Cold-Start Multi-hop Reasoning by Hierarchical Guidance and Self-verification

Mayi Xu[1], Ke Sun[1], Yongqi Li[1], and Tieyun Qian[1,2](✉)

[1] School of Computer Science, Wuhan University, Wuhan, China
{xumayi,sunke1995,liyongqi,qty}@whu.edu.cn

[2] Intellectual Computing Laboratory for Cultural Heritage, Wuhan University, Wuhan, China

Abstract. Multi-hop reasoning has attracted wide attention for knowledge graph (KG) completion since it can provide interpretable reasoning paths. Most prior multi-hop reasoning studies assume the KGs are static with fixed entities. However, in real applications, KGs are often dynamic since new entities will emerge continuously in the form of new fact triplets. In this paper, we are particularly interested in *the cold-start scenario toward dynamic KGs* to facilitate more practical multi-hop reasoning, which aims to explore the reasoning paths between emerging entities and existing entities. There are two challenging issues arising from this scenario: **i) lacking precise guidance** since available information for emerging entities is extremely limited in the cold-start scenario, **ii) lacking explicit path** since the emerging entities and existing ones are isolated. To address these issues, we propose a generation-based model, namely SelfHier, to explore the reasoning paths by hierarchical guidance and self-verification strategies. The *hierarchical guidance strategy* guides the reasoning process using hierarchical fine-grained sub-relations and coarse-grained clusters. The *self-verification strategy* constructs explicit reasoning paths by supplementing some missing fact triplets. Experimental results prove that SelfHier performs well in the cold-start scenario on dynamic KGs and also significantly outperforms existing multi-hop reasoning methods in the standard scenario on static KGs.

Keywords: Multi-hop Reasoning · Cold-start · Hierarchical Guidance · Self-verification

1 Introduction

Knowledge graphs such as Freebase [1] and NELL [2] store fact triplets in the form of (*head entity*, *relation*, *tail entity*), which benefit various knowledge-driven applications. However, existing KGs suffer from serious incompleteness in reality, which limits their practicability. Therefore, Knowledge Graph Completion

This work was supported by the grants from the National Natural Science Foundation of China (NSFC) project (No. 62276193, 41971347). It was also supported by the Joint Laboratory on Credit Science and Technology of CSCI-Wuhan University.

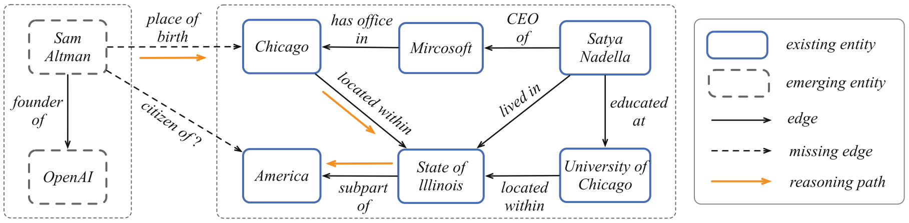

Fig. 1. An example of multi-hop reasoning in the cold-start scenario.

(KGC) has been proposed to reason the missing fact triplets, such as predicting the *tail entity* given the query *head entity* and *relation*.

For a long time in the past, embedding-based methods such as TransE [3], RotatE [4], and ConE [5] have achieved excellent performance. However, the drawbacks of these approaches are obvious since they reason in a black-box manner and can not provide interpretable reasoning paths.

To realize the interpretability, Deeppath [6] formulates the KGC as a multi-hop reasoning task. As Fig. 1 shows, given a query (*Sam Altman*, *citizen of*, ?), multi-hop reasoning methods try to predict the tail entity "*America*" with a reasoning path. Most existing multi-hop reasoning works [7–10] adopt the Reinforcement Learning (RL) framework to model the reasoning process as a Markov Decision Process, where the agents walk on KGs to search the target tail entities. Recently, SQUIRE [11] employs the generative framework to reason the paths and missing fact triplets in an end-to-end fashion, which achieves state-of-the-art (SOTA) performance.

Despite their effectiveness, existing multi-hop reasoning studies mainly focus on reasoning over static KGs with fixed entities. However, in reality, the KGs are essentially dynamic, and new entities will emerge continuously in the form of new fact triplets. For instance, the NELL KG has been extracting new fact triplets from the web since January 2010, with new entities emerging simultaneously. Completing the edges between emerging entities and existing entities is crucial to the development of KGs, but it is not easy, especially when they are isolated from each other. As Fig. 1 shows, the emerging entity "*Sam Altman*" is isolated from existing entities, without direct or indirect interaction. In such cases, reasoning a path between them becomes a cold-start scenario, as they are unseen from each other during training. In this paper, we are particularly interested in *the cold-start scenario toward dynamic KGs* to facilitate more practical multi-hop reasoning. Specifically, the cold-start scenario is set to explore interpretable reasoning paths between emerging entities and existing entities that are isolated from each other on dynamic KGs. Nevertheless, conducting multi-hop reasoning in the cold-start scenario is not trivial due to the following challenges:

- **lacking precise guidance.** Since available information for emerging entities is extremely limited in the cold-start scenario, it is difficult to construct precise guidance to handle those complex reasoning processes such as distinguishing semantically similar relations and eliminating entities with unrelated attributes.

- **lacking explicit paths.** Since emerging entities and existing entities are isolated, there is an absence of explicit paths on dynamic KGs that can connect them. Hence, most prior methods are unable to reason a correct path due to their reasoning process totally relying on existing edges.

To overcome the above challenges, we propose the generation-based model, namely SelfHier, to explore the reasoning paths by hierarchical guidance and self-verification strategies. The hierarchical guidance strategy guides the reasoning process by fine-grained sub-relations and coarse-grained clusters, which contribute to distinguishing semantically similar relations and eliminating entities with unrelated attributes, respectively. The self-verification strategy is proposed to solve the absence of explicit reasoning paths by pre-exploring some missing fact triplets to bridge the gap between emerging entities and existing entities.

