To Copy Rather Than Memorize: A Vertical Learning Paradigm for Knowledge Graph Completion

Rui Li[1*], **Xu Chen**[2†], **Chaozhuo Li**[3*†], **Yanming Shen**[1], **Jianan Zhao**[4], **Yujing Wang**[5], **Weihao Han**[5], **Hao Sun**[5], **Weiwei Deng**[5], **Qi Zhang**[5], **Xing Xie**[3]
[1]Dalian University of Technology, [2]Renmin University of China,
[3]Microsoft Research Asia, [4]Université de Montréal, [5]Microsoft
xu.chen@ruc.edu.cn, cli@microsoft.com

Abstract

Embedding models have shown great power in knowledge graph completion (KGC) task. By learning structural constraints for each training triple, these methods *implicitly memorize* intrinsic relation rules to infer missing links. However, this paper points out that the multi-hop relation rules are hard to be reliably memorized due to the inherent deficiencies of such implicit memorization strategy, making embedding models underperform in predicting links between distant entity pairs. To alleviate this problem, we present Vertical Learning Paradigm (VLP), which extends embedding models by allowing to *explicitly copy* target information from related factual triples for more accurate prediction. Rather than solely relying on the implicit memory, VLP directly provides additional cues to improve the generalization ability of embedding models, especially making the distant link prediction significantly easier. Moreover, we also propose a novel relative distance based negative sampling technique (ReD) for more effective optimization. Experiments demonstrate the validity and generality of our proposals on two standard benchmarks. Our code is available at https://github.com/rui9812/VLP.

1 Introduction

Knowledge graphs (KGs) structurally represent human knowledge as a collection of factual triples. Each triple (h, r, t) represents that there is a relation r between head entity h and tail entity t. With the massive human knowledge, KGs facilitate a myriad of downstream applications (Xiong et al., 2017). However, real-world KGs such as Freebase (Bollacker et al., 2008) are far from complete (Bordes et al., 2013). This motivates substantial research on the knowledge graph completion (KGC) task, i.e., automatically inferring missing triples.

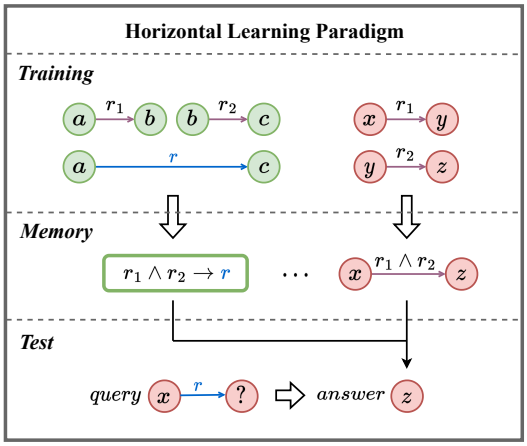


Figure 1: Learning paradigm of embedding models.

As an effective solution for KGC, embedding model learns representations of entities and relations with pre-designed relation operations. For example, TransE (Bordes et al., 2013) represents relations as translations between head and tail entities. RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) model the three-way interactions in each triple. RotatE (Sun et al., 2019), QuatE (Zhang et al., 2019) and DualE (Cao et al., 2021) represent relations as rotations in different dimensions. Rot-Pro (Song et al., 2021) further introduces the orthogonal projection for each relation.

Essentially, embedding models learn structural constraints for every factual triple during the training period. For example, for each training triple (h, r, t), TransE constrains that the head embedding $\mathbf{h}$ plus the relation embedding $\mathbf{r}$ equals the tail embedding $\mathbf{t}$. Such single-triple constraints empower embedding models to implicitly perceive (i.e., memorize) the high-order entity connections and intrinsic relation rules (Sun et al., 2019). As shown in Figure 1, by imposing the structural constraints (e.g., $\mathbf{h} + \mathbf{r} = \mathbf{t}$ in TransE) on the five training triples, embedding models can memorize the entity connection $(x, r_1 \wedge r_2, z)$ and the relation rule $r_1 \wedge r_2 \rightarrow r$. In this way, the missing link

[*] Work done during Rui Li's internship at Microsoft Research Asia. Both authors contributed equally to this research.
[†] Corresponding authors.

Model	Score Function	$g(\mathbf{W}_{r,1}\mathbf{h}+\mathbf{b}_r, \mathbf{W}_{r,2}\mathbf{t})$				Space
		$\mathbf{W}_{r,1}$	$\mathbf{b}_r$	$\mathbf{W}_{r,2}$	$g(\mathbf{q},\mathbf{k})$	
RESCAL (Nickel et al., 2011)	$\mathbf{h}^\top\mathbf{W}_r\mathbf{t}$	$\mathbf{I}$	$\mathbf{0}$	$\mathbf{W}_r$	$\mathbf{q}^\top\mathbf{k}$	$\mathbb{R}$
TransE (Bordes et al., 2013)	$-\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert$	$\mathbf{I}$	$\mathbf{r}$	$\mathbf{I}$	$-\lVert\mathbf{q}-\mathbf{k}\rVert$	$\mathbb{R}$
TransR (Lin et al., 2015)	$-\lVert\mathbf{W}_r\mathbf{h}+\mathbf{r}-\mathbf{W}_r\mathbf{t}\rVert$	$\mathbf{W}_r$	$\mathbf{r}$	$\mathbf{W}_r$	$-\lVert\mathbf{q}-\mathbf{k}\rVert$	$\mathbb{R}$
DistMult (Yang et al., 2015)	$\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\mathbf{t}$	$\mathrm{diag}(\mathbf{r})$	$\mathbf{0}$	$\mathbf{I}$	$\mathbf{q}^\top\mathbf{k}$	$\mathbb{R}$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\bar{\mathbf{t}})$	$\mathrm{diag}(\mathbf{r})$	$\mathbf{0}$	$\mathbf{I}$	$\mathrm{Re}(\mathbf{q}^\top\bar{\mathbf{k}})$	$\mathbb{C}$
RotatE (Sun et al., 2019)	$-\lVert\mathbf{h}\circ\mathbf{r}-\mathbf{t}\rVert$	$\mathrm{diag}(\mathbf{r})$	$\mathbf{0}$	$\mathbf{I}$	$-\lVert\mathbf{q}-\mathbf{k}\rVert$	$\mathbb{C}$

Table 1: The score functions and GSF settings of several models, where $\circ$ denotes the Hadamard product.

