KGCRR: An Effective Metric-Driven Knowledge Graph Completion Framework by Designing a Novel Upper Bound Function with Adaptive Approximation to Reciprocal Rank

Kuan Xu[1, 2], Kuo Yang*[1, 2], Jian Liu[4], Xiangkui Lu[1, 3], Jun Wu[1, 3], Xuezhong Zhou*[1, 2]

[1]School of Computer Science & Technology, Beijing Jiaotong University, Beijing, 100044, China
[2]Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, 100044, China
[3] MoE Key Lab of Big Data & Artificial Intelligence in Transportation, Beijing Jiaotong University, Beijing 100044, China
[4]University of Science and Technology Beijing, Beijing, 100083, China
{xukuan, kuoyang, luxkui, wuj, xzzhou}@bjtu.edu.cn, jian.liu@ustb.edu.cn

Abstract

Knowledge Graph Embedding (KGE) methods have achieved great success in predicting missing links in knowledge graphs, a task also known as Knowledge Graph Completion (KGC). Under this task, the Reciprocal Rank (RR) of ground-truth items serve as a key indicator for evaluating the method's performance. However, most existing studies have overlooked the inconsistency between the ranking metric, RR, and the optimization objective functions, resulting in sub-optimal KGC performance. To address this issue, we propose a KGC framework called KGCRR, which introduces an objective function named CRR that serves as an upper bound to RR. By introducing the parameter-pressure ρ to adjust the sigmoid function, CRR achieves a better approximation to RR compared to existing objective functions. We theoretically prove that by adjusting ρ, CRR can achieve a more effective approximation to RR. By narrowing the discrepancy with RR and alleviating the gradient vanishing issue associated with the direct optimization of RR loss, CRR demonstrates an advantage in optimizing RR. CRR serves as a plug-and-play objective, capable of seamless integration into various KGE methods. Through extensive experiments conducted on FB15k-237 and WN18RR datasets, we have obtained promising results, with an average improvement of 19.06% in MRR, indicating that CRR significantly enhances the performance of existing methods.

1 Introduction

Knowledge Graphs (KGs) play a crucial role in enhancing accurate information retrieval and supporting decision-making processes in the real world (Rossi et al. 2021; Chen et al. 2024). However, even the most advanced KGs still suffer from incompleteness (Shen, Zhang, and Cheng 2022; Xie and Ge 2023; Meilicke et al. 2024), which restricts practical applications and has spurred research into Knowledge Graph Completion (KGC), a task aimed at inferring and filling in the most plausible missing facts to enrich KGs.

In KGs, each fact is represented by a triple in the form of (*head entity*, *relation*, *tail entity*) or (h, r, t) for short. The task of KGC involves predicting the missing h or t given the other components of the triple (Tang et al. 2022; Xie and Ge 2023). Under this task, for each missing triple, candidate entities are scored and ranked in descending order with the top-n (e.g., 1, 10) items selected to complete the knowledge graph (Chen et al. 2024). In recent years, numurous Knowledge Graph Embedding (KGE) methods (Bordes et al. 2013; Trouillon et al. 2016; Chao et al. 2021; Ge et al. 2023) have achieved great success in KGC. Under this task, the ranking metric Reciprocal Rank (RR) of ground-truth items serves as a key indicator for evaluating the method's performance. However most existing methods overlook the inconsistency between the ranking metric and the optimization objective functions, such as Log-Likelihood (LL) loss and margin-based loss (Ge et al. 2024), as shown in Figure 1, resulting in a scenario where the model's training is not directly aligned with the desired metric for KGC. This misalignment may lead to a sub-optimal performance, where the model might optimize for a lower loss but fail to improve in terms of the ranking metrics like MRR[1], which are critical

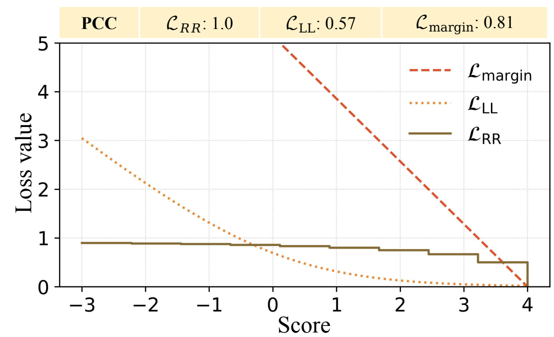

Figure 1: Illustration of the inconsistency between the ranking metric and optimization objective, where score indicates the predicted score of ground-truth item. $\mathcal{L}_{\mathrm{margin}}$: the margin-based loss; $\mathcal{L}_{\mathrm{LL}}$: the Log-Likelihood loss; $\mathcal{L}_{\mathrm{RR}}$: the Reciprocal Rank loss; PCC: Pearson correlation coefficient. The inconsistency arises between the RR and optimization objectives e.g., Log-Likelihood loss and margin-based loss.

*the Corresponding author

[1]MRR is the mean value of Reciprocal Ranks

for evaluating the quality of KGC.

An intuitive approach to address this issue is to directly optimize the metric objective. Numerous existing works have pursued this strategy on other tasks, such as 'Smooth-AP'(Brown et al. 2020) and 'Smooth-AUC'(Tang, Luo, and Wu 2022). However, this approach is not sufficiently effective for RR optimization. As shown in Figure 2, with the Rank increases, the curve of the RR loss approaches a constant function, represented as $\mathcal{L} = 1$, thereby diminishing its discriminative capability. Specifically, the gradient change of it approaches 0, presenting a challenge to the convergence process with gradient-descent methods. Given that direct optimization of RR is not effective, a common approach is to optimize an upper bound function of RR (Bubeck et al. 2015), e.g., the Rank loss, to indirectly achieve optimization of RR. However, although the Rank loss serves as an upper bound for the RR loss, a great gap exists between these two functions, as shown in Figure 2. Consequently, directly optimizing the Rank loss is also impractical. Furthermore, the Rank loss does not sufficiently emphasize the importance of top ranking positions[2], which is a priority for the RR metric. An effective upper bound function should closely approximate the original function to ensure that the solution obtained by optimizing the upper bound provides a good solution to the original problem (Mohapatra et al. 2018).