In **hierarchical guidance** strategy, a relation is divided into multiple fine-grained sub-relations with different semantics, and entities with similar attributes are gathered into a coarse-grained cluster. The prediction of relations and entities will be guided by the prediction of fine-grained sub-relations and coarse-grained clusters since they are easier to distinguish and eliminate, respectively. For instance, the "*subpart of*" in (*State of Illinois*, *subpart of*, *America*) and (*youtube*, *subpart of*, *google product*) show different semantics. After it is divided into two sub-relations which describe the geographical relation and ownership relation, respectively, it will be more distinguishable from the relation "*located within*". Furthermore, if the target entity is "*America*", we should select an entity in the cluster "Country" and eliminate those entities in "Company" and "Person".

In **self-verification** strategy, inspired by the behavior that humans always verify whether their former reasoning process misses some steps by reasoning once again on the same query, we imitate it to conduct the reasoning process again on the former training query and try to find those missing fact triplets. Specifically, the generation-based framework may generate some unknown fact triplets as a step of reasoning paths since its selection space is unconstrained. We argue that if these reasoning paths containing unknown fact triplets arrive at the target tail entities with the highest probability, these unknown fact triplets may be missing in KGs originally. Hence, we further extract these unknown fact triplets to KGs to construct explicit reasoning paths.

We evaluate our model on both dynamic and static KGs. The experimental results show that our model performs well in the cold-start scenario and even outperforms existing methods in the standard scenario. Furthermore, the ablation studies show the effectiveness of both hierarchical guidance and self-verification strategies.

2 Related Work

Knowledge graph embedding methods [3–5,12,13] learn distributed representations of entities and relations from structure information, and further leverage score functions to measure the likelihood of each triplet. Despite their effectiveness, embedding-based methods can not provide interpretable information.

Multi-hop reasoning is an emerging task for KGC, which aims to find the target tail entities with interpretable reasoning paths. Rule-based methods [14,15] automatically induce logical rules from the KG and predict missing fact triplets by matching queries to the rules. Although rule-based methods such as AnyBURL [16] achieve remarkable performance, they are hard to generalize in practice due to the limitation of symbolic representation. RL-based methods model the reasoning process as a Markov Decision Process (MDP), where the agent walks on KGs to search the target entities. Deeppath [6] first adopts the RL framework to search the reasoning paths and target relations given head entities and tail entities. MINERVA [7] proposes a more difficult and practical task to find the target tail entities while given the relations and head entities. Following this work, most RL-based methods [9,10,16,17] are devoted to tackling the sparse rewards problem or trying to design a more efficient policy network. Recently, the generation-based method SQUIRE [11] introduces the generative framework to find the target entities and reasoning paths in an end-to-end fashion. By leveraging the rule-enhanced and iterative training strategy, SQUIRE achieves current state-of-the-art performance. However, these methods focus on static KGs and overlook the challenge of conducting multi-hop reasoning on dynamic KGs. We are particularly interested in the cold-start scenario toward dynamic KGs to facilitate more practical multi-hop reasoning and further design an effective generation-based model SelfHier which achieves SOTA performance in both the prior standard scenario and the cold-start scenario.

3 Methodology

Knowledge Graph. A KG is defined as a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and relation set, respectively. A KG $\mathcal{G}$ contains a set of fact triplets defined as $\mathcal{T} = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where h, r, and t represent the head entity, the relation, and the tail entity, respectively. The static KGs contain fixed entities with connectivity. The dynamic KGs contain some emerging entities, which are isolated from existing entities.

Multi-hop Reasoning. Given a query $(h, r, ?)$, multi-hop reasoning aims to predict the target tail entity t through a generated n-hop reasoning path τ : $h \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \cdots \xrightarrow{r_n} e_n$, where e_i and r_i represent the entity and the relation in the path τ. The last entity e_n in τ is treated as the predicted target tail entity. In the standard scenario, the t is not isolated from h on static KGs. In the cold-start scenario, the t is isolated from h on dynamic KGs, where they will not belong to the emerging entities or existing entities simultaneously.

3.1 Model Framework

To tackle the problem of lacking precise guidance and explicit paths in the cold-start scenario, we propose a generative model, namely SelfHier, of which the

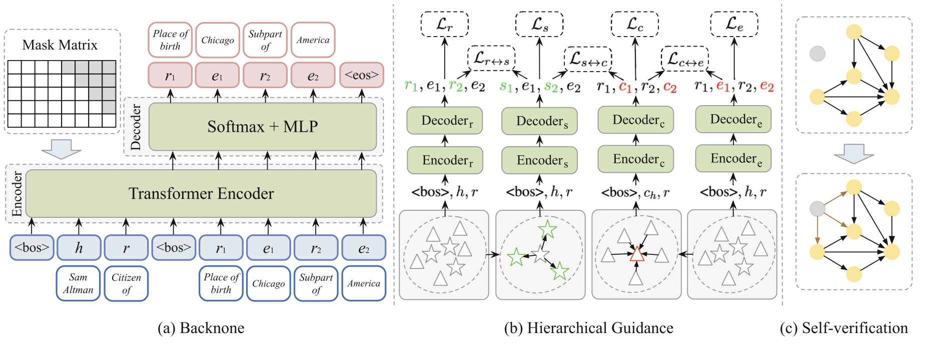

Fig. 2. SelfHier model overview. △: entity. △: cluster. ☆: relation. ☆: sub-relation. ●: existing entity. ●: emerging entity. ⟶: edges completed by self-verification. ⟶: existing edge. (Color figure online)

overall framework is shown in Fig. 2. Apparently, SelfHier mainly consists of three components: the backbone, the hierarchical guidance strategy, and the self-verification strategy. The backbone module autoregressively generates the reasoning path, which follows the pioneer generative method [11]. Based on the backbone, the hierarchical guidance strategy is developed to guide the reasoning process to distinguish semantically similar relations and eliminate entities with unrelated attributes. Furthermore, the self-verification strategy is proposed to construct explicit reasoning paths, by pre-exploring some missing fact triplets.

