person as the first argument and a company as the second argument of $R$. However, note that the role of $\mathbf{R}_1$ extends beyond simply checking the types of the entities that can fill the first argument of a relation. For our example above, not all people are CEOs and $\mathbf{R}_1$ evaluates the likelihood of a person to be selected as the first argument of the ceo-of relation. $Z_c$ is a normalisation coefficient such that $\sum_{h \in \mathcal{V}} p(h \mid R, \boldsymbol{c}) = 1$, where the vocabulary $\mathcal{V}$ is the set of all entities in the KG.

After generating $h$, the state of our random walker changes to $\boldsymbol{c}' = \boldsymbol{c}_{k+1}$, and we next generate the second argument of $R$ with the probability given by (2).

$$p(t \mid R, \boldsymbol{c}') = \frac{1}{Z_{c'}} \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right). \tag{2}$$

Here, $\mathbf{R}_2 \in \mathbb{R}^{d \times d}$ is a relation-specific orthogonal matrix that evaluates the appropriateness of $t$ as the second argument of $R$. $Z_{c'}$ is a normalisation coefficient such that $\sum_{t \in \mathcal{V}} p(t \mid R, \boldsymbol{c}') = 1$. Following our previous example of is-ceo-of relation, $\mathbf{R}_2$ evaluates the likelihood of an organisation to be a company with a CEO position. Importantly, $\mathbf{R}_1$ and $\mathbf{R}_2$ are representations of the relation $R$ and independent of the entities. Therefore, we consider ($\mathbf{R}_1$ and $\mathbf{R}_2$) to collectively represent the embedding of $R$. Orthogonality of $\mathbf{R}_1, \mathbf{R}_2$ is a requirement for the mathematical proof and also acts as a regularisation constraint to prevent overfitting by restricting the relational embedding space (Tang et al., 2020). Intuitively, orthogonality of the relation embedding matrices ensures that the length of the head and tail entity embeddings are not altered during the generation of the tuple. Orthogonal transformation has been shown to improve the performance of relation representation in prior work. For example, Tang et al. (2020) apply orthogonal transformation that extends RotatE (Sun et al., 2019) to model complex relations (e.g., N-to-N). For relationships in word embedding space, Ethayarajh (2019) use orthogonal transformation for analogical reasoning between words. The performance of hypernymy prediction through orthogonal projections has been improved as shown in Wang et al. (2019).

The knowledge vector $\boldsymbol{c}_k$ performs a *slow* random walk (meaning $\boldsymbol{c}_{k+1}$ is obtained from $\boldsymbol{c}_k$ by adding a small random displacement vector) such that the head and tail entities of a relation are generated under similar knowledge vectors. More specifically, we assume that $||\boldsymbol{c}_k - \boldsymbol{c}_{k+1}|| \leq \epsilon_2$ for some small $\epsilon_2 > 0$. This is a realistic assumption for generating the two entity arguments in the same relational triple because, if the knowledge vectors were significantly different in the two generation steps, then it is likely that the corresponding relations are also different, which would not be coherent with the above-described generative process. Moreover, we assume that the knowledge vectors are distributed uniformly in the unit sphere and denote the distribution of knowledge vectors by $\mathcal{C}$.

To relate KGEs with the connections in the graph, we must estimate the probability that $h$ and $t$ satisfy the relation $R$, $p(h, t \mid R)$, which can be ob-

tained by taking the expectation of $p(h, t \mid R, \boldsymbol{c}, \boldsymbol{c}')$ w.r.t. $\boldsymbol{c}, \boldsymbol{c}' \sim \mathcal{C}$ given by (3).

$$p(h, t \mid R) = \mathbb{E}_{\boldsymbol{c},\boldsymbol{c}'}\left[p(h, t \mid R, \boldsymbol{c}, \boldsymbol{c}')\right] \tag{3}$$
$$= \mathbb{E}_{\boldsymbol{c},\boldsymbol{c}'}\left[p(h \mid R, \boldsymbol{c}) p(t \mid R, \boldsymbol{c}')\right] \tag{4}$$
$$= \mathbb{E}_{\boldsymbol{c},\boldsymbol{c}'}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right)}{Z_{c'}}\right]. \tag{5}$$

Here, partition functions are given by

$$Z_c = \sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \tag{6}$$
$$Z_{c'} = \sum_{t \in \mathcal{V}} \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right) \tag{7}$$

(4) follows from our two-step generative process where the generation of $h$ and $t$ in each step is independent given the relation and the corresponding knowledge vectors.

Computing the expectation in (5) is generally difficult because of the two partition functions $Z_c$ and $Z_{c'}$. However, Lemma 1 shows that the partition functions are narrowly distributed around a constant value for all $c$ (or $c'$) values with high probability.

**Lemma 1 (Concentration Lemma).** *If the entity embedding vectors satisfy the Bayesian prior $\boldsymbol{v} = s\hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{v}}$ is from the spherical Gaussian distribution, and $s$ is a scalar random variable, which is always bounded by a constant $\kappa$, then the entire ensemble of entity embeddings satisfies that:*

$$\Pr_{c \sim \mathcal{C}}[(1-\epsilon_z)Z \leq Z_c \leq (1+\epsilon_z)Z] \geq 1 - \delta, \tag{8}$$

*for $\epsilon_z = O(1/\sqrt{n})$, and $\delta = \exp(-\Omega(\log^2 n))$, where $n \geq d$ is the number of entities in a given KG and $Z_c$ is the partition function for $c$ given by $\sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)$.*

Refer to Appendix A for the proof of the concentration lemma. We empirically investigate the relationship between the performance of the KGEs and the degree to which Lemma 1 is satisfied in subsection 5.1. Under the conditions required to satisfy Lemma 1, the following main theorem of this paper holds:

**Theorem 1.** *Suppose that the entity embeddings satisfy (1). Then, we have*

$$\log p(h, t \mid R) = \frac{\left|\left|{\mathbf{R}_1}^\top \boldsymbol{h} + {\mathbf{R}_2}^\top \boldsymbol{t}\right|\right|_2^2}{2d} - 2\log Z \pm \epsilon. \tag{9}$$

*for $\epsilon = O(1/\sqrt{n}) + \widetilde{O}(1/d)$, where*

$$Z = Z_c = Z_{c'}. \tag{10}$$

*Proof sketch:* Let $F$ be the event that both $c$ and $c'$ are within $(1 \pm \epsilon_z)Z$. Then, from Lemma 1 and the union bound, event $F$ happens with probability at least $1 - 2\exp(-\Omega(\log^2 n))$. The R.H.S. of (5) can be split into two parts $T_1$ and $T_2$ according to whether $F$ happens or not.

$$p(h, t \mid R) = \underbrace{\mathbb{E}_{c,c'}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_2 \boldsymbol{c}'\right)}{Z_{c'}} \mathbf{1}_F\right]}_{=T_1} + \underbrace{\mathbb{E}_{c,c'}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_2 \boldsymbol{c}'\right)}{Z_{c'}} \mathbf{1}_{\bar{F}}\right]}_{=T_2}. \tag{11}$$

$T_1$ can be approximated as given by (12).

$$T_1 = \frac{1 \pm \mathcal{O}(\epsilon_z)}{Z^2} \mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right)\right] \tag{12}$$

On the other hand, $T_2$ can be shown to be a constant, independent of $d$, given by (13).

$$|T_2| = \exp(-\Omega(\log^{1.8} n)) \tag{13}$$

The vocabulary size $n$ of real-world KGs is typically over $10^5$, for which $T_2$ becomes negligibly small. Therefore, it suffices to consider only $T_1$. Because of the slowness of the random walk we have $\boldsymbol{c} \approx \boldsymbol{c}'$.

