Rethinking Regularization Methods for Knowledge Graph Completion

Linyu Li^1, **Zhi Jin**^{1*}, **Yuanpeng He**^{1}, **Dongming Jin**^{1}, **Haoran Duan**^{2},
Zhengwei Tao^1, **Xuan Zhang**^{1}, **Jiandong Li**^{1}
^{1}School of Computer Science, Peking University, Beijing, China
^{2}Wuhan University, Wuhan, China
*corresponding author
linyuli@stu.pku.edu.cn, zhijin@pku.edu.cn

Abstract

Knowledge graph completion (KGC) has attracted considerable attention in recent years because it is critical to improving the quality of knowledge graphs. Researchers have continuously explored various models. However, most previous efforts have neglected to take advantage of regularization from a deeper perspective and therefore have not been used to their full potential. This paper rethinks the application of regularization methods in KGC. Through extensive empirical studies on various KGC models, we find that carefully designed regularization not only alleviates overfitting and reduces variance but also enables these models to break through the upper bounds of their original performance. Furthermore, we introduce a novel sparse-regularization method that embeds the concept of rank-based selective sparsity into the KGC regularizer. The core idea is to selectively penalize those components with significant features in the embedding vector, thus effectively ignoring many components that contribute little and may only represent noise. Various comparative experiments on multiple datasets and multiple models show that the SPR regularization method is better than other regularization methods and can enable the KGC model to further break through the performance margin.

1 Introduction

Knowledge graphs (KGs) [Ji et al., 2021; Liang et al., 2024a] are graph-structured representations that model the real world. KGs have achieved significant milestones in a variety of domains, such as large language models (LLM) [Pan et al., 2024a], computer vision (CV) [Gong et al., 2024], recommender systems [Gao et al., 2022; Guo et al., 2020], and biomedicine mining [Fang et al., 2023; Ma et al., 2024; Xia et al.]. However, due to the continuous growth of knowledge and the limitations inherent in KG acquisition techniques, existing KGs [Bollacker et al., 2008] [Vrandečić and Kr otzsch, 2014] generally suffer from incompleteness. Therefore, investigating effective Knowledge Graph Completion (KGC) models is crucial for enhancing the quality and usability of KGs.

With the continuous efforts of researchers, many excellent KGC models [Bordes et al., 2013; Luo et al., 2023a; Liang et al., 2024b, 2023; Li et al., 2025; Qi et al., 2023; Cui et al., 2024; Wang et al., 2024; Yao et al., 2019; Zhang et al., 2024] have been born. However, it should be noted that most current studies ignore the overfitting problem of the KGC model in practical applications. That is, the KGC model learns the noise and specific relationship patterns in the training set, resulting in poor performance on new datasets that have not been seen before. This phenomenon seriously limits the generalization ability of the model and the upper limit of the model's performance.

In order to better understand and ultimately solve the overfitting problem in the KGC model, this paper conducted an extensive experimental study on the KGC model. First, it systematically evaluates

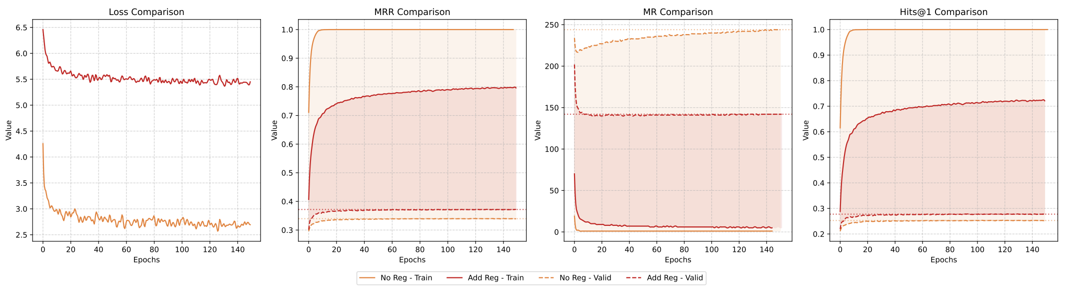

Figure 1: Comparison of the overfitting performance of the GIE [Cao et al., 2022a] model without regularization and with regularization training on the FB15K-237 dataset: Visual comparison of the model Loss, MRR, MR, and Hits@1 indicators as they change with Epoch. The shaded part represents the difference between the training set and the validation set. By adding the regularization method, the variance is significantly reduced.

and analyzes the role of regularization techniques in existing KGC models. Our experimental analysis covers various types of KGC models, especially translation-based embedding models, tensor decomposition-based models, and neural network-based models. Through extensive empirical studies on these different types of models in multiple KG datasets, we found that many of the existing KGC models had overfitting problems, as evidenced by a large discrepancy between their high training performance and low validation performance. Taking Figure 1 as an example, we observe that the GIE [Cao et al., 2022a] model quickly achieves nearly perfect training accuracy without regularization, but suffers significantly lower performance in the validation set, clearly indicating an overfitting problem and suggesting the need to apply regularization techniques. Moreover, without regularization, the MR index gradually increases, which further illustrates worsening stability in model predictions and poor generalization, limiting practical applicability. In contrast, when regularization is added, as shown in Figure 1, the convergence speed slows down, requiring more training epochs. Although the training performance decreases slightly, the validation performance significantly improves, which aligns precisely with our ultimate goal of achieving better generalization rather than merely optimizing training performance, clearly highlighting regularization's effectiveness in mitigating overfitting and enhancing model practicality. Similarly, as presented in Table 1 in the experimental section, adding regularization further enhances the upper bound of model performance on the test set. Limited by the length of the paper, please refer to Appendix A.4 for more empirical studies.

Some existing regularization methods for KGC usually constrain the complexity of the model by minimizing the norm of the embedding vector [Friedland and Lim, 2018]. For example, the Frobenius norm regularizer [Nickel et al., 2011; Yang et al., 2015] is widely used in various models due to its simplicity and ease of use. In subsequent developments, regularizers such as N3 [Lacroix et al., 2018], DURA [Zhang et al., 2020a; Wang et al., 2022], RE[Cao et al., 2022b], and VIR [Xiao and Cao, 2024] have gradually been designed. However, although these regularizations have achieved good results, they are generally only suitable for KGC models based on tensor decomposition. To overcome this limitation, we proposed a KGC regularization method called SPR. With a very simple idea, it can support more types of KGC models, including GNN-based, translation-based, tensor decomposition-based KGC models, and Temporal Knowledge Graph Completion (TKGC) models, and achieve better results. In summary, our contributions are mainly reflected in three aspects:

- We conducted an extensive empirical study of existing KGC methods and found that many KGC models have overfitting problems. In addition, we pointed out that the reasonable application of regularization technology in existing KGC technology can greatly alleviate the overfitting problem in KGC models, thereby enhancing the generalization ability of the model on different data sets and breaking the upper bound of the model's performance.

- We propose a very simple and efficient regularization method, SPR, for the KGC model, which uses the idea of sparsity to selectively discard some unimportant items and selectively retain the operation items to achieve the purpose of regularization.

- We applied SPR to various KGC models and conducted a large number of experiments on multiple datasets. The experimental results and our theoretical analysis demonstrate the effectiveness of our proposed method.

2 Related Work

Translation-based Models Well-known examples include TransE [Bordes et al., 2013], TransH [Wang et al., 2014], TransR [Lin et al., 2015], RotatE [Sun et al., 2019], HAKE [Zhang et al., 2020b], GIE [Cao et al., 2022a], ROTH [Chami et al., 2020], and ConE [Bai et al., 2021]. For additional details, see Appendix A.2.

Tensor-decomposition based Models (TDB) A knowledge graph can be written as a third-order tensor $\mathcal{A} \in \{0,1\}^{|E|\times|R|\times|E|}$ whose entry $\mathcal{A}_{ijk} = 1$ if the triple (e_i, r_k, e_j) exists. CP decomposition [Hitchcock, 1927] approximates this tensor by a sum of d rank-1 tensors: $\mathcal{A} \approx \sum_{f=1}^{d} \mathbf{h}_f \circ \mathbf{r}_f \circ \mathbf{t}_f$, where $\circ$ is the outer product. Scoring functions of many popular KGC models—including DistMult [Yang et al., 2014], ComplEx [Trouillon et al., 2016], SimplE [Kazemi and Poole, 2018], RESCAL [Nickel et al., 2011], and TuckER [Balažević et al., 2019] can be interpreted in this framework.

Regularization Methods Adding a penalty term to the loss keeps models from overfitting [Santos and Papa, 2022]. Standard choices such as L2 or Frobenius norms [Cortes et al., 2012] remain common in KGC, but newer strategies target the peculiarities of knowledge graphs. Examples include DURA [Zhang et al., 2020a; Wang et al., 2022], ER [Cao et al., 2022b], VIR [Xiao and Cao, 2024], and N3 [Lacroix et al., 2018]. Most of these were designed for TDB models; our work shows that overfitting is a widespread issue in KGC models and introduces a simpler, more generalizable regularization method.

3 Methodology

3.1 Preliminaries

Knowledge Graph Let the set of entities be denoted as $\mathcal{V} = \left\{v_1, \ldots, v_{|\mathcal{V}|}\right\}$, and the set of relations as $\mathcal{P} = \left\{\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{P}|}\right\}$. A knowledge graph is defined as a set of triples $\mathcal{T} \subseteq \mathcal{V} \times \mathcal{P} \times \mathcal{V}$, where each triple $(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)$ corresponds to a head entity, a relation, and a tail entity, respectively. The knowledge graph (KG) is then represented as the tuple $\mathcal{G} = (\mathcal{V}, \mathcal{P}, \mathcal{T})$, comprising the set of entities, relations, and observed triples.

Knowledge Graph Completion Given an observed knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{P}, \mathcal{T})$, the task of KGC is to infer the missing but plausible facts, denoted by $\mathcal{T}^\star$. The mainstream KGC method first assigns a vector embedding in real or complex space to each entity and relationship, and constructs a scoring function:

$$\phi_e : \mathcal{V} \to \mathbb{R}^{d_e} \text{ or } \mathbb{C}^{d_e}, \quad \phi_\rho : \mathcal{P} \to \mathbb{R}^{d_\rho} \text{ or } \mathbb{C}^{d_\rho} \tag{1}$$

$$s : \mathcal{V} \times \mathcal{P} \times \mathcal{V} \longrightarrow \mathbb{R}, \quad (v_h, \rho_r, v_t) \mapsto s\left(v_h, \rho_r, v_t\right) \tag{2}$$

which evaluates the plausibility of any candidate triple. For completion queries of the form $\langle v_i, \rho_j, ?\rangle$, the model substitutes the missing entity with each $v \in \mathcal{V}$ in turn and computes the corresponding score. Entities with the highest scores are top candidates for completing the triple.

3.2 The SPR Regularizer Method

Existing KGC regularizers such as DURA [Zhang et al., 2020a], ER [Cao et al., 2022b], and VIR [Cao et al., 2024] have proven effective for CP/ComplEx style tensor-decomposition models. But generality in the KGC model is not enough. SPR is designed precisely for this broader setting: by sparsifying the regularization target, we focus the penalty on the truly influential coordinates. The KGC model with and without regularization is shown in Figure 2.