3.2 Backbone

The backbone is a generative model, as shown in Fig. 2 (a), which adopts the classic encoder-decoder architecture. Inputting the query $q = (\text{<bos>}, h, r)$ as the source sequence, the Transformer encoder learns contextualized hidden representation. Sequently, the MLP autoregressively decodes the reasoning paths τ token by token. During training, we maximize the cross-entropy loss as follows:

$$\mathcal{L} = -\sum_{(q,\tau)\in\mathcal{A}} \frac{1}{|\tau|} \sum_{k=1}^{|\tau|} \sum_{i=1}^{|V|} \alpha_i \log p\left(i \mid q, \tau_{<k}\right) \tag{1}$$

where $\mathcal{A}$ is the training set of all (q, τ) training pairs, $\tau_{<k}$ denotes the former k-1 tokens in τ, $|\tau|$ is the number of tokens in τ, $|V|$ is the size of vocabulary V, α_i is a label-smoothing hyperparameter to avoid overfitting, $\alpha_i = \epsilon$ for target tokens, and $\alpha_i = \frac{1-\epsilon}{|V|-1}$ for other tokens, and ϵ ranges from 0 to 1.

3.3 Hierarchical Guidance

Guidance of Fine-Grained Sub-relations. Prior methods assume that the semantics of various relations are different, while the different relations may be

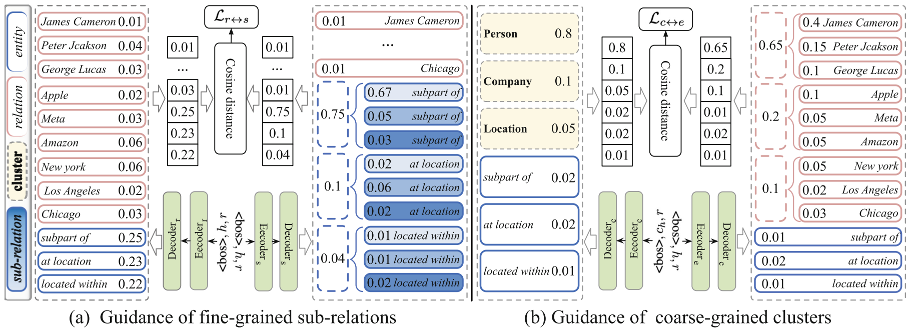

Fig. 3. Example of hierarchical guidance. (a) Distinguishing semantically similar relations by the guidance of fine-grained sub-relations. (b) Eliminating entities with unrelated attributes by the guidance of coarse-grained clusters.

semantically similar in reality. To distinguish those semantically similar relations with overlapped representation, a relation is divided into multiple fine-grained sub-relations by measuring their difference in context. To be specific, we use the TransE [3] to learn the general relation representation $\mathbf{r}$. The specific relation representation $\hat{\mathbf{r}}$ in (h, r, t) can be calculated by $\hat{\mathbf{r}} = \mathbf{t} - \mathbf{h}$ since a correct fact triplet should satisfy $\mathbf{h} + \mathbf{r} = \mathbf{t}$ in TransE. The cosine distance d between the specific relation representation $\hat{\mathbf{r}}$ and the general relation representation $\mathbf{r}$ is adopted to show the level of similarity between them. If the r in different fact triplets have close distance d, they are similar in real semantics naturally. For instance, the relation "*subpart of*" in (*Fairfax*, *subpart of*, *Virginia*) and (*Mcallen*, *subpart of*, *Texas*) has close d, where both of them emphasize the geographical relationship. Oppositely, their d is far from the d of relation "*subpart of*" in (*Micron*, *subpart of*, *Steven appleton*), which emphasizes the ownership relationship. For each relation r_i, supposing the max and the minimal d in all fact triplets containing r_i is d^{max} and d^{min}, respectively. We split the section $[d^{min}, d^{max}]$ to M equal sub-sections, and the relation r_i in specific fact triplet is converted to its sub-relation s_j when its d locate in j-th sub-section. For simplicity, the M is pre-defined as a hyperparameter, while it can be dynamically adjusted for different relations, which we leave for future research. For further reasoning among sub-relations, we convert the $(q, \tau) \in \mathcal{A}$ to $(q, \tau^s) \in \mathcal{A}^s$ as follow:

$$\tau : r_1, e_1, r_2, \cdots, e_n, \text{<eos>} \Longrightarrow \tau^s : s_1, e_1, s_2, \cdots, e_n, \text{<eos>} \tag{2}$$

where the relations r_i are converted to their corresponding sub-relations s_j.

As shown in Fig. 3 (a), the $(q, \tau) \in \mathcal{A}$ are used to train Encoder$_r$-Decoder$_r$ while the $(q, \tau^s) \in \mathcal{A}^s$ are used to train Encoder$_s$-Decoder$_s$. The particular dimensions of their output embeddings correspond to the probability of the relations and sub-relations, respectively. For the output of Decoder$_s$, we sum the probabilities of those sub-relations divided from the same relation, and further align the probabilities of corresponding relations output by Decoder$_r$ with them.