(x, r, z) can be inferred at test time without any explicit prompt. We refer to this single-triple learning paradigm as Horizontal Learning Paradigm (HLP), since the relation rules are implicitly induced by the horizontal paths between head and tail entities.

However, this paper shows that the HLP-based embedding models are hard to reliably memorize the multi-hop relation rules, which is attributed to inevitable single-triple bias and high-demanding memory capacity. The unreliable multi-hop relation rules in the implicit memory cannot serve as rational basis for prediction, leading to the inferior performance of embedding models in predicting links between distant entity pairs. This brings us a question: *is there a general paradigm for embedding models to alleviate this problem of HLP and achieve superior performance?*

We give an affirmative answer by presenting Vertical Learning Paradigm (VLP), which endows embedding models with the ability to explicitly consult related factual triples (i.e., vertical references) for more accurate prediction. Specifically, to answer $(h, r, ?)$, VLP first selects N relevant reference queries in the training graph, and then treats their ground-truth entities as the reference answers for embedding models to jointly predict the target t. This learning process can be viewed as an *explicit copy* strategy, which is different from the *implicit memorization* strategy of HLP, making it significantly easier to predict distant links. Moreover, to effectively optimize the models, we further propose a novel Relative Distance based negative sampling technique (ReD), which can generate more informative negative samples and reduce the toxicity of false negative samples. Note that VLP and ReD are both general techniques and can be widely applied to various embedding models. Our contributions are summarized as follows:

- We show that existing embedding models underperform in predicting links between distant entity pairs, since they are hard to reliably memorize the multi-hop relation rules.
- We present a novel learning paradigm named VLP, which can empower embedding models to leverage explicit references as cues for more accurate prediction.
- We further propose a new relative distance based negative sampling technique named ReD for more effective optimization.
- We conduct in-depth experiments on two standard benchmarks, demonstrating the validity and generality of the proposed techniques.

2 Preliminaries

To elicit our proposal from a general paradigm perspective, we give a bird's eye view of existing embedding models in this section. We first review the problem setup of KGC task. Afterwards, we summarize a generalized score function of embedding models and describe how the models learn to predict new links (i.e., horizontal learning paradigm).

2.1 Problem Setup

Given the entity set $\mathcal{E}$ and relation set $\mathcal{R}$, a knowledge graph can be formally defined as a collection of factual triples $\mathcal{D} = \{(h, r, t)\}$, in which head/tail entities $h, t \in \mathcal{E}$ and relation $r \in \mathcal{R}$. KGC task aims to infer new links by answering a query $(h, r, ?)$ or $(?, r, t)$. As an effective tool for this task, embedding model learns representations of entities and relations to measure each candidate's plausibility with a pre-designed score function.

2.2 Generalized Score Function

Based on a series of previous works (Nickel et al., 2011; Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Yang et al., 2015; Trouillon et al., 2016; Sun et al., 2019; Gao et al., 2020; Song et al., 2021), we summarize a generalized score function (GSF) of embedding models. To facilitate presentation, we only describe the query case of $(h, r, ?)$, while $(?, r, t)$ can be similarly conduced.

Given a query $(h, r, ?)$ and a candidate answer t, GSF first maps the head embedding $\mathbf{h} \in \mathbb{X}^{d_e}$ to the query embedding $\mathbf{q} \in \mathbb{X}^{d_r}$ with a relation-specific linear transformation:

$$\mathbf{q} = \mathbf{W}_{r,1}\mathbf{h} + \mathbf{b}_r, \tag{1}$$

where $\mathbb{X} \in \{\mathbb{R}, \mathbb{C}\}$ is the embedding space, d_e and d_r are the embedding dimensions of entities and relations, $\mathbf{W}_{r,1} \in \mathbb{X}^{d_r \times d_e}$ and $\mathbf{b}_r \in \mathbb{X}^{d_r}$ denote the relation-specific projection matrix and bias vector.

Then, GSF uses another linear function to generate the answer embedding $\mathbf{k} \in \mathbb{X}^{d_e}$ from the tail embedding $\mathbf{t} \in \mathbb{X}^{d_e}$:

$$\mathbf{k} = \mathbf{W}_{r,2}\mathbf{t}, \tag{2}$$

where $\mathbf{W}_{r,2} \in \mathbb{X}^{d_r \times d_e}$ denotes the relation transformation matrix for tail projections.

Finally, the plausibility score of the triple (h, r, t) is calculated by a similarity function g:

$$score = g(\mathbf{q}, \mathbf{k}). \tag{3}$$

By combining the above three steps, we formally define the generalized score function f_g as follows:

$$f_g(h, r, t) = g(\mathbf{W}_{r,1}\mathbf{h} + \mathbf{b}_r, \mathbf{W}_{r,2}\mathbf{t}). \tag{4}$$

With different choices of $\mathbf{W}_{r,1}$, $\mathbf{b}_r$, $\mathbf{W}_{r,2}$ and g, GSF can be instantiated as specific score functions of existing models. Table 1 exhibits several popular methods and their corresponding GSF settings.