The basic idea of SPR regularization is to "sparse" the elements that are input to the regularization term so that only important components (i.e., components with large squared magnitudes) are penalized,

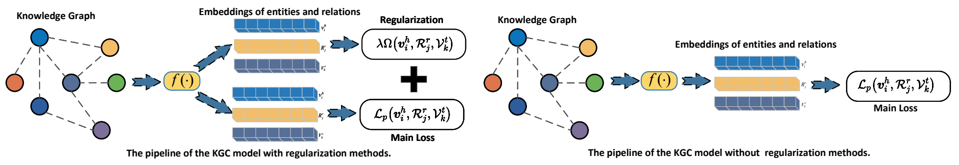
Knowledge Graph
Embeddings of entities and relations
Regularization
$\lambda\Omega(v_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)$
$\mathcal{L}_p(v_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)$
Main Loss
The pipeline of the KGC model with regularization methods.
Knowledge Graph
Embeddings of entities and relations
$\mathcal{L}_p(v_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)$
Main Loss
The pipeline of the KGC model without regularization methods.

Figure 2: The pipeline of the KGC model with/without regularization methods.

while many almost negligible components (which may just be noise) are discarded. The sparsification is achieved by sorting the (non-negative) squared terms and "masking" those whose cumulative sum is lower than δ and "discarding" them (i.e., zeroing them). One could also envision making δ adaptive, for example, as a function of the current gradient norm. However, for clarity, we use a fixed δ in this version. SPR applies sparsity regularization exclusively to the input entities, relations, and their interactive elements, effectively mitigating the risk of model overfitting.

Sorting and Select First, let's define the binary mask $\mathcal{M}(\boldsymbol{x}) \in \{0,1\}^D$. For any non-negative vector (entity, relation, or their interaction representation) whose element $\boldsymbol{x} \in \mathbb{R}^D_{\geq 0}$(for example $\boldsymbol{x} = {\boldsymbol{v}_i^h}^2, {\boldsymbol{v}_i^t}^2$, or ${\boldsymbol{v}_i^h}^2 \odot {\mathcal{R}_j^r}^2$). Let $\{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \ldots, \boldsymbol{x}_{(D)}\}$ be the entries of $\boldsymbol{x}$ arranged in non-decreasing order. Let $\boldsymbol{S}$ be the largest integer such that:

$$\sum_{i=1}^{\boldsymbol{S}} \boldsymbol{x}_{(i)} \leq \delta \tag{3}$$

Then,the sparsified version of $\boldsymbol{x}$ is given by:

$$\boldsymbol{x}_{\text{sparse}} = \boldsymbol{x} \odot (1 - \mathcal{M}(\boldsymbol{x})) \tag{4}$$

which means that the small components (those for which $\mathcal{M}(\boldsymbol{x})_j = 1$) are set to zero.

Lemma 1 Sparsification Error Bound *Let $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^n_{\geq 0}$ be a nonnegative vector and suppose that the operator select-small$(\boldsymbol{x}, \delta)$ produces a binary mask $\mathcal{M} \in \{0,1\}^n$ .Such that the cumulative sum of the "dropped" entries is bounded by δ, i.e., $\sum_{i:\mathcal{M}_i=1} \boldsymbol{x}_i \leq \delta$. Then the difference between the full sum and the sparsified sum is bounded as:*

$$\left| \sum_{j=1}^{D} \boldsymbol{x}_j - \sum_{j=1}^{D} (\boldsymbol{x}_j)_{\text{sparse}} \right| \leq \delta \tag{5}$$

Proof. Please refer to Appendix A.5.

Sparsified Norms of Entities Using the sparsification operator defined above, we now start defining each term in the regularizer. For a given triple $\left(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right)$, there is the following sparse regularization term:

$$\|\mathcal{V}_i^h\|^2_{\text{sparse}} = \left\| {\boldsymbol{v}_i^h}^2 \odot \left(1 - \mathcal{M}\left({\boldsymbol{v}_i^h}^2\right)\right) \right\| = \sum_{d=1}^{D} {\boldsymbol{v}_{i\,d}^h}^2 \left(1 - \mathcal{M}\left({\boldsymbol{v}_{i\,d}^h}^2\right)\right) \tag{6}$$

$$\|\mathcal{V}_k^t\|^2_{\text{sparse}} = \left\| {\boldsymbol{v}_k^t}^2 \odot \left(1 - \mathcal{M}\left({\boldsymbol{v}_k^t}^2\right)\right) \right\| = \sum_{d=1}^{D} {\boldsymbol{v}_{k\,d}^t}^2 \left(1 - \mathcal{M}\left({\boldsymbol{v}_{k\,d}^t}^2\right)\right) \tag{7}$$

Sparsified Interaction Terms For the interaction between the head entity and the tail entity, we have the following sparse regularization term:

$$\|\mathcal{V}_i^h \odot \mathcal{R}_j^r\|^2_{\text{sparse}} = \left\| \left({\mathcal{V}_i^h}^2 \odot {\mathcal{R}_j^r}^2\right) \odot \left(1 - \mathcal{M}\left({\mathcal{V}_i^h}^2 \odot {\mathcal{R}_j^r}^2\right)\right) \right\| = \sum_{d=1}^{D} {\mathcal{V}_{i\,d}^h}^2 {\mathcal{R}_{j\,d}^r}^2 \left(1 - \mathcal{M}\left({\mathcal{V}_{i\,d}^h}^2 \odot {\mathcal{R}_{j\,d}^r}^2\right)\right) \tag{8}$$

$$\|\mathcal{V}_k^t \odot \mathcal{R}_j^r\|^2_{\text{sparse}} = \left\| \left({\mathcal{V}_k^t}^2 \odot {\mathcal{R}_j^r}^2\right) \odot \left(1 - \mathcal{M}\left({\mathcal{V}_k^t}^2 \odot {\mathcal{R}_j^r}^2\right)\right) \right\|_1 = \sum_{d=1}^{D} {\mathcal{V}_{k\,d}^t}^2 {\mathcal{R}_{j\,d}^r}^2 \left(1 - \mathcal{M}\left({\mathcal{V}_{k\,d}^t}^2 \odot {\mathcal{R}_{j\,d}^r}^2\right)\right) \tag{9}$$

After obtaining the above sparse regularization term, we can get the SPR regularization term of the complete triple as:

$$\mathrm{SPR}(\mathcal{V}_i^h,\ \mathcal{R}_j^r,\ \mathcal{V}_k^t) = \left\|\mathcal{V}_i^h\right\|_{\text{sparse}}^2 + \left\|\mathcal{V}_i^t\right\|_{\text{sparse}}^2 + \left\|\mathcal{V}_i^h \odot \mathcal{R}_j^r\right\|_{\text{sparse}}^2 + \left\|\mathcal{V}_k^t \odot \mathcal{R}_j^r\right\|_{\text{sparse}}^2 \tag{10}$$

When training a batch of size B triplets, the overall regularization loss is:

$$\Omega_{\text{SPR}} = \frac{1}{|B|} \sum_{(\boldsymbol{v}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t) \in B} \mathrm{SPR}(\mathcal{V}_i^h,\ \mathcal{R}_j^r,\ \mathcal{V}_k^t) \tag{11}$$

For a given triplet, the general form of the scoring function for distance-based embedding (DB) models and tensor factorization-based (TFB) KGC models can be formalized as:

$$f_{\mathcal{R}_j^r}\left(\mathcal{V}_i^h, \mathcal{V}_k^t\right) = \phi_\rho\left(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right) = \phi_\rho\left(\left\langle \mathcal{V}_i^h \mathcal{R}_j^r, \mathcal{V}_k^t \right\rangle\right). \tag{12}$$

among them, ϕ_ρ represents the linear transformation of entity $\mathcal{V}_i^h$ through relation $\mathcal{R}_j^r$. $\langle\cdot\rangle$ represents the spatial distance function and similarity function of $\mathcal{V}_i^h$ and $\mathcal{V}_k^t$ in the DB model and the TFB model, respectively. Generally speaking, the prediction function of the traditional KGC model is:

$$\begin{gathered}\mathcal{L}_p = \sum_{(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t) \in \{\mathcal{G} \cup \mathcal{G}'\}} \log\left(1 + \exp\left(\Theta_{(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)} \cdot f_{\mathcal{R}_j^r}\left(\mathcal{V}_i^h, \mathcal{V}_k^t\right)\right)\right) \\ \text{in which, } \Theta_{(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)} = \begin{cases} 1 \text{ for } \left(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right) \in \mathcal{G} \\ -1 \text{ for } \left(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right) \in \mathcal{G}' \end{cases}\end{gathered} \tag{13}$$

where $\mathcal{G} \cup \mathcal{G}'$ denotes the set of positive and negative samples for the triple after applying a random replacement operation following the method in [Bordes et al., 2013]. If the model is combined with a regularization method, the basic optimization paradigm is as follows:

$$\min \sum_{(\boldsymbol{v}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t) \in \{\mathcal{G} \cup \mathcal{G}'\}} \mathcal{L}_p\left(\boldsymbol{v}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right) + \lambda\Omega\left(\boldsymbol{v}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t\right) \tag{14}$$

where $\Omega(\cdot)$ represents the regularization method, and λ represents the regularization coefficient. Our method hopes to be as universal as possible in the KGC model and as simple and universal as possible.

Score Function CP [Kazemi and Poole, 2018] is a classic bilinear method that decomposes the knowledge graph tensor into the sum of rank 1 tensors, which is fully expressive and highly parameter efficient. The score function is as follows:

$$f_{\text{CP}}(h, r, t) = \sum_{k=1}^{d} a_{h,k} b_{t,k} c_{r,k} \tag{15}$$

where $a_{h,k}$ is the k^{th} component of the subject side embedding $\mathbf{a}_h \in \mathbb{R}^d$; $b_{t,k}$ is the k^{th} component of the object side embedding $\mathbf{b}_t \in \mathbb{R}^d$; $c_{r,k}$ is the k^{th} component of the relation vector $\mathbf{c}_r \in \mathbb{R}^d$; d is the embedding dimensionality; the sum is a standard inner product.

ComplEx [Trouillon et al., 2016] extends DistMult[Yang et al., 2014] into the complex plane so it can natively model asymmetric relations while keeping $O(d)$ parameters per relation. The real part of a tri-linear Hermitian product becomes the plausibility score. The score function is as follows:

$$f_{\text{ComplEx}}(h, r, t) = \operatorname{Re}\left(\sum_{k=1}^{d} h_k r_k \overline{t_k}\right). \tag{16}$$

where $h_k, r_k, t_k \in \mathbb{C}$ are the k^{th} complex components of entity and relation embeddings; $\overline{t_k}$ is the complex conjugate of t_k;$\operatorname{Re}(\cdot)$ extracts the real part, turning a complex score into a real scalar.

