Soft Reasoning Paths for Knowledge Graph Completion

Yanning Hou[1], **Sihang Zhou**[1*], **Ke Liang**[2], **Lingyuan Meng**[2], **Xiaoshu Chen**[2], **Ke Xu**[3], **Siwei Wang**[2], **Xinwang Liu**[2], **Jian Huang**[1]

[1]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China

[2] College of Computer Science and Technology, National University of Defense Technology, Changsha, China

[3]School of Artifcial Intelligence, Anhui University, Hefei, China

{hyn241513, sihangjoe}@gmail.com,
liangke200694@126.com, mly_edu@163.com, xschenranker@gmail.com, xuke@ahu.edu.cn,
{wangsiwei13, xinwangliu, huang_jian}@nudt.edu.cn

Abstract

Reasoning paths are reliable information in knowledge graph completion (KGC) in which algorithms can find strong clues of the actual relation between entities. However, in real-world applications, it is difficult to guarantee that computationally affordable paths exist toward all candidate entities. According to our observation, the prediction accuracy drops significantly when paths are absent. To make the proposed algorithm more stable against the missing path circumstances, we introduce soft reasoning paths. Concretely, a specific learnable latent path embedding is concatenated to each relation to help better model the characteristics of the corresponding paths. The combination of the relation and the corresponding learnable embedding is termed a soft path in our paper. By aligning the soft paths with the reasoning paths, a learnable embedding is guided to learn a generalized path representation of the corresponding relation. In addition, we introduce a hierarchical ranking strategy to make full use of information about the entity, relation, path, and soft path to help improve both the efficiency and accuracy of the model. Extensive experimental results illustrate that our algorithm outperforms the compared state-of-the-art algorithms by a notable margin. Our code will be released at https://github.com/7HHHHH/SRP-KGC.

1 Introduction

Knowledge graphs (KGs) have emerged as a foundational framework for organizing and utilizing structured information in mission-critical domains, including question answering [Sun *et al.*, 2019a; Dinan *et al.*, 2019], recommendation systems [Huang *et al.*, 2018], and information retrieval [Edge *et al.*, 2024]. Structurally, KGs are composed of triples (h, r, t), where h denotes the head entity, r specifies the semantic relationship and t identifies the tail entity. However, despite their practical importance, KGs often exhibit incompleteness. This inherent limitation underscores the importance of Knowledge Graph Completion (KGC) techniques, which play a pivotal role in automating the knowledge graph construction and validation processes.

Existing knowledge graph completion methods can be broadly categorized into two main categories: embedding-based methods [Bordes *et al.*, 2013; Sun *et al.*, 2019b; Balazevic *et al.*, 2019] and text-based methods [Wang *et al.*, 2022a; Qiao *et al.*, 2023; Chen *et al.*, 2023]. With the advent of language models, their advanced linguistic understanding capabilities have significantly improved the performance of text-based methods. Taking full advantage of the semantic relationships between candidate tail entities and the query, text-based approaches have gained widespread adoption due to their substantial improvements in accuracy.

Recent studies [Iwamoto and Kameiwa, 2024; Zha *et al.*, 2021] have investigated the incorporation of reasoning path information into text-based knowledge graph completion, where reasoning paths serve as valuable indicators for predicting entity relationships, leading to significant improvements in prediction accuracy. However, our empirical analysis reveals that these algorithms exhibit a marked decrease in performance when reasoning paths are not available. Furthermore, through a detailed statistical evaluation, we found that approximately 82% of the triples in the WN18RR test set and roughly 27% of the triples in the FB15K-237 test set do not contain valid 2-hop or 3-hop reasoning paths. This observation suggests that a substantial portion of entities lack accessible reasoning paths that could be utilized for relation prediction. Consequently, this limitation severely constrains the performance ceiling of these methods. Moreover, prior path-based methods require reasoning path searches and ranking for all candidate tail entities to achieve high prediction accuracy, resulting in long testing times and limiting their practical application in real-world scenarios.

To address the aforementioned challenges, we propose a knowledge graph completion method based on soft reasoning

[*]Corresponding author.

paths (SRP-KGC). Specifically, the proposed soft reasoning paths are formed by combining relations and learnable embeddings. By assigning an independent learnable embedding to each type of relation and then aligning it with the paths of that relation, our approach enables the modeling of various path information corresponding to the same relation using soft reasoning paths. In cases where reasoning paths are missing, soft reasoning paths effectively fill the gaps, thereby enhancing the stability and robustness of the algorithm in such scenarios.

Additionally, to improve the scalability of the algorithm and mitigate the negative impact of extensive path searches on efficiency while maintaining the accuracy of the ranking, we propose a hierarchical ranking strategy. This approach utilizes a combination of relation, reasoning path, and soft reasoning path evaluation metrics to perform tiered filtering, effectively ensuring the scalability of the algorithm for test entities. Our contributions are summarized as follows:

- We identify an overlooked issue of performance degradation in path-based algorithms when paths are missing and propose a KGC method based on soft reasoning paths that enhances the algorithm's stability against the candidate entities whose path information is absent.
- We propose a hierarchical ranking method based on relations, reasoning paths, and soft reasoning paths, which alleviate the scalability defect of the path-based algorithm and enhances its practical value.
- Extensive experimental results demonstrate that the soft reasoning paths constructed based on trainable embeddings can effectively narrow the semantic gap between relations and their corresponding holistic reasoning paths, while enhancing the discriminative ability of relational representations in path discrimination.