Therefore, for RR optimization, inspired by (Lu, Wu, and Yuan 2023), we propose a KGC framework called **KGCRR**, which introduces an objective function named **CRR** that serves as an upper bound to achieve a Close approximation to RR in this study. CRR applies a nonlinear transformation to Rank loss to emphasize top-ranking positions and incorporates a parameter-pressure ρ, to achieve a close approximation to RR loss while maintaining global differentiability. We theoretically proved that by adjusting ρ, CRR can achieve a more effective approximation to RR. By narrowing the discrepancy with RR loss, and mitigating the gradient vanishing issue associated with direct optimization of RR loss, the CRR loss demonstrates an advantage in optimizing RR compared with other losses. CRR can be dynamically adapted and serves as a plug-and-play objective, capable of being seamlessly integrated into various KGE models. Through extensive experiments conducted on FB15k-237 and WN18RR datasets, we have obtained promising results, with an average improvement of 19.06% in MRR, indicating that CRR has the general potential to enhance the performance of existing KGC methods.

Our contributions of this work are as follows:

- We propose a new upper bound function of RR, named **CRR**, and theoretically prove that by adjusting parameter temperature-ρ, CRR can achieve more effective approximation to RR compared to existing objective functions;
- CRR is dynamically adaptable and serves as a plug-and-play objective, capable of being seamlessly integrated into various KGE methods, yielding an average improvement of 19.06% in MRR on FB15k-237 and WN18RR;

[2]Gradient changes at top-ranking positions (e.g., 1, 10) are more significant compared to lower-ranking positions (e.g., 20, 30).

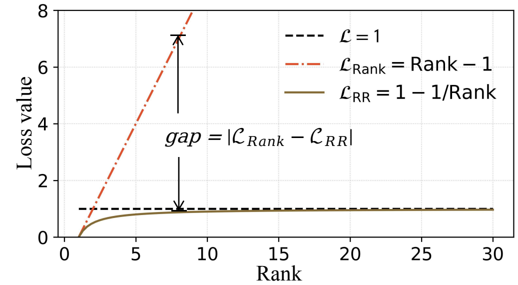

Figure 2: Illustration of the Rank loss and RR loss. $\mathcal{L}_{\text{Rank}}$: the Rank loss; $\mathcal{L}_{\text{RR}}$: the RR loss. Due to gradient vanishing, directly optimizing $\mathcal{L}_{\text{RR}}$ is not sufficiently effective. The upper bound function $\mathcal{L}_{\text{Rank}}$ is also not working, since a great gap exists between $\mathcal{L}_{\text{Rank}}$ and $\mathcal{L}_{\text{RR}}$. And $\mathcal{L}_{\text{Rank}}$ is limited in emphasis on top ranking positions, which $\mathcal{L}_{\text{RR}}$ prioritizes.

- Extensive experiments conducted on FB15k-237 and WN18RR datasets demonstrate the effectiveness of CRR, highlighting its capability in enhancing KGC.

2 Related Work

Knowledge graph embedding methods have demonstrated great success in the task of KGC. The geometric models interpret relations as geometric transformations that map the head entity to the tail entity within a latent space. Notable examples include RotatE (Sun et al. 2019), TransE (Bordes et al. 2013) and others (Chao et al. 2021; Ge et al. 2023). Compared with geometric models, semantic matching models such as ComplEx (Trouillon et al. 2016), DistMult (Yang et al. 2014), and ConvE (Dettmers et al. 2018) capture and utilize the latent semantic information of entities and relations in vector space, thereby enhancing confidence in the learned embeddings. Besides these, tensor decomposition models such as SimplE (Kazemi and Poole 2018) and TuckER (Balažević, Allen, and Hospedales 2019), as well as the analogical inference model ANALOGY (Liu, Wu, and Yang 2017), have also achieved success in KGC. In semantic matching models, adapting graph neural networks for KGE has also gained significant attention in recent years (Schlichtkrull et al. 2018; Li et al. 2022). However, despite the variety of existing methods, research targeting the optimization of RR for KGC task remains underexplored.

The field of metric optimization remains an active area of research (Brown et al. 2020; Tang, Luo, and Wu 2022). In the context of enhancing information retrieval quality (Rolínek et al. 2020; Cakir et al. 2019; Li 2011), various strategies have been proposed, including direct minimization of Average Precision (AP) loss (Oh Song et al. 2016; Henderson and Ferrari 2017), optimization of a smooth upper bound for AP (Mohapatra et al. 2018), and optimization of a close approximation of AP (Brown et al. 2020). Similarly, significant advancements have been achieved in optimizing the Area Under the ROC Curve (Tang, Luo, and Wu 2022; Shen, Yang, and Gao 2020; Herschtal and Raskutti 2004), highlighting the great progress in this domain. All of these provide valuable insights for the optimization of RR in

KGC within this study.

3 Preliminaries

A knowledge graph can be represented as $\mathcal{K} = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, wherein h, r, and t symbolize the *head entity*, *relation*, and *tail entity* of the triple (h, r, t) respectively. $\mathcal{E}$ and $\mathcal{R}$ correspond to the sets of entities and relations respectively. In this study, the KGC problem is defined to deduce the most likely missing triple from $\{(h, r, t)|t \in \mathcal{E} \wedge (h, r, t) \notin \mathcal{K}\}$ for each incomplete triple $(h, r, ?)$. The benchmark metric for evaluating the quality of KGC is the Mean Reciprocal Rank (MRR) of ground-truth items.