The process is depicted in Fig. 3 (a), and the cosine distance loss is maximized to align their probabilities as follows:

$$\mathcal{L}_{r\leftrightarrow s} = -\sum_{\substack{(q,\tau)\in\mathcal{A}\\(q,\tau^s)\in\mathcal{A}^s}} \sum_{k=1}^{|\tau|/2} \mathrm{d}\Big(p(\tau_{2k-1} \mid q, \tau_{<2k-1}), p(\tau^s_{2k-1} \mid q, \tau^s_{<2k-1}) \cdot \mathbf{E}_{s\rightarrow r}\Big) \quad (3)$$

where $\mathbf{E}_{s\rightarrow r}$ is a mapping matrix set to map the probability of the sub-relations to their corresponding relations. Note that we only align them in $2k$-1-th step, in which we should generate the relation. The cosine distance is calculated as follows:

$$\mathrm{d}(\mathbf{X},\mathbf{Y}) = 1 - \frac{\mathbf{X}\cdot\mathbf{Y}}{\|\mathbf{X}\|\|\mathbf{Y}\|} = 1 - \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} + \sqrt{\sum_{i=1}^{n}(y_i)^2}} \quad (4)$$

Guidance of Coarse-Grained Clusters. To eliminate entities with unrelated attributes, we formulate the coarse-grained cluster by gathering the entities with similar features together. Those entities belonging to the same cluster always share similar attributes. For instance, if the target entity is "*James Cameron*", we should select an entity in the cluster "Person" and eliminate those entities in "Company" and "Location". Specifically, we learn the embeddings of entities by TransE, then use the K-means [18] algorithm to obtain K clusters, where the K is a pre-defined hyperparameter. For reasoning among clusters, we convert the $(q,\tau)\in\mathcal{A}$ to $(q^c,\tau^c)\in\mathcal{A}^c$ as follow:

$$q : <\text{bos}>, h, r \Longrightarrow q^c : <\text{bos}>, c_h, r \quad (5)$$

$$\tau : r_1, e_1, r_2, \cdots, e_n, <\text{eos}> \Longrightarrow \tau^c : r_1, c_1, r_2, \cdots, c_n, <\text{eos}> \quad (6)$$

where the entities h and e_i are converted to their corresponding clusters c_h and c_j, respectively.

As shown in Fig. 3 (b), the $(q,\tau)\in\mathcal{A}$ are used to train Encoder$_e$-Decoder$_e$ while the $(q^c,\tau^c)\in\mathcal{A}^c$ are used to train Encoder$_c$-Decoder$_c$.

The particular dimension of their output representation corresponds to the probability of the entities and clusters. As the process when aligning the relations to their corresponding sub-relations, the cosine distance loss is maximized to align the probabilities of the entities with their corresponding clusters as follows:

$$\mathcal{L}_{c\leftrightarrow e} = -\sum_{\substack{(q,\tau)\in\mathcal{A}\\(q^c,\tau^c)\in\mathcal{A}^c}} \sum_{k=1}^{|\tau^c|/2} \mathrm{d}\Big(p(\tau^c_{2k} \mid q^c, \tau^c_{<2k}), p(\tau_{2k} \mid q, \tau_{<2k}) \cdot \mathbf{E}_{e\rightarrow c}\Big) \quad (7)$$

where $\mathbf{E}_{e\rightarrow c}$ is a mapping matrix set to map the probability of the entities to their corresponding clusters. Note that we only align them in $2k$-th step, in which we should generate the entity.

Ensemble of Coarse-Grained Clusters and Fine-Grained Sub-relations. To ensemble the hierarchical guidance during training, we also align the output of Decoder$_s$ and Decoder$_c$ at each step as follows:

$$\mathcal{L}_{s\leftrightarrow c} = -\sum_{\substack{(q^c,\tau^c)\in\mathcal{A}^c\\(q,\tau^s)\in\mathcal{A}^s}} \sum_{k=1}^{|\tau^c|} \mathrm{d}\Big(p(\tau^c_k \mid q^c, \tau^c_{<k}), p(\tau^s_k \mid q, \tau^s_{<k}) \cdot \mathbf{E}_{e\rightarrow c} \cdot \mathbf{E}_{s\rightarrow r}\Big) \quad (8)$$

Finally, the overall loss of our model is:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_e + \lambda(\mathcal{L}_s + \mathcal{L}_c) + \beta(\mathcal{L}_{r\leftrightarrow s} + \mathcal{L}_{c\leftrightarrow e} + \mathcal{L}_{s\leftrightarrow c}) \tag{9}$$

where the $\mathcal{L}_r, \mathcal{L}_e, \mathcal{L}_s, \mathcal{L}_c$ are losses of generating reasoning paths, which are formulated as Eq. 1, the $\mathcal{L}_{r\leftrightarrow s}, \mathcal{L}_{c\leftrightarrow e}, \mathcal{L}_{s\leftrightarrow c}$ are the hierarchical guidance losses, the λ and β are hyperparameters to control the loss ratio.

To ensemble the hierarchical guidance during generating, the Encoder$_r$-Decoder$_r$ and Encoder$_e$-Decoder$_e$ alternately generate the entities and relations. As shown in Fig. 4, the Encoder$_e$-Decoder$_e$ focuses on generating the next relation since it is guided by sub-relations, while the Encoder$_r$-Decoder$_r$ is devoted to generating the next entity since it is guided by clusters. In addition, the Encoder$_r$-Decoder$_r$ and Encoder$_e$-Decoder$_e$ also generate the paths alone, and the final tail entity is voted by the above three generative manners.

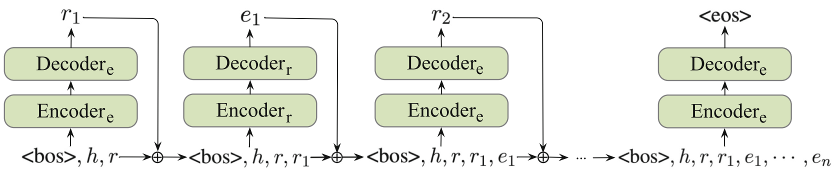

Fig. 4. Alternately decoding strategy during generating.