2 Related Work

2.1 Knowledge Graph Completion

Existing methods for knowledge graph completion (KGC) fall into two categories: embedding-based and text-based.Embedding-based methods encode entities and relations as vectors. Translational models (e.g., TransE [Bordes *et al.*, 2013], TransH [Wang *et al.*, 2014]) are efficient but weak in modeling complex patterns. Tensor models like ComplEx [Trouillon *et al.*, 2016] handle diverse relations but scale poorly. Graph neural networks (e.g., CompGCN [Vashishth *et al.*, 2019]) incorporate neighbor information to improve representations, though they require careful architecture design.Text-based methods leverage textual context. KG-BERT [Yao *et al.*, 2021] encodes triples as text for classification. SimKGC [Wang *et al.*, 2022a] and C-LMKE [Wang *et al.*, 2022b] use contrastive learning for better discrimination. However, these methods often rely only on (h, r) and candidate entity text similarity, ignoring richer auxiliary cues.

2.2 Reasoning Path in KGC

Reasoning is crucial for accurate knowledge graph completion (KGC). Unlike traditional embedding-based methods, reasoning path-based approaches capture higher-order relations by exploring paths that reflect semantic or logical connections. GraIL [Teru *et al.*, 2020] uses GNNs to assess path-relation relevance, BERTRL [Zha *et al.*, 2021] encodes reasoning paths and candidate triples with BERT, and ReDistLP [Iwamoto and Kameiwa, 2024] aggregates multiple paths for prediction. These methods excel in inductive KGC but struggle when reasoning paths are missing or candidate triples are abundant.

2.3 Prompt Tuning

Prompt tuning, through the use of prompts, enables pre-trained language models (PLMs) to achieve exceptional performance across various downstream tasks with minimal computational cost. CSProm-KG [Chen *et al.*, 2023] is the first work to incorporate prompt tuning into KGC tasks. By applying prefix tuning in conjunction with GNNs, it effectively completes the KGC task under low-parameter conditions. AutoKG [Zhu *et al.*, 2023] also explores the application of prompt engineering within the knowledge graph domain. A frequently overlooked aspect of prompt tuning is its capacity to learn general representations of data during the training process, a feature that our method leverages. This enables our model not only to handle specific tasks but also to extract and utilize general patterns from the data, thereby enhancing the model's generalization ability and overall performance.

3 Method

3.1 Problem Statement

Given a knowledge graph $G = \{(h, r, t) \mid h, t \in E, r \in R\}$, where E and R are the set of entities and relations of the KG, respectively. h and t are the head and tail entities, while r is the relation between them. The KGC task aims to predict the missing triples. In the entity ranking evaluation protocol, tail entity prediction $(h, r, ?)$ ranks all entities based on h and r, while head entity prediction $(?, r, t)$ does the same. In this paper, we follow the SimKGC [Wang *et al.*, 2022a] setup and add an inverse triple (t, r^{-1}, h) for each triple (h, r, t), simplifying the task to only tail entity prediction.

3.2 Network Framework Based on Contrastive Learning

The proposed SRP-KGC method is based on a dual-encoder contrastive learning architecture and consists of three main components. First, we use multi-type positive samples for contrastive learning, introducing reasoning paths during the training phase to guide the model in enhancing its ability to discriminate reasoning paths. Next, we introduce soft reasoning paths, and by aligning soft reasoning paths with reasoning paths, we guide the model to learn generalized path representations of the corresponding relations to alleviate the issue of missing reasoning paths. Finally, during the testing phase, to fully utilize the information from entities, relations, reasoning paths, and soft reasoning paths, we introduce a hierarchical ranking strategy, combining multiple sources of information to further improve the accuracy of predictions.