RESCAL [Nickel et al., 2011] learns one dense matrix per relation so every pair of latent features can interact. The score function is as follows:

$$f_{\text{RESCAL}}(h, r, t) = \mathbf{h}^\top M_r \mathbf{t}. \tag{17}$$

where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ are entity vectors shared across relations; $M_r \in \mathbb{R}^{d \times d}$ is the relation-specific matrix;$\mathbf{h}^\top M_r \mathbf{t}$ is a quadratic form producing the triple score.

GIE [Cao et al., 2022a], a typical embedding-based model, embeds every entity in three curved spaces (Euclidean $\mathcal{E}$, hyperbolic $\mathcal{H}$, hyperspherical $\mathcal{S}$) and scores a triple by interacting those geometries via cross-space geodesic distances. The score function is as follows:

$$f_{\text{GIE}}(h, r, t) = -\left(d_c\left(\operatorname{Inter}\left(E_h, H_h, S_h\right), \mathbf{t}\right) + d_c\left(\operatorname{Inter}\left(E_t, H_t, S_t\right), \mathbf{h}\right)\right) + b \tag{18}$$

where E_h, H_h, S_h are the head (tail) coordinates in each manifold; Inter $(\cdot)$ is the learned attention-weighted fusion of the three points; d_c is the manifold-geodesic with curvature c;b is a trainable bias.

4 Experiment

In order to rigorously answer the following five questions, we carefully designed relevant experiments in five aspects to prove the contribution of our article.

- **Q1: Necessity of Regularization in KGC Models.** Does integrating regularization methods into KGC models effectively mitigate overfitting? Is regularization necessary for KGC models?
- **Q2: Comparative Performance.** How does our proposed SPR regularization approach compare to existing regularization methods across diverse experimental conditions?
- **Q3: Generalization Capability.** Is the SPR method versatile enough to enhance performance consistently across various types of KGC models?
- **Q4: Sensitivity Analysis.** How robust is SPR's performance to variations in hyperparameters?
- **Q5: Embedding Quality Visualization.** Does the SPR method visibly improve embedding representations at the entity level, as demonstrated through embedding space visualization?

4.1 Experiment Setting

Table 1: KGC main experiment results of adding regularization methods to different models on WN18RR, FB15K-237, and YAGO3-10 datasets. All experiments were run on the same machine. In the table, boldface marks the best result, an underline tagged with _ marks the second-best result, and $-$ means that this regularization method is incompatible with the baseline model and produces a "NAN" error during training.

MODEL	WN18RR				FB15k-237				YAGO3-10			
	MRR	Hits1	Hits3	Hits10	MRR	Hits1	Hits3	Hits10	MRR	Hits1	Hits3	Hits10
CP	0.438	0.416	0.465	0.485	0.327	0.239	0.359	0.502	0.567	0.495	0.609	0.695
CP-F2	0.449	0.421	0.474	0.506	0.332	0.250	0.369	0.517	0.571	0.499	0.617	0.699
CP-N3	0.469	0.432	0.485	0.541	0.354	0.260	0.389	0.543	0.573	0.499	0.618	0.705
CP-DURA	0.471	0.433	0.488	0.545	0.36	0.266	0.395	0.55	0.577	0.504	0.623	0.709
CP-ER	0.475	0.436	0.489	0.549	0.358	0.263	0.394	0.549	0.576	0.5021	0.620	0.706
CP-SPR	**0.479**	**0.44**	**0.491**	**0.555**	**0.366**	**0.272**	**0.404**	**0.558**	**0.582**	**0.509**	**0.627**	**0.713**
ComplEx	0.457	0.426	0.471	0.518	0.350	0.260	0.384	0.532	0.569	0.495	0.614	0.703
ComplEx-F2	-	-	-	-	-	-	-	-	-	-	-	-
ComplEx-N3	0.488	0.445	0.505	0.57	0.360	0.268	0.399	0.551	0.574	0.498	0.618	0.710
ComplEx-DURA	0.488	0.447	0.502	0.568	0.365	0.270	0.403	0.555	0.581	0.506	0.624	0.716
ComplEx-ER	0.486	0.444	0.502	0.566	0.369	0.275	0.408	0.561	0.584	0.512	0.626	0.712
ComplEx-SPR	**0.491**	**0.451**	**0.509**	**0.572**	**0.371**	**0.277**	**0.409**	**0.566**	**0.584**	**0.514**	**0.628**	**0.719**
GIE	0.464	0.434	0.479	0.529	0.341	0.2536	0.3728	0.5198	0.574	0.503	0.617	0.699
GIE-F2	-	-	-	-	-	-	-	-	-	-	-	-
GIE-N3	0.491	0.452	0.507	0.576	0.364	0.269	0.400	0.555	0.579	0.508	0.623	0.710
GIE-DURA	0.488	0.449	0.503	0.571	0.355	0.262	0.395	0.547	0.586	0.515	0.623	0.714
GIE-ER	0.49	0.45	0.505	0.574	0.351	0.259	0.388	0.541	0.585	0.517	0.626	**0.721**
GIE-SPR	**0.497**	**0.455**	**0.514**	**0.581**	**0.367**	**0.272**	**0.402**	**0.558**	**0.593**	**0.522**	**0.629**	0.720
RESCAL	0.444	0.419	0.453	0.491	0.350	0.262	0.382	0.525	0.551	0.471	0.598	0.695
RESCAL-F2	-	-	-	-	-	-	-	-	-	-	-	-
RESCAL-N3	-	-	-	-	-	-	-	-	-	-	-	-
RESCAL-DURA	0.495	0.454	0.511	0.576	0.365	0.273	0.403	0.546	0.563	0.486	0.608	0.704
RESCAL-ER	0.493	0.452	0.505	0.575	0.363	0.270	0.401	0.549	0.569	0.502	0.611	0.709
RESCAL-SPR	**0.499**	**0.456**	**0.518**	**0.581**	**0.367**	**0.276**	**0.407**	**0.551**	**0.575**	**0.511**	**0.619**	**0.713**

Datasets. This study selected five widely used knowledge graph benchmark datasets: WN18RR [Dettmers et al., 2018] , FB15K-237 [Toutanova and Chen, 2015], YAGO3-10 [Mahdisoltani et al., 2013],Kinship [Kemp et al., 2006] and UMLS [McCray, 2003]. Please see the Appendix for a detailed introduction to the dataset.

Implementation Details. All our experiments were conducted on a server equipped with 1 TB of RAM, an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, and 8 V100 GPUs with 32 GB each. For more details on the model's hyperparameters and configurations, please refer to Appendix A.2.

Evaluation Metrics. Our experiment uses the three most common evaluation indicators of KGC, namely MRR, HITS@1, HITS@3, and HITS@10. They represent the predictive ability of the model. The larger the value, the better the effect. The detailed calculation process is shown in Appendix A.2

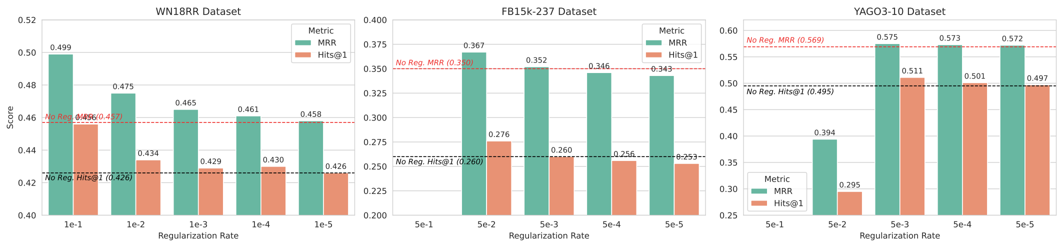

Figure 3: Analysis of the Impact of Regularization Rate on Model Performance for WN18RR, FB15K-237 and YAGO3-10 Datasets.

4.2 Main Performance (RQ 1 & RQ 2)

Table 1 shows the comprehensive experimental results of KGC based on various models (CP, ComplEx, GIE, RESCAL) and applying different regularization methods on different datasets (WN18RR, FB15K-237, YAGO3-10). A detailed analysis of these results reveals the superior performance of our SPR regularization method and the significant impact of regularization on the KGC model.

KGC models without regularization usually face problems such as overfitting. They perform well on the training data, but poorly on the validation and test data. This is reflected in the relatively low scores of the base model. Unregularized models (such as CP, ComplEx, GIE, etc.) generally perform poorly on all datasets, especially on high-precision indicators such as Hits@1 and Hits@3, which indicates that they are prone to overfitting the training data or cannot learn high-quality embedding representations. After adding the regularization method, the model's performance is significantly improved. Taking the CP model on the WN18RR dataset as an example, the MRR metric of CP-SPR is 0.479. Compared with the basic model, the MRR is significantly improved by about 9.4%. Similar trends are observed in other models and datasets. For example, in the ComplEx model in the FB15K-237 dataset, the MRR of the basic ComplEx is 0.35, while ComplEx-SPR increases it to 0.371, achieving a significant improvement. This can now answer **Q1 and Q2** very well. In the appendix, we provide more empirical studies to prove the important role of regularization methods in the KGC model.

In order to fully verify the effectiveness of the SPR method in the future, after testing large-scale datasets such as YAGO3-10 in the main experiment above, we continued to use two small datasets, UMLS and Kinship, to verify the main experiment. The experimental results are shown in Table 2. In all datasets and compared to regularized baseline methods such as F2, N3, DURA, and ER, SPR always maintains the best or near-best performance on most models and indicators. This generally high effect verifies that selectively regularizing only the important components in the embedding space is effective and advantageous. It also shows that SPR can effectively alleviate the overfitting problem in the KGC model, thereby improving the embedding representation ability of the model to improve the model's prediction effect. The simplicity and effectiveness of this SPR sparsification method provide valuable insights for the future development of the KGC method.

Table 2: KGC Results on UMLS and Kinship Datasets

Model	**UMLS**		**Kinship**	
	MRR	Hits@1	MRR	Hits@1
CP-NA	0.868	0.780	0.872	0.784
CP-DURA	0.881	0.788	0.885	0.792
CP-N3	0.879	0.782	0.885	0.819
CP-ER	0.883	0.791	0.888	0.820
CP-SPR	**0.885**	**0.792**	**0.889**	**0.821**
ComplEx-NA	0.897	0.822	0.875	0.783
ComplEx-DURA	0.904	0.841	**0.888**	0.796
ComplEx-N3	0.907	0.843	0.883	0.786
ComplEx-ER	**0.910**	0.844	0.887	0.795
ComplEx-SPR	0.909	**0.847**	0.887	**0.820**

4.3 Sensitivity experiments (RQ4)

In this chapter, we conduct an in-depth investigation of the effects of applying SPR regularization across multiple datasets, examining how different regularization rates influence model performance measured by MRR and Hits@1 on each dataset. The experimental results are shown in Figure 3 and Figure 7 in Appendix A.3. Our findings reveal several important insights:

Regularization method is a "double-edged sword": performance varies non-monotonically with the regularization rate. Both excessively high and excessively low regularization rates can degrade performance, even yielding worse results than using no regularization at all. On certain datasets, such as FB15K-237 and YAGO3-10, an overly large regularization rate imposes such a severe penalty on the model that it fails to learn any meaningful semantic information about entities. This leads to embedding collapse, with metric values dropping below 0.1.