2.3 Horizontal Learning Paradigm

With the pre-defined score functions, embedding models commonly follow the horizontal learning paradigm, which constructs the single-edge constraints to implicitly memorize high-order entity connections and intrinsic relation rules.

Take RotatE to process the triples in Figure 1 as an example. By imposing the rotation constraints on three triples (a, r_1, b), (b, r_2, c) and (a, r, c), RotatE is able to perceive a two-hop entity connection and further induce a two-hop relation rule:

$$\begin{cases} \mathbf{b} = \mathbf{a} \circ \mathbf{r}_1 \\ \mathbf{c} = \mathbf{b} \circ \mathbf{r}_2 \\ \mathbf{c} = \mathbf{a} \circ \mathbf{r} \end{cases} \Rightarrow \mathbf{r} = \mathbf{r}_1 \circ \mathbf{r}_2. \tag{5}$$

Similarly, the high-order connection can also be captured by constraining (x, r_1, y) and (y, r_2, z):

$$\begin{cases} \mathbf{y} = \mathbf{x} \circ \mathbf{r}_1 \\ \mathbf{z} = \mathbf{y} \circ \mathbf{r}_2 \end{cases} \Rightarrow \mathbf{z} = \mathbf{x} \circ \mathbf{r}_1 \circ \mathbf{r}_2. \tag{6}$$

Finally, by combining Equation (5) and (6), RotatE is capable of inferring the missing link (x, r, z).

3 Motivation

The motive of our work originates from an observation that embedding models underperform in predicting links between distant entity pairs (refer to Appendix A for more details). Since the effectiveness of embedding models is largely determined by the ability to learn intrinsic relation rules (Sun et al., 2019; Song et al., 2021; Li et al., 2022), such inferior performance reveals that the models are hard to memorize the multi-hop relation rules. We attribute this deficiency to the *multi-hop bias accumulation* and *high-demanding memory capacity* in the implicit memorization strategy of HLP.

Multi-hop Bias Accumulation The HLP-based embedding models implicitly perceive the multi-hop relation rules by constraining each training edge as shown in Section 2.3. Nevertheless, the single-edge constraints inevitably have biases during the optimization, which will accumulate with the increase of relation hops. This bias accumulation makes the memorized relation rules unreliable, leading to the deficient generalization ability for link prediction between distant entities. Concretely, considering the single-edge biases, the rule learning process in Equation (5) can be rewritten as:

$$\begin{cases} \mathbf{b} = \mathbf{a} \circ \mathbf{r}_1 \circ \boldsymbol{\epsilon}_1 \\ \mathbf{c} = \mathbf{b} \circ \mathbf{r}_2 \circ \boldsymbol{\epsilon}_2 \\ \mathbf{c} = \mathbf{a} \circ \mathbf{r} \circ \boldsymbol{\epsilon}_0 \end{cases} \Rightarrow \mathbf{r} = \mathbf{r}_1 \circ \mathbf{r}_2 \circ \boldsymbol{\epsilon}_{abc}, \tag{7}$$

where $\boldsymbol{\epsilon}_{abc} = \boldsymbol{\epsilon}_0^{-1} \circ \boldsymbol{\epsilon}_1 \circ \boldsymbol{\epsilon}_2$ is the cumulative bias. Note that $\boldsymbol{\epsilon}_{abc}$ is triple-dependent, which makes it intractable for other queries, e.g., $(x, r, ?)$ in Figure 1, to rely on this rule for prediction.

High-demanding Memory Capacity The HLP-based models essentially learn the general rules from the relation paths between head and tail entities. With the increase of path length, the quantity of different paths (or rules) expands exponentially (Wang et al., 2021). This requires intensive memory to memorize the whole crucial relation rules. However, the modeling capacity of embedding models is insufficient to meet this requirement. Since these models constrain basic edges to form long-range paths following the bottom-up design of HLP, they are more inclined to memorize the low-order rules and forget the high-order rules.

Design Goal We seek to develop a general technique to alleviate the "*Hard to Memorize*" problem of existing embedding models.

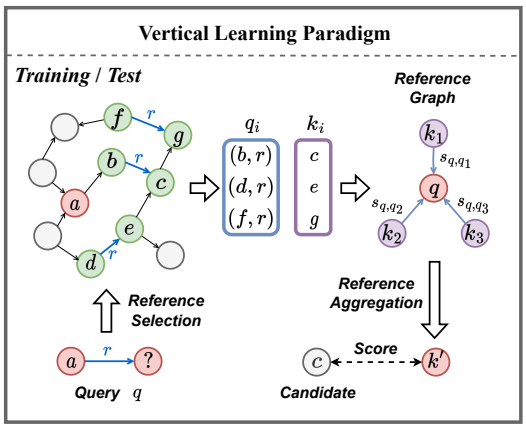


Figure 2: Vertical learning paradigm consisting of reference query selection, reference graph construction and reference answer aggregation.

A straight-forward strategy is to directly extract and process the enclosing subgraph between head and tail entities (Teru et al., 2020), which can avoid the multi-hop bias accumulation. However, such a sophisticated procedure needs to be executed once for each candidate triple, which brings enormous training and test time costs. For example, GraIL (Teru et al., 2020) takes about 1 month to infer on the full FB15k-237 test set (Zhu et al., 2021). Moreover, the enclosing subgraph extraction is also constrained by the path length, severely harming the performance of link prediction.

Therefore, this paper aims to propose a general framework which can: (1) alleviate the deficiency of HLP; (2) enjoy the merits of validity and generality with tractable computational costs.

4 Methodology

4.1 Vertical Learning Paradigm

Inspired by the notion that "*to copy is easier than to memorize*" (Khandelwal et al., 2020), we propose a vertical learning paradigm for KGC task. Different from the implicit memorization strategy of HLP, VLP provides embedding models with the ability to reference related triples as cues for prediction, which can be viewed as an explicit copy strategy.