Using the law of total expectation we can write $T_1$ as follows:

$$T_1 = \frac{1 \pm \mathcal{O}(\epsilon_z)}{Z^2} \mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right)\right]\right]$$
$$= \frac{1 \pm \mathcal{O}(\epsilon_z)}{Z^2} \mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) A(c)\right] \tag{14}$$

where $A(c) := \mathbb{E}_{c'|c}\left[\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}'\right)\right]$. Doing some further evaluations we show that

$$A(c) = (1 \pm \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \tag{15}$$

Plugging (51) back in (14) provides the claim of the theorem. Detailed proof is shown in Appendix B. □

The relationship given by (9) indicates that head and tail entity embeddings are first transformed respectively by ${\mathbf{R}_1}^\top$ and ${\mathbf{R}_2}^\top$, and the squared $\ell_2$ norm of the sum of the transformed vectors is proportional to the probability $p(h, t \mid R)$.

## 4 Learning KG Embeddings

In this section, we derive a training objective from Theorem 1 that we can then optimise to learn KGEs. The goal is to empirically validate the theoretical result by evaluating the learnt KGEs. KGs represent

information about relations between two entities in the form of *relational triples*. The joint probability $p(h, R, t)$ given by Theorem 1 is useful for determining whether a relation $R$ exists between two given entities $h$ and $t$. For example, if we know that with a high probability that $R$ holds between $h$ and $t$, then we can append $(h, R, t)$ to the KG. The task of expanding KGs by predicting missing links between entities or relations is known as the *link prediction* problem (Trouillon et al., 2016). In particular, if we can automatically append such previously unknown knowledge to the KG, we can expand the KG and address the knowledge acquisition bottleneck.

To derive a criteria for determining whether a link must be predicted among entities and relations, let us consider a relational triple $(h, R, t) \in \mathcal{D}$ that exists in a given KG $\mathcal{D}$. We call such relational triples as *positive* triples because from the assumption it is known that $R$ holds between $h$ and $t$. On the other hand, consider a *negative* relational triple $(h', R, t') \in \mathcal{D}$ formed by, for example, randomly perturbing a positive triple. A popular technique for generating such (pseudo) negative triples is to replace $h$ or $t$ with a randomly selected different instance of the same entity type. As an alternative for random perturbation, Cai and Wang (2018) proposed a method for generating negative instances using adversarial learning. Here, we are not concerned about the actual method used for generating the negative triples but assume a set of negative triples, $\bar{\mathcal{D}}$, generated using some method, to be given.

Given a positive triple $(h, R, t) \in \mathcal{D}$ and a negative triple $(h', R, t') \in \bar{\mathcal{D}}$, we would like to learn KGEs such that a higher probability is assigned to $(h, R, t)$ than that assigned to $(h', R, t')$. We can formalise this requirement using the likelihood ratio given by (16).

$$\frac{p(h, R, t)}{p(h', R, t')} \geq \eta \tag{16}$$

Here, $\eta > 1$ is a threshold that determines how higher we would like to set the probabilities for the positive triples compares to that of the negative triples.

By taking the logarithm of both sides in (16) we obtain

$$\begin{aligned} \log p(h, R, t) - \log p(h', R, t') &\geq \log \eta \\ \log \eta + \log p(h', R, t') - \log p(h, R, t) &\leq 0 \end{aligned} \tag{17}$$

If a positive triple $(h, R, t)$ is correctly assigned a higher probability than a negative triple $p(h', R, t')$, then the left hand side of (17) will be negative, indicating that there is no *loss* incurred during this classification task. Therefore, we can re-write (17) to obtain the *marginal loss* (Bordes et al., 2013, 2011), $L(\mathcal{D}, \bar{\mathcal{D}})$, a popular choice as a learning objective in prior work in KGE, as shown in (18).

$$\begin{aligned} &L(\mathcal{D}, \bar{\mathcal{D}}) = \\ &\sum_{\substack{(h,R,t)\in\mathcal{D} \\ (h',R,t')\in\bar{\mathcal{D}}}} \max\left(0, \log\eta + \log p(h', R, t') - \log p(h, R, t)\right) \\ &= \max\Big(0, 2d\log\eta + \left|\left|{\mathbf{R}_1}^\top \boldsymbol{h}' + {\mathbf{R}_2}^\top \boldsymbol{t}'\right|\right|_2^2 \\ &\quad - \left|\left|{\mathbf{R}_1}^\top \boldsymbol{h} + {\mathbf{R}_2}^\top \boldsymbol{t}\right|\right|_2^2\Big) \end{aligned} \tag{18}$$

We can assume $2d \log \eta$ to be the *margin* for the constraint violation.

Theorem 1 requires $\mathbf{R}_1$ and $\mathbf{R}_2$ to be orthogonal. To reflect this requirement, we add two $\ell_2$ regularisation terms $\left|\left|{\mathbf{R}_1}^\top \mathbf{R}_1 - \mathbf{I}\right|\right|_2^2$ and $\left|\left|{\mathbf{R}_2}^\top \mathbf{R}_2 - \mathbf{I}\right|\right|_2^2$ respectively with regularisation coefficients $\lambda_1$ and $\lambda_2$ to the objective function given by (18). In our experiments, we compute the gradients (18) w.r.t. each of the parameters $\boldsymbol{h}$, $\boldsymbol{t}$, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ and use stochastic gradient descent (SGD) for optimisation. Considering that negative triples are generated via random perturbation, it is important to consider multiple negative triples during training to better estimate the classification boundary. This approach can be easily extended to learn from multiple negative triples as shown in Appendix C.

## 5 Empirical validation

To empirically evaluate the theoretical result stated in Theorem 1, we learn KGEs (denoted by **RelWalk**) by minimising the marginal loss objective derived in section 4. We use the FB15k237, FB13 (subsets of *Freebase*) and WN18RR (a subset of *WordNet*) datasets, which are standard benchmarks for KGE. We use the standard training, validation and test splits. Statistics about the datasets and training details are in Appendix D. **RelWalk** is implemented in the open-source toolkit OpenKE (Han et al., 2018) and the code and learnt KGEs will are publicly available[1].