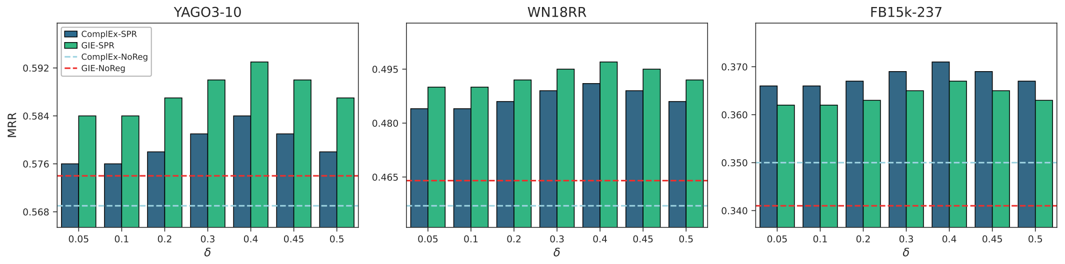

Figure 4: Impact of SPR threshold δ on GIE and ComplEx models across WN18RR, FB15k-237, and YAGO3-10 datasets.

From the comprehensive experimental results, on all these KG datasets, as long as the regularization rate is properly selected, the use of regularization technology can generally improve the link prediction performance of the model (MRR and Hits@1 indicators). This strongly indicates that regularization helps alleviate the overfitting of the model during training and improves the generalization ability of the model.

The performance improvement brought by regularization varies across different datasets. The improvement is more obvious on the WN18RR and FB15k-237 datasets, while the improvement is relatively small on the YAGO3-10 dataset. This may be related to the characteristics of the dataset itself (such as scale, sparsity, relation complexity, etc.) and the fit between the model and the dataset. We will further explore the potential correlation between the characteristics of the dataset and the regularization method in the appendix.

To quantify the impact of the threshold δ in the SPR regularization method, we conducted parameter sensitivity experiments on three datasets using two backbone baseline models, ComplEx and GIE. We varied δ from 0.05 to 0.50 in increments of 0.05. The experimental results are shown in Figure 4. Overall, the results exhibit a pattern resembling a normal distribution, with δ=0.4 yielding the peak (i.e., the best performance). These findings suggest that both too small and too large a sparsification threshold δ can adversely affect model performance, but both still outperform having no regularization at all. Therefore, it is necessary to select an appropriate δ value based on the specific model being used.

4.4 Scalability Experiment (RQ3)

To answer **Q3**, we conducted additional experiments using various types of KGC models: the embedding-based model GIE, the tensor decomposition-based model ComplEx, the graph neural network–based model CompGCN, and a TKGC model. The experimental results for the CompGCN variants are presented in Figure 5. It is evident that regularization is absolutely critical; without any form of regularization, both CompGCN-TransE (CompGCN-T) and CompGCN-ConvE (CompGCN-C) yield relatively low MRR scores (e.g., 0.216 and 0.302 on FB15k-237; 0.146 and 0.436 on WN18RR). Introducing dropout alone leads to substantial improvements across the board. For instance, on FB15k-237, the MRR of CompGCN-T increases by 59.3%, and that of CompGCN-C increases by 18.9%; on WN18RR, CompGCN-T improves by 34.9%, and CompGCN-C by 8.7%. Our proposed SPR regularization method goes a step further by penalizing specific entity–relation interactions, thereby encouraging the model to learn genuinely robust and non-redundant features. This targeted regularization enables GNN-based models to surpass their conventional performance ceiling and achieve even better results.

In addition, we further explored the application of regularization methods in TKGC [Wang et al., 2023; Cai et al., 2022]. We tested the TKGC model HGE [Pan et al., 2024b] on two datasets ICEWS14 [García-Durán et al., 2018] and ICEWS05-15 [García-Durán et al., 2018]. The main experimental results are shown in Table 3. The experimental results clearly show that regularization is also very important in the task of TKGC. Compared with the baseline model (HGE+NOREG) that does not use regularization at all, the introduction of the SPR regularization method brings significant performance improvements on both datasets. Specifically, on the ICEWS14 and ICEWS05-15 datasets, the SPR method achieved improvements of approximately 4.2% and 5.4% in the MRR metric compared to the non-regularized baseline. For the Hits@10 metric, the improvements were around 6.9% and 5.9%, respectively. Other evaluation metrics also showed relative gains ranging from 2.7% to 4.6%. The same SPR regularization method also performs better than the existing N3 regularization method on both datasets. These quantitative improvement ratios clearly show that SPR regularization significantly improves the upper bound of the model compared to the non-regularization method. For additional visualizations of the TKGC model training trends, please refer to Appendix 4.

Table 3: TKGC Results on ICEWS14 and ICEWS05-15 Datasets.

Models	ICEWS14				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
HGE+NoReg	0.596	0.514	0.648	0.739	0.653	0.579	0.704	0.785
HGE+N3	0.617	0.526	0.673	0.783	0.684	0.599	0.733	0.829
HGE+SPR	**0.621**	**0.528**	**0.678**	**0.790**	**0.688**	**0.602**	**0.736**	**0.831**

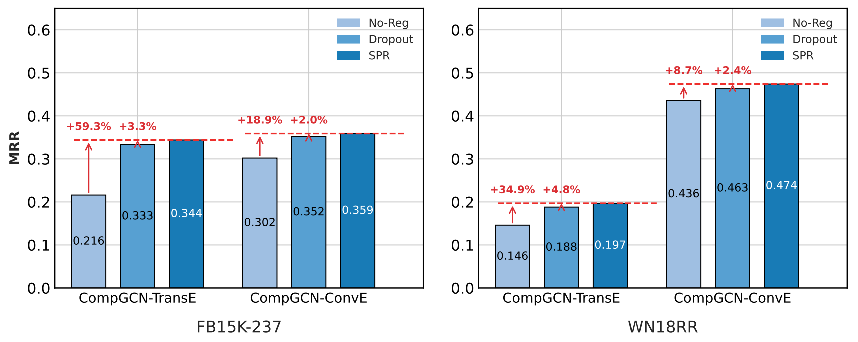

Figure 5: Comparing the Impact of Regularization on GNN-based KGC models CompGCN: MRR Performance of CompGCN-TransE and CompGCN-ConvE on FB15K-237 and WN18RR Datasets with No-Reg, Dropout, and SPR Settings.

4.5 Visual Analytics (RQ5)

To evaluate the quality of entity embeddings learned by our proposed SPR method, we used T-SNE to visualize the FB15K-237 dataset. We randomly selected 10 different entities and projected their 50-dimensional embeddings into a low-dimensional space. As shown in Visualization Figure 6, the embeddings obtained using SPR regularization show significantly better clustering than those generated using DURA regularization, ER regularization, or without any regularization. In the SPR graph, data points corresponding to the same entity form tight and well-separated clusters, while data points from different entities have less overlap. This clear separation shows that SPR can effectively capture the unique semantic features of each entity and obtain more discriminative and higher-quality entity embeddings.

5 Conclusion

In this paper, we rethink the application of regularization methods in mainstream KGC models.

Our first major contribution is conducting extensive empirical studies to investigate the practical performance of regularization methods in KGC models. Through these empirical studies, we observed that the issue of overfitting is frequently overlooked in KGC models; further extensive experiments demonstrate that applying

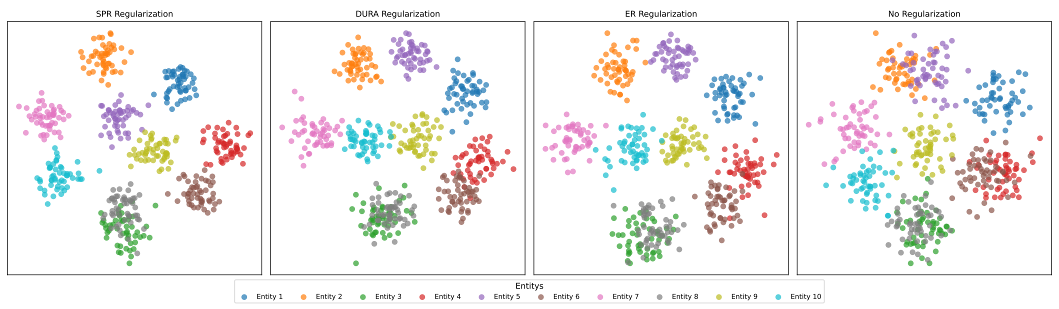

Figure 6: Visualization experiment of the ComplEx model using different regularization methods and no regularization method on the WN18RR dataset.

appropriate regularization techniques to KGC models not only alleviates overfitting and reduces variance but also enables the models to surpass their original performance upper bounds.

Our second major contribution is to revisit the overfitting problem in KGC models and propose SPR, a novel yet simple regularization method tailored specifically for KGC models. SPR mitigates potential overfitting by selectively penalizing the most important components in the entity and relation embeddings, while discarding less influential and potentially noisy elements. Comparative experiments across multiple datasets and various KGC models (including translation-based, tensor decomposition-based, and GNN-based models etc.) consistently show that SPR outperforms existing regularization methods, significantly improving model effectiveness and mitigating overfitting. To support future research and development of KGC regularization methods, we will provide the source code and data.

References

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, Fuchun Sun, and Kunlun He. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024a.

Biao Gong, Shuai Tan, Yutong Feng, Xiaoying Xie, Yuyuan Li, Chaochao Chen, Kecheng Zheng, Yujun Shen, and Deli Zhao. Uknow: A unified knowledge protocol with multimodal knowledge graph datasets for reasoning and vision-language pre-training. *Advances in Neural Information Processing Systems*, 37: 9612–9633, 2024.

Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Graph neural networks for recommender system. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1623–1625, 2022.

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34 (8):3549–3568, 2020.

Yin Fang, Qiang Zhang, Ningyu Zhang, Zhuo Chen, Xiang Zhuang, Xin Shao, Xiaohui Fan, and Huajun Chen. Knowledge graph-enhanced molecular contrastive learning with functional prompt. *Nature Machine Intelligence*, 5(5):542–553, 2023.

Tengfei Ma, Yujie Chen, Wen Tao, Dashun Zheng, Xuan Lin, Cheong-Iao Pang, Yiping Liu, Yijun Wang, Longyue Wang, Bosheng Song, et al. Learning to denoise biomedical knowledge graph for robust molecular interaction prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z Li. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*.

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

Denny Vrandečić and Markus Kr otzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*, 2023a.

Ke Liang, Yue Liu, Hao Li, Lingyuan Meng, Suyuan Liu, Siwei Wang, Sihang Zhou, and Xinwang Liu. Clustering then propagation: Select better anchors for knowledge graph embedding. *Advances in Neural Information Processing Systems*, 37:9449–9473, 2024b.

Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):226–238, 2023.