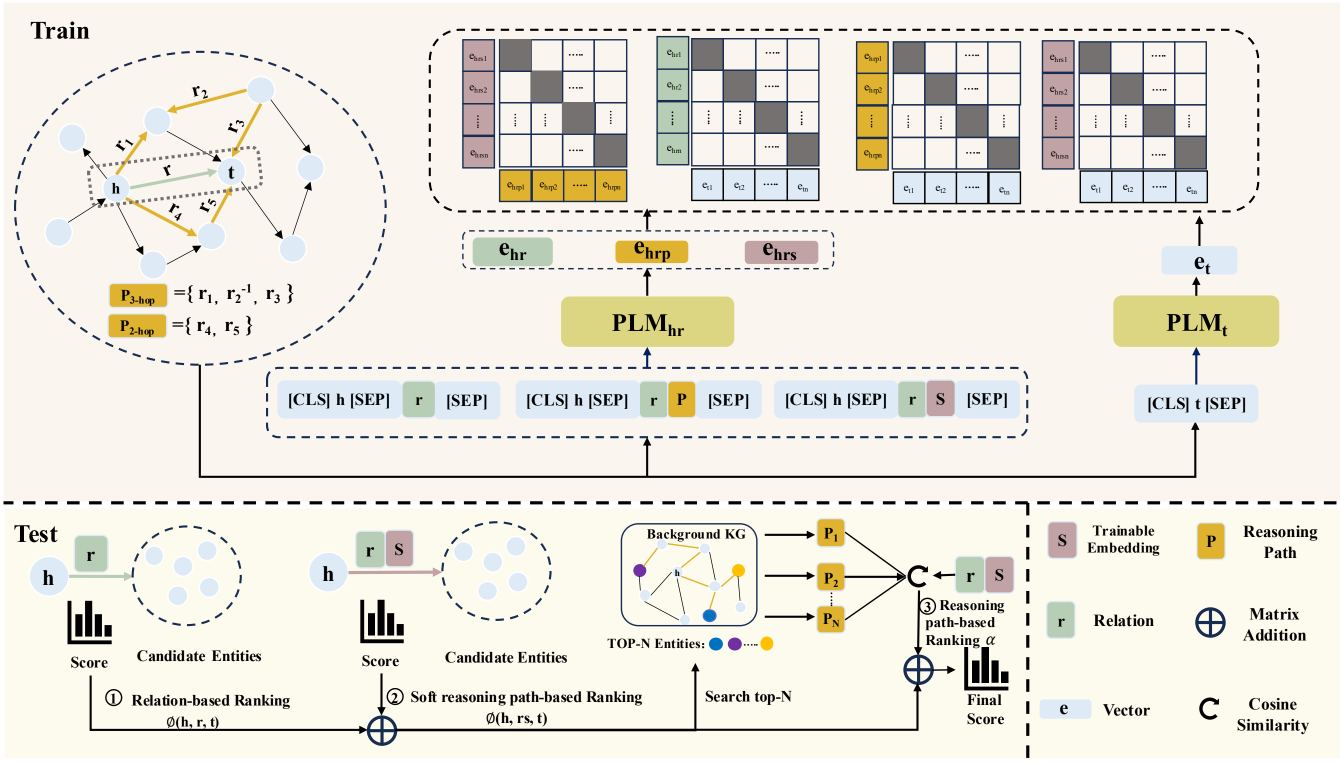

Figure 1: SRP-KGC Framework: During the training process, we introduced three types of positive samples. By incorporating these diverse positive samples, the model's ability to understand reasoning paths was enhanced, while the soft reasoning path learns the generalized representation of reasoning paths. In the testing phase, we employed a hierarchical ranking strategy, combining information from entities, relations, soft reasoning paths, and reasoning paths to further improve the model's accuracy.

3.3 Multi-Type Positive Samples

In the contrastive learning framework, we use three types of positive samples: relation positive samples, reasoning path positive samples, and soft reasoning path positive samples. Relation positive samples are triples (h, r, t) where the head and tail entities are directly related by relation r. Reasoning path positive samples replace the direct relation with a reasoning path from h to t, while soft reasoning path positive samples involve learning a generalized representation of the reasoning path through trainable embeddings, which will be explained in the next section.

Reasoning paths are the foundation of our approach. To ensure their generalization, we focus on relations and ignore entity information, represented as $p = \{r_1, r_2, \dots\}$. Paths are classified based on the number of hops n, with 2-hop and 3-hop paths being considered. We use a path constraint resource allocation algorithm from [Lin *et al.*, 2015] to compute the confidence of each path and retain the highest-scoring ones. Additionally, we add the original relations as prefixes to each path to improve their expressiveness, creating a composite representation.

$$I_r^p = [CLS]h[SEP]r[SEP] \tag{1}$$

$$I_{rp2}^p = [CLS]h[SEP]rp_2[SEP] \tag{2}$$

$$I_{rp3}^p = [CLS]h[SEP]rp_3[SEP] \tag{3}$$

In our framework, each relation or reasoning path, combined with the corresponding head entity, forms query texts (I_r^p, I_{rp2}^p, I_{rp3}^p). These query texts are paired with the correct tail entity t to generate positive samples. Then, these text pairs are processed through two BERT modules: the relation-aware module ($Bert_{hr}$) encodes the query text, generating embeddings e_{hr}, e_{hrp2}, and e_{hrp3}. The entity-specific module ($Bert_t$) independently encodes the tail entity and generates the embedding e_t.

$$\mathcal{L}_{\text{hr\_t}} = \mathcal{L}(e_{hr}, e_t), \mathcal{L}_{\text{hp\_t}} = \mathcal{L}(e_{hrp2}, e_t) + \mathcal{L}(e_{hrp3}, e_t) \tag{4}$$

Here, $\mathcal{L}$ represents the loss function, and its specific form will be introduced in detail later.

3.4 Soft Reasoning Paths

To alleviate the issue of absent reasoning paths, we introduce soft reasoning paths. Specifically, for each relation r or inverse relation r^{-1}, we append a trainable embedding $\mathbf{S}_r \in \mathbb{R}^{d_{out} \times m}$ to it. By concatenating trainable embedding with the original relation representation, we construct soft reasoning paths that are capable of generalizing path semantics. During training, we design a contrastive learning objective to align the soft reasoning path embedding e_{hrs} with the encoded authentic reasoning paths e_{hrp2} and e_{hrp3} in vector space. This alignment guides the soft reasoning paths to learn generalized path patterns of relations from a limited set of path samples. This design allows soft reasoning paths to simulate latent reasoning logic through generalized representations during testing, even when reasoning paths are absent, thus significantly mitigating the impact of missing paths on prediction performance.