We conduct two evaluation tasks: *link prediction* (predict the missing head or tail entity in a given triple $(h, R, ?)$ or $(?, R, t)$) (Bordes et al., 2011) and *triple classification* (predict whether a relation $R$ holds between $h$ and $t$ in a given triple

[1] https://github.com/LivNLP/Relational-Walk-for-Knowledge-Graphs

| Method | FB15K237 MRR | MR | H@1 | H@3 | H@10 | WN18RR MRR | MR | H@1 | H@3 | H@10 |
|---|---|---|---|---|---|---|---|---|---|---|
| TransE$^{\bullet}$ | 0.294 | 347 | - | - | 0.465 | 0.226 | 3384 | - | - | 0.50 |
| TransD$^{\triangleleft}$ | 0.280 | - | - | - | 0.453 | - | - | - | - | 0.43 |
| DistMult$^{\star}$ | 0.241 | 254 | 0.155 | 0.263 | 0.419 | 0.430 | 5110 | 0.39 | 0.44 | 0.49 |
| ComplEx$^{\star}$ | 0.247 | 339 | 0.158 | 0.275 | 0.428 | 0.440 | 5261 | 0.41 | 0.46 | 0.51 |
| ConvE (Dettmers et al., 2017a) | 0.325 | 244 | 0.237 | **0.356** | 0.501 | 0.430 | 4187 | 0.40 | 0.44 | 0.52 |
| CP-N3 (Lacroix et al., 2018) | **0.360** | - | - | - | **0.540** | **0.470** | - | - | - | **0.54** |
| RelWalk | 0.329 | **105** | **0.243** | 0.354 | 0.502 | 0.451 | **3232** | **0.42** | **0.47** | 0.51 |

| Method | Accuracy |
|---|---|
| Structured$^{\diamond}$ | 75.2 |
| TransE$^{\diamond}$ | 81.5 |
| TransR (Lin et al., 2015) | 82.5 |
| TransG (Xiao et al., 2016) | 87.3 |
| NTN (Socher et al., 2013) | 87.2 |
| RelWalk | **88.6** |

Table 2: Results of link prediction (left) and triple classification on FB13 (right). Results marked with [$\star$] are taken from (Dettmers et al., 2017a), [$\bullet$] from (Nguyen et al., 2016),[$\triangleleft$] from (Cai and Wang, 2018) and [$\diamond$] from (Wang et al., 2014). All other results for the baselines are taken from their original papers.

| Relation | H@10 | $\nu_R$ | $\sigma_c$ | $\sigma_{c'}$ | $\sqrt{\sigma_c^2+\sigma_{c'}^2}$ |
|---|---|---|---|---|---|
| hypernym | 0.188 | 3.249 | 68.89 | 64.41 | 94.31 |
| derivational | 0.955 | 1.690 | 63.44 | 65.33 | 91.07 |
| instance_hypernym | 0.541 | 0.362 | 63.11 | 64.56 | 90.28 |
| also_see | 0.670 | 0.234 | 70.76 | 61.51 | 93.76 |
| member_meronym | 0.281 | 4.389 | 63.78 | 66.09 | 91.84 |
| synset_domain_topic | 0.513 | 0.727 | 65.66 | 65.48 | 92.73 |
| has_part | 0.247 | 0.548 | 66.21 | 66.50 | 93.84 |
| domain_usage | 0.688 | 0.045 | 65.24 | 63.16 | 90.81 |
| domain_region | 0.442 | 0.065 | 67.53 | 66.31 | 94.64 |
| verb_group | 0.974 | 0.038 | 64.22 | 63.19 | 90.09 |
| similar_to | 1.000 | 0.111 | 63.67 | 63.96 | 90.25 |
| Correlations | | $-0.51$ | $-0.39$ | $-0.49$ | $-0.70$ |

Table 3: Empirical analysis of the concentration of the partitioning functions and the orthogonality of the relation embeddings, and their Pearson correlation coefficients against H@10 for the relations in WN18RR.

$(h, R, t)$) (Socher et al., 2013). We evaluate the performance in the link prediction task using mean reciprocal rank (**MRR**), mean rank (**MR**) (the average of the rank assigned to the original head or tail entity in a corrupted triple) and hits at ranks 1, 3 and 10 (**H@1, 3, 10**), whereas in the triple classification task we use **accuracy** (percentage of the correctly classified test triples). We only report scores under the *filtered* setting (Bordes et al., 2013), which removes all triples appeared in training, validating and testing sets from candidate triples before obtaining the rank of the ground truth triple. In link prediction, we consider all entities that appear in the corresponding argument in the entire knowledge graph as candidates.

In Table 2 we compare the KGEs learnt by **RelWalk** against prior work using the published results. For triple classification, **RelWalk** reports the best performance on FB13, outperforming all methods compared. For the link prediction results as shown in Table 2, we see that **RelWalk** obtains competitive performance on both WN18RR and FB15K237 under all evaluation measures. In particular, it is outperformed by the KGE method proposed by Lacroix et al. (2018) (**CP-N3**), which uses nuclear 3-norm regularisers with canonical tensor decomposition. Interestingly, the improvement against structured embeddings (SE) is consistent and interesting because the scoring function of SE closely resembles that of RelWalk as we can redefine $\mathbf{R}_2$ with the negative sign. However, SE learns KGEs that *minimise* the $\ell_{1,2}$ norm whereas according to (9) we must *maximise* the probability for relational triples in a knowledge graph. WN18RR excludes triples from WN18 that are simply inverted between train and test partitions (Toutanova and Chen, 2015; Dettmers et al., 2017b), making it a difficult dataset for link prediction using simple memorisation heuristics. **RelWalk**'s consistent good performance on both versions of this dataset shows that it is considering the global structure in the KG when learning KGEs.

We note that our goal in this paper is *not* to claim SoTA for KGE but to provide a theoretical understanding with empirical validation. To this end, the experimental results support our theoretical claim and emphasise the importance of theoretically motivating the KGE scoring function design process.

### 5.1 Orthogonality and Concentration

Our theoretical analysis depends on two main assumptions: (a) concentration of the partition function $Z_c$ (Lemma 1), and (b) the orthogonality of the relation embedding matrices $\mathbf{R}_1, \mathbf{R}_2$. In this section, we empirically study the relationship between these assumptions and the performance of RelWalk.

Given $\mathbf{R}_1$ and $\mathbf{R}_2$ learnt by RelWalk for a particular $R$, we can measure the degree to which the orthogonality, $\nu_R$, is satisfied by the sum of the non-diagonal elements (19).

$$\nu_R = \sum_{i \neq j} |\mathbf{R}_1{}^{\top}\mathbf{R}_1|_{ij} + |\mathbf{R}_2{}^{\top}\mathbf{R}_2|_{ij} \tag{19}$$

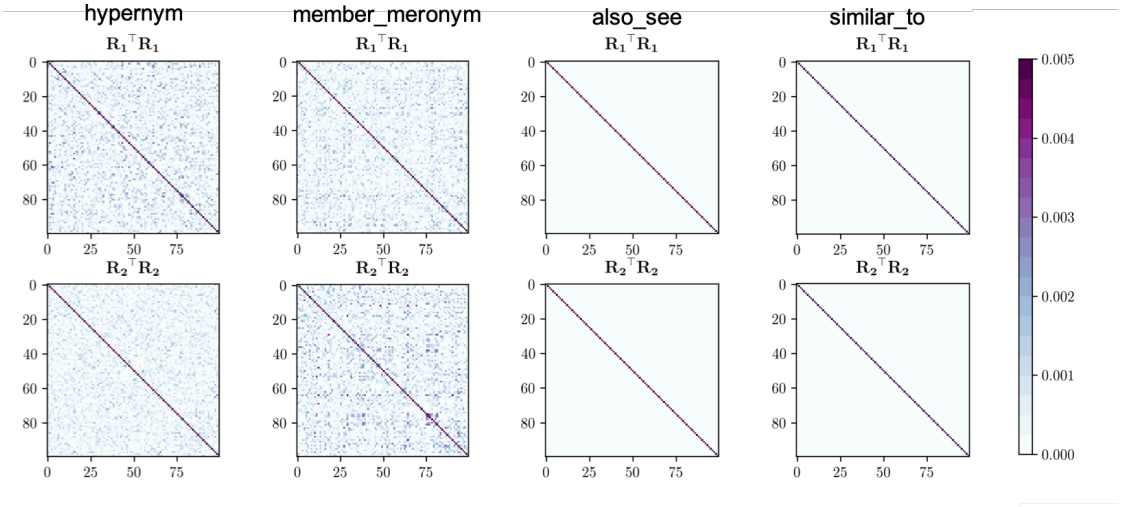


Figure 1: Heatmap visualisation of the orthogonality in different relation embeddings from the WN18RR.