Linyu Li, Zhi Jin, Xuan Zhang, Haoran Duan, Jishu Wang, Zhengwei Tao, Haiyan Zhao, and Xiaofeng Zhu. Multi-view riemannian manifolds fusion enhancement for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Kunxun Qi, Jianfeng Du, and Hai Wan. Learning from both structural and textual knowledge for inductive knowledge graph completion. *Advances in Neural Information Processing Systems*, 36:26546–26562, 2023.

Yuanning Cui, Zequn Sun, and Wei Hu. A prompt-based knowledge graph foundation model for universal in-context reasoning. *Advances in Neural Information Processing Systems*, 37:7095–7124, 2024.

Jiapu Wang, Sun Kai, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410, 2024.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

Yichi Zhang, Zhuo Chen, Lingbing Guo, Yajing Xu, Wen Zhang, and Huajun Chen. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 233–242, 2024.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Geometry interaction knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 5521–5529, 2022a.

Shmuel Friedland and Lek-Heng Lim. Nuclear norm of higher-order tensors. *Mathematics of Computation*, 87 (311):1255–1281, 2018.

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 3104482–3104584, 2011.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR, 2018.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. Duality-induced regularizer for tensor factorization based knowledge graph completion. *Advances in Neural Information Processing Systems*, 33:21604–21615, 2020a.

Jie Wang, Zhanqiu Zhang, Zhihao Shi, Jianyu Cai, Shuiwang Ji, and Feng Wu. Duality-induced regularizer for semantic matching knowledge graph embeddings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1652–1667, 2022.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, and Qingming Huang. Er: equivariance regularizer for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5512–5520, 2022b.

Changyi Xiao and Yixin Cao. Knowledge graph completion by intermediate variables regularization. *Advances in Neural Information Processing Systems*, 37:110218–110245, 2024.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

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

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3065–3072, 2020b.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2005.00545*, 2020.

Yushi Bai, Zhitao Ying, Hongyu Ren, and Jure Leskovec. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. *Advances in Neural Information Processing Systems*, 34:12316–12327, 2021.

Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31, 2018.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*, 2019.

Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (Csur)*, 54(10s):1–25, 2022.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*, 2012.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, 56(6):1–42, 2024.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Kristina Toutanova and Danqi Chen. 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, 2015.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, 2013.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

Alexa T McCray. An upper-level ontology for the biomedical domain. *Comparative and Functional genomics*, 4 (1):80–84, 2003.

Jiapu Wang, Boyue Wang, Meikang Qiu, Shirui Pan, Bo Xiong, Heng Liu, Linhao Luo, Tengfei Liu, Yongli Hu, Baocai Yin, et al. A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2308.02457*, 2023.

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. Temporal knowledge graph completion: A survey. *arXiv preprint arXiv:2201.08236*, 2022.

Jiaxin Pan, Mojtaba Nayyeri, Yinan Li, and Steffen Staab. Hge: embedding temporal knowledge graphs in a product space of heterogeneous geometric subspaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8913–8920, 2024b.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*, 2018.

Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*, 2023b.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 2002.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

A Appendix

A.1 Limitations and Future Works

Although in this paper, we have demonstrated through a large number of experiments that regularization methods are crucial for the KGC model. And the simple and easy-to-use SPR regularization method we proposed has also achieved very good results. However, we also noticed two limitations of this paper and areas for improvement. First of all, it is well known that transform-based pre-trained language models and large language models have been very popular in recent years. Now, there is also a series of KGC models based on natural language processing. For example, LLM-DA [Wang et al., 2024], LSTK-TELM [Qi et al., 2023], ChatRule [Luo et al., 2023b], etc. However, since many non-open-source large models are still difficult for us to reproduce, we have not yet conducted thorough ablation and experiments on KGC models based on NLP, and have not yet thoroughly verified whether such methods require regularization methods. We will further explore the application of regularization methods on more types of KGC models in future work. Secondly, we are currently using a manual grid search method to determine the SPR sparsification threshold δ. This fixed value will limit the adaptability of the model to a certain extent. In the future, we will explore how to use adaptive forms to automatically select the best threshold δ of the model. This method may be able to better use regularization methods to further break through the upper limit of the KGC model effect.

A.2 RELATED WORK

Translation-based Embedding Models This type of model[Cao et al., 2024] embeds entities and relations in a low-dimensional vector space and regards relations as some kind of translation or geometric transformation operation between entity vectors. Representative models include TransE and its variants. The central idea of this type of model is very intuitive: embed both entities and relations in KG into a low-dimensional continuous vector space. For an existing fact triple (e_i, r_k, e_j), the model assumes that the embedding vector of the head entity e_i plus the embedding vector of the relation r_k should be "close" to the embedding vector of the tail entity t. Mathematically, it can be expressed as $e_i + r_k \approx e_j$. The "closeness" is usually quantified by some distance metric $D(e_i + r_k, e_j)$, such as the Euclidean distance. The design of the scoring function is often based on this distance. The smaller the distance, the higher the rationality score of the triple. This geometric intuition of interpreting relations as translation operations in vector space gives this type of model simplicity, ease of understanding, and better scalability.Typical representatives of this type of model include TransE [Bordes et al., 2013], TransH [Wang et al., 2014], TransR [Lin et al., 2015], RotatE [Sun et al., 2019], HAKE [Zhang et al., 2020b], GIE [Cao et al., 2022a], ROTH [Chami et al., 2020], ConE [Bai et al., 2021] and other models.

Tensor decomposition-based models (TDB) The foundation of this type of method is to represent a knowledge graph as a third-order tensor $\mathcal{A}$. If the knowledge graph contains $|E|$ entities and $|R|$ relations, then $\mathcal{A}$ is a $|E| \times |R| \times |E|$ tensor. The elements of the tensor $\mathcal{A}_{ijk}$ are typically binary: if the triple (e_i, r_k, e_j) exists in the knowledge graph, then $\mathcal{A}_{ijk} = 1$; otherwise, it is 0. The CP model approximates an N-order tensor as a sum of several rank-1 tensors. For a third-order tensor $\mathcal{A}$, the CP decomposition [Hitchcock, 1927] can be written as $\mathcal{A} \approx \sum_{f=1}^{d} \mathbf{h}_f \circ \mathbf{r}_f \circ \mathbf{t}_f$, where d is the rank of the decomposition (often equal to the embedding dimension), and $\mathbf{h}_f, \mathbf{r}_f, \mathbf{t}_f$ are the vectors corresponding to the head entity, relation and tail entity of the f-th factor, respectively. The symbol $\circ$ denotes the outer product. Many KGC models based on tensor decomposition, such as DistMult [Yang et al., 2014], ComplEx [Trouillon et al., 2016], SimplE [Kazemi and Poole, 2018], RESCAL [Nickel et al., 2011], and TuckER [Balažević et al., 2019], have scoring functions closely related to CP decomposition.

Regularization Method Regularization [Santos and Papa, 2022] is an important machine learning technique that aims to constrain the complexity of the model and prevent the model from learning the noise in the training data by adding a penalty term to the model's loss function. Regularization can effectively reduce the variance of the model and improve its effect and robustness on test data. Although regularization has been proven to be effective in general machine learning tasks, its application in the specific field of KGC and its impact on different KGC models still need to be further studied. Classic L2 [Cortes et al., 2012] or Frobenius norm regularization is still the default configuration of many KGC models. As researchers continue to explore regularization methods, many new methods have been proposed. These include DURA [Zhang et al., 2020a; Wang et al., 2022], which uses the idea of duality to reduce the representation of entities and relationships, ER [Cao et al., 2022b], which uses the idea of equivariance to reduce the representation of entities and relationships, VIR [Xiao and Cao, 2024], which uses the setting of intermediate variables to reduce entities and relationships, and N3 [Lacroix et al., 2018], which uses the tensor kernel p norm (p=3) to regularize explicit triple interactions. Although these regularization methods have achieved results in KGC tasks, their audience models are still very niche and are usually limited to TDB models. In this paper, we found that more than just the TDB model has serious overfitting problems, so we want to propose a more general and simpler KGC regularization method.

A.3 Experiment Setting

Dataset This study selected five widely used knowledge graph benchmark datasets: WN18RR [Dettmers et al., 2018] (improved from the WordNet subset WN18 [Bordes et al., 2013], which solves the data leakage problem by removing reversible relations and supports common sense reasoning), FB15K-237 [Toutanova and Chen, 2015] (constructed by deleting redundant reverse relations from the Freebase subset, suitable for general knowledge reasoning), YAGO3-10 [Mahdisoltani et al., 2013] (entities with more than 10 categories of relations screened from multilingual YAGO3, containing 1.17 million high-attribute density triples), Kinship [Kemp et al., 2006] (describing the high-density kinship relationships of Australian tribes) and UMLS [McCray, 2003] (based on the American Medical Standard Terminology System). ICEWS14 [García-Durán et al., 2018] and ICEWS05-15 [García-Durán et al., 2018] are two datasets widely used in the study of temporal knowledge graphs. Both datasets provide important resources for the study of temporal knowledge graphs with their high-quality event data and clear time tags. The statistics of the dataset are shown in Table 4 and Table 5.

Table 4: Dataset Statistics

Dataset	#Entity	#Relation	#Train	#Valid	#Test
WN18RR [Dettmers et al., 2018]	40,943	11	86,835	3,034	3,314
FB15K-237 [Toutanova and Chen, 2015]	14,541	237	272,115	17,535	20,466
YAGO3-10 [Mahdisoltani et al., 2013]	123,182	37	1,079,040	5,000	5,000
Kinship [Kemp et al., 2006]	104	26	8,544	1,068	1,074
UMLS [McCray, 2003]	135	46	5,216	652	661

Table 5: Statistical of ICEWS14 and ICEWS05-15 datasets.

Datasets	#Entities	#Relations	#Timestamps	#Time Span	#Training	#Validation	#Test	#Granularity	#Category
ICEWS14 [García-Durán et al., 2018]	6,869	230	365	01/01/2014–12/31/2014	72,826	8,941	8,963	24 hours	Interpolation
ICEWS05-15 [García-Durán et al., 2018]	10,094	251	4,017	01/01/2005–12/31/2015	368,962	46,275	46,092	24 hours	Interpolation

Evaluation metrics To evaluate the performance of the SPR method in KGC models, we utilize five primary metrics: Mean Reciprocal Rank (MRR), Hits@1, Hits@3, Hits@10, and the Mean Reciprocal (MR). MRR evaluates the ranking quality of the prediction results. For each test query (typically a triplet in a missing link prediction task), the model generates a ranked list of candidate entities and MRR quantifies the average inverse rank of correct predictions. Hits@N measures the proportion of correct entities ranked within the top N positions, reflecting the model's precision at different truncation points. These metrics are complemented by MR, which quantifies prediction stability: smaller MR values indicate a lower fluctuation in model outputs across evaluations, reflecting greater robustness. The definitions of these metrics are as follows.

$$\mathbf{MRR} = \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{\text{rank}i} = \frac{1}{|S|}\left(\frac{1}{\text{rank}1}+\frac{1}{\text{rank}2}+\ldots+\frac{1}{\text{rank}|S|}\right)$$

where S denotes the set of triples, $|S|$ represents the number of triples in the set, and $rank_i$ is the link prediction rank of the i-th triple.

$$\mathbf{MR} = \frac{1}{|S|}\sum_{i=1}^{|S|}\text{rank}_i = \frac{1}{|S|}\left(\text{rank}_1+\text{rank}_2+\ldots+\text{rank}_{|S|}\right)$$

where MR represents the average ranking of all triples. It reflects the average ranking of all triples by the link prediction model. The smaller the ranking, the more accurate the model's prediction of the triples, which also means that the fluctuation of the model's prediction is smaller. Therefore, the smaller the MR, the better the performance of the model.

$$\mathbf{Hits}@N = \frac{1}{|S|}\sum_{i=1}^{|S|}\mathbb{I}\left(\text{rank}_i \leq n\right)$$

The function II$(\cdot)$ used in our analysis serves as an indicator function. It returns a value of 1 when the specified condition is met and 0 otherwise. In our experiments, this indicator function is evaluated at various thresholds, with common values of n being 1, 3, or 10. These values correspond to the widely used metrics in KGC tasks: Hits@1, Hits@3, and Hits@10, respectively.