More concretely, we present the overall pipeline of VLP in Figure 2. Given a query $(h, r, ?)$, the procedure of predicting tail t can be divided into reference query selection, reference graph construction and reference answer aggregation.

Reference Query Selection For the input query $q = (h, r, ?)$, the VLP-based models first select N entity-relation pairs (h_i, r) in the training graph as the reference queries $\{q_i\}_{i=1}^{N}$, which can provide relevant semantics for prediction. For example, to answer (*Jill Biden*, *lives_in*, ?), we can reference the answer-known query (*Joe Biden*, *lives_in*, ?) for target information, since *Joe Biden* and *Jill Biden* are highly related. One intuitive way for the reference selection is to choose the top-k entities in terms of the cosine similarity between h and all entities involved in relation r during the optimization. Nevertheless, this approach incurs high computational costs and is intractable. Numerically, the time complexity of such similarity calculation is $O(n_r d_e)$, where n_r is the number of r-involved entities and $n_r \approx |\mathcal{E}| \gg d_e$ in the worst case.

In this work, inspired by the small world principle (Newman, 2001; Liben-Nowell and Kleinberg, 2007), in which related individuals are connected by short chains (e.g., *Joe Biden* and *Jill Biden* are directly connected by the marriage relationship), we introduce the graph distance based approach for efficient reference query selection. Specifically, we select N r-involved entities $\{h\}_{i=1}^{N}$ closest to h in terms of their relative graph distance (i.e., the shortest path length on the training graph). The corresponding ground-truth targets t_i of the reference queries $q_i = (h_i, r, ?)$ are referred as reference answers. In this way, VLP-based models can pre-retrieve N related references for every input query, thus incurring no additional computational cost for training and inference.

Reference Graph Construction After the efficient reference retrieval, we construct an edge-attributed reference graph to integrate the selected N reference queries and their corresponding answers with the input query. As shown in Figure 2, the input query q is regarded as the central node, and the reference answers t_i are treated as the N neighbors. VLP-based models aims to leverage the explicit reference answers for prediction. However, since there is no guarantee that t_i is the same as the target tail t, it is unreasonable to directly copy t_i without any modification. For example, to answer (*England*, *capital_is*, ?), we cannot directly copy the answer of (*France*, *capital_is*, ?).

Therefore, we introduce the query similarity s_{q,q_i} as the edge attribute between q and t_i. By considering the query differences, VLP-based models are able to adaptively copy the reference answers. For example, to answer the input (*England*, *capital_is*, ?), we can adjust the target information from *Paris* in terms of the difference between (*France*, *capital_is*, ?) and the input query.

Reference Answer Aggregation With the constructed reference graph, VLP-based models learn to explicitly gather target information from neighbor answers for prediction. Specifically, based on the generalized functions summarized in Section 2.2, the central node embedding $\mathbf{q}$ and neighbor node embedding $\mathbf{k}_i$ can be defined as:

$$\begin{aligned} \mathbf{q} &= \mathbf{W}_{r,1}\mathbf{h} + \mathbf{b}_r, \\ \mathbf{k}_i &= \mathbf{W}_{r,2}\mathbf{t}_i. \end{aligned} \tag{8}$$

The edge embedding $\mathbf{s}_{q,q_i}$ (i.e., query similarity embedding) can be further defined as:

$$\begin{aligned} \mathbf{s}_{q,q_i} &= \mathbf{q} - \mathbf{q}_i \\ &= \mathbf{W}_{r,1}(\mathbf{h} - \mathbf{h}_i). \end{aligned} \tag{9}$$

Then, combining the neighbor nodes and edge attributes, VLP-based models aggregate the reference answers to generate the final embedding $\mathbf{t}'$:

$$\begin{aligned} \mathbf{t}' &= \sigma(\mathbf{W}_{agg}[\mathbf{t}_N, \mathbf{q}]), \\ \mathbf{t}_N &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{W}_{node}\mathbf{k}_i + \mathbf{W}_{edge}\mathbf{s}_{q,q_i}), \end{aligned} \tag{10}$$

where $\sigma(\cdot)$ is a nonlinear activation function (e.g., tanh), $[\cdot,\cdot]$ is the concatenate operation, $\mathbf{W}_{agg}$, $\mathbf{W}_{node}$ and $\mathbf{W}_{edge}$ are shared projection matrices. The output $\mathbf{t}'$ should be close to the target tail embedding $\mathbf{t}$ in the latent space, whose score can be revealed by the cosine similarity:

$$f_c(h,r,t) = \frac{\mathbf{t}^\top \mathbf{t}'}{\|\mathbf{t}\|\|\mathbf{t}'\|} \tag{11}$$

We highlight that the VLP's aggregating strategy in Equation (10) differs from GNN-based methods (Vashishth et al., 2020; Bansal et al., 2019; Shang et al., 2019; Schlichtkrull et al., 2018). For each query $(h, r, ?)$, regardless of whether the reference query is a neighbor of h in the training graph, VLP-based models can directly attend to the reference answer throughout the entire training set.

Score Function For each triple (h, r, t) in the test sets, to alleviate the deficiency of HLP and predict more accurately, we integrate the vertical score f_c with the horizontal score f_g to form the final score function f with a weight hyper-parameter λ:

$$f(h,r,t) = f_c(h,r,t) + \lambda f_g(h,r,t). \tag{12}$$

Note that VLP can be widely applied to various embedding models, since the reference aggregation is designed on the generalized score function.