If a matrix $\mathbf{A}$ is orthogonal, then the non-diagonal elements of the inner-product $\mathbf{A}^\top\mathbf{A}$ will contain zeros. Therefore, the smaller the $\nu_R$ values, more orthogonal the relation embeddings will be. We measure $\nu_R$ values for the 11 relation types in the WN18RR dataset as shown in Table 3. From Table 3 we see that $\nu_R$ values are indeed small for different relation types indicating that the orthogonality requirement is satisfied as expected. Interestingly, a moderately high (-0.515) negative Pearson correlation between H@10 and $\nu_R$ shows that orthogonality correlates with the better the performance.

To visualise how the orthogonality affects different relation types, we plot the elements in $\mathbf{R}_1{}^\top\mathbf{R}_1$ and $\mathbf{R}_2{}^\top\mathbf{R}_2$ for four relations in the WN18RR dataset in Figure 1 for $100 \times 100$ dimensional relational embeddings. For the two relations also_see and similar_to we see that the corresponding inner-products are sparse except in the main diagonal, compared to that in hypernym and member_meronym relations. On the other hand, according to Table 3 the H@10 values for also_see and similar_to are higher than that for hypernym and member_meronym as implied by the negative correlation.

To test for the concentration of the partition function, for a relation $R$ we compute $Z_c$ and $Z_{c'}$ values using respectively (6) and (7) over a set of randomly sampled 10000 head or tail entities. We compute the standard deviations $\sigma_c$ and $\sigma_{c'}$ respectively for the distributions of $Z_c$ and $Z_{c'}$ and their geometric means as shown in Table 3. We observed a Gaussian-like distributions for the partition functions for different relations for which smaller standard deviations indicate stronger concentration around the mean. Interestingly, from Table 3 we see a negative correlation between H@10 and the standard deviations indicating that the performance of RelWalk depends on the validity of the concentration assumption.

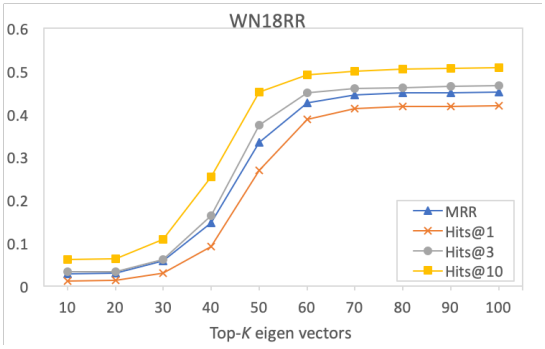


Figure 2: Results for the approximated relation embeddings for link prediction on WN18RR.

### 5.2 Compression of Embeddings

To reduce the amount of memory required for KGEs, especially with a large KG, compressing KGEs has been studied recently (Sachan, 2020). **RelWalk** uses (orthogonal) matrices to represent relations, which require more parameters compared to a vector representation of the same dimensionality of a relation. Prior work studying lower-rank decomposition of KGEs have shown that, although linear embeddings of graphs can require prohibitively large dimensionality to model certain types of relations (Nickel et al., 2014) (e.g. sameAs), nonlinear embeddings can mitigate this

problem (Bouchard et al., 2015). In this section, we propose memory-efficient low-rank approximations to the **RelWalk** embeddings.

From the definition of orthogonality it follows that the relation embeddings $\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{R}^{d \times d}$ learnt by **RelWalk** for a particular relation $R$ means that $\mathbf{R}_1, \mathbf{R}_2$ are both full-rank and cannot be factorised as the product of two lower rank matrices. This prevents us from directly applying matrix decomposition methods such as non-negative matrix factorisation on the learnt relation embeddings to obtain low-rank approximations. Therefore, we subtract the identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$ from the relation embedding $\mathbf{R} (\in \{\mathbf{R}_1, \mathbf{R}_2\})$ and factorise the remainder $\mathbf{R}' \in \mathbb{R}^{n \times n}$ as the product of two low-rank matrices using the eigendecomposition of $\mathbf{R}'$ as given by (20).

$$\begin{aligned} \mathbf{R} &= \mathbf{I} + \mathbf{R}' \\ &= \mathbf{I} + \mathbf{U}_R \mathbf{D} {\mathbf{U}_R}^\top \\ &\approx \mathbf{I} + \sum_{k=1}^{K} \mathbf{D}_{(k,k)} \mathbf{U}_{R(k,:)} \mathbf{U}_{R(:,k)} \end{aligned} \tag{20}$$

Here, $\mathbf{U}$ is the matrix formed by arranging the eigenvectors of $\mathbf{R}'$ as columns, and $\mathbf{D}$ is a diagonal matrix containing the eigenvalues of $\mathbf{R}'$ in the descending order. We can then use the largest $K \leq d$ eigenvalues and corresponding eigenvectors to obtain a rank-$K$ approximation in the sense of minimum Frobenius distance between $\mathbf{R}'$ and its rank-$K$ approximation. In the case we use $K$ factors in the approximation, we must store $dK$ real numbers corresponding to the $d$-dimensional eigen vectors per each of the $K$ components as opposed to $d^2$ real numbers in $\mathbf{R}$. The compression ratio in this case becomes $dK/d^2 = K/d$. When $K \ll d$, this results in a significant compression.

To empirically evaluate the trade-off between the number of singular vectors used in the compression and the accuracy of the learnt relation embeddings, we use the approximated relation embeddings for link prediction on WN18RR as shown in Figure 2 (similar trend was observed for FB15K237). We use $d = 100$ dimensional relation embeddings learnt by **RelWalk** and approximate using top-$K$ eigenvectors. From Figure 2 we see for $K > 60$ components the performance saturates in both datasets. On the other hand, we need at least $K = 30$ components to get any meaningful accuracy for link prediction on these two datasets. With $K = 60$ and $d = 100$ this approximation results in an $60\%$ compression ratio.

## 6 Conclusion

We proposed **RelWalk**, a generative model of KGE and derived a theoretical relationship between the probability of a triple consisting of head, tail entities and the relation that exists between those two entities, and the embedding vectors for the two entities and embeddings matrices for the relation. In RelWalk, we represented entities by vectors and relations by matrices. We then proposed a learning objective based on the theoretical relationship we derived to learn entity and relation embeddings from a given knowledge graph. Experimental results on a link prediction and a triple classification tasks show that **RelWalk** outperforms several previously proposed KGE learning methods. The key assumptions of **RelWalk** are validated by empirically analysing the relationship between such assumptions and the performance of the learnt embeddings from a KG. Moreover, we studied the compressibility of the learnt relation embeddings and discovered that using only $60\%$ of the components, we can approximate the relation embeddings without any significant loss in performance.