A.3.1 Implementation Details

All our experiments were conducted on a server equipped with 1T RAM, an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, and 8 V100 GPUs with 32 GB each.

All baseline models were reproduced from the GitHub of their respective original papers (including DURA[1], ER[2], GIE[3], HGE[4], CompGCN[5], and other models), and the experiments were run on the same machine using the optimal hyperparameters given in their papers. We provide Algorithm 1 and Algorithm 2 to help readers better understand the idea of SPR regularization. For the task of TKGC, we use HGE here. We only change the regularization term of the entity and relationship interaction, but not the temporal regularization term.

The optimal hyperparameters employed in this study were determined systematically via grid search. Specifically, the parameter δ was selected from the set $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.45, 0.5\}$, the regularization rate was tuned from the candidate set $\{5 \times 10^{-1}, 5 \times 10^{-2}, 5 \times 10^{-3}, 5 \times 10^{-4}, 5 \times 10^{-5}\}$, and the embedding dimension was optimized over the range $\{50, 100, 400, 1000, 2000, 3000, 4000\}$. The final selection of hyperparameters was based on performance metrics achieved through validation experiments.

Algorithm 1 SPR Regularization for a Triple

```
 1: function SPR($\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t, \delta$)
    Input:
      $\mathcal{V}_i^h \in \mathbb{R}^D$: Embedding vector of the head entity
      $\mathcal{R}_j^r \in \mathbb{R}^D$: Embedding vector of the relation
      $\mathcal{V}_k^t \in \mathbb{R}^D$: Embedding vector of the tail entity
      $\delta > 0$: Threshold parameter controlling sparsity
    Output:
      SPR regularization term (scalar) for the triple
 2:   $\mathbf{x}_1 \leftarrow (\mathcal{V}_i^h)^2$                                   ▷ Element-wise square of head entity embedding
 3:   $\mathbf{x}_2 \leftarrow (\mathcal{V}_k^t)^2$                                   ▷ Element-wise square of tail entity embedding
 4:   $\mathbf{x}_3 \leftarrow (\mathcal{V}_i^h)^2 \odot (\mathcal{R}_j^r)^2$          ▷ Element-wise product of squared head and relation embeddings
 5:   $\mathbf{x}_4 \leftarrow (\mathcal{V}_k^t)^2 \odot (\mathcal{R}_j^r)^2$          ▷ Element-wise product of squared tail and relation embeddings
 6:   spr $\leftarrow 0$                                                          ▷ Initialize SPR regularization term
 7:   for $k = 1$ to $4$ do                                                      ▷ Iterate over the four computed vectors
 8:     $\mathbf{x} \leftarrow \mathbf{x}_k$                                          ▷ Select the current vector
 9:     Sort $\mathbf{x}$ in non-decreasing order: $\mathbf{x}_{(1)} \leq \mathbf{x}_{(2)} \leq \cdots \leq \mathbf{x}_{(D)}$    ▷ Order elements
10:     Compute cumulative sum: $c_i = \sum_{j=1}^{i} \mathbf{x}_{(j)}$ for $i = 1$ to $D$    ▷ Calculate running total
11:     Find the largest $S$ such that $c_S \leq \delta$                             ▷ Determine elements to mask
12:     if $S < D$ then                                                          ▷ Check if some elements exceed threshold
13:       spr $\leftarrow$ spr $+ \sum_{i=S+1}^{D} \mathbf{x}_{(i)}$                 ▷ Add unmasked elements to SPR
14:     end if
15:   end for
16:   return SPR regularization term                                            ▷ Return total SPR regularization term
17: end function
```

[1]https://github.com/MIRALab-USTC/KGE-DURA

[2]https://github.com/Lion-ZS/ER

[3]https://github.com/Lion-ZS/GIE

[4]https://github.com/NacyNiko/HGE

[5]https://github.com/malllabiisc/CompGCN

Algorithm 2 Compute SPR Regularization for a Batch

```
1: function COMPUTE SPR REGULARIZATION(batch, $\delta$)
   Input:
      batch $= \{(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)\}$: Set of triples with head entity $v_h$, relation $\rho_r$, tail entity $v_t$
      $\delta > 0$: Threshold parameter for sparsity regularization
   Output:
      Average SPR regularization term (scalar) across the batch
2:    total_spr $\leftarrow 0$                                    ▷ Initialize accumulator for total SPR
3:    for each triple $(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t)$ in batch do      ▷ Process each triple
4:       Fetch embeddings: $\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t$ for $v_h, \rho_r, v_t$      ▷ Retrieve vectors
5:       total_spr $\leftarrow$ total_spr + SPR$(\mathcal{V}_i^h, \mathcal{R}_j^r, \mathcal{V}_k^t, \delta)$      ▷ Accumulate SPR
6:    end for
7:    $\Omega_{\text{SPR}} \leftarrow \frac{\text{total\_spr}}{|\text{batch}|}$                 ▷ Compute average SPR
8:    return $\Omega_{\text{SPR}}$                          ▷ Return batch regularization term
9: end function
```

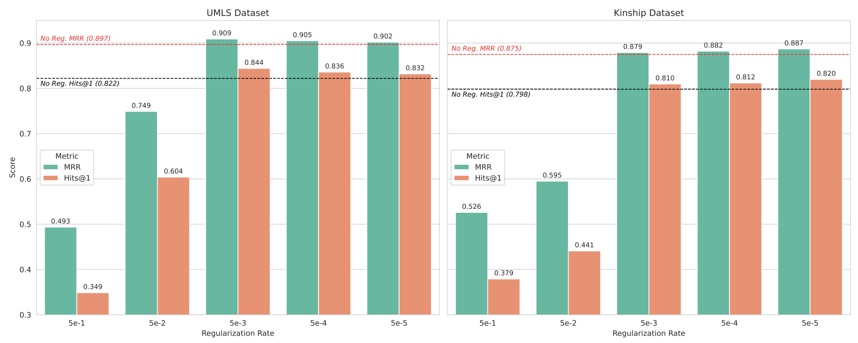

Figure 7: Analysis of the Impact of Regularization Rate on Model Performance for UMLS and Kinship Datasets

A.4 Further Empirical Research

A.4.1 Analyzing the Impact of Regularization on KGC Model Training Curves

All the provided Figure.8, Figure.9, and Figure.11 consistently show that there is a very serious overfitting problem when training knowledge graph completion models such as CP, ComplEx, RESCAL, and HGE without SPR regularization. This means that many existing types of KGC models have this problem of overfitting (including models based on tensor decomposition, models based on embedding, models based on graph neural networks, and TKGC models). In various datasets (WN18RR, YAGO3-10, FB15k-237, ICEWS14), models trained without regularization ("No Reg") quickly achieved near-perfect scores on training data (MRR and Hits@1 close to 1.0). However, their performance on validation set data ("No Reg - Valid") is still significantly lower, and often quickly stagnates at a suboptimal level. This large discrepancy between high-trained model prediction metrics and low validation accuracy is a clear indication of severe overfitting, where the model memorizes the training examples but fails to learn generalizable patterns for unseen data.

In contrast, adding regularization ("Add Reg") significantly improves the generalization ability of the model. All Figure.8, Figure.9, and Figure.11 consistently show that the "Add Reg - Valid" curve achieves significantly higher MRR and Hits@1 scores on the validation set than the "No Reg - Valid" curve. This suggests that regularization helps the model learn more generalizable capabilities. Although the regularized models may converge slightly slower and have slightly lower peak performance on the training set than the unregularized models, they achieve significantly higher scores on the validation set for all tested models and datasets. The gap between training and validation performance is significantly narrowed, indicating that regularization effectively mitigates overfitting. Therefore, these results highlight the important role of regularization in developing robust knowledge graph completion models that perform well not only on the training data but, more importantly, on new, unseen data.

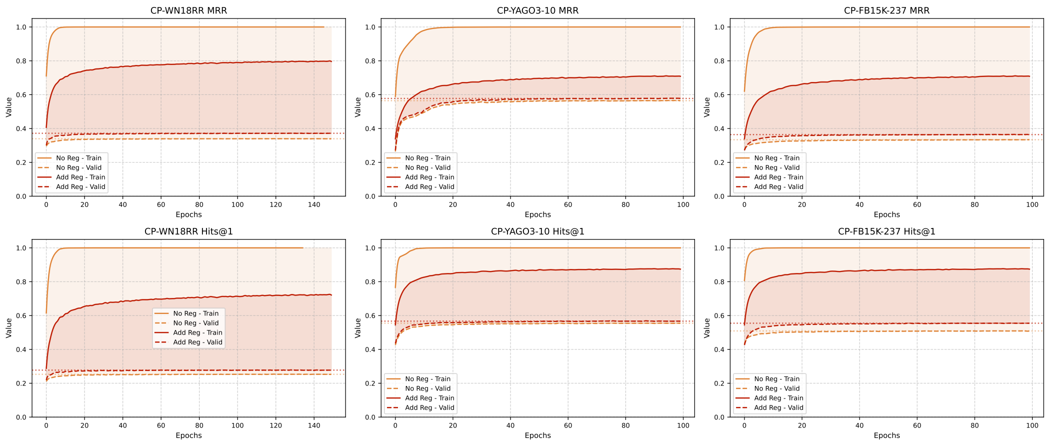
Figure 8: Visualization experiment of the CP model using different regularization methods and no regularization method on the FB15K-237, WN18RR, and YAGO3-10 datasets.