Complexity Analysis Compared with the vanilla embedding models, the VLP-based models only bring a few additional parameters, i.e., the shared aggregation matrices in Equation (10). Therefore, the VLP-based models have the same space complexity as the HLP-based models, i.e., $O(|\mathcal{E}|\, d_e)$. In the aspect of time cost for processing single triple, the time complexity of vanilla embedding models is $O(d_r d_e)$, derived from the generalized score function in Equation (4). The VLP-based models require the same computation for each reference, which produces the time complexity of $O(N d_r d_e)$. Such computation is tractable since a small N (no more than 8) is enough for VLP-based models to achieve high performance in the experiments.

4.2 Optimization

During training, we jointly optimize f_c and f_g by a two-component loss function with coefficient α:

$$L = L_1 + \alpha L_2. \tag{13}$$

.

For the former one, we use the cross-entropy between predictions and labels as training loss:

$$L_1 = -\sum_{i=1}^{|\mathcal{E}|} y_i \log p_i, \tag{14}$$

where p_i and y_i are the i-th components of $\mathbf{p}$ and $\mathbf{y}$, respectively; $\mathbf{p} \in \mathbb{R}^{|\mathcal{E}|}$ is calculated by applying the softmax function to the "1-to-All" (Lacroix et al., 2018a) results of f_c; $\mathbf{y} \in \mathbb{R}^{|\mathcal{E}|}$ is the one-hop vector that indicates the position of true label.

For the later one, negative sampling has been proved quite effective in extensive works (Song et al., 2021; Sun et al., 2019). Formally, for a positive triple (h, r, t), we first sample a set of entities $\{t'_i\}_{i=1}^{l}$ (or $\{h'_i\}_{i=1}^{l}$) based on the *pre-sampling weights* p_0 to construct negative triples (h, r, t'_i) (or (h'_i, r, t)). With these samples, a negative sampling loss is designed to optimize embedding models:

$$\begin{aligned} L_2 = &-\sum_{i=1}^{l} p_1(h'_i, r, t'_i) \log \sigma(-f(h'_i, r, t'_i) - \gamma) \\ &- \log \sigma(\gamma + f(h, r, t)), \end{aligned} \tag{15}$$

where γ is a pre-defined margin, σ is the sigmoid function, l denotes the number of negative samples, (h'_i, r, t'_i) is a negative sample against (h, r, t). Importantly, $p_1(h'_i, r, t'_i)$ is the *post-sampling weight*, which determines the proportion of (h'_i, r, t'_i) in the current optimization.

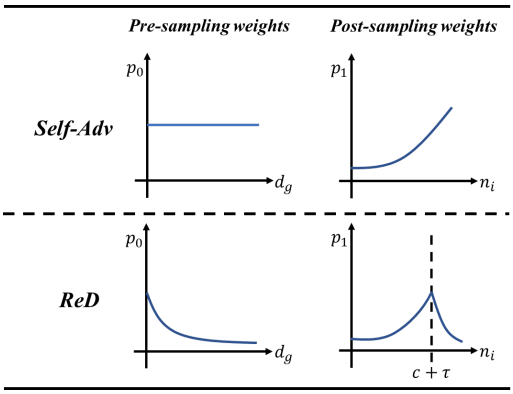


Figure 3: Comparison between ReD and Self-Adv.

As shown in Figure 3, recent works (Song et al., 2021; Chao et al., 2021; Gao et al., 2020; Sun et al., 2019) utilize the self-adversarial technique (Self-Adv), in which the pre-sampling weights follow a uniform distribution and the post-sampling weights increase with the negative scores. Differently, in this work, we propose a new approach named ReD based on the relative distance, which can draw more informative negative samples and reduce the toxicity of false negative samples.

For the pre-sampling weights, considering the deficiency of embedding models as described in Section 3, the distant entities are usually hard to be predicted as the target answer. It reveals a rational priori, i.e., distant entities are more likely to form easy (meaningless) negative triples. This inspires us to sample more hard (informative) negative triples based on the relative graph distance d_g. As shown in Figure 3, the pre-sampling weight in ReD decreases with the increase of graph distance between head and tail entities. Formally, for a training query $(h, r, ?)$, we pre-sample entities t' to construct negatives from the following distribution:

$$p_0(h, r, t') = \frac{\exp -\alpha_0 d_g(h, t')}{\sum_{i=1}^{|\mathcal{E}|} \exp -\alpha_0 d_g(h, t'_i)}, \quad (16)$$

where α_0 is the pre-sampling temperature, $d_g(\cdot, \cdot)$ outputs the relative graph distance between two entities. Note that the calculation of $d_g(\cdot, \cdot)$ is a one-time preprocessing step, which will not bring additional training overhead.

For the post-sampling weights, Self-Adv assigns greater weights to high scoring negative triples in Equation (15), which makes the optimization focus more on hard negatives. However, this monotonically increasing strategy ignores the issue of false negatives, since triples with higher scores are more likely to be correct. A more rational posteriori is that the easy negatives are underscored and the false negatives are overscored. In this work, we use the relative latent distance between the positive and negative samples to determine whether the negative score is too low or too high. Specifically, ReD defines the post-sampling weights as a distribution that first rises and then falls as the negative score increases. As shown in Figure 3, if the negative score is significantly greater than (or less than) the positive score, this negative sample is more likely to be false (or easy), and thus be assigned a small weight in the Equation 15. Formally, based on the positive score $c = f_g(h, r, t)$ and negative score $n_i = f_g(h'_i, r, t'_i)$, the post-sampling weight in ReD is defined as:

$$p_1(h'_i, r, t'_i) = \frac{\exp w(h'_i, r, t'_i)}{\sum_j \exp w(h'_j, r, t'_j)},$$
$$w(h'_i, r, t'_i) = \begin{cases} \alpha_1 n_i, & n_i \le c + \tau \\ \alpha_1 c - \alpha_2 m_i, & n_i > c + \tau \end{cases},$$
$$m_i = n_i - c - \tau, \quad (17)$$

where α_1 and α_2 are the post-sampling temperatures. By combining the sampling weights in Equation (16) and (17), ReD is able to generate and process higher quality negatives for optimization.