## Acknowledgement

Yuichi Yoshida and Ken-ichi Kawarabayashi are supported by JSPS KAKENHI Grant Number JP18H05291.


## References


Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint arXiv:1502.03520*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016a. A latent variable model approach to pmi-based word embeddings. *TACL*, 4:385–399.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016b. Rand-walk: A latent variable model approach to word embeddings. *arXiv*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of SIGMOD*, pages 1247 – 1250.

Danushka Bollegala, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2018. Using $k$-way Co-occurrences for Learning Word Embeddings. In *Proc. of AAAI*.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhenko. 2013. Translating embeddings for modeling multi-relational data. In *Proc. of NIPS*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proc. of AAAI*.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *Proc. of Knowlege Representation and Reasoning: Integrating Symbolic and Neural Approaches: Papers from the 2015 AAAI Spring Symposium*, pages 6–9.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proc. of NAACL*, pages 1470–1480.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017a. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017b. Convolutional 2D Knowledge Graph Embeddings.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. In *Proc. of ACL*, pages 110–121.

Kawin Ethayarajh. 2019. Rotate king to get queen: Word relationships as orthogonal transformations in embedding space. 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)*, pages 3503–3508, Hong Kong, China. Association for Computational Linguistics.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838, Seattle, Washington, USA. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proc. of KDD*.

Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proc. of EMNLP*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proc. of ACL*, pages 687–696.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proc. of ICML*, pages 2863–2872.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026, Jeju Island, Korea. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proc. of AAAI*, pages 2181–2187.

Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representation in vector space. In *Proc. of ICLR*.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 777–782, Atlanta, Georgia.

Dat Quoc Nguyen. 2017. An overview of embedding models of entities and relationships for knowledge base completion.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proc. of NAACL-HLT*, pages 460–466.

Maximilian Nickel, Xueyan Jiang, and Volker Tresp. 2014. Reducing the rank in relational factorization models by including observable patterns. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1179–1187. Curran Associates, Inc.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proc. of ICML*, pages 809–816.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proc. of KDD*, pages 701–710.

Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. 2018. Rdf2vec: Rdf graph embeddings and their applications. *Semantic Web*, (Preprint):1–32.

Mrinmaya Sachan. 2020. Knowledge graph embedding compression. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2681–2691, Online. Association for Computational Linguistics.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of NIPS*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020. Orthogonal relation transforms with graph context modeling for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2713–2722, Online. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proc. of 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proc. of ICML*.

Chengyu Wang, Yan Fan, Xiaofeng He, and Aoying Zhou. 2019. A family of fuzzy orthogonal projection models for monolingual and cross-lingual hypernymy prediction. In *The World Wide Web Conference*, pages 1965–1976.

Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *TKDE*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proc. of AAAI*, pages 1112 – 1119.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. Transg : A generative model for knowledge graph embedding. In *Proc. of ACL*, pages 2316–2325.

Bishan Yang, Wen-tau Yih, Xiadong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. of ICLR*.

Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. A translation-based knowledge graph embedding preserving logical property of relations. In *Proc. of NAACL*, pages 907–916.

## A Proof of the Concentration Lemma

To prove the concentration lemma, we show that the mean $\mathbb{E}_{\boldsymbol{h}}[Z_c]$ of $Z_c$ is concentrated around a constant for all knowledge vectors $\boldsymbol{c}$ and its variance is bounded. If $\mathbf{P}$ is an orthogonal matrix and $\boldsymbol{x}$ is a vector, then $\left|\left|\mathbf{P}^\top \boldsymbol{x}\right|\right|_2^2 = (\mathbf{P}^\top \boldsymbol{x})^\top (\mathbf{P}^\top \boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{P}\mathbf{P}^\top \boldsymbol{x} = ||\boldsymbol{x}||_2^2$, because $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$. Therefore, from (6) and the orthogonality of the relational embeddings, we see that $\mathbf{R}_1 \boldsymbol{c}$ is a simple rotation of $\boldsymbol{c}$ and does not alter the length of $\boldsymbol{c}$. We represent $\boldsymbol{h} = s_h \hat{\boldsymbol{h}}$, where $s_h = ||\boldsymbol{h}||$ and $\hat{\boldsymbol{h}}$ is a unit vector (i.e. $\left|\left|\hat{\boldsymbol{h}}\right|\right|_2 = 1$) distributed on the spherical Gaussian with zero mean and unit covariance matrix $\mathbf{I}_d \in \mathbb{R}^{d \times d}$. Let $s$ be a random variable that has the same distribution as $s_h$. Moreover, let us assume that $s$ is upper bounded by a constant $\kappa$ such that $s \leq \kappa$. From the assumption of the knowledge vector $\boldsymbol{c}$, it is on the unit sphere as well, which is then rotated by $\mathbf{R}_1$.

We can write the partition function using the inner-product between two vectors $\boldsymbol{h}$ and $\mathbf{R}_1 \boldsymbol{c}$, $Z_c = \sum_{h \in \mathcal{V}} \exp\left(\boldsymbol{h}^\top (\mathbf{R}_1 \boldsymbol{c})\right)$. Arora et al.(Arora et al., 2016a) showed that (Lemma 2.1 in their paper) the expectation of a partition function of this form can be approximated as follows:

$$\mathbb{E}_{\boldsymbol{h}}[Z_c] = n\mathbb{E}_{\boldsymbol{h}}[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)] \tag{21}$$

$$\geq n\mathbb{E}_{\boldsymbol{h}}[1 + \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}] = n. \tag{22}$$

where $n = |\mathcal{V}|$ is the number of entities in the vocabulary. (21) follows from the expectation of a sum and the independence of $\boldsymbol{h}$ and $\mathbf{R}_1$ from $\boldsymbol{c}$. The inequality of (22) is obtained by applying the Taylor expansion of the exponential series and the

final equality is due to the symmetry of the spherical Gaussian. From the law of total expectation, we can write

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}}[Z_c] &= n\mathbb{E}_{\boldsymbol{h}}[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)] \\ &= n\mathbb{E}_{s_h}\left[\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right]\right].\end{aligned} \tag{23}$$

where, $x = \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$. Note that conditioned on $s_h$, $\boldsymbol{h}$ is a Gaussian random variable with variance $\sigma^2 = s_h^2$. Therefore, conditioned on $s_h$, $x$ is a random variable with variance $\sigma^2 = \sigma_h^2$. Using this distribution, we can evaluate $\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right]$ as follows:

$$\begin{aligned}&\mathbb{E}_{x|s_h}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right] \\ &= \int_x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp(x) dx \\ &= \int_x \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\sigma^2)^2}{2\sigma^2} + \sigma^2/2\right) dx \\ &= \exp(\sigma^2/2).\end{aligned} \tag{24}$$

Therefore, it follows that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}}[Z_c] &= n\mathbb{E}_{s_h}[\exp(\sigma^2/2)] \\ &= n\mathbb{E}_{s_h}[\exp(s_h^2/2)] = n\exp(s^2/2),\end{aligned} \tag{25}$$

where $s$ is the variance of the $\ell_2$ norms of the entity embeddings. Because the set of entities is given and fixed, both $n$ and $\sigma$ are constants, proving that $\mathbb{E}_{\boldsymbol{h}}[Z_c]$ does not depend on $c$.