3.4 Self-verification

To solve the problem that there is no explicit path between emerging entities and existing ones, an intuitive solution is pre-completing some missing fact triplets to bridge this gap before conducting the multi-hop reasoning. However, it is impractical for previous KGC methods since they need to know two elements of the missing fact triplet before completing it. Inspired by the behavior that humans always verify whether their former reasoning process misses some steps by reasoning once again on the same query, we propose the self-verification strategy to reason again on the former training query to find those missing fact triplets, which does not need to know two elements of it before completing it.

Specifically, the generative framework may generate some unknown fact triplets in reasoning paths since its selection space is unconstrained. We argue that if these reasoning paths arrive at the target tail entities with the highest probability, these unknown fact triplets may be missing for KGs from the beginning, which contribute to constructing explicit reasoning paths. Specifically, we first train the model on the training set $(q, \tau) \in \mathcal{A}$. Then, inputting the same query $q \in \mathcal{A}$ to the model once again, the generated paths which arrive at the target tail entity with the highest probability are picked. We collect the fact triplets in these paths which not exist in the original KG $\mathcal{T}$ and filter those fact triplets whose frequencies are less than a pre-defined threshold T. These filtered fact triplets $\mathcal{T}^{new}$ are added to the KGs to construct more explicit paths.

4 Experiments

4.1 Experimental Setup

Experimental Knowledge Graphs. In the standard scenario, we conduct experiments on four static KGs, including FB15K237 [19], NELL995[1] [6], FB15K23720 and NELL23K [8]. The first two are considered dense KGs, while the latter two are sparse KGs. Additionally, to formulate the cold-start scenario, we carefully construct some dynamic KGs by following these rules strictly: Firstly, keeping the connectivity of the existing entities. Secondly, ensuring the disconnectedness between emerging entities and existing entities. Thirdly, the head entity and tail entity will not belong to emerging entities or existing entities simultaneously during testing. A detailed overview of the KGs is provided in Table 1. For instance, the NELL23K-40% denotes that split 40% emerging entities and retain 60% existing entities on NELL23K. Furthermore, the split ratio of NELL23K is different from FB15K23720 due to the connectivity of existing entities will be broken when setting a lower ratio in NELL23K.[2]

Table 1. Dataset statistics of different KGs.

KGs type	KGs	Existing Entities	Relations	Facts	Mean degree	Emerging Entities
Static	FB15K237	14,505	237	272,115	18.76	0
	NELL995	62,706	198	117,937	1.88	0
	FB15K23720	13,166	237	54,423	4.13	0
	NELL23K	22,925	200	25,445	1.11	0
Dynamic	FB15K23720-50%	6,583	237	38,013	3.46	4,393
	FB15K23720-20%	10,532	237	50,144	4.31	1,100
	FB15K23720-10%	11,849	237	52,938	4.29	502
	NELL23K-60%	9,170	200	20,772	1.07	10,128
	NELL23K-50%	11,462	200	22,440	1.09	9,033
	NELL23K-40%	13,755	200	24,345	1.111	8,211

Baselines. For embedding-based models, we compare against TransE [3], ConvE [13], RotatE [4], TuckER [12], and ConE [5]. As for multi-hop reasoning models, we compare against MINERVA [7], MultiHopKG [17], AnyBURL [16], DacKGR [8], RuleGuider [9], CURL [10], and current SOTA model SQUIRE [11].

Evaluation Protocol. We follow the same evaluation protocol as most multi-hop reasoning methods [7,8,11]. Specifically, we report the results in terms of the Hit@1, 3, and 10 metrics, as well as the mean reciprocal rank (MRR) score, for the link prediction task.

[1] We use the same version as [11] considering the inconsistent split in previous studies.
[2] The data and code are available at: https://github.com/NLPWM-WHU/SelfHier.

Implementation Details. We utilize the rule-based method AnyBURL [16] to construct ground-truth reasoning paths as prior generation-based method [11] do. The Adam optimizer [20] is used for training our model. For evaluation, we follow the same process as [11] which involves using beam search and self-consistency [21] to decode the reasoning paths and target entities. The models are trained with five different seeds, and the results are averaged.

Table 2. Experiment results in the standard scenario. **Bold**: the best score of multi-hop reasoning models. <u>Underlined</u>: the second-best score of multi-hop reasoning models. †: the results are retrieved from [11]. ‡: the results reported by our reproduction based on their released code and the best hyperparameters. We reproduce CURL since its performance is not reported in most KGs previously. We also reproduce SQUIRE for the significance test. The reproduction results of SQUIRE are almost the same as those in their paper. *: the improvements of our SelfHier over the best baseline is significant at $p < 0.01$.