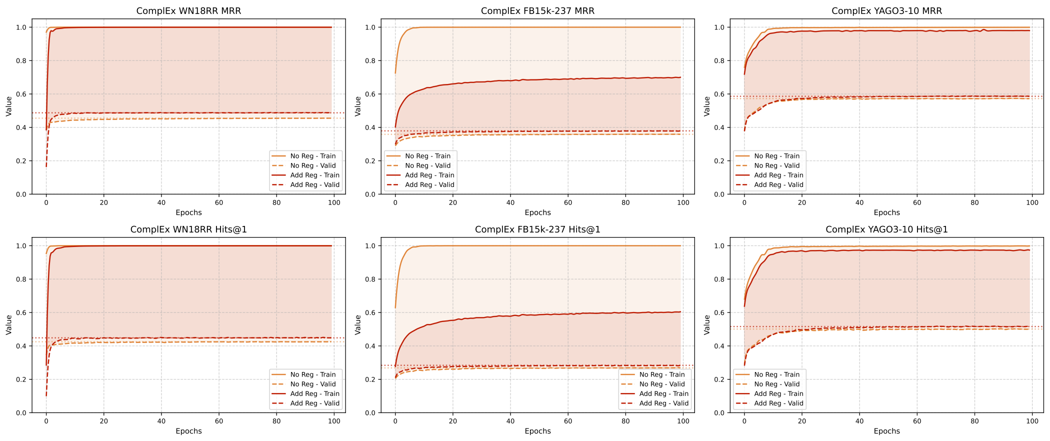
Figure 9: Visualization experiment of the ComplEx model using different regularization methods and no regularization method on the FB15K-237, WN18RR, and YAGO3-10 datasets.

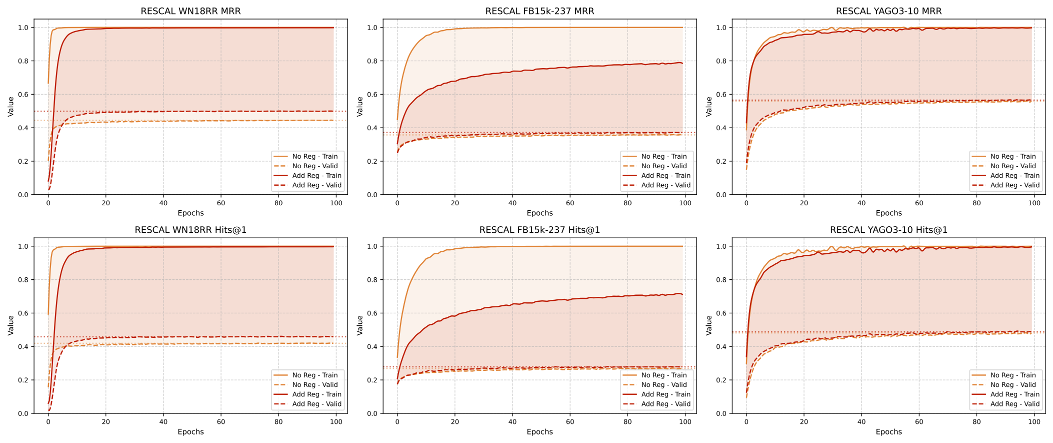

Figure 10: Visualization experiment of the RESCAL model using different regularization methods and no regularization method on the FB15K-237, WN18RR, and YAGO3-10 datasets.

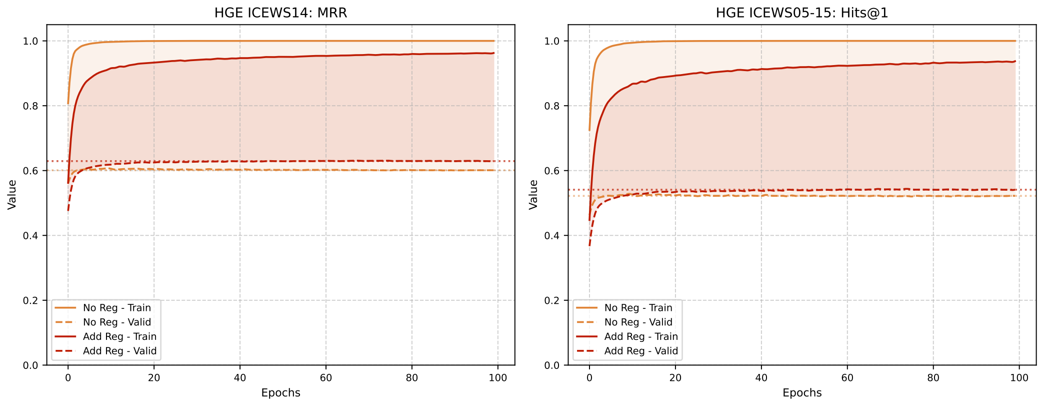

Figure 11: Visualization experiment of the HGE model using different regularization methods and no regularization method on the ICEWS14 and ICEWS05-15 datasets.

A.4.2 How Datasets Composition Influences Regularization Performance

We noticed that the improvement ratios of multiple models on the three main datasets are not the same, and there seems to be some underlying rules. In order to explore the rules of the datasets, we give Table 6 as a reference. In the above, we conducted a deeper exploration of the composition structure of the three main datasets, WN18RR, FB15K-237, and YAGO3-10. Since the data volume of UNLS and kinship is too small and the topological structure is particularly simple, we can consider them as toy datasets, so we will not explore them in depth again.

Table 6: Dataset Statistics and #Train/#Rel

Dataset	Training Triples	Entities	Relations	Avg. (#Train/#Rel)
FB15K-237	272,115	14,541	237	$\approx$ 1148 (Sparsest)
WN18RR	86,835	40,943	11	$\approx$ 7894
YAGO3-10	1,079,040	123,182	37	$\approx$ 29,163 (Densest)

The overall trend of training set size and regularization improvement First, we sort the training set sizes from small to large: WN18RR (about 87,000) < FB15K-237 (about 270,000) < YAGO3-10 (about 1.08 million). We found that when observing the data of YAGO3-10 (the largest training set), no matter which model (CP, ComplEx, GIE, RESCAL), the percentage of Hits@1 improvement is generally the lowest (mostly between 2%-4%, and some indicators are slightly higher). We noticed that the data of WN18RR (the smallest training set) generally has the highest or close to the highest percentage of Hits@1 improvement (mostly between 5%-18%). FB25K-237 is right in the middle. Through the above observations, we can preliminarily believe that large-scale datasets with more training sets themselves provide richer and more diverse knowledge. The KGC model is less likely to "solidify" the noise and special samples in the training during the learning process; that is, the risk of overfitting is relatively low. Therefore, the benefits brought by the addition of regularization methods are relatively small. On the contrary, the role of regularization is more critical on smaller training set data sets.

A.5 Technical Appendices and Proofs

Lemma 1 (Sparsification Error Bound). Let $A = (A_1, A_2, \ldots, A_n)^\top \in \mathbb{R}^n_{\geq 0}$ be a nonnegative vector and suppose that the operator select-small(A, δ) produces a binary mask $m \in \{0, 1\}^n$ .Such that the cumulative sum of the "dropped" entries is bounded by δ, i.e., $\sum_{i:m_i=1} A_i \leq \delta$. Then the difference between the full sum and the sparsified sum is bounded as:

$$\left| \sum_{i=1}^{n} A_i - \sum_{i=1}^{n} A_i \left(1 - m_i\right) \right| \leq \delta$$

Proof. We first consider the non-negative vector A. Observing the non-negativity of its components, it is natural to arrange the entries in ascending order. More precisely, let π be a permutation of $\{1, \ldots, n\}$ such that the sequence satisfies:

$$A_{\pi(1)} \leq A_{\pi(2)} \leq \cdots \leq A_{\pi(n)}$$

Using this ordering, we define an index S as the maximum integer for which the cumulative sum of the smallest S entries does not exceed the threshold δ; that is:

$$S = \max \left\{ k \in \{0, 1, \ldots, n\} : \sum_{s=1}^{k} A_{\pi(s)} \leq \delta \right\}$$

The operator select-small(A, δ) then produces the mask m by setting:

$$m_{\pi(s)} = 1 \quad \text{for } s = 1, 2, \ldots, S,$$
$$m_{\pi(s)} = 0 \quad \text{for } s = S + 1, \ldots, n.$$

Based on the construction, one can immediately conclude that:

$$\sum_{i:m_i=1} A_i = \sum_{s=1}^{S} A_{\pi(s)} \leq \delta$$

We now discuss the quantity for which we wish to establish bounds. The sparsification is defined as:

$$\sum_{i=1}^{n} A_i \left(1 - m_i\right),$$

which represents the sum of those components of A for which $m_i = 0$. Consequently, the difference between the full sum and the sparsified sum is given by:

$$\sum_{i=1}^{n} A_i - \sum_{i=1}^{n} A_i (1 - m_i) = \sum_{i=1}^{n} A_i m_i.$$

Since m_i is nonzero only for the indices corresponding to the S smallest elements (as determined by π), this expression is exactly equal to :

$$\sum_{s=1}^{S} A_{\pi(s)}$$

which by the definition of S is bounded by δ. For completeness, we also assume by contradiction that the absolute difference exceeds δ, that is:

$$\left| \sum_{i=1}^{n} A_i - \sum_{i=1}^{n} A_i (1 - m_i) \right| > \delta$$

Since all entries of A are nonnegative, the absolute value may be removed, and we obtain:

$$\sum_{i:m_i=1} A_i > \delta$$

This, however, directly contradicts the construction of the mask m, which ensures that the sum of the masked (dropped) entries does not exceed δ. Hence, our assumption must be false, and it follows that:

$$\left| \sum_{i=1}^{n} A_i - \sum_{i=1}^{n} A_i (1 - m_i) \right| \le \delta$$

In summary, by carefully sorting the entries of A and choosing the mask m to drop precisely those entries whose cumulative sum is controlled by δ, we have rigorously shown that the error introduced by sparsification is bounded by δ.

A.6 Theoretical Analysis

A.6.1 SPR vs. Dropout Theory

In this section we analyse two regularisation schemes: classical Dropout [Srivastava et al., 2014] and selective parameter regularisation (SPR). With Dropout we zero individual embedding coordinates according to an i.i.d. Bernoulli mask that is *independent* of their current values. Denoting the mask by $\mathbf{m} \sim \mathrm{Bernoulli}(1-p)^d$ and choosing the *non-inverted* implementation (i.e. no division by $1-p$ during training), the stochastic forward pass $\tilde{\mathbf{x}} = \mathbf{x} \odot \mathbf{m}$ induces the deterministic quadratic penalty:

$$\mathbb{E}_{\mathbf{m}}\big[\|\tilde{\mathbf{x}}\|_2^2\big] = (1-p)\,\|\mathbf{x}\|_2^2$$

By contrast, SPR employs a *deterministic* discard mask $\mathcal{M}(\mathbf{x}) \in \{0,1\}^d$ with the convention $\mathcal{M}_d = 1$ if the d-th coordinate is pruned. The largest coordinates are kept until the *discarded energy* $\sum_{d:\,\mathcal{M}_d=1} x_d^2$ does not exceed a threshold $\delta > 0$. The vector that enters the loss is therefore

$$\mathbf{x}_{\text{keep}} \;=\; \mathbf{x} \odot \big(1 - \mathcal{M}(\mathbf{x})\big), \qquad \|\mathbf{x}_{\text{keep}}\|_2^2 \;=\; \sum_{d:\,\mathcal{M}_d=0} x_d^2.$$

Because the SPR mask depends on the current data whereas the Dropout mask is random, the two methods exhibit fundamentally different statistics and optimisation behaviour.