Next, we calculate the variance $\mathbb{V}_{\boldsymbol{h}}[Z_c]$ as follows:

$$\begin{aligned}\mathbb{V}_{\boldsymbol{h}}[Z_c] &= \sum_h \mathbb{V}_{\boldsymbol{h}}[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)] \\ &\leq n\mathbb{E}_{\boldsymbol{h}}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\right] \\ &= n\mathbb{E}_{s_h}\left[\mathbb{E}_{x|s_h}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right]\right].\end{aligned} \tag{26}$$

Because $2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$ is a Gaussian random variable with variance $4\sigma^2 = 4s_h^2$ from a similar calculation as in (24) we obtain,

$$\mathbb{E}_{x|s_h}\left[\exp\left(2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mid s_h\right] = \exp(2\sigma^2). \tag{27}$$

By substituting (27) in (26) we have that

$$\mathbb{V}_{\boldsymbol{h}}[Z_c] \leq n\mathbb{E}_{s_h}\left[\exp\left(2\sigma^2\right)\right] = n\mathbb{E}_{s_h}\left[\exp(2s^2)\right] \leq \Lambda n \tag{28}$$

for $\Lambda = \exp(8\kappa^2)$ a constant bounding $s \leq \kappa$ as stated. From above, we have bounded both the mean and variance of the partition function by constants that are independent of the knowledge vector. Note that neither $\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)$ nor $\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)$ are sub-Gaussian nor sub-exponential. Therefore, standard concentration bounds derived for sub-Gaussian or sub-exponential random variables cannot be used in our analysis. However, the argument given in Appendix A.1 in (Arora et al., 2016b) for a partition function with bounded mean and variance can be directly applied to $Z_c$ in our case, which completes the proof of the concentration lemma. From the symmetry between $h$ and $t$, the concentration Lemma is also applies for the partition function $Z_{c'} = \sum_{t \in \mathcal{V}} \left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)$.

## B Proof of RelWalk Theorem

Let us consider the probabilistic event that $(1-\epsilon_z)Z \leq Z_c \leq (1+\epsilon_z)Z$ to be $F_c$ and $(1-\epsilon_z)Z \leq Z_{c'} \leq (1+\epsilon_z)Z$ to be $F_{c'}$. From Lemma 1 we have $\Pr[F_c] \geq 1-\delta$. Then from the union bound we have,

$$\begin{aligned}\Pr[\bar{F}_c \cup \bar{F}_{c'}] &\leq \Pr[\bar{F}_c] + \Pr[\bar{F}_{c'}] \\ &= 1 - \Pr[F_c] + 1 - \Pr[F_{c'}] \\ &= 2\delta.\end{aligned} \tag{29}$$

where $\bar{F}$ is the complement of event $F$. Moreover, let $F$ be the probabilistic event that both $F_c$ and $F_{c'}$ being True. Then from $\Pr[F] = 1 - \Pr[\bar{F}_c \cup \bar{F}_{c'}]$ we have, $\Pr[F] \geq 1 - 2\exp\left(-\Omega\left(\log^2 n\right)\right)$. The R.H.S. of (5) can be split into two parts $T_1$ and $T_2$ according to whether $F$ happens or not.

$$\begin{aligned}p(h,t \mid R) = &\underbrace{\mathbb{E}_{\boldsymbol{c},\boldsymbol{c'}}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)}{Z_{c'}} \mathbf{1}_F\right]}_{T_1} \\ &+ \underbrace{\mathbb{E}_{\boldsymbol{c},\boldsymbol{c'}}\left[\frac{\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)}{Z_c} \frac{\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)}{Z_{c'}} \mathbf{1}_{\bar{F}}\right]}_{T_2}.\end{aligned} \tag{30}$$

Here, $\mathbf{1}_F$ and $\mathbf{1}_{\bar{F}}$ are indicator functions of the events $F$ and $\bar{F}$ given as follows:

$$\mathbf{1}_F = \begin{cases} 1 & \text{if } F \text{ is True,} \\ 0 & \text{otherwise,} \end{cases} \tag{31}$$

$$\mathbf{1}_{\bar{F}} = \begin{cases} 0 & \text{if } F \text{ is True,} \\ 1 & \text{otherwise.} \end{cases} \tag{32}$$

Let us first show that $T_2$ is negligibly small.

For two real integrable functions $\psi_1(x)$ and $\psi_2(x)$ in $[a, b]$, the Cauchy-Schwarz's inequality states that

$$\left[\int_a^b \psi_1(x)\psi_2(x)dx\right]^2 \leq \int_a^b [\psi_1(x)]^2 dx \int_a^b [\psi_2(x)]^2 dx. \tag{33}$$

Applying (33) to $T_2$ in (30) we have:

$$\begin{aligned}
&\left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_c Z_{c'}}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\mathbf{1}_{\bar{F}}\right]\right)^2 \\
&\leq \left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{\bar{F}}\right]\right) \times \\
&\quad \left(\mathbb{E}_{c,c'}\left[\frac{1}{Z_{c'}^2}\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbf{1}_{\bar{F}}\right]\right) \\
&= \left(\mathbb{E}_{c}\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]\right) \times \\
&\quad \left(\mathbb{E}_{c'}\left[\frac{1}{Z_{c'}^2}\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbb{E}_{c|c'}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)
\end{aligned} \tag{34}$$

Note that $Z_c \geq 1$ because $Z_c$ is the sum of positive numbers and if $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0$ for at least one of the $h \in \mathcal{V}$, then the total sum will be greater than 1. Therefore, by dropping $Z_c$ term from the denominator we can further increase the first term in (34) as given by (35).

$$\begin{aligned}
&\mathbb{E}_c\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&\leq \mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]
\end{aligned} \tag{35}$$

Let us split the expectation on the R.H.S. of (35) into two cases depending on whether $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0$ or otherwise, indicated respectively by $\mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)}$ and $\mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} \leq 0)}$.

$$\begin{aligned}
&\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&= \mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&+ \mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} \leq 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]
\end{aligned} \tag{36}$$

The second term of (36) is upper bounded by

$$\mathbb{E}_{c,c'}\left[\mathbf{1}_{\bar{F}}\right] \leq \exp\left(-\Omega(\log^2 n)\right) \tag{37}$$

The first term of (36) can be bounded as follows:

$$\begin{aligned}
&\mathbb{E}_c\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&\leq \mathbb{E}_c\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbf{1}_{(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} > 0)} \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right] \\
&\leq \mathbb{E}_c\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]
\end{aligned} \tag{38}$$

where $\alpha > 1$. Therefore, it is sufficient to bound $\mathbb{E}_c\left[\exp(\alpha \boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c})^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]$ when $||\boldsymbol{h}|| = \Omega(\sqrt{d})$.