Model	FB15K237				NELL995				FB15K23720				NELL23K			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE†	42.5	32.0	47.5	63.5	37.1	20.9	47.3	65.4	26.3	17.8	28.8	43.4	17.9	7.6	20.8	37.9
ConvE†	43.8	34.2	48.3	62.7	54.2	44.9	59.4	70.9	26.4	18.7	28.4	42.2	27.9	19.3	30.1	46.7
RotatE†	42.6	32.1	47.4	63.5	51.3	41.1	57.0	70.8	26.5	18.5	28.6	43.0	21.7	14.1	23.2	36.8
TuckER†	45.1	35.7	49.5	63.5	51.1	42.2	55.6	68.2	24.6	17.8	26.1	38.4	20.7	14.3	22.4	33.8
ConE†	44.6	34.5	49.0	64.5	54.3	44.8	60.2	71.5	27.4	19.3	29.8	43.7	23.4	15.8	24.9	40.0
MINERVA†	27.5	19.9	30.6	43.3	39.1	29.3	44.9	57.5	12.3	7.0	13.3	23.6	15.1	10.1	15.9	24.7
MultiHopKG†	40.7	32.7	44.3	56.4	46.7	38.8	51.2	60.9	23.1	16.7	25.0	36.1	17.8	12.4	18.8	29.7
AnyBURL†	-	30.0	40.5	54.4	-	38.9	52.1	62.8	-	15.9	24.0	35.9	-	14.0	20.3	29.2
RuleGuider†	38.7	29.7	42.8	56.3	41.7	34.4	47.6	58.2	9.4	4.2	9.4	21.0	11.2	3.0	14.0	27.3
DacKGR†	34.7	27.4	38.2	49.3	42.1	34.7	46.4	55.4	24.6	18.0	27.0	38.6	19.7	13.3	21.1	33.7
CURL‡	27.6	20.1	30.8	42.8	40.1	30.2	45.9	58.7	13.5	8.0	14.6	24.9	16.2	11.0	17.1	26.0
SQUIRE‡	<u>43.2</u>	<u>34.1</u>	<u>47.5</u>	<u>61.5</u>	<u>51.9</u>	<u>43.5</u>	<u>57.0</u>	<u>68.1</u>	<u>25.1</u>	<u>17.9</u>	<u>27.7</u>	<u>40.5</u>	<u>24.5</u>	<u>16.6</u>	<u>26.8</u>	<u>41.3</u>
SelfHier(ours)	**47.4***	**38.8***	**51.6***	**64.0***	**56.7***	**49.3***	**61.0***	**71.0***	**28.8***	**21.0***	**31.1***	**44.9***	**28.6***	**20.2***	**30.8***	**46.6***

4.2 Main Results

Table 2 presents the results of our SelfHier model and the baseline models in the standard scenario, while Table 3 displays the comparison results in the cold-start scenario.

Our SelfHier model achieves new state-of-the-art performance in the standard scenario, outperforming prior multi-hop reasoning models with a significant margin. This demonstrates that SelfHier is highly effective in conducting multi-hop reasoning on static KGs with fixed entities. Compared to the best baseline model, SelfHier achieves improvements of 9.7%, 9.2%, 14.7%, and 16.7% in MRR, and 13.8%, 13.3%, 17.3%, and 21.7% in Hit@1 across the four static KGs. Additionally, our model performs better than embedding-based models in most metrics for all static knowledge graphs, indicating that SelfHier can provide superior performance while retaining interpretability in the standard scenario.

In the cold-start scenario, we compare SelfHier with SQUIRE and DacKGR models, as other models are unable to handle this scenario entirely. SelfHier outperforms the baselines significantly on all dynamic KGs, regardless of the

Table 3. Experiment results in the cold-start scenario. The markers are same as those in Table 2.

Model	FB15K23720-10%				FB15K23720-20%				FB15K23720-50%			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
DacKGR	18.1	<u>15.6</u>	19.2	23.4	16.6	<u>14.2</u>	18.0	21.1	14.5	12.1	15.4	19.3
SQUIRE	<u>19.4</u>	14.5	<u>22.3</u>	<u>28.6</u>	<u>18.5</u>	<u>14.2</u>	<u>20.1</u>	<u>27.2</u>	<u>17.1</u>	<u>12.2</u>	<u>18.7</u>	<u>27.4</u>
SelfHier(ours)	**22.4***	**18.7***	**23.2***	**30.4***	**21.2***	**17.0***	**22.8***	**28.7***	**20.3***	**15.5***	**21.6***	**30.1***
Model	NELL23K-40%				NELL23K-50%				NELL23K-60%			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
DacKGR	14.0	8.8	14.9	23.5	13.2	8.2	14.8	22.5	11.2	6.0	12.4	21.0
SQUIRE	<u>15.3</u>	<u>10.8</u>	<u>16.8</u>	<u>23.7</u>	<u>14.2</u>	<u>10.2</u>	<u>14.3</u>	<u>22.7</u>	<u>13.5</u>	<u>7.9</u>	<u>14.3</u>	<u>24.3</u>
SelfHier(ours)	**19.2***	**12.4***	**22.4***	**31.4***	**18.0***	**11.7***	**19.7***	**29.5***	**17.7***	**11.6***	**19.1***	**29.3***

proportion of emerging entities. Notably, in the most stringent cold-start scenario of FB15K23720-50% and NELL23K-60% knowledge graphs, SelfHier achieves 18.7% and 31.1% improvement in MRR, and 27.0% and 46.8% improvement in Hit@1, respectively, compared to the best baseline model. These improvements are even more significant than their corresponding standard scenario.

In conclusion, our SelfHier model is a practical and widely-adaptive method that performs well not only in the cold-start scenario but also achieves state-of-the-art performance in the standard scenario.

Table 4. Ablation studies in the standard scenario. "w/o": removing the corresponding strategy. "w/o all strategies": removing hierarchical guidance and self-verification strategies simultaneously. **Bold**: the best score among different variants.