In the context of current issue, let the set $\mathcal{T} = \{t_1, t_2, ..., t_{n-1}, t_n\}$ represents the pool of potential entities for each incomplete triple $(h, r, ?)$, with n denoting the total count of elements in $\mathcal{T}$. For each incomplete triple, the KGC model is designed to generate a score vector $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_{n-1}, \hat{y}_n]$, where $\hat{y}_i$ is the predicted score of triple (h, r, t_i). A higher score indicates a greater likelihood of the existence of a triple. The corresponding ground truth can be presented as a one-hot vector $\mathbf{y} = [y_1, y_2, ..., y_{n-1}, y_n]$, where $\mathbf{y} \in \{0, 1\}^n$. Following the filter setting (Dettmers et al. 2018), for each incomplete triple, there exists only one ground-truth item within the entity set $\mathcal{T}$.

Given the output vector $\hat{\mathbf{y}}$ and ground truth $\mathbf{y}$, for the ground-truth item in each triple, the Rank can be computed as follows:

$$\mathrm{Rank} = \sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{I}\{\hat{y}_k - \hat{y}_j\}\right), \tag{1}$$

$$\mathbb{I}\{x\} = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0. \end{cases} \tag{2}$$

For convenience, Equation (1) can be implemented by computing a difference matrix $D \in \mathbb{R}^{n \times n}$:

$$D = \begin{bmatrix} \hat{y}_1 & \cdots & \hat{y}_n \\ \vdots & \ddots & \vdots \\ \hat{y}_1 & \cdots & \hat{y}_n \end{bmatrix} - \begin{bmatrix} \hat{y}_1 & \cdots & \hat{y}_1 \\ \vdots & \ddots & \vdots \\ \hat{y}_n & \cdots & \hat{y}_n \end{bmatrix}. \tag{3}$$

Thus, the computation of Rank can be reformulated as:

$$\mathrm{Rank} = \sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{I}\{D_{jk}\}\right). \tag{4}$$

Then, the RR, defined as the reciprocal of the Rank, can be obtained by:

$$\mathrm{RR} = \frac{1}{\mathrm{Rank}} \tag{5}$$

$$= \frac{1}{\sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{I}\{D_{jk}\}\right)}. \tag{6}$$

Given a set of incomplete triples $\mathcal{I}$, the MRR is calculated across all triples as follows:

$$\mathrm{MRR} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathrm{RR}_i, \tag{7}$$

where RR_i represents the RR score of triple i as derived from Equation (5).

4 The Proposed Method

In this section, we first introduce the construction process of CRR. Following this, we conduct a comparative analysis with existing objective functions to highlight the effectiveness of CRR. Specifically, CRR demonstrates better consistency with RR compared to existing objective functions.

4.1 CRR

Construction Process of CRR Inspired by previous work on AUC (Tang, Luo, and Wu 2022) and AP (Brown et al. 2020) optimization, in the pursuit of optimizing the RR metric, the straightforward approach is the minimization of RR loss, expressed as:

$$\mathcal{L}_{\mathrm{RR}} = 1 - \frac{1}{\mathrm{Rank}} \tag{8}$$

$$= 1 - \frac{1}{\sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{G}(D_{jk}, \tau)\right)}, \tag{9}$$

$$\mathbb{G}(x, \tau) = \frac{1}{1 + e^{\frac{-x}{\tau}}} \equiv \mathbb{I}\{x\}, \tag{10}$$

where D is difference matrix of predicted scores. $\mathbb{G}(x, \tau)$ is the sigmoid function with temperature τ, which is employed as a replacement for the Indicator function $\mathbb{I}\{x\}$ to tackle the non-differentiability challenge with gradient-descent methods (Tang, Luo, and Wu 2022). However, as discussed in Section 1, directly optimizing RR loss presents challenges due to its limited discriminative capacity, making the convergence process difficult with gradient-descent methods.

As Bubeck et al. elucidate in their work on convex optimization (Bubeck et al. 2015), when direct optimization of a function presents challenges, an effective approach is to optimize a surrogate upper bound of the function instead. This strategy not only simplifies the optimization process but also ensures that improvements in the surrogate directly translate to enhancements in the original function's performance. Consequently, to address this challenge, optimizing its upper bound-the Rank loss may serve as a plausible alternative, which can be defined as follows:

$$\mathcal{L}_{\mathrm{Rank}} = \mathrm{Rank} - 1 \tag{11}$$

$$= \sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{G}(D_{jk}, \tau)\right) - 1. \tag{12}$$

However, the issue with Rank loss lies in its limited approximation to the RR loss, making it insufficiently effective when optimized. An effective upper bound function should closely approximate the original function to ensure that the solution obtained by optimizing the upper bound provides a good solution to the original problem (Camponogara, Nazari et al. 2015; Mohapatra et al. 2018). Therefore, we aim to develop an objective function that offers a more effective approximation to the RR loss to achieve better RR outcome.

To effectively simulate the behavior of the indicator function, prior researchs (Brown et al. 2020; Tang, Luo, and Wu 2022) employs a small value for the temperature τ (e.g., 0.001). However, our findings indicate that as τ increases,

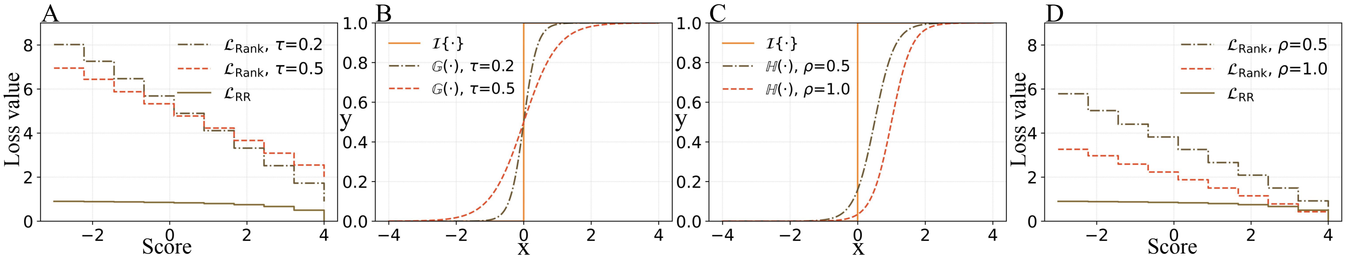

Figure 3: The impact of parameter temperature-τ and pressure-ρ on Rank loss, where score indicates the predicted score of ground-truth item. Figures **A** and **B** show the effects of the temperature parameter on Rank loss smoothing, as well as its limitations in approximating RR loss when the ground-truth item has a high score, due to the inherent constraints of the sigmoid function. Figures **C** and **D** show that the parameter pressure-ρ enhances the sigmoid function's approximation to the Indicator function when $x < 0$ by shifting the sigmoid function, thereby reducing the gap between Rank loss and RR loss and improving its approximation to RR loss. The pressure ρ is fixed to 0.0 in **A** and **B**. The temperature τ is fixed to 0.1 in **C** and **D**.