5 Experiment

5.1 Experimental Setup

Datasets We evaluate our proposal on two widely-used benchmarks: WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015). More details can be found in Appendix B.

Baselines To verify the effectiveness and generality of our proposal, we combine the proposed techniques with three representative embedding models DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) and RotatE (Sun et al., 2019). For performance comparison, we select a series of embedding models as baselines in Table 2.

Implementation Details We fine-tune the hyperparameters with the grid search on the validation sets. Please see Appendix C for more details.

5.2 Main Results

The experimental results are reported in Table 2. Compared to DistMult, ComplEx and RotatE, all three VLP-based versions achieve consistent and significant improvements on both datasets. For example, on WN18RR and FB15k-237 datasets, RotatE-VLP outperforms RotatE with 2.2% and

Model	WN18RR				FB15k-237			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE (Bordes et al., 2013)†	.226	-	-	.501	.294	-	-	.465
ConvE (Dettmers et al., 2018)	.43	.40	.44	.52	.325	.237	.356	.501
A2N (Bansal et al., 2019)	.45	.42	.46	.51	.317	.232	.348	.486
QuatE (Zhang et al., 2019)	.481	.436	.500	.564	.311	.221	.342	.495
CompGCN (Vashishth et al., 2020)	.479	.443	.494	.546	.355	.264	.390	.535
PairRE (Chao et al., 2021)	.455	.413	.469	.539	.348	.254	.384	.539
DualE (Cao et al., 2021)	.482	.440	.500	.561	.330	.237	.363	.518
Rot-Pro (Song et al., 2021)	.457	.397	.482	.577	.344	.246	.383	.540
CAKE (Niu et al., 2022)	-	-	-	-	.321	.226	.355	.515
REP (Wang et al., 2022)	.488	.439	.505	**.588**	.354	.262	.388	.540
ReflectE (Zhang et al., 2022)	.488	.450	.501	.559	.358	.263	.396	**.546**
DistMult (Yang et al., 2015)◇	.439	.392	.453	.534	.308	.220	.337	.485
DistMult-VLP	.462	.421	.474	.545	.347	.256	.379	.528
ComplEx (Trouillon et al., 2016)◇	.466	.423	.484	.552	.328	.235	.354	.511
ComplEx-VLP	.494	.450	.508	.580	.354	.258	.396	.536
RotatE (Sun et al., 2019)	.476	.428	.492	.571	.338	.241	.375	.533
RotatE-VLP	**.498**	**.455**	**.514**	.582	**.362**	**.271**	**.397**	.542

Table 2: Link prediction results on WN18RR and FB15k-237. Best results are in **bold** and second best results are underlined. [†]: Results are taken from (Nguyen et al., 2018). [◇]: we re-evaluate DistMult and ComplEx based on the open source codes from (Sun et al., 2019), achieving better results than those reported in the original papers.

Distance d_{ht}	1 (47.7%)	2 (12.7%)	3 (29.3%)	4 (10.3%)
DistMult	0.971	0.331	0.293	0.039
DistMult-VLP	0.989	0.345	0.328	0.053
Relative Imp.	**+1.9%**	**+4.2%**	**+11.9%**	**+35.9%**
ComplEx	0.979	0.367	0.396	0.058
ComplEx-VLP	0.985	0.400	0.449	0.102
Relative Imp.	**+0.6%**	**+9.0%**	**+13.4%**	**+75.9%**
RotatE	0.986	0.375	0.378	0.091
RotatE-VLP	0.991	0.391	0.456	0.111
Relative Imp.	**+0.5%**	**+4.3%**	**+20.6%**	**+22.0%**

Table 3: MRR on each distance split of WN18RR.

Relation Name	RotatE	QuatE	RotatE-VLP
hypernym	0.154	0.172	**0.191**
instance_hypernym	0.324	0.362	**0.376**
member_meronym	0.255	0.236	**0.269**
synset_domain_topic_of	0.334	0.395	**0.411**
has_part	0.205	0.210	**0.220**
member_of_domain_usage	0.277	0.372	**0.375**
member_of_domain_region	0.243	0.140	**0.391**
derivationally_related_form	0.957	0.952	**0.958**
also_see	0.627	0.607	**0.635**
verb_group	0.968	0.930	**0.968**
similar_to	1.000	1.000	**1.000**

Table 4: MRR on each relation of WN18RR.

2.4% absolute improvements in MRR, respectively. Such obvious gains reveal that the vertical contexts generally inject valuable information into the embedding models for more accurate prediction.

Moreover, one can further see that ComplEx-VLP and RotatE-VLP perform competitively with the SOTA baselines. Specifically, RotatE-VLP surpasses all the baselines in terms of most metrics over both datasets; ComplEx-VLP also achieves promising performance on FB15k-237 compared with the baselines. The superior performance further confirms the effectiveness of our proposal.

5.3 Fine-grained Performance Analysis

Performance on Distance Splits Table 3 reports the performance of three VLP-based models on the distance splits defined in Appendix A. One can observe that: (1) the VLP-based embedding models outperform the vanilla models across all the distance splits; (2) the VLP models achieve greater relative improvement on the split with larger d_{ht}. For example, as d_{ht} increases from 1 to 4, RotatE-VLP achieves 0.5%, 4.3%, 20.6% and 22.0% relative improvements over RotatE on the MRR metric, respectively. This reveals that the explicit vertical contexts can significantly alleviate the limitations of memory strategy in the embedding models.

Performance on Each Relation To verify the modeling capacity of our proposal from a fine-grained perspective, we explore the performance of VLP-based models on each relation of WN18RR following (Zhang et al., 2019). As shown in Table 4, compared to RotatE and QuatE, RotatE-VLP surpasses them on all the 11 relation types, confirming that the explicit reference aggregation brings superior modeling capacity.