Model Variants	FB15K237				NELL995				FB15K23720				NELL23K			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
SelfHier	**47.4**	**38.8**	**51.6**	**64.0**	**56.7**	**49.3**	**61.0**	**71.0**	**28.8**	**21.0**	**31.1**	**44.9**	**28.6**	**20.2**	**30.8**	**46.6**
w/o sub-relations	46.3	37.8	50.5	62.9	55.0	47.6	59.0	69.5	27.5	19.9	29.8	43.8	26.9	19.2	29.0	43.1
w/o clusters	46.5	37.9	50.8	63.3	54.9	47.6	58.4	69.5	27.4	19.8	29.9	43.7	26.9	19.1	29.2	43.2
w/o self-verification	43.5	34.5	47.8	61.7	52.7	44.2	57.4	68.6	26.3	18.9	28.8	41.9	27.8	19.3	30.4	45.9
w/o all strategies	41.5	32.7	45.8	59.4	50.6	42.2	55.6	66.4	24.4	17.3	27.2	39.2	23.5	15.7	25.9	39.4

4.3 Ablation Studies

We conduct ablation studies on the hierarchical guidance and self-verification strategies, and the results on the standard and cold-start scenarios are shown in Table 4 and Table 5, respectively. These results indicate that all strategies are critical to improving the performance of our model in both standard and cold-start scenarios. We also observe that the score in Hit@10 slightly increases on some cold-start scenarios, which may be due to the guidance of clusters constraining the diversity of entities severely. Furthermore, incorporating both coarse-grained clusters and fine-grained sub-relations into the guidance is beneficial to the overall performance of our model. Although the performance is significantly impacted when both the self-verification and hierarchical guidance strategies are removed, the model still outperforms the best RL-based method, demonstrating the effectiveness of the generation-based framework for conducting multi-hop reasoning, and its potential for further research in the community. Overall, our ablation studies demonstrate the effectiveness of both hierarchical guidance and self-verification strategies.

Table 5. Ablation studies in the cold-start scenario. The markers are same as those in Table 4.

Model Variants	FB15K23720-10%				FB15K23720-20%				FB15K23720-50%			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
SelfHier	**22.4**	**18.7**	**23.2**	30.4	**21.2**	**17.0**	**22.8**	28.7	**20.3**	**15.5**	**21.6**	**30.1**
w/o sub-relations	21.5	17.5	22.6	30.1	20.5	16.8	21.4	28.0	19.4	14.9	20.5	28.2
w/o clusters	21.3	17.5	22.3	**30.7**	20.6	16.2	22.6	**28.8**	19.2	14.6	20.7	28.4
w/o self-verification	20.7	17.2	21.1	28.6	19.3	14.7	21.1	28.2	19.2	14.8	20.3	28.1
w/o all strategies	19.2	14.8	19.6	28.0	17.6	13.1	18.9	27.7	17.0	12.4	18.1	27.1
Model Variants	NELL23K-40%				NELL23K-50%				NELL23K-60%			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
SelfHier	**19.2**	**12.4**	**22.4**	**31.4**	**18.0**	**11.7**	**19.7**	29.5	**17.7**	**11.6**	**19.1**	**29.3**
w/o sub-relations	17.8	10.3	21.1	29.6	17.1	10.7	19.5	27.7	16.2	10.1	18.7	27.4
w/o clusters	16.7	9.5	18.6	29.9	16.4	9.8	18.2	**29.7**	16.1	9.4	18.7	28.7
w/o self-verification	18.7	11.9	21.4	31.4	17.1	10.7	19.1	29.1	16.8	10.7	18.4	28.4
w/o all strategies	14.8	8.0	16.0	28.9	13.5	8.2	14.6	23.6	13.2	7.2	14.7	24.9

4.4 Interpretability Evaluation

Following [11,22], we manually annotate the interpretability score for paths generated by MultiHopKG, SQUIRE, and SelfHier model. We randomly select 100 queries and choose the generated reasoning path with the highest probability that leads to the gold tail entity. Then, two experts score it alone based on whether it is convincing to them, where 1, 0.5, and 0 scores for paths that are reasonable, partially reasonable, and unreasonable, respectively. If there is an inconsistency of their score, another expert who is more familiar with this task makes the decision finally[3] The evaluation result on FB15K237 is reported in Table 6. We observe that SelfHier achieves higher scores on both metrics, suggesting that our model can generate more reasonable paths and facilitate explainable multi-hop reasoning.

Table 6. Interpretability evaluation. The interpretability score is the sum score of all reasoning paths, and the reasonable rate is the ratio of those paths gaining 1 score.

Model	MultiHopKG	SQUIRE	SelfHier
Interpretability score	14.0	24.5	29.5
Reasonable rate	5.0%	10.0%	13.0%

[3] The three experts are well-versed in this task, and they are blind to which model generates the path when scoring. The full annotation results are available at https://github.com/NLPWM-WHU/SelfHier/blob/main/annotation.csv.

Additionally, we conduct a study on the dependability of unknown fact triplets within the reasoning paths generated by SelfHier and SQUIRE. To achieve this, we train them on FB15K23720 KG and collect fact triplets that are not present in it but are generated during the reasoning process. We evaluate whether these fact triplets are included in the more comprehensive FB15K237 KG. The coverage statistics are presented in Table 7, demonstrating that our SelfHier model is capable of generating more reliable fact triplets compared to SQUIRE.

Table 7. Coverage statistics.

Model	Unknown Num	Covered Num	Cover Ratio
SQUIRE	11891	1236	10.4%
SelfHier	13502	3717	27.5%

4.5 Model Complexity

To investigate the model complexity, we compare the time-consuming and memory-usage between MultiHopKG, SQUIRE, and our SelfHier model. Firstly, we measure the training time across graph sizes ranging from 5×10^4 to 25×10^4 nodes. As shown in Fig. 5, more time will be consumed by our SelfHier model compared to SQUIRE, but less compared to MultiHopKG. The seq2seq architecture of our model can avoid the iterative trial-and-error process of reinforcement learning methods, resulting in reduced time consumption. Furthermore, our model is more effective in memory-usage. Overall, our model achieves a good trade-off between performance and complexity.