the Rank loss curve becomes smoother (Figure 3A), bringing its trend more closely in line with that of the RR loss to some extent. However, another issue that arises is that Rank loss exhibits characteristics of global smoothness. This results in a scenario where, when the predicted score of the ground-truth item is higher (i.e., ranking closer to the top), the Rank loss does not closely approximate the RR loss as effectively. The underlying cause of this issue stems from the limitation of sigmoid function in approximating the Indicator function effectively (Figure 3B), particularly when the ground-truth items achieve high scores.

To mitigate this issue, we introduce a new parameter, termed 'pressure ρ'. Thus, the function $\mathbb{G}(x, \tau)$ can be reformulated as follows:

$$\mathbb{H}(x;\rho;\tau) = \frac{1}{1 + e^{\frac{\rho - x}{\tau}}}. \tag{13}$$

As evident from Equation (13), the primary role of the pressure ρ is to control the shift of the sigmoid function. When pressure parameter ρ is set to 0 and temperature parameter τ is set to 1, respectively, Equation (13) simplifies to a sigmoid function. As the pressure parameter ρ increases, it forces the sigmoid function to more closely approximate the Indicator function when $D_{jk} < 0$ (Figure 3C, $x < 0$). This adjustment facilitates the smoothing of Rank loss, and enhances its approximation to RR loss when the ground-truth items are highly scored (Figure 3D). Additionally, due to RR's emphasis on top ranking positions, inspired by (Jones 1973), a nonlinear transformation function is applied to to accommodate this characteristic.

Building upon the aforementioned, we propose a new objective function, named **CRR**, designed specifically to achieve better approximation to RR as follows:

$$\mathcal{L}_{\text{CRR}} = T\left(\sum_{j=1}^{n} y_j \left(1 + \sum_{k=1}^{n} \mathbb{H}(D_{jk}, \rho, \tau)\right)\right). \tag{14}$$

The CRR loss, shown in Figure 4, is a nonlinear transformation of the Rank loss. T is a transformation function, instantiated as a log function. This new approach is designed to overcome the limitations of existing objective functions, particularly in the ability to approximate RR loss effectively.

Theory Analysis and Validation of Parameter ρ The parameter ρ is a key factor in controlling the approximation of CRR to RR. In this section, we will theoretically analyze the role of ρ mentioned previously. Specifically, as ρ increases, the gap between CRR and RR narrows, whereas τ does not exhibit this property, as shown in Figure 3A, D.

First, we need to demonstrate that $\mathcal{L}_{\text{CRR}} > \mathcal{L}_{\text{RR}}$ holds when $\rho = 0$. Given that $\mathbf{y} = [y_1, y_2, ..., y_{n-1}, y_n]$ is a one-hot vector $\mathbf{y} \in \{0,1\}^n$, and $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_{n-1}, \hat{y}_n]$, where $\hat{y}_i$ is the predicted score. Consider $y_t = 1$ with all other values set to 0, and the corresponding score can be denoted as $\hat{y}_t$. The $\mathcal{L}_{\text{RR}}$ and $\mathcal{L}_{\text{CRR}}$ can be simplified as follows:

$$\mathcal{L}_{\text{RR}} = 1 - \frac{1}{(1 + \sum_{k=1}^{n} \frac{1}{1+e^{\frac{-\hat{x}_k}{\tau}}})}, \tag{15}$$

$$\mathcal{L}_{\text{CRR}} = \ln(1 + \sum_{k=1}^{n} \frac{1}{1 + e^{\frac{\rho - \hat{x}_k}{\tau}}}), \tag{16}$$

where $\hat{x}_k = \hat{y}_k - \hat{y}_t$. For clarity, we define $s(\text{x}) = \sum_{k=1}^{n} \frac{1}{1+e^{\frac{-\hat{x}_k}{\tau}}}$. It is evident that $s(\text{x}) > 0$. To demonstrate that $\mathcal{L}_{\text{CRR}} > \mathcal{L}_{\text{RR}}$ with $\rho = 0$, we begin by defining:

$$F(s(\text{x})) = \mathcal{L}_{\text{CRR}} - \mathcal{L}_{\text{RR}} \tag{17}$$

$$= \ln(1 + s(\text{x})) + \frac{1}{1 + s(\text{x})} - 1, \tag{18}$$

where $F(0) = 0$. Next we calculate the derivatives of $F(s(\text{x}))$ with respect to $s(\text{x})$, specifically:

$$\frac{d}{ds(\text{x})} F(s(\text{x})) = \frac{s(\text{x})}{(1 + s(\text{x}))^2} > 0. \tag{19}$$

Therefore, $F(s(\text{x}))$ is monotonically increasing. Due to $s(\text{x})$ is always greater than 0 and $F(0) = 0$, it follows that $F(s(\text{x}))$ is also greater than zero. Therefore, it can be concluded that $\mathcal{L}_{\text{CRR}} > \mathcal{L}_{\text{RR}}$ when $\rho = 0$.