Let us denote by $z$ the random variable $2\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$. Moreover, let $r(z) = \mathbb{E}_{c'|z}[\mathbf{1}_{\bar{F}}]$, which is a function of $z$ between $[0, 1]$. We wish to upper bound $\mathbb{E}_c[\exp(z)r(z)]$. The worst-case $r(z)$ can be quantified using a continuous version of Abel's inequality (proved as Lemma A.4 in (Arora et al., 2015)), we can upper bound $\mathbb{E}_c\left[\exp(z)r(z)\right]$ as follows:

$$\mathbb{E}_c\left[\exp(z)r(z)\right] \leq \mathbb{E}\left[\exp(z)\mathbf{1}_{[t,+\infty]}(z)\right] \tag{39}$$

where $t$ satisfies that $\mathbb{E}_c[\mathbf{1}_{[t,+\infty]}(z)] = \Pr[z \geq t] = \mathbb{E}_c[r(z)] \leq \exp(-\Omega(\log^2 n))$. Here, $\mathbf{1}_{[t,+\infty]}(z)$ is a function that takes the value 1 when $z \geq t$ and zero elsewhere. Then, we claim $\Pr_c[z \geq t] \leq \exp(-\Omega(\log^2 n))$ implies that $t \geq \Omega(\log^{.9} n)$.

If $c$ was distributed as $\mathcal{N}(0, \frac{1}{d}\mathbf{I})$, this would be a simple tail bound. However, as $c$ is distributed uniformly on the sphere, this requires special care, and the claim follows by applying the tail bound for the spherical distribution given by Lemma A.1 in (Arora et al., 2015) instead. Finally, applying Corollary A.3 in (Arora et al., 2015), we have:

$$\begin{aligned}
\mathbb{E}[\exp(z)r(z)] &\leq \mathbb{E}[\exp(z)\mathbf{1}_{[t,+\infty]}(z)] \\
&= \exp(-\Omega(\log^{1.8} n))
\end{aligned} \tag{40}$$

From a similar argument as above we can obtain the same bound for $c'$ as well. Therefore, $T_2$ in (30) can be upper bounded as follows:

$$\begin{aligned}
&\mathbb{E}_{c,c'}\left[\frac{1}{Z_c Z_{c'}}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\mathbf{1}_{\bar{F}}\right] \\
&\leq \left(\mathbb{E}_c\left[\frac{1}{Z_c^2}\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)^2 \mathbb{E}_{c'|c}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)^{1/2} \times \\
&\quad \left(\mathbb{E}_{c'}\left[\frac{1}{Z_{c'}^2}\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)^2 \mathbb{E}_{c|c'}\left[\mathbf{1}_{\bar{F}}\right]\right]\right)^{1/2} \\
&\leq \exp(-\Omega(\log^{1.8} n))
\end{aligned} \tag{41}$$

Because $n = |\mathcal{V}|$, the size of the entity vocabulary, is large (ca. $n > 10^5$) in most knowledge graphs, we can ignore the $T_2$ term in (30).

Combining the above analysis of $T_2$ term with (30) we obtain an upper bound for $p(h, t \mid r)$ given by (42).

$$\begin{aligned}
p(h, t \mid R) &\leq (1+\epsilon_z)^2 \frac{1}{Z^2}\mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\mathbf{1}_F\right] \\
&\quad + |\mathcal{D}|\exp(-\Omega(\log^{1.8} n)) \\
&= (1+\epsilon_z)^2 \frac{1}{Z^2}\mathbb{E}_{c,c'}\left[\exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right)\exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right)\right] + \delta_0
\end{aligned} \tag{42}$$

where $|\mathcal{D}|$ is the number of relational tuples $(h, r, t)$ in the KB and $\delta_0 = |\mathcal{D}| \exp(-\Omega(\log^{1.8} n)) \leq \exp(-\Omega(\log^{1.8} n))$ by the fact that $Z \leq \exp(2\kappa)n = O(n)$, where $\kappa$ is the upper bound on $\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}$ and $\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}$, which is regarded as a constant.

On the other hand, we can lower bound $p(h, t \mid r)$ as given by (43).

$$
\begin{aligned}
p(h,t \mid R) &\geq (1-\epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'} \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] \\
&\geq (1-\epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'} \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] \\
&\quad - |\mathcal{D}| \exp(-\Omega(\log^{1.8} n)) \\
&\geq (1-\epsilon_z)^2 \frac{1}{Z^2} \mathbb{E}_{c,c'} \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] - \delta_0
\end{aligned} \tag{43}
$$

Taking the logarithm of both sides, from (42) and (43), the multiplicative error translates to an additive error given by (44).

$$
\begin{aligned}
\log p(h,t \mid R) &= \log \left( \mathbb{E}_{c,c'} \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \mathbb{E}_{c'|c} \left[ \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) A(c) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z)
\end{aligned} \tag{44}
$$

where $A(c) \coloneqq \mathbb{E}_{c'|c} \left[ \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right]$.

We assumed that $\boldsymbol{c}$ and $\boldsymbol{c}'$ are on the unit sphere and $\mathbf{R}_1$ and $\mathbf{R}_2$ to be orthogonal matrices. Therefore, $\mathbf{R}_1 \boldsymbol{c}$ and $\mathbf{R}_2 \boldsymbol{c}'$ are also on the unit sphere. Moreover, if we let the upper bound of the $\ell_2$ norm of the entity embeddings to be $\kappa' \sqrt{d}$, then we have $||\boldsymbol{h}|| \leq \kappa' \sqrt{d}$ and $||\boldsymbol{t}|| \leq \kappa' \sqrt{d}$. Therefore, we have

$$
\langle \mathbf{R}_1 \boldsymbol{h}, \boldsymbol{c}' - \boldsymbol{c} \rangle \leq ||\boldsymbol{h}|| \left|\left| \boldsymbol{c} - \boldsymbol{c}' \right|\right| \leq \kappa' \sqrt{d} \left|\left| \boldsymbol{c} - \boldsymbol{c}' \right|\right| \tag{45}
$$

Then, we can upper bound $A(c)$ as follows:

$$
\begin{aligned}
A(c) &= \mathbb{E}_{c'|c} \left[ \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c'}\right) \right] \\
&= \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c} \left[ \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 (\boldsymbol{c'} - \boldsymbol{c})\right) \right] \\
&\leq \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c} \left[ \exp\left(\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \\
&\leq (1 + \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)
\end{aligned} \tag{46}
$$

For some $\epsilon_2 > 0$. The last inequality holds because

$$
\begin{aligned}
&\mathbb{E}_{c|c'} \left[ \exp\left(\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \\
&= \int \exp\left(\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) p(c'|c) dc' \\
&= \underbrace{\exp(\kappa' \sqrt{d})}_{\geq 1} \underbrace{\int \exp(\left|\left| \boldsymbol{c} - \boldsymbol{c'} \right|\right|) p(c'|c) dc'}_{\geq 1} \\
&= 1 + \epsilon_2
\end{aligned} \tag{47}
$$

To obtain a lower bound on $A(c)$ from the first-order Taylor approximation of $\exp(x) \geq 1 + x$ we observe that:

$$
\begin{aligned}
&\mathbb{E}_{c|c'} \left[ \exp\left(\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \\
&\quad + \mathbb{E}_{c|c'} \left[ \exp\left(-\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \geq 2.
\end{aligned} \tag{48}
$$

Therefore, from our model assumptions we have

$$
\mathbb{E}_{c|c'} \left[ \exp\left(-\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \geq 1 - \epsilon_2 \tag{49}
$$

Hence,

$$
\begin{aligned}
A(c) &= \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c} \left[ \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 (\boldsymbol{c'} - \boldsymbol{c})\right) \right] \\
&\geq \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \mathbb{E}_{c'|c} \left[ \exp\left(-\kappa' \sqrt{d} \left|\left| \boldsymbol{c'} - \boldsymbol{c} \right|\right|\right) \right] \\
&\geq (1 - \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right)
\end{aligned} \tag{50}
$$