Performance on Mapping Properties Table 5 exhibits the performance of our proposal on different relation mapping properties (Sun et al., 2019) in FB15k-237. We observe that RotatE-VLP consistently outperforms RotatE across all RMP types. Such advanced performance owes to the powerful modeling capability of the explicit copy strategy.

Task	RMPs	RotatE	RotatE-VLP
Predicting Head (MRR)	1-to-1	0.498	**0.504**
	1-to-N	0.475	**0.478**
	N-to-1	0.088	**0.126**
	N-to-N	0.260	**0.286**
Predicting Tail (MRR)	1-to-1	0.490	**0.499**
	1-to-N	0.071	**0.093**
	N-to-1	0.747	**0.770**
	N-to-N	0.367	**0.388**

Table 5: MRR on mapping properties in FB15k-237.

5.4 Impact of Reference Quantity

VLP aggregates target information from N references pre-selected before training. We investigate the impact of N on the performance (MRR) of VLP-based models. Figure 4 shows the results on WN18RR dataset. As expected, all three VLP-based models with more vertical references achieve better performance than the ones with fewer references, since the aggregation of sufficient references brings the superior modeling capacity. Moreover, we can observe that the models can achieve high performance with N less than 10, making the computation tractable as discussed in Section 4.1.

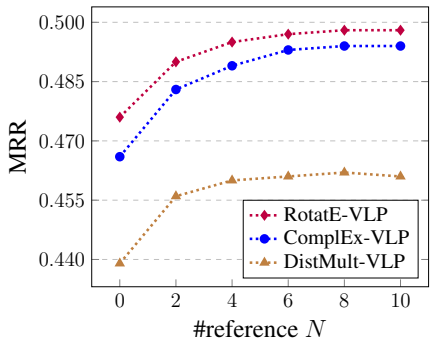


Figure 4: Impact of reference quantity on WN18RR.

5.5 Ablation Study of ReD

To explore the effectiveness of the proposed ReD, we conduct ablation studies on the pre-sampling and post-sampling parts of the three VLP-based models. Table 6 shows the detailed results. We can observe that the removal of any part reduces the performance, which demonstrates that ReD makes the model focus more on meaningful negative samples for more effective optimization. Moreover, we also integrate ReD with original embedding models to verify the generality of this technique. Please refer to Appendix D for more results.

6 Related Work

Embedding models can be roughly categorized into distance based models and semantic matching models (Chao et al., 2021).

	WN18RR		FB15k-237	
Model	MRR	H@10	MRR	H@10
DistMult-VLP	0.462	0.545	0.347	0.528
w/o pre.	0.456	0.537	0.338	0.518
w/o post.	0.458	0.542	0.344	0.525
ComplEx-VLP	0.494	0.580	0.354	0.536
w/o pre.	0.491	0.579	0.344	0.529
w/o post.	0.493	0.580	0.345	0.531
RotatE-VLP	0.498	0.582	0.362	0.542
w/o pre.	0.493	0.578	0.355	0.540
w/o post.	0.496	0.580	0.359	0.539

Table 6: Ablation study of ReD.

Distance based models use the Euclidean distance to measure the plausibility of each triple. A series of work is conducted along this line, such as TransE (Bordes et al., 2013) TransH (Wang et al., 2014), TransR (Lin et al., 2015), RotatE (Sun et al., 2019), PairRE (Chao et al., 2021), Rot-Pro (Song et al., 2021), ReflectE (Zhang et al., 2022) and so on. TransE and RotatE are the most representative distance-based models, which represent relations as translations and rotations, respectively. Semantic matching models utilize multiplicative functions to score each triple, including RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), QuatE (Zhang et al., 2019), DualE (Cao et al., 2021) and so on. Typically, RESCAL (Nickel et al., 2011) defines each relation as the tensor decomposition matrix. DistMult (Yang et al., 2015) simplifies the relation matrices to be diagonal for preventing overfitting. However, existing embedding models essentially follow the horizontal learning paradigm, underperforming in predicting links between distant entities.

Moreover, some advanced techniques are proposed to improve embedding models, such as graph encoders (Schlichtkrull et al., 2018; Shang et al., 2019; Vashishth et al., 2020; Wang et al., 2022) and regularizers (Lacroix et al., 2018b). Note that our proposals are orthogonal to these techniques, and one can integrate them for better performance.

7 Conclusion

In this paper, we present a novel learning paradigm named VLP for KGC task. VLP can be viewed as an explicit copy strategy, which allows embedding models to consult related triples for explicit references, making it much easier to predict distant links. Moreover, we also propose ReD, a new negative sampling technique for more effective optimization. The in-depth experiments on two datasets demonstrate the validity and generality of our proposals.

Limitations

Although our proposal enjoys the advantages of validity and generality, there are still two major limitations. First, VLP cannot directly generalize to the inductive setting, since VLP is defined based on the score functions of transductive embedding models. One potential direction is to design an inductive reference selector for emerging entities. Second, how to efficiently select more helpful references for prediction is still an open challenge. We expect future studies to mitigate these issues.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62276044, and also Sponsored by CAAI-Huawei MindSpore Open Fund.

References

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 4387–4392.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 6894–6902.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. Pairre: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4360–4369.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1811–1818.

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *International Conference on Information and Knowledge Management*, pages 385–394.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018a. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2869–2878.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018b. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2869–2878.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, Xing Xie, and Qi Zhang. 2022. House: Knowledge graph embedding with householder parameterization. In *International Conference on Machine Learning*, pages 13209–13224.

David Liben-Nowell and Jon M. Kleinberg. 2007. The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.*, 58(7):1019–1031.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Mark EJ Newman. 2001. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–333.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816.