Next we will prove that as ρ increases, the gap between CRR and RR will narrow, i.e., $\mathcal{L}_{\text{CRR}} - \mathcal{L}_{\text{RR}}$ will decrease. To achieve this objective, we treat ρ as an variable and define:

$$E(\rho) = \mathcal{L}_{\text{CRR}} - \mathcal{L}_{\text{RR}}. \tag{20}$$

Since $\mathcal{L}_{\mathrm{RR}}$ is independent of ρ, we have $\frac{d}{d\rho}E(\rho) = \frac{d}{d\rho}\mathcal{L}_{\mathrm{CRR}}$. According to chain rule (Rudin et al. 1964), we can obtain:

$$\frac{d}{d\rho}u_k(\rho) = -\frac{e^{\frac{\rho-\hat{x}_k}{\tau}}}{\tau(1+e^{\frac{\rho-\hat{x}_k}{\tau}})^2} < 0, \tag{21}$$

$$\frac{d}{d\rho}E(\rho) = \frac{1}{1+\sum_{k=1}^{n} u_k(\rho)} \cdot \sum_{k=1}^{n} \frac{d}{d\rho}u_k(\rho) < 0, \tag{22}$$

where $u_k(\rho) = \frac{1}{1+e^{\frac{\rho-\hat{x}_k}{\tau}}} > 0$. Thus, it is proven that $E(\rho)$ is a monotonically decreasing function and its monotonicity is independent of $\hat{x}_k$ and τ. Consequently, as ρ increases, the gap between $\mathcal{L}_{\mathrm{CRR}}$ and $\mathcal{L}_{\mathrm{RR}}$ decreases.

4.2 Comparison to Margin-based and Log-Likelihood losses

Margin-based, Log-Likelihood (LL) and Binary Cross Entropy (BCE) losses are widely recognized as foundational objective functions in the field of KGC. Given that the Binary Cross-Entropy loss can be considered a simplification of the LL loss for the current task, in this study, we focus our comparison on LL and margin-based losses, which is shown in Figure 4. Following the definitions outlined in Section 3, the formulations for margin-based and LL losses are presented as follows:

$$\mathcal{L}_{\mathrm{margin}} = \sum_{j=1}^{n} y_j \left(\sum_{k=1}^{n} max(D_{jk}; -\gamma) + \gamma \right), \tag{23}$$

$$\mathcal{L}_{\mathrm{LL}} = \sum_{j=1}^{n} y_j \log(1+e^{-\hat{y}_j}) + (1-y_j)\log(1+e^{\hat{y}_j}), \tag{24}$$

where $\gamma > 0$ is a margin hyperparameter.

Under the metric of RR, optimizing for LL loss or margin-based loss proves to be sub-optimal. This is attributed to the fact that LL loss primarily aims to maximize the predicted probabilities of positive samples, whereas the goal of margin-based loss is to ensure that the scores of positive samples exceed those of negative samples by at least a margin γ. Both of them, however, overlook the rank of the ground-truth item, which is the focal point of the RR metric. To conduct different losses comparision, we plot the losses curves based on the ground-truth item score, as shown in Figure 4. From the figure, it is evident that an increase in the predicted score results in a more pronounced decrease in the loss value for mis-ordered items within the curve of RR. However, both LL loss and margin-based loss fail to account for this distinctive characteristic of RR, thereby manifesting an inconsistency with RR. Contrastingly, the behavior of the CRR loss curve is closely aligned with that of RR. By calculating the pearson correlation coefficient between RR and each of the CRR loss, LL loss, and margin-based loss, it reveal that the CRR loss more accurately approximates RR, when compared to both LL and margin-based losses.

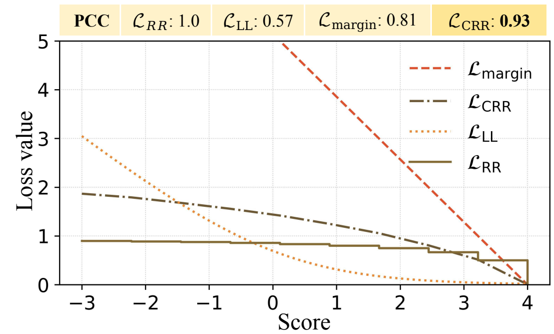

Figure 4: Comparison of different losses, where score indicates the predicted score of ground-truth item. $\mathcal{L}_{\mathrm{CRR}}$: the CRR loss ($\rho$=0.3, τ=0.1); $\mathcal{L}_{\mathrm{margin}}$: the margin-based loss (γ=4.0); $\mathcal{L}_{\mathrm{LL}}$: the Log-Likelihood loss; $\mathcal{L}_{\mathrm{RR}}$: the RR loss; PCC: Pearson correlation coefficient. The CRR achieves better PCC and demonstrates greater consistency with RR, compared to both margin-based and Log-Likelihood losses.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#Train	#Valid	#Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134

Table 1: Statistics of FB15k-237 and WN18RR. $\lvert\mathcal{E}\rvert$: the number of entities; $\lvert\mathcal{R}\rvert$: the number of relations.

5 Results

5.1 Experimental Settings

Datasets and Evaluation Metrics Experiments are conducted on FB15k-237 and WN18RR (Toutanova et al. 2015; Dettmers et al. 2018) datasets, which are widely used for evaluating KGC. More details about datasets can be found in Table 1. For evaluation, we report MRR and Hits@K(H@K, K$\in \{1, 10\}$) to evaluate the performance of methods[3] in the filtered setting (Bordes et al. 2013; Dettmers et al. 2018).

Baselines To investigate the performance of CRR in comparison with Log-Likelihood and margin-based losses, we integrate these three loss functions into several well-established methods implemented by OpenKE (Han et al. 2018). These methods include the translation-based model TransE (Bordes et al. 2013), semantic matching models DistMult (Yang et al. 2014) and ComplEx (Trouillon et al. 2016), the analogical inference model ANALOGY (Liu, Wu, and Yang 2017), and the tensor decomposition model SimplE (Kazemi and Poole 2018), which are among the most popular methods used in KGC related studies (Zheng et al. 2021; Gao, Ding, and Xu 2022; Li et al. 2023; Luo et al. 2024). Additionally, we include two state-of-the-art methods with available code, PairRE (Chao et al. 2021) and CompoundE (Ge et al. 2023), for an extensive comparison.