Specifically, we first construct sentence pairs I_r^p. After token embedding, we append a trainable embedding $\mathbf{S_r}$ to the embedding vector of the relation. We define the soft reasoning path as I_{rs}^p.

$$I_{rs}^p = [CLS]h[SEP]rS_r[SEP], S_r = W_2 \cdot (\mathrm{ReLU}(W_1 \cdot x_r)) \tag{5}$$

where $x_r \in \mathbb{R}^{d_{in} \times m}$, m denotes the number of relations, and d_{in} represents the dimensionality of the trainable embeddings we define. $W_1 \in \mathbb{R}^{d_h \times d_{in}}$ and $W_2 \in \mathbb{R}^{d_{out} \times d_h}$ are trainable weight matrices. d_h denotes the dimensionality of the hidden layer, while d_{out} represents the dimensionality of the output layer. The output layer is typically expressed as $l \times 768$, where l is the number of trainable embeddings, and 768 is the default dimensionality of BERT input embeddings. A corresponding x_r is assigned to each relation. Specifically, if the knowledge graph contains 247 types of relations, there will be a total of 247×2 instances of x_r (accounting for both forward and inverse relations). Notably, W_1 and W_2 are shared parameters across all relations. The soft reasoning path plays a role in learning the representations of reasoning paths to better capture complex reasoning information.

$$\mathcal{L}_{\mathrm{hrs\_t}} = \mathcal{L}(e_{hrs}, e_t) \tag{6}$$

$$\mathcal{L}_{\mathrm{hrs\_p}} = \mathcal{L}(e_{hrs}, e_{hp2}) + \mathcal{L}(e_{hrs}, e_{hp3}) \tag{7}$$

Here, e_{hrs} is the result of encoding I_{rs}^p with $Bert_{hr}$.

3.5 Hierarchical Ranking

During the testing phase, we predict the tail entities using the known head entities and relations. In this process, in addition to the relational information, we can also leverage the soft reasoning paths learned during training (i.e., (h, r) and (h, rs)). By employing a dual-encoder architecture, we preprocess all candidate tail entities, enabling efficient and rapid computation. Although reasoning paths are highly valuable, performing path searches for every candidate entity would incur substantial computational costs. For instance, in the Wikidata5M dataset, each triple contains 4,594,485 candidate tail entities, making exhaustive computation impractical. To address this challenge, our approach strikes a balance between computational cost and performance through a hierarchical ranking strategy. During the reasoning phase, we first perform a quick filtering using the relations and soft reasoning paths, and then conduct reasoning path searches only for the high-confidence candidate entities.

$$\mathrm{Logits} = \phi(h, r, t) + \phi(h, rs, t), \hat{E} = \text{Top-N}(\mathrm{Logits}) \tag{8}$$

Here, we define $\phi(h, r, t) = \cos(\mathbf{e}_{hr}, \mathbf{e}_t) \in [-1, 1]$, and similarly, $\phi(h, rs, t) = \cos(\mathbf{e}_{hrs}, \mathbf{e}_t) \in [-1, 1]$. Next, we select the top N candidate entities with the highest scores for the current triple (h, r), and perform path searches in the known graph between the head entity and these candidate entities. Here, N is a tunable ranking parameter that can be adjusted flexibly based on the characteristics of the dataset.

$$Path_2, Path_3 = \mathrm{Search}(h, \hat{E}) \tag{9}$$

Here, we still limit the search to only 2-hop and 3-hop paths, i.e., $Path_2$ and $Path_3$. After combining the searched paths with the head entity and passing them through Bert_h, we obtain the embeddings $\mathbf{e}_{hp2}$ and $\mathbf{e}_{hp3}$. Then, we calculate the similarity between these two vectors and $\mathbf{e}_{hrs}$ by computing the cosine similarity, yielding a value $\alpha \in [-1, 1]$. From these results, we select the one with the highest score.

$$\alpha = \max\left(\cos(\mathbf{e}_{hp2}, \mathbf{e}_{hrs}), \cos(\mathbf{e}_{hp3}, \mathbf{e}_{hrs})\right) \tag{10}$$

We add the obtained α values to the high-confidence candidate entities in order to further optimize the results. This adjustment allows the model to prioritize the most relevant entities, improving the overall performance of the KGC task.

3.6 Loss Function

In the training process, to further enhance the generalizability of the knowledge learned by the soft reasoning paths, inspired by [Khosla *et al.*, 2020], we improve the InfoNCE loss function. We extend InfoNCE [Chen *et al.*, 2020] to handle multiple positive samples simultaneously by maximizing the likelihood of these positive samples, thus integrating shared semantic information. This modification allows the model to better capture diverse patterns in the data, improving its performance on KGC tasks.

$$\mathcal{L} = -\frac{1}{|P|} \sum_{r* \in P} \log \frac{e^{\phi(h, r*, t)/\tau}}{\sum_{i=1}^{|\mathcal{N}|} e^{\phi(h, r*, t_i)/\tau}} \tag{11}$$

Here, P is the set of all previously mentioned positive samples, that is, the relation r, 2-hop path rp_2, 3-hop path rp_3, and the soft reasoning path rs. At the same time, we retain the temperature parameter τ to balance the importance between the samples. In addition to the in-batch negative samples, we do not introduce any additional negative samples.

$$\mathcal{L}_{all} = w_1 \mathcal{L}_{\mathrm{hr\_t}} + w_2 \mathcal{L}_{\mathrm{hp\_t}} + w_3 \mathcal{L}_{\mathrm{hrs\_t}} + w_4 \mathcal{L}_{\mathrm{hrs\_p}} \tag{12}$$

Where w_i is tunable hyper-parameters for adapting to specific knowledge graph characteristics.