Therefore, from (47) and (50) we have

$$
A(c) = (1 \pm \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \tag{51}
$$

Plugging $A(c)$ back in (44) results in $\log p(h, t \mid r)$ equal to:

$$
\begin{aligned}
&\log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) A(c) \right] \pm \delta_0 \right) - 2\log Z + 2\log(1 \pm \epsilon_z) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) (1 \pm \epsilon_2) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c}\right) \exp\left(\boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left(\boldsymbol{h}^\top \mathbf{R}_1 \boldsymbol{c} + \boldsymbol{t}^\top \mathbf{R}_2 \boldsymbol{c}\right) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2) \\
&= \log \left( \mathbb{E}_c \left[ \exp\left( \left( {\mathbf{R}_1}^\top \boldsymbol{h} + {\mathbf{R}_2}^\top \boldsymbol{t} \right)^\top \boldsymbol{c} \right) \right] \pm \delta_0 \right) \\
&\quad - 2\log Z + 2\log(1 \pm \epsilon_z) + \log(1 \pm \epsilon_2)
\end{aligned} \tag{52}
$$

Note that $\boldsymbol{c}$ has a uniform distribution over the unit sphere. In this case, from Lemma A.5 in (Arora et al., 2015), (53) holds approximately.

$$
\begin{aligned}
&\mathbb{E}_c \left[ \exp\left( \left( {\mathbf{R}_1}^\top \boldsymbol{h} + {\mathbf{R}_2}^\top \boldsymbol{t} \right)^\top \boldsymbol{c} \right) \right] \\
&= (1 \pm \epsilon_3) \exp\left( \frac{\left|\left| {\mathbf{R}_1}^\top \boldsymbol{h} + {\mathbf{R}_2}^\top \boldsymbol{t} \right|\right|_2^2}{2d} \right)
\end{aligned} \tag{53}
$$

where $\epsilon_3 = \tilde{O}(1/d)$. Plugging (53) in (52) we have that

$$\log p(h,t \mid R) = \frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d} + O(\epsilon_z) + O(\epsilon_2) + O(\epsilon_3) + O(\delta_0') - 2\log Z \tag{54}$$

where $\delta_0' = \delta_0$.

$\left(\mathbb{E}_c\left[\exp\left((\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t})^\top \boldsymbol{c}\right)\right]\right)^{-1} = \exp(-\Omega(\log^{1.8} n))$. Therefore, $\delta_0'$ can be ignored. Note that $\epsilon_3 = \tilde{O}(1/d)$ and $\epsilon_z = \tilde{O}(1/\sqrt{n})$ by assumption. Therefore, we obtain that

$$\log p(h,t \mid R) = \frac{\left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2}{2d} + O(\epsilon_z) + O(\epsilon_2) + \tilde{O}(1/d) - 2\log Z \tag{55}$$

## C Learning with Multiple Negative Triples

This approach can be easily extended to learn from multiple negative triples as follows. Let us consider that we are given a positive triple, $(h, R, t)$ and a set of $K$ negative triples $\{(h_k', R, t_k')\}_{k=1}^K$. We would like our model to assign a probability, $p(h, t \mid R)$, to the positive triple that is higher than that assigned to any of the negative triples. This requirement can be written as (56).

$$p(h,t|R) \geq \max_{k=1,\ldots,K} p(h_k', t_k' \mid R) \tag{56}$$

We could further require the ratio between the probability of the positive triple and maximum probability over all negative triples to be greater than a threshold $\eta \geq 1$ to make the requirement of (56) to be tighter.

$$\frac{p(h,t \mid R)}{\max_{k=1,\ldots,K} p(h_k', t_k' \mid R)} \geq \eta \tag{57}$$

By taking the logarithm of (57) we obtain

$$\log p\left(h,t \mid R\right) - \log\left(\max_{k=1,\ldots,K} p\left(h_k', t_k' \mid R\right)\right) \geq \log(\eta) \tag{58}$$

Therefore, we can define the marginal loss for a misclassification as follows:

$$L\left((h,R,t), \{(h_k', R, t_k')\}_{k=1}^K\right) = \max\left(0, \log\left(\max_{k=1,\ldots,K} p(h_k', t_k' \mid R)\right) + \log(\eta) - \log p\left(h,t \mid R\right)\right) \tag{59}$$

However, from the monotonicity of the logarithm we have $\forall x_1, x_2 > 0$, if $\log(x_1) \geq \log(x_2)$ then $x_1 \geq x_2$. Therefore, the logarithm of the maximum can be replaced by the maximum of the logarithms in (59) as shown in (60).

$$L\left((h,R,t), \{(h_k', R, t_k')\}_{k=1}^K\right) = \max\left(0, \max_{k=1,\ldots,K} \log\left(p\left(h_k', t_k' \mid R\right)\right) + \log(\eta) - \log p\left(h,t \mid R\right)\right) \tag{60}$$

By substituting (9) for the probabilities in (60) we obtain the rank-based loss given by (61).

$$L\left((h,R,t), \{(h_k', R, t_k')\}_{k=1}^K\right) = \max\left(0, 2d\log(\eta) + \max_{k=1,\ldots,K} \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h}_k' + \mathbf{R}_2{}^\top \boldsymbol{t}_k'\right|\right|_2^2 - \left|\left|\mathbf{R}_1{}^\top \boldsymbol{h} + \mathbf{R}_2{}^\top \boldsymbol{t}\right|\right|_2^2\right) \tag{61}$$

In practice, we can use $p(h_k', t_k' \mid R)$ to select the negative triple with the highest probability for training with the positive triple.

| Dataset | #R | #E | Train | Test | Val. |
|---|---|---|---|---|---|
| FB15K237 | 237 | 14,541 | 272,115 | 17,535 | 20,466 |
| WN18RR | 11 | 40,943 | 86,835 | 3,134 | 3,034 |
| FB13 | 13 | 75,043 | 316,232 | 23,733 | 5,908 |

Table 4: Statistics of the datasets

## D Training Details

The statistics of the benchmark datasets are shown in Table 4. We selected the initial learning rate ($\alpha$) for SGD in $\{0.01, 0.001\}$, the regularisation coefficients $(\lambda_1, \lambda_2)$ for the orthogonality constraints of relation matrices in $\{0, 1, 10, 100\}$. The number of randomly generated negative triples $n_{\text{neg}}$ for each positive example is varied in $\{1, 10, 20, 50, 100\}$ and $d \in \{50, 100\}$. Optimal hyperparameter settings were: $\lambda_1 = \lambda_2 = 10$, $n_{\text{neg}} = 100$ for all the datasets, $\alpha = 0.001$ for FB15K237 and FB13, $\alpha = 0.01$ for WN18RR. For FB15K237 and WN18RR $d = 100$ was the best, whereas for FB13 $d = 50$ performed best. Negative triples are generated by replacing a head or a tail entity in a positive triple by a randomly selected entity and learn KGEs. We train the model until convergence or at most 1000 epochs over the training data where each epoch is divided into 100 mini-batches. The best model is selected by early stopping based on the performance of the learnt embeddings on the validation set (evaluated after each 20 epochs).