Implementation Details All experiments are executed on a Linux server equipped with Nvidia RTX 3090 GPUs with 24GB RAM. We use Adagrad as the optimizer, with learning

[3] Optimizing MRR can indirectly improve H@K.

Dataset	Method	$\mathcal{L}_{\text{margin}}$			$\mathcal{L}_{\text{LL}}$			$\mathcal{L}_{\text{Rank}}$			$\mathcal{L}_{\text{RR}}$			$\mathcal{L}_{\text{CRR}}$ (ours)		
		MRR	H@10	H@1	MRR	H@10	H@1	MRR	H@10	H@1	MRR	H@10	H@1	MRR	H@10	H@1
FB15k-237	TransE	0.287	0.477	0.191	0.213	0.365	0.136	0.224	0.451	0.105	0.172	0.345	0.078	**0.313**	**0.507**	**0.216**
	DistMult	0.241	0.405	0.160	0.247	0.428	0.158	0.301	0.469	0.217	0.234	0.400	0.161	**0.322**	**0.498**	**0.234**
	ComplEx	0.253	0.425	0.168	0.248	0.417	0.162	0.300	0.473	0.214	0.250	0.408	0.155	**0.334**	**0.518**	**0.243**
	ANALOGY	0.216	0.435	0.176	0.306	0.481	0.219	0.300	0.486	0.220	0.280	0.450	0.190	**0.332**	**0.512**	**0.242**
	SimplE	0.246	0.411	0.165	0.266	0.419	0.189	0.301	0.465	0.210	0.246	0.387	0.176	**0.321**	**0.498**	**0.233**
	PairRE	0.369	0.552	0.276	0.275	0.431	0.198	0.383	0.571	0.289	0.373	0.560	0.281	**0.416**	**0.598**	**0.319**
	CompoundE	0.376	0.540	0.293	0.320	0.487	0.223	0.358	0.522	0.274	0.346	0.510	0.264	**0.428**	**0.602**	**0.337**
WN18RR	TransE	0.226	0.501	0.012	0.161	0.360	0.065	0.168	0.386	0.001	0.167	0.380	0.001	**0.241**	**0.533**	**0.066**
	DistMult	0.396	0.487	0.343	0.425	0.484	0.392	0.414	0.458	0.390	0.300	0.363	0.310	**0.432**	**0.506**	**0.400**
	ComplEx	0.418	0.493	0.377	0.453	0.514	0.421	0.432	0.474	0.412	0.388	0.405	0.377	**0.459**	**0.519**	**0.428**
	ANALOGY	0.407	0.472	0.373	0.443	0.517	0.408	0.433	0.482	0.395	0.398	0.429	0.383	**0.447**	**0.520**	**0.413**
	SimplE	0.410	0.487	0.367	0.422	0.486	0.387	0.417	0.466	0.392	0.355	0.361	0.352	**0.431**	**0.504**	**0.400**
	PairRE	0.422	0.483	0.391	0.428	0.486	0.393	0.408	0.448	0.387	0.395	0.429	0.377	**0.464**	**0.551**	**0.421**
	CompoundE	0.474	0.551	0.431	0.458	0.512	0.410	0.412	0.467	0.383	0.364	0.435	0.381	**0.484**	**0.552**	**0.446**

Table 2: Performance comparison of methods utilizing different objective functions in terms of H@{1,10} and MRR on FB15k-237 and WN18RR datasets, where the percentages indicate the relative improvement in MRR achieved by our CRR loss over margin-based and Log-Likelihood losses. The most superior outcomes are denoted in **bold**.

rate explored within {1e-4, 5e-4, 1e-3, 5e-3, 0.01, 0.05, 0.1}. The dimensions of the embeddings are varied among {100, 200, 300, 400, 500}, ensuring a comprehensive search space for optimal model performance. We standardize the batch size across all models to 100, the parameter-pressure ρ is investigated across {0.0, 0.1, 0.2, 0.3, 0.4, 0.5} and the temperature τ is explored more extensively within {1e-3, 0.01, 0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0}. For reliability, we conduct five independent experiments with the optimized hyperparameters and report the mean results. The relevant code can be found at https://github.com/XKKuan/KGCRR.

5.2 Performance Comparison

The results of baseline methods with different objective functions are presented in Table 2. The best performance in each method is highlighted in bold. It shows that CRR surpasses both Log-Likelihood and margin-based losses in most cases across the two datasets, which indicates its potential of general capabilities for performance enhancement in KGC. Specifically, using CRR as the loss function leads to an average improvement of 26.49% in MRR compared to margin-based loss and 32.31% compared to LL loss on the FB15k-237 dataset. On the WN18RR dataset, CRR achieves average improvements of 7.51% and 9.93% over margin-based loss and LL loss, respectively. Moreover, the experimental results clearly indicate that the choice of objective function should be tailored to the specific dataset and model. For instance, on the WN18RR dataset, ComplEx with Log-Likelihood loss outperforms the margin-based loss. However, this trend does not hold on the FB15k-237 dataset. In contrast, the CRR loss, with its parameter ρ, offers superior adaptability, allowing for dynamic adjustments to achieve optimal performance across datasets and models. Furthermore, the results reveal that on the FB15k-237 dataset, the CRR loss leads to a greater improvement in model performance compared to the WN18RR dataset. This may be owing to the fact that the FB15k-237 dataset has fewer candidate entities for each triple, which helps CRR rank the ground-truth items more effectively. In addition, experiments on baselines with Rank loss and RR loss are also presented in Table 2. A notable result is that methods using CRR loss outperform their counterparts using RR loss and Rank loss in performance. This superiority can be primarily attributed to the CRR's ability to mitigate the gradient vanishing issue commonly encountered with RR loss, while also offering a closer approximation to RR loss compared to Rank loss. Moreover, unlike Rank loss, CRR loss maintains top-sensitivity, making it particularly effective for enhancing MRR outcomes. During our experiments, we also observed that CRR exhibits good stability and mitigates the gradient vanishing issue associated with directly optimizing RR, which will be detailed in Section 5.5. All results clearly demonstrate the superiority of CRR.