Guanglin Niu, Bo Li, Yongfei Zhang, and Shiliang Pu. 2022. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2867–2877.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, volume 10843, pages 593–607.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3060–3067.

Tengwei Song, Jie Luo, and Lei Huang. 2021. Rotpro: Modeling transitivity by projection in knowledge graph embedding. In *Advances in Neural Information Processing Systems*, pages 24695–24706.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations*.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9448–9457.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 2071–2080.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations*.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2021. Relational message passing for knowledge graph completion. In *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1697–1707.

Huijuan Wang, Siming Dai, Weiyue Su, Hui Zhong, Zeyang Fang, Zhengjie Huang, Shikun Feng, Zeyu Chen, Yu Sun, and Dianhai Yu. 2022. Simple and effective relation-based embedding propagation for knowledge representation learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2755–2761.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1271–1279.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations*.

Qianjin Zhang, Ronggui Wang, Juan Yang, and Lixia Xue. 2022. Knowledge graph embedding by reflection transformation. *Knowl. Based Syst.*, 238:107861.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2731–2741.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems*, pages 29476–29490.

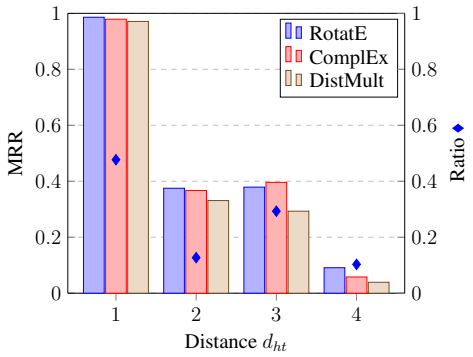
A Experimental Observation

The motive of our work originates from an experimental observation, which shows that embedding models underperform in predicting links between distant entity pairs. Specifically, according to the relative graph distance d_{ht} between head and tail entities of each test triple, we divide the test sets of WN18RR and FB15k-237 into four splits. Three representative embedding models (DistMult, ComplEx and RotatE) are tested on each split.

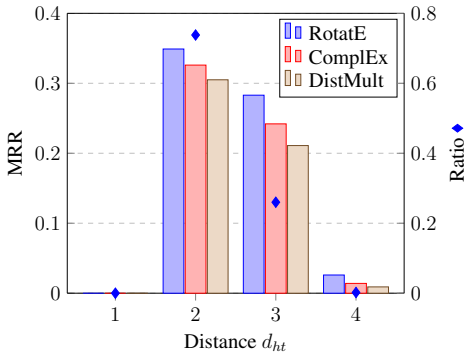
Figure 5 summarizes the detailed MRR results and split ratios on the two datasets. We can observe that all three embedding models achieve promising results in link prediction between close entities, while the performance drops significantly in the prediction between distant entities. For example, on the split where $d_{ht} = 1$ in WN18RR, RotatE achieves excellent performance (MRR of 0.986), while on the split where $d_{ht} = 2$, the performance of RotatE decreases by about 62% (MRR of 0.375).

B Datasets

Table 7 summarizes the detailed statistics of two benchmark datasets. WN18RR (Dettmers et al.,



(a) MRR and ratio for each split in WN18RR



(b) MRR and ratio for each split in FB15k-237

Figure 5: The MRR results of three popular embedding models (DistMult, ComplEx and RotatE) tested on each distance split in WN18RR and FB15k-237 datasets. The blue diamond marks denote the ratio of each test split.

Dataset	WN18RR	FB15k-237
#entity	40,943	14,541
#relation	11	237
#training	86,835	272,115
#validation	3,034	17,535
#test	3,134	20,466

Table 7: Statistics of five standard benchmarks.

Hyperparameter	Search Space
b	$\{256, 512, 1024\}$
d	$\{500, 1000\}$
$\alpha_0, \alpha_1, \alpha_2$	$\{0.1, 0.5, 1.0, 1.5\}$
λ	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$
γ	$\{4, 6, 8, 11, 15\}$

Table 8: Hyperparameter search space.

2018) and FB15k-237 (Toutanova and Chen, 2015) datasets are subsets of WN18 (Bordes et al., 2013) and FB15k (Bordes et al., 2013) respectively with inverse relations removed. WN18 is extracted from WordNet (Miller, 1995), a database featuring lexical relations between words. FB15k is extracted from Freebase (Bollacker et al., 2008), a large-scale KG containing general knowledge facts.

C Implementation Details

We use Adam (Kingma and Ba, 2015) as the optimizer and fine-tune the hyperparameters on the validation dataset. The hyperparameters are tuned by the grid search, including batch size b, embedding dimension d, negative sampling temperatures $\{\alpha_i\}_{i=0}^{2}$, loss weight λ and fixed margin γ. The hyper-parameter search space is shown in Table 8.

D Embedding Models with ReD

To verify the generality of the proposed negative sampling technique ReD, we integrate ReD with three representative embedding models (i.e., DistMult, ComplEx and RotatE) for KGC task. As shown in Table 9, compared to Self-Adv, the embedding models combined with ReD achieve better performance on both datasets, since ReD guarantees more informative negative samples from both pre-sampling and post-sampling stages.

	WN18RR		FB15k-237	
Model	MRR	H@10	MRR	H@10
DistMult-Adv	0.439	0.534	0.308	0.485
DistMult-ReD	0.445	0.539	0.315	0.491
ComplEx-Adv	0.466	0.552	0.328	0.511
ComplEx-ReD	0.470	0.554	0.335	0.516
RotatE-Adv	0.476	0.571	0.338	0.533
RotatE-ReD	0.478	0.572	0.344	0.536

Table 9: Results of DistMult, ComplEx and RotatE with different negative sampling techniques. X-Adv denotes the embedding model X combined with Self-Adv.