4 Experiments

In this section, we evaluate the overall performance of SRP-KGC and the effectiveness of its individual modules. The experiments aim to answer the following four research questions:

- RQ1. How does the proposed SRP-KGC perform compared to the state-of-the-art methods under both transductive and inductive settings? (see Section 4.2)
- RQ2. Will the introduction of soft paths improve the discriminability of the reasoning path embedding? (see Section 4.3)
- RQ3. How does the soft reasoning path perform when reasoning paths are missing or present? (see Section 4.4)
- RQ4. How does hierarchical ranking work? Is it effective? (see Section 4.5)

Methods	WN18RR				FB15k-237				Wikidata5M-Trans			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Embedding-based methods												
TransE	24.3	4.3	44.1	53.2	27.9	19.8	37.6	44.1	25.3	17.0	31.1	39.2
ComplEx	44.9	40.9	46.9	53.0	27.8	19.4	29.7	45.0	28.2	22.6	-	39.7
RotatE	47.6	42.8	49.2	57.1	33.8	24.1	37.5	53.3	29.0	23.4	32.2	39.0
ConvE	45.6	41.9	47.0	53.1	31.2	22.5	34.1	49.7	-	-	-	-
CompGCN	48.1	44.8	49.2	54.8	35.5	26.4	39.0	53.5	-	-	-	-
TuckER	47.0	44.3	48.2	52.6	35.8	26.6	39.4	54.4	-	-	-	-
CompoundE	49.2	45.2	51.0	57.0	35.0	26.2	39.0	54.7	-	-	-	-
KPACL	52.7	48.2	54.7	61.3	36.0	26.6	39.5	<u>54.8</u>	-	-	-	-
RotatE-VLP	49.8	45.5	51.4	58.2	<u>36.2</u>	27.1	39.7	54.2	-	-	-	-
Text-based methods												
KG-BERT	21.6	4.1	30.2	52.4	-	-	-	42.0	-	-	-	-
StAR	40.1	24.3	49.1	70.9	29.6	20.5	32.2	48.2	-	-	-	-
KG-S2S	57.4	53.1	59.5	66.1	33.6	25.7	37.3	49.8	-	-	-	-
C-LMKE	61.9	52.3	67.1	78.9	30.6	21.8	33.1	48.4	-	-	-	-
SimKGC	67.1	58.7	<u>73.1</u>	81.7	33.3	24.6	36.2	51.0	35.3	30.1	37.4	44.8
CSProm-KG	57.5	52.2	59.6	67.8	35.8	26.9	39.3	53.8	<u>38.0</u>	<u>34.3</u>	<u>39.9</u>	44.6
LP-BERT	48.2	34.3	56.3	75.2	31.0	22.3	33.6	49.0	-	-	-	-
GS-KGC	-	34.6	51.6	-	-	<u>28.0</u>	<u>42.6</u>	-	-	-	-	-
GHN	<u>67.8</u>	<u>59.6</u>	71.9	<u>82.1</u>	33.9	25.1	36.4	51.8	36.4	31.7	38.0	<u>45.3</u>
SRP-KGC	**70.5**	**63.6**	**74.4**	**83.1**	**43.1**	**35.3**	**46.1**	**58.5**	**40.9**	**36.6**	**43.0**	**48.8**

Table 1: Main results on WN18RR,FB15k-237 and Wikidata5M-Trans datasets. Bold numbers represent the best and underlined numbers represent the second best.

4.1 Experimental Settings

We evaluated our method on three commonly used datasets: WN18RR, FB15k-237, and Wikidata5M-Trans. Detailed information about these datasets is shown in Table 2. During the evaluation on these datasets, the candidate entities included all entities in the respective datasets. In addition, [Teru *et al.*, 2020] extracted four inductive versions (v1,v2,v3,v4) of datasets for both WN18RR and FB15k-237. When testing on these inductive datasets, we followed the conventional setup and used only 50 candidate entities that included the target tail entity for fair comparison. Due to space constraints, the detailed descriptions of the inductive datasets are provided in the appendix.

Dataset	# Ent	# Rel	# train	# valid	# test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
Wikidata5M-Trans	4,594,485	822	20,614,279	5,163	5,163

Table 2: Statistics of the datasets.

We adopted the text-based model SimKGC [Wang *et al.*, 2022a] as our baseline, retaining the BERT parameter settings from the original paper. Our implementation was built using PyTorch. Hyperparameters w_i were optimized via grid search over the set $\{0.2, 0.4, 0.6, 0.8, 1\}$. All experiments ran on 4 NVIDIA RTX 4090 24GB GPUs. Evaluation used four automated metrics:MRR: Mean reciprocal rank of test triples; Hit@k: Proportion of correct entities in top-k predictions ($k = 1, 3, 10$). The detailed hyperparameters can be found in the appendix.