5.3 Impact of Different ρ Values

The parameter ρ is crucial for controlling how closely CRR approximates RR. To validate the impact of different ρ values on the performance of CRR, we conducted parameter sensitivity experiments on FB15k-237 and WN18RR using TransE, DistMult, and ComplEx, as shown in Figure 5.

The results clearly show that the parameter ρ significantly impacts the performance of CRR. As ρ increases from 0.0, the performance measured by MRR undergoes a process from improvement to decline. This corresponds to the process where the curve of the CRR function approaches that of the RR function. From the experimental results, it is clear that through the approximation of the RR function, CRR can effectively improve the methods' performance. However, further increasing the value of ρ leads to a decline in performance, which might be caused by the CRR loss generating a gradient vanishing issue similar to that observed with RR loss. This implies that the CRR loss should strike a

	Method	TransE		DistMult		ComplEx		ANALOGY		SimplE		PairRE		CompoundE	
		MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
FB15k-237	CRR	**0.313**	**0.216**	**0.322**	**0.234**	**0.334**	**0.243**	**0.332**	**0.242**	**0.321**	**0.233**	**0.416**	**0.319**	**0.428**	**0.337**
	w/o T	0.310	0.212	0.311	0.227	0.324	0.236	0.326	0.235	0.317	0.231	0.407	0.312	0.414	0.321
WN18RR	CRR	**0.241**	**0.066**	**0.432**	**0.400**	**0.459**	**0.428**	**0.447**	**0.413**	**0.431**	**0.400**	**0.464**	**0.421**	**0.484**	**0.446**
	w/o T	0.207	0.009	0.423	0.390	0.451	0.421	0.443	0.411	0.430	0.397	0.462	0.420	0.468	0.420

Table 3: Effectiveness of transformation function. w/o T: the CRR without transformation function.

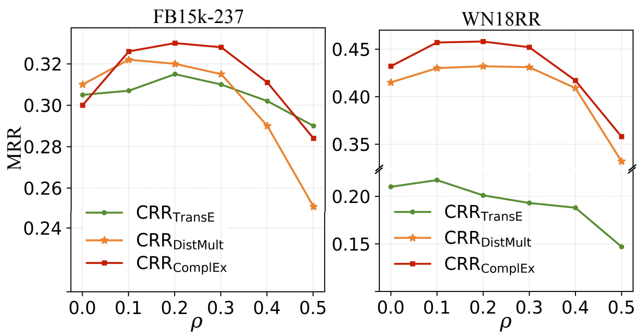

Figure 5: Impact of ρ values on methods utilizing CRR loss.

balance between preserving enough gradients and approximating the RR loss to get the best result.

5.4 Effectiveness of Transformation Function

The transformation function is important for preserving top-ranking sensitivity for CRR. To validate the effectiveness of the transformation function, instantiated as a log function inspired by the work on index term weighting (Jones 1973), we conduct ablation experiments for CRR with and without the use of a transformation function, as shown in Table 3. From the results, it is evident that the transformation function is beneficial for enhancing the performance of the model using CRR loss. The reasons behind this result are twofold: On one hand, the nonlinear transformation function allows the CRR loss to better align with the characteristics of RR, which is more sensitive to changes in top-ranking positions compared to bottom-ranking ones. On the other hand, the log function may further enable the CRR loss to closely approximate the RR function. All of these findings may inspire us to further explore different transformation functions in future.

5.5 Stability Verification

The stability of the objective function plays a critical role in the model learning process. To evaluate the stability of the CRR loss, we monitor the loss variations across various models, including TransE, DistMult, and ComplEx, each trained with the CRR loss. Additionally, we compare the performance of various loss functions, including margin-based, Log-Likelihood, CRR, and RR losses, using TransE as the baseline. All experiments are conducted on the FB15k-237 and WN18RR datasets, as shown in Figure 6.

The experimental results clearly demonstrate that when CRR is used as the objective function, different baseline

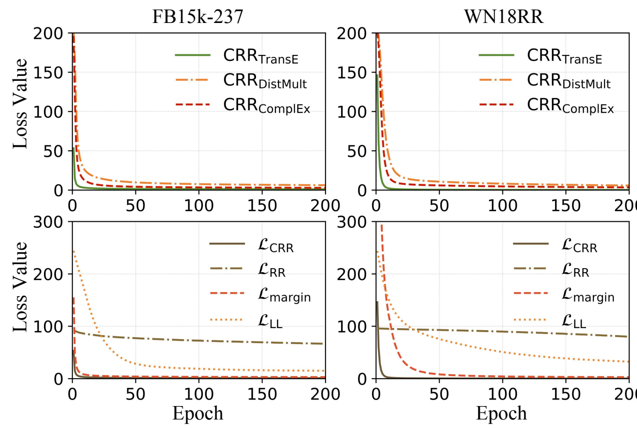

Figure 6: Stability verification of CRR loss across different models and comparison of various objective functions. The results indicate that CRR maintains consistent stability throughout the learning process and exhibits a more rapid convergence rate compared to other objective functions.

models exhibit consistent stability throughout the learning process. And during the comparative experiments with different loss functions, CRR demonstrate a more rapid convergence rate compared to the other losses. Furthermore, we observe that the RR loss converges more slowly than the others, likely due to the gradient vanishing problem, particularly when some ground-truth items are ranked lower in the early stages of training. All of these findings collectively highlight the superiority of CRR as a loss function.