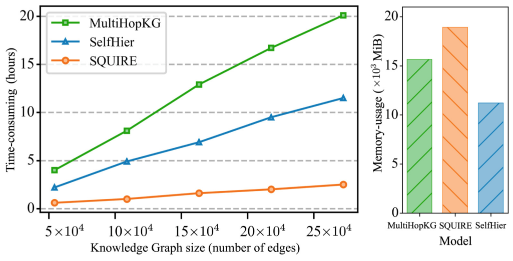

Fig. 5. Model complexity comparison.

4.6 Case Study

As Table 8 shows, we analyze two cases in the standard and cold-start scenarios, respectively. For each scenario, we choose the reasoning paths at Hit@1 predicted by the top-3 best models for comparison. Given Query1 in the standard scenario, MultiHopKG and SQURIE struggle to distinguish those semantically similar relations and eliminate entities with unrelated attributes, while SelfHier can reason a path toward the target tail entity correctly since the *hierarchichal-guidance* is equipped to solve the dilemma. Given Query2 in the cold-start scenario, the existing entity "*Brooklyn Dodgers*" is isolated from the emerging entity "*Baseball*". DacKGR and SQUIRE not only have trouble in distinguishing semantically similar relations and eliminating entities with unrelated attributes due to the lack of precise guidance, but are also troubled by the lacking of an explicit path. However, our self-verification can pre-explore the edge "*team plays against*" between "*Brooklyn Dodgers*" and "*Colorado Rochies*", which merges the gap between them. Overall, our SelfHier is more effective in both standard and cold-start scenarios.

Table 8. Case study. $\longrightarrow$: the edge completed by self-verification. $\dashrightarrow$: the edge completed by DacKGR or SQUIRE during reasoning. $\longrightarrow$: the edge existing in KGs.

Standard	Query1: ($U.K$, $contains$, ?)$\Longrightarrow$ Target1: *Borough of Chesterfield*	Prediction
MultiHopKG:	$U.K \xrightarrow{contains} England \xrightarrow{contains} University\ of\ Sheffield$	✗
SQUIRE:	$U.K \xrightarrow{at\ location^{-1}} England \xrightarrow{contains} Greater\ London \overset{contains}{\dashrightarrow} River\ Thames$	✗
SelfHier:	$U.K \xrightarrow{contains} England \xrightarrow{at\ location^{-1}} Borough\ of\ Chesterfield$	✓
Cold-start	Query2: ($Brooklyn\ Dodgers$, $team\ plays\ sport$, ?)$\Longrightarrow$ Target2: *Baseball*	Prediction
DacKGR:	$Brooklyn\ Dodgers \xrightarrow{team\ home\ stadium} Ebbets\ Field \overset{proxy\ for^{-1}}{\dashrightarrow} Chicago$	✗
SQUIRE:	$Brooklyn\ Dodgers \overset{team\ plays\ sport}{\dashrightarrow} Football$	✗
SelfHier:	$Brooklyn\ Dodgers \xrightarrow{team\ plays\ against} Colorado\ Rockies \xrightarrow{team\ plays\ sport} Baseball$	✓

5 Conclusion

In this paper, we are interested in the cold-start scenario toward dynamic KGs to facilitate more practical multi-hop reasoning. To solve the problem of lacking precise guidance and explicit paths, we design hierarchical guidance and self-verification strategies. The hierarchical guidance can distinguish semantically similar relations and eliminate entities with unrelated attributes by the guidance of fine-grained sub-relations and coarse-grained clusters. Furthermore, self-verification is able to construct more explicit reasoning paths by pre-exploring some missing fact triplets. Finally, the experiments demonstrate that our SelfHier model achieves SOTA performance in both standard and cold-start scenarios.

Ethical Statement. The data and code used in our experiments are all open-source resources for research purposes. The paper is free from copyright or other intellectual property issues.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250 (2008)
2. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E., Mitchell, T.: Toward an architecture for never-ending language learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 24, pp. 1306–1313 (2010)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)

4. Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2018)
5. Bai, Y., Ying, Z., Ren, H., Leskovec, J.: Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. In: Advances in Neural Information Processing Systems, vol. 34, pp. 12316–12327 (2021)
6. Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: a reinforcement learning method for knowledge graph reasoning. In: Empirical Methods in Natural Language Processing (2017)
7. Das, R., et al.: Go for a walk and arrive at the answer: reasoning over paths in knowledge bases using reinforcement learning. In: International Conference on Learning Representations (2018)
8. Lv, X., et al.: Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5694–5703 (2020)
9. Lei, D., Jiang, G., Gu, X., Sun, K., Mao, Y., Ren, X.: Learning collaborative agents with rule guidance for knowledge graph reasoning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8541–8547 (2020)
10. Zhang, D., Yuan, Z., Liu, H., Xiong, H., et al.: Learning to walk with dual agents for knowledge graph reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 5932–5941 (2022)
11. Bai, Y., et al.: Squire: a sequence-to-sequence framework for multi-hop knowledge graph reasoning. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 1649–1662 (2022)
12. Balažević, I., Allen, C., Hospedales, T.: 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), pp. 5185–5194 (2019)
13. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
14. Rocktäschel, T., Riedel, S.: End-to-end differentiable proving. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
15. Yang, F., Yang, Z., Cohen, W.W.: Differentiable learning of logical rules for knowledge base reasoning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
16. Meilicke, C., Chekol, M.W., Ruffinelli, D., Stuckenschmidt, H.: Anytime bottom-up rule learning for knowledge graph completion. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3137–3143 (2019)
17. Lin, X.V., Socher, R., Xiong, C.: Multi-hop knowledge graph reasoning with reward shaping. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3243–3253 (2018)
18. McQueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
19. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (2015)
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

21. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
22. Lv, X., et al.: Is multi-hop reasoning really explainable? Towards benchmarking reasoning interpretability. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8899–8911 (2021)