4.2 Performance Comparison with SOTA Method

In this study, we conducted a comparative analysis of SRP-KGC, comparing it with both embedding-based and text-based approaches. The embedding-based methods include TransE [Bordes *et al.*, 2013], ComplEx [Trouillon *et al.*, 2016], RotatE [Sun *et al.*, 2019b], ConvE [Dettmers *et al.*, 2017], TuckER [Balazevic *et al.*, 2019], CompoundE [Ge *et al.*, 2023], KRACL [Tan *et al.*, 2022], and RotatE-VLP [Li *et al.*, 2023]. On the other hand, the text-based methods include KG-BERT [Yao *et al.*, 2021], StAR [Wang *et al.*, 2021], KG-S2S [Chen *et al.*, 2022], C-LMKE [Wang *et al.*, 2022b], SimKGC [Wang *et al.*, 2022a], CSProm-KG [Chen *et al.*, 2023], LP-BERT [Li *et al.*, 2022], GS-KGC [Yang *et al.*, 2024] and GHN [Qiao *et al.*, 2023].

The main results are summarized in Table 1. Several conclusions can be drawn from these findings. Firstly, our method outperforms previous works across all metrics on the three datasets. Specifically, on the WN18RR dataset, our SRP-KGC improves the MRR and Hits@1 metrics by 4% and 6.7%, respectively. On Wikidata5M-Trans, it improves by 7.6% and 6.7%. Notably, on FB15k-237, our SRP-KGC improves by 19% and 26.1%. These results indicate that our SRP-KGC method demonstrates strong competitiveness in knowledge graphs with both sparse and dense topologies, as well as in large-scale knowledge graphs.

To further explore the generalization capability of our method, we conducted experiments under the inductive KGC setting. The datasets used include WN18RR (v1, v2, v3, v4) and FB15k-237 (v1, v2, v3, v4), which were extracted by [Teru *et al.*, 2020]. Due to space constraints, we

Methods	WN18RR_ind				
	V1	V2	V3	V4	AVG
GraIL	82.4	78.6	58.4	73.4	73.2
SimKGC	95.8	97.2	**96.2**	97.4	96.7
GLAR	93.6	94.7	93.3	92.4	93.5
SRP-KGC	**97.8**	**98.9**	96.1	**98.4**	**97.8**

Table 3: The Hits@10 of WN18RR under inductive scenario. The optimal values of each metric are marked in blod.

compared SRP-KGC with the following three approaches: GraIL [Teru *et al.*, 2020], which is one of the most classic methods for completing KGC tasks using reasoning paths; SimKGC [Wang *et al.*, 2022a], which has a similar structure; and GLAR [Xie *et al.*, 2024], the current state-of-the-art method. For a fair comparison, we adopted the experimental setup of GraIL, retaining only 50 candidate entities containing the target tail entity by default, and used Hits@10 as the evaluation metric. Tables 3 and Tables 4 present the experimental results on these two datasets.

Methods	FB15k-237_ind				
	V1	V2	V3	V4	AVG
GraIL	64.2	81.8	95.7	89.3	82.7
SimKGC	90.5	93.2	91.0	90.1	91.2
GLAR	91.3	96.6	96.0	**96.4**	95.1
SRP-KGC	**96.3**	**98.1**	**96.2**	95.3	**96.5**

Table 4: The Hits@10 of FB15k-237 under the inductive scenario. The optimal values of each metric are marked in bold.

The experiments demonstrate that SRP-KGC improves upon the best-performing method by 1.1% and 1.5%, respectively, in these two tasks. Through this approach, the model is able to effectively capture the underlying patterns of reasoning paths, demonstrating strong generalization ability even when handling previously unseen entities.

4.3 Ability to Comprehend the Reasoning Path

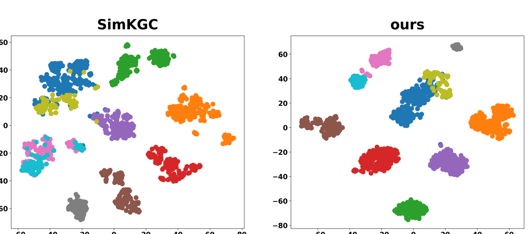

Figure 2: Visualization of embeddings with the different head entities and relations using t-SNE under the settings of SimKGC and SRP-KGC. In the visualization, points with the same color represent embeddings that share the same target tail entity.

We compared the discriminative features obtained from different head entities pointing to the same tail entity through different paths and validated the effectiveness of incorporating multiple types of positive samples into the training process to enhance the model's ability to understand reasoning paths. In the experiment, we selected 10 tail entities that are highly relevant to triples from the FB15k-237 test set. For these tail entities and their related triples, we performed path searches and combined the resulting paths with their corresponding head entities. Subsequently, we encoded these combinations using BERT models trained with SimKGC and SRP-KGC. We visualized the encoded outcomes using t-SNE in Figure 2. The visualization shows that after training with SRP-KGC, the embeddings for the same tail entity are significantly closer in the embedding space, demonstrating better feature discriminability.

4.4 Effectiveness of Soft Reasoning Paths

To validate the effectiveness of the soft reasoning path, we designed a series of comparative experiments, focusing on its performance in different scenarios. Specifically, we conducted comparisons under two conditions based on the existence of reasoning paths, and analyzed the impact of introducing the soft reasoning path:

Testing settings	MRR	Hits@1	Hits@3	Hits@10
R	26.9	18.0	29.6	44.1
RS	**33.2**	**24.4**	**36.5**	**51.0**

Table 5: For the comparison of results in the absence of reasoning paths, R represents testing using relationships, while RS represents testing using soft reasoning paths.