6 Conclusion

In this study, we introduce KGCRR, a KGC framework with the goal of optimizing Reciprocal Rank effectively by designing a new upper bound, CRR. Extensive experiments and theoretical analysis indicate that, by incorporating the pressure parameter ρ and transformation function, CRR achieves a closer approximation to Reciprocal Rank, displaying greater consistency compared to existing objective functions. Its plug-and-play nature allows for seamless integration with various KGE methods. Through extensive experiments on FB15k-237 and WN18RR, the superiority of CRR is well-demonstrated. Some future directions include exploring more flexible objective functions for KGC.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. U23B2062, 82174533, 82374302, 82204941), the National Key Research and Development Program (Nos. 2023YFC3502604), the Natural Science Foundation of Beijing (No. L232033), the Noncommunicable Chronic Diseases-National Science and Technology Major Project (2023ZD0505700), the State Key Laboratory on Technologies for Chinese Medicine Pharmaceutical Process Control and Intelligent Manufacture (SKL2024Z0102).

References

Balažević, I.; Allen, C.; and Hospedales, T. M. 2019. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

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

Brown, A.; Xie, W.; Kalogeiton, V.; and Zisserman, A. 2020. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European Conference on Computer Vision*, 677–694. Springer.

Bubeck, S.; et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4): 231–357.

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1861–1870.

Camponogara, E.; Nazari, L. F.; et al. 2015. Models and algorithms for optimal piecewise-linear function approximation. *Mathematical Problems in Engineering*, 2015.

Chao, L.; He, J.; Wang, T.; and Chu, W. 2021. PairRE: Knowledge Graph Embeddings via Paired Relation Vectors. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Chen, Z.; Zhang, Y.; Fang, Y.; Geng, Y.; Guo, L.; Chen, X.; Li, Q.; Zhang, W.; Chen, J.; Zhu, Y.; et al. 2024. Knowledge graphs meet multi-modal learning: A comprehensive survey. *arXiv preprint arXiv:2402.05391*.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Gao, Z.; Ding, P.; and Xu, R. 2022. KG-Predict: A knowledge graph computational framework for drug repurposing. *Journal of biomedical informatics*, 132: 104133.

Ge, X.; Wang, Y. C.; Wang, B.; and Kuo, C.-C. J. 2023. Compounding geometric operations for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6947–6965.

Ge, X.; Wang, Y. C.; Wang, B.; Kuo, C.-C. J.; et al. 2024. Knowledge Graph Embedding: An Overview. *APSIPA Transactions on Signal and Information Processing*, 13(1).

Han, X.; Cao, S.; Xin, L.; Lin, Y.; Liu, Z.; Sun, M.; and Li, J. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In *Proceedings of EMNLP*.

Henderson, P.; and Ferrari, V. 2017. End-to-end training of object class detectors for mean average precision. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, 198–213. Springer.

Herschtal, A.; and Raskutti, B. 2004. Optimising area under the ROC curve using gradient descent. In *Proceedings of the twenty-first international conference on Machine learning*, 49.

Jones, K. S. 1973. Index term weighting. *Information storage and retrieval*, 9(11): 619–633.

Kazemi, S. M.; and Poole, D. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31.

Li, H. 2011. A short introduction to learning to rank. *IEICE Transactions on Information and Systems*, 94(10): 1854–1862.

Li, R.; Cao, Y.; Zhu, Q.; Bi, G.; Fang, F.; Liu, Y.; and Li, Q. 2022. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 5781–5791.

Li, S.; Wong, K. W.; Zhu, D.; and Fung, C. C. 2023. Drug-CoV: a drug-origin knowledge graph discovering drug repurposing targeting COVID-19. *Knowledge and Information Systems*, 65(12): 5289–5308.

Liu, H.; Wu, Y.; and Yang, Y. 2017. Analogical inference for multi-relational embeddings. In *International conference on machine learning*, 2168–2178.

Lu, X.; Wu, J.; and Yuan, J. 2023. Optimizing Reciprocal Rank with Bayesian Average for improved Next Item Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2236–2240.

Luo, H.; Yin, W.; Wang, J.; Zhang, G.; Liang, W.; Luo, J.; and Yan, C. 2024. Drug-drug interactions prediction based on deep learning and knowledge graph: A review. *Iscience*.

Meilicke, C.; Chekol, M. W.; Betz, P.; Fink, M.; and Stuckeschmidt, H. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 33(1): 131–161.

Mohapatra, P.; Rolinek, M.; Jawahar, C.; Kolmogorov, V.; and Kumar, M. P. 2018. Efficient optimization for rank-based loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3693–3701.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4004–4012.

Rolínek, M.; Musil, V.; Paulus, A.; Vlastelica, M.; Michaelis, C.; and Martius, G. 2020. Optimizing rank-based metrics with blackbox differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7620–7630.

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

Rudin, W.; et al. 1964. *Principles of mathematical analysis*, volume 3. McGraw-hill New York.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Shen, S.-Q.; Yang, B.-B.; and Gao, W. 2020. AUC optimization with a reject option. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5684–5691.

Shen, T.; Zhang, F.; and Cheng, J. 2022. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 109597.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Tang, S.; Luo, F.; and Wu, J. 2022. Smooth-AUC: Smoothing the Path Towards Rank-based CTR Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2400–2404.

Tang, Z.; Pei, S.; Zhang, Z.; Zhu, Y.; Zhuang, F.; Hoehndorf, R.; and Zhang, X. 2022. Positive-unlabeled learning with adversarial data augmentation for knowledge graph completion. *arXiv preprint arXiv:2205.00904*.

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

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, 2071–2080.

Xie, T.; and Ge, Y. 2023. Enhance Knowledge Graph Embedding by Mixup. *IEEE Transactions on Knowledge and Data Engineering*.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Zheng, S.; Rao, J.; Song, Y.; Zhang, J.; Xiao, X.; Fang, E. F.; Yang, Y.; and Niu, Z. 2021. PharmKG: a dedicated knowledge graph benchmark for bomedical data mining. *Briefings in bioinformatics*, 22(4): bbaa344.