Capacity (Rademacher-complexity) comparison For linear or bilinear KGC scorers the empirical Rademacher complexity is bounded by a constant multiple of the expected ℓ_2-norm of the *effective* embeddings. Let C be the Lipschitz constant of the scorer. Dropout yields

$$\mathfrak{R}_{\text{drop}} \;\le\; C\,\sqrt{1-p}\;\mathbb{E}\big[\|\mathbf{x}\|_2\big],$$

whereas SPR gives

$$\mathfrak{R}_{\text{SPR}} \;\le\; C\,\mathbb{E}\Big[\|\mathbf{x}\|_2\,\sqrt{1 - \frac{\delta}{\|\mathbf{x}\|_2^2}}\Big].$$

Whenever $\delta < p\,\|\mathbf{x}\|_2^2$ which covers the typical settings in this paper $p \approx 0.2$–0.5 and $\delta \le 0.4\,\|\mathbf{x}\|_2^2$, we have $\mathfrak{R}_{\text{SPR}} < \mathfrak{R}_{\text{drop}}$. The smaller capacity tightens the generalisation bound and explains SPR's consistently lower test error.

Optimal in class property of SPR Define $\mathcal{S}_\delta = \{\mathbf{z} \in \mathbb{R}^D_{\geq 0} \mid \|\mathbf{x} - \mathbf{z}\|_1 \leq \delta\}$. SPR chooses $\mathbf{x}_{\text{sparse}} = \arg\min_{\mathbf{z} \in \mathcal{S}_\delta} \|\mathbf{z}\|_2^2$, because hard-thresholding the smallest coordinates uniquely solves this constrained problem. It therefore achieves the minimum possible ℓ_2 penalty among all vectors that distort $\mathbf{x}$ by at most δ. Dropout produces a random $\tilde{\mathbf{x}} \in \mathcal{S}_\infty$ whose expected norm is strictly larger, so the regularised objective is never lower in expectation.

Concrete signal–noise example Consider an embedding with k informative coordinates of magnitude $a \gg 0$ and $n - k$ noisy ones of magnitude ϵ. With $\delta < (n - k)\epsilon^2$, SPR removes *all* noise and keeps all signal, giving a penalty of ka^2. Dropout retains each coordinate with probability $1 - p$, so the expected penalty is $(1 - p)\big(ka^2 + (n - k)\epsilon^2\big)$, while roughly pk signal dimensions are lost, degrading the score. Because $a \gg \epsilon$, SPR achieves both a smaller penalty and larger retained signal.

Empirical corroboration We evaluate Dropout and SPR on two variants of the CompGCN framework (TransE and ConvE scorers). Without regularisation, the models reach MRR values of 0.216 and 0.302, respectively. Dropout improves these to 0.333 and 0.352. SPR pushes them further to 0.359 and 0.371 about 7–8% relative improvement over Dropout. These empirical results support the theoretical analysis: SPR's lower variance, reduced capacity, and optimal masking translate into consistent performance gains.

Proposition 1 *Let $\ell(\theta)$ be the unregularised loss and $\mathbf{x}(\theta)$ the parameter-dependent feature vector. Dropout draws $\mathbf{m} \sim$ Bernoulli$(1 - p)^D$, whereas SPR uses the deterministic selector $\mathcal{M}\big(\mathbf{x}(\theta)\big)$ defined earlier. Conditioning on current parameters θ,*

$$\mathrm{Var}_{\mathbf{m}}\Big[\nabla_\theta\big(\ell(\theta) + \lambda\|\mathbf{m} \odot \mathbf{x}\|_2^2\big)\Big] = \lambda^2 p(1 - p)\|\mathbf{x}(\theta)\|_2^2, \qquad \mathrm{Var}\Big[\nabla_\theta\big(\ell(\theta) + \lambda\|\mathcal{M}(\mathbf{x}) \odot \mathbf{x}\|_2^2\big)\Big] = 0.$$

Proof. Because Dropout masks are sampled independently of the data and the parameters, they act as exogenous noise. Writing the Dropout-regularised objective as $L_{\text{drop}}(\theta, \mathbf{m}) = \ell(\theta) + \lambda\|\mathbf{m} \odot \mathbf{x}(\theta)\|_2^2$, the gradient decomposes into

$$\nabla_\theta L_{\text{drop}} = \underbrace{\nabla_\theta \ell(\theta)}_{\text{data / parameters}} + \lambda \mathbf{m} \odot \mathbf{x}(\theta),$$

so $\mathrm{Var}_{\mathbf{m}}\big[\nabla_\theta L_{\text{drop}}\big] = \lambda^2 p(1 - p)\|\mathbf{x}(\theta)\|_2^2$, where $m_d \in \{0, 1\}$. In contrast, the SPR objective $L_{\text{SPR}}(\theta) = \ell(\theta) + \lambda\|\mathcal{M}(\mathbf{x}(\theta)) \odot \mathbf{x}(\theta)\|_2^2$ contains no external randomness once θ is fixed, so its gradient is deterministic and the conditional variance is zero.

A.7 Model Complexity and Generalization

To assess Sparse Peak Regularization (SPR)'s effect on generalization, we employ Rademacher complexity, a key measure in statistical learning theory that bounds the generalization error of a hypothesis class. For a Knowledge Graph Completion (KGC) scoring function $f_\theta(v_h, r, v_t)$ with Lipschitz constant C, the empirical Rademacher complexity $\mathfrak{R}(\mathcal{H})$ of the hypothesis class $\mathcal{H}$ is influenced by the expected norm of the embeddings, providing insight into the model's capacity to generalize.

For Dropout, a widely used stochastic regularization technique, the Rademacher complexity is bounded as:

$$\mathfrak{R}_{\text{drop}} \leq C\sqrt{1 - p}\,\mathbb{E}[\|\mathbf{x}\|_2],$$

where p is the dropout probability, and $\mathbf{x}$ denotes the embedding vectors. This bound reflects Dropout's mechanism of randomly masking components, reducing the effective embedding norm by a factor of $\sqrt{1 - p}$ in expectation [Srivastava et al., 2014].

For SPR, which deterministically retains the larger components of the embeddings, the complexity is bounded as:

$$\mathfrak{R}_{\text{SPR}} \leq C\,\mathbb{E}\left[\sqrt{\sum_{d:\mathcal{M}_d(\mathbf{x})=0} x_d^2}\right] = C\,\mathbb{E}[\|\mathbf{x}_{\text{sparse}}\|_2],$$

where $\mathcal{M}_d(\mathbf{x}) = 0$ indicates that the d-th component is retained, and $\mathbf{x}_{\text{sparse}}$ is the sparse embedding after masking. Since the sum of the squared magnitudes of the masked components is constrained by $\sum_{d:\mathcal{M}_d(\mathbf{x})=1} x_d^2 \leq \delta$, we have:

$$\|\mathbf{x}_{\text{sparse}}\|_2^2 = \|\mathbf{x}\|_2^2 - \sum_{d:\mathcal{M}_d(\mathbf{x})=1} x_d^2 \geq \|\mathbf{x}\|_2^2 - \delta.$$

Thus, $\mathfrak{R}_{\text{SPR}} \leq C\,\mathbb{E}\left[\sqrt{\|\mathbf{x}\|_2^2 - \delta}\right]$. The effectiveness of this bound compared to Dropout depends on δ and the distribution of $\mathbf{x}$. For instance, a small δ may yield a tighter bound under specific conditions, enhancing generalization.

Moreover, SPR acts as an adaptive sparsity-inducing regularizer. By masking smaller components, it reduces the effective dimensionality of the embedding space, which is advantageous in high-dimensional settings where many dimensions may represent noise rather than signal [Hastie et al., 2009]. This selective retention of dominant components mirrors feature selection strategies, enhancing the model's focus on informative features [Guyon and Elisseeff, 2003].

Consider an embedding $\mathbf{x}$ with signal components of large magnitudes and noise components of small magnitudes. SPR's masking of the smallest components resembles hard thresholding in signal processing, known to enhance the signal-to-noise ratio by eliminating noise, thereby improving generalization [Donoho, 2002]. In contrast, Dropout's random masking may discard informative components, potentially reducing its effectiveness at preserving the signal.

Thus, through selective feature retention and signal-to-noise enhancement, SPR demonstrates a strong capacity to control model complexity and enhance generalization in KGC tasks compared to Dropout.

A.7.1 Selective Retention of Informative Components

Proposition 2: *SPR selectively retains the most significant components of the embeddings, potentially leading to better generalization by focusing on informative features while discarding noise.*

Proof. By design, SPR masks the smallest components of $\mathbf{x}$, which are more likely to represent noise or less informative features. This ensures the model prioritizes significant dimensions, enhancing the signal-to-noise ratio and improving generalization.

A.8 Optimization Dynamics

We investigate SPR's influence on optimization by analyzing the gradient of its regularization term and its implications for convergence. For $\Omega_{\text{SPR}} = \|\mathbf{x}_{\text{sparse}}\|_2^2 = \sum_{d:\mathcal{M}_d(\mathbf{x})=0} x_d^2$, treating the mask $\mathcal{M}(\mathbf{x})$ as fixed per iteration (a common approximation in proximal gradient methods), the gradient is:

$$\nabla_{x_d}\Omega_{\text{SPR}} = 2x_d(1 - \mathcal{M}_d(\mathbf{x})),$$

which is zero for masked components ($\mathcal{M}_d(\mathbf{x}) = 1$) and $2x_d$ for retained components. This selective penalization targets larger components, unlike L2 regularization's uniform gradient:

$$\nabla_{x_d}\Omega_{\text{L2}} = 2x_d,$$

which shrinks all dimensions equally [Bishop and Nasrabadi, 2006]. Discussing the implications of selective penalization. This selective approach offers key benefits:

Noise Reduction: By penalizing dominant dimensions, SPR prevents overfitting to noise or spurious correlations, common in complex datasets like knowledge graphs. Unlike L2 regularization, SPR preserves smaller components that may represent meaningful signals.

Effective Dimensionality Reduction: Masked components have zero regularization gradient, effectively freezing those dimensions during updates. This reduces the parameter space, akin to pruning, simplifying the optimization landscape.

Additionally, SPR's deterministic nature yields zero conditional variance in the gradient, unlike Dropout's stochastic noise [Srivastava et al., 2014]. This stability enhances gradient update reliability in stochastic gradient descent, potentially accelerating convergence [Bottou et al., 2018].

In summary, SPR's selective and deterministic regularization mitigates overfitting, reduces the optimization space, and stabilizes gradients, making it robust for training KGC models.

A.9 Empirical Support

Empirical results (Table 1) validate these insights, showing SPR enhances metrics like MRR and Hits@k across KGC models and datasets. For example, on WN18RR, SPR improves the CP model's MRR from 0.438 to 0.449, supporting reduced overfitting and improved generalization.

In conclusion, SPR's deterministic and selective regularization offers superior control over model complexity, gradient stability, and optimization efficiency, as evidenced by theoretical analysis and empirical gains.