Without reasoning paths: In this scenario, previous models rely solely on the direct relationships within the triples for prediction, which fails to provide more effective information. Our approach introduces soft reasoning paths, and we conduct a comparative analysis. We collected 5,560 triplets from the FB15k-237 test set that do not have reasoning paths, and performed a separate analysis. As shown in Table 5, the proposed Soft reasoning paths, in the case of missing paths, showed improvements over traditional methods by 23.4%, 35.6%, 23.3%, and 15.6% on the MRR, Hits@1, Hits@3, and Hits@10 metrics, respectively. These results demonstrate that Soft reasoning paths, by learning the representation of the same relationship under different paths, effectively alleviate the issue of missing reasoning paths in KGC tasks.

With reasoning paths: In this scenario, existing path-based methods determine the target tail entity by the correlation between the reasoning path and the target relation. However, reasoning paths are often stacks of relationships, resulting in a significant semantic gap from the target relation. To alleviate this, we compared the correlation between reasoning paths and relations, as well as the correlation between reasoning paths and soft reasoning paths, to evaluate the role of soft reasoning paths in reducing the semantic gap.

Replacing the soft reasoning path with relationships and applying it to the final step of our proposed hierarchical ranking strategy is an effective comparative method. We conducted comparisons on the structurally relatively dense FB15k-237 dataset, as shown in Table 6, indicate that using Soft reasoning paths, compared to using relationships, led to improvements of 23.3%, 30.5%, 23.2%, and 15.9% in MRR, Hit@1, Hit@3, and Hit@10, respectively.

Rank settings	MRR	Hits@1	Hits@3	Hits@10
R	33.8	26.2	36.1	48.5
RS	**41.7**	**34.2**	**44.5**	**56.2**

Table 6: The comparison of results during the ranking phase using relationships and soft reasoning paths is as follows: R represents relationships, and RS represents Soft reasoning paths.

To further validate the effectiveness of soft reasoning paths, we collected 14,806 triples with reasoning paths and searched for their 2-hop and 3-hop paths. We performed relation prediction using both relations and soft reasoning paths for these paths. Specifically, the embedding vectors $\mathbf{e}_r$ of the relations involved in these triples, as well as the soft reasoning paths $\mathbf{e}_{rs}$ corresponding to each relation, were computed. Subsequently, we encoded the embeddings of these reasoning paths using the same encoder to obtain $\mathbf{e}_p$. Finally, we calculated the similarities between $\mathbf{e}_p$ and $\mathbf{e}_r$, as well as between $\mathbf{e}_p$ and $\mathbf{e}_{rs}$ and evaluated the results within their respective sets. As shown in Table 7, compared to using only relations, the use of soft reasoning paths improved the Hits@10, F1, and ROC-AUC metrics by 4.6%, 4.8%, and 2.1%, respectively. Furthermore, if soft reasoning paths were not used during training, the corresponding improvements in the metrics would be 15.6%, 94.7%, and 6.3%, respectively.

Training Settings	Testing Settings	Hits@10	F1	ROC-AUC
w/o RS	R	74.5	19.0	49.5
w RS	R	82.3	35.3	51.5
w RS	RS	**86.1**	**37.0**	**52.6**

Table 7: Relation prediction performance across training and testing configurations. R represents relationships, and RS represents soft reasoning paths.

4.5 Case Study

Contact, language film, English Language			Answer
information	Top 3 candidate entities	probabilities	Rank
(h,r)	Greek Language Japanese Language Hebrew Language	0.547 0.543 0.530	7
(h,r)+(h,rs)	Japanese Language **English Language** Greek Language	1.034 1.025 1.021	2
(h,r)+(h,rs)+(p)	**English Language** Japanese Language Greek Language	1.458 1.262 1.249	1

Table 8: The rankings and scores predicted by the model under different information conditions. The target entity is indicated in bold.

To further illustrate the effectiveness of the hierarchical ranking, we selected "Contac" as the head entity and "language film" as the relation for a prediction experiment. Specifically: Using (h, r) (head entity and relation) as the query, we calculated the similarity with all candidate entities, resulting in a rank of 7. Next, we added (h, rs) (head entity and soft reasoning path) as the query and performed similarity calculations again, improving the rank to 2. Finally, we conducted a search for the reasoning path and calculated the similarity between the reasoning paths and soft reasoning paths. Adding this score to the original score further improved the final rank to 1. This process demonstrates that integrating multiple types of information can effectively improve the accuracy of the model's predictions.

5 Limitations

Although SRP-KGC enhances KGC tasks by introducing soft reasoning paths, this leads to increased computational demands during ranking. To assess this, we examined the trade-off between performance gains and computational costs.Comparing SRP-KGC with BERTRL and SimKGC, SRP-KGC strikes the best balance between speed and accuracy. BERTRL takes 60 seconds per batch for a 3.5 MRR improvement, while SRP-KGC achieves an 8.1 MRR boost in just 15 seconds. Despite SRP-KGC requiring five times more processing time than SimKGC (which has a 0.8 MRR improvement), it offers ten times greater performance gains. The results show that SRP-KGC effectively enhances model precision through strategic computational allocation, surpassing BERTRL in efficiency and SimKGC in effectiveness. (The BERTRL results are from our reproduced experiments. We used our hierarchical ranking strategy to handle many candidate entities; without it, testing would take about 36 minutes per batch.)

Ranking Time Per Batch (512)		
Methods	Time	Ability (MRR)
SimKGC	3s	32.8→33.6
BERTRL	60s	32.0→35.5
SRP-KGC	15s	33.6→41.7

Table 9: Comparison of ranking time and performance with SimKGC and BERTRL in FB15k-237.

6 Conclusion

This paper proposes the SRP-KGC, which effectively alleviate issues such as missing reasoning paths, semantic gaps, and scalability in existing KGC tasks. By introducing learnable embeddings to construct soft reasoning paths and employing a hierarchical ranking strategy to fully leverage the available information, SRP-KGC significantly outperforms existing methods across multiple datasets, demonstrating its potential in large-scale KGC tasks. Although there is an increase in computational overhead, the substantial performance improvement indicates clear advantages of the method. Future research will focus on optimizing computational efficiency and further reducing time costs to enhance the practical applicability of the method.

Acknowledgments

This work was supported by the Huxiang Young Talents in Science and Technology Innovation Project (No. 2024RC3148).

References

[Balazevic *et al.*, 2019] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2019.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2013.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[Chen *et al.*, 2022] Chen Chen, Yufei Wang, Bing Li, and Kwok-Yan Lam. Knowledge is flat: A seq2seq generative framework for various knowledge graph completion. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*. International Committee on Computational Linguistics, 2022.

[Chen *et al.*, 2023] Chen Chen, Yufei Wang, Aixin Sun, Bing Li, and Kwok-Yan Lam. Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 2023.

[Dettmers *et al.*, 2017] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI Conference on Artificial Intelligence*, 2017.

[Dinan *et al.*, 2019] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Edge *et al.*, 2024] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130, 2024.

[Ge *et al.*, 2023] Xiou Ge, Yun Cheng Wang, Bin Wang, and C.-C. Jay Kuo. Compounding geometric operations for knowledge graph completion. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

[Huang *et al.*, 2018] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 2018.

[Iwamoto and Kameiwa, 2024] Yuki Iwamoto and Ken Kameiwa. Predicting from a different perspective in reranking model for inductive knowledge graph completion. *ArXiv*, abs/2405.16902, 2024.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *ArXiv*, abs/2004.11362, 2020.

[Li *et al.*, 2022] Da Li, Boqing Zhu, Sen Yang, Kele Xu, Ming Yi, Yukai He, and Huaimin Wang. Multi-task pretraining language model for semantic network completion. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 2022.

[Li *et al.*, 2023] Rui Li, Xu Chen, Chaozhuo Li, Yanming Shen, Jianan Zhao, Yujing Wang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, and Xing Xie. To copy rather than memorize: A vertical learning paradigm for knowledge graph completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6335–6347. Association for Computational Linguistics, 2023.

[Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *ArXiv*, abs/1506.00379, 2015.

[Qiao *et al.*, 2023] Zile Qiao, Wei Ye, Dingyao Yu, Tong Mo, Weiping Li, and Shikun Zhang. Improving knowledge graph completion with generative hard negative mining. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5866–5878. Association for Computational Linguistics, 2023.

[Sun *et al.*, 2019a] Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2019.

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

2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[Tan *et al.*, 2022] Zhaoxuan Tan, Zilong Chen, Shangbin Feng, Qingyue Zhang, Qinghua Zheng, Jundong Li, and Minnan Luo. Kracl: Contrastive learning with graph context modeling for sparse knowledge graph completion. *Proceedings of the ACM Web Conference 2023*, 2022.

[Teru *et al.*, 2020] Komal K. Teru, Etienne G. Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457. PMLR, 2020.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. *ArXiv*, abs/1606.06357, 2016.

[Vashishth *et al.*, 2019] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Pratim Talukdar. Composition-based multi-relational graph convolutional networks. *ArXiv*, abs/1911.03082, 2019.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI Conference on Artificial Intelligence*, 2014.

[Wang *et al.*, 2021] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. Structure-augmented text representation learning for efficient knowledge graph completion. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2021.

[Wang *et al.*, 2022a] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Association for Computational Linguistics, 2022.

[Wang *et al.*, 2022b] Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. Language models as knowledge embeddings. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 2022.

[Xie *et al.*, 2024] Zhiwen Xie, Yi Zhang, Guangyou Zhou, Jin Liu, Xinhui Tu, and Jimmy Xiangji Huang. One subgraph for all: Efficient reasoning on opening subgraphs for inductive knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*, 36:8914–8927, 2024.

[Yang *et al.*, 2024] Rui Yang, Jiahao Zhu, Jianping Man, Hongze Liu, Li Fang, and Yi Zhou. Gs-kgc: A generative subgraph-based framework for knowledge graph completion with large language models. *Information Fusion*, 2024.

[Yao *et al.*, 2021] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193, 2021.

[Zha *et al.*, 2021] Hanwen Zha, Zhiyu Chen, and Xifeng Yan. Inductive relation prediction by bert. *ArXiv*, abs/2103.07102, 2021.

[Zhu *et al.*, 2023] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web (WWW)*, 27:58, 2023.