GMH: A General Multi-hop Reasoning Model for KG Completion

Yao Zhang[1] **Hongru Liang**[2] **Adam Jatowt**[3] **Wenqiang Lei**[4]
Xin Wei[5] **Ning Jiang**[5] **Zhenglu Yang**[1*]

[1]TKLNDST, CS, Nankai University, China, [2]Sichuan University, China,
[3]University of Innsbruck, Austria, [4]National University of Singapore, Singapore,
[5]Mashang Consumer Finance Co., Ltd., China
`yaozhang@mail.nankai.edu.cn`, `lianghongru@scu.edu.cn`,
`adam.jatowt@uibk.ac.at`, `wenqianglei@gmail.com`,
`{xin.wei02, ning.jiang02}@msxf.com`, `yangzl@nankai.edu.cn`

Abstract

Knowledge graphs are essential for numerous downstream natural language processing applications, but are typically incomplete with many facts missing. This results in research efforts on multi-hop reasoning task, which can be formulated as a search process and current models typically perform short distance reasoning. However, the long-distance reasoning is also vital with the ability to connect the superficially unrelated entities. To the best of our knowledge, there lacks a general framework that approaches multi-hop reasoning in mixed long-short distance reasoning scenarios. We argue that there are two key issues for a general multi-hop reasoning model: i) *where to go*, and ii) *when to stop*. Therefore, we propose a general model which resolves the issues with three modules: 1) the local-global knowledge module to estimate the possible paths, 2) the differentiated action dropout module to explore a diverse set of paths, and 3) the adaptive stopping search module to avoid over searching. The comprehensive results on three datasets demonstrate the superiority of our model with significant improvements against baselines in both short and long distance reasoning scenarios.

1 Introduction

Knowledge graphs (KGs) have become the preferred technology for representing, sharing and adding factual knowledge to many natural language processing applications like recommendation (Wang et al., 2019; Lei et al., 2020) and question answering (Huang et al., 2019; Zhang et al., 2018). KGs store triple facts *(head entity, relation, tail entity)* in the form of graphs, where entities are represented as nodes and relations are represented as labeled edges between entities (e.g., Figure 1 (a)). Although popular KGs already contain millions of facts, e.g., YAGO (Suchanek et al., 2007)

and Freebase (Bollacker et al., 2008), they are far from being complete considering the amount of existing facts and the scope of continuously appearing new knowledge. This has become the performance bottleneck of many KG-related applications, triggering research efforts on the multi-hop reasoning task.

The multi-hop reasoning task can be formulated as a search process, in which the search agent traverses a logical multi-hop path to find the missing tail entity of an incomplete triple in KG. As shown in Figure 1 (b), the two-hop path *Stephen Curry* $\xrightarrow{plays\ for}$ *Golden State Warriors* $\xleftarrow{plays\ for}$ *Klay Thompson* is searched to reason *Klay Thompson* as the missing entity of *(Stephen Curry, teammate, ⋆)*, where ⋆ denotes the missing tail entity. Multi-hop reasoning methods (Xiong et al., 2017; Das et al., 2018) have been proposed to model the search process as a sequential decision problem in reinforcement learning (RL) framework. (Lin et al., 2018) further optimized the reward function of RL framework based on (Das et al., 2018). However, these works have only scratched the surface of multi-hop reasoning as they focus only on short distance reasoning scenarios (e.g., the two-hop case in Figure 1 (b)).

We observe that the **long distance reasoning scenarios** are vital in the development of multi-hop reasoning and KG-related applications, because two superficially unrelated entities may be actually deeply connected over a long distance. With the significant expansion of KGs, the incompleteness of KG becomes more prominent, and long distance scenarios are rapidly increasing. As shown in Figure 1 (c), the missing entity *James Harden* in the incomplete triple *(Stephen Curry, opponent, ⋆)* is inferred by a long reasoning process, i.e., a four-hop path. Moreover, in practice, the long and short distance reasoning scenarios are mixed. The ideal multi-hop reasoning model should be competent
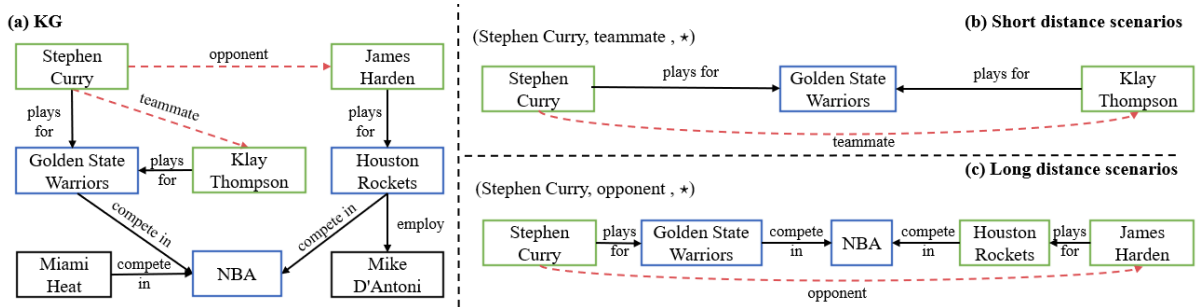
---
*Corresponding author.

Figure 1: Examples of (a) an incomplete knowledge graph, (b) a short distance scenario (two-hop) about the reasoning of *(Stephen Curry, teammate, ⋆)*, and (c) a long distance scenario (four-hop) about the reasoning of *(Stephen Curry, opponent, ⋆)*. The dotted lines refer to the relations of incomplete triples and solid lines refer to existing relations. The green, blue and black boxes represent the entities of the incomplete triples, the entities in the reasoning paths and the unrelated entities, respectively. As it can be seen, the long distance reasoning is needed and more complex than the short distance reasoning. Best viewed in color.

on mixed short and long distances. Specifically, we argue that there are two key issues in the traverse of KG that need to be resolved:

i) *Where to go?* The search agent needs to decide where to go at next search step, i.e., selecting an edge connected with the current node. Selecting the positive edge means that the agent will move towards the target node, otherwise, it will move away from the target. When the search distance increases, the issue becomes more challenging because the agent needs to make more decisions.

ii) *When to stop?* The search agent needs to decide when to stop the search because the exact search steps cannot be known in advance. An ideal search agent needs to stop at a suitable time to avoid over searching and adapt to realistic reasoning scenarios with mixed short and long distances.

To this end, we propose a **G**eneral **M**ulti-**H**op reasoning model, termed GMH, which solves the two above-listed issues in three steps: 1) the local-global knowledge fusion module fuses the local knowledge learnt from history path and the global knowledge learnt from graph structure; 2) the differentiated action dropout module forces the search agent to explore a diverse set of paths from a global perspective; and 3) the adaptive stopping search module uses a self-loop controller to avoid over searching and resource wasting. We train the policy network with RL and optimize the reward to find the target entity effectively. In summary, the main contributions of this work are as follows:

• We observe that the long distance reasoning scenarios are vital in the development of multi-hop reasoning, and argue that an ideal multi-hop rea-

soning model should be competent on mixed long-short distance reasoning scenarios.

• We propose a general multi-hop reasoning model, GMH, which can solve two key issues in mixed long-short distance reasoning scenarios: i) where to go and ii) when to stop.

• We evaluate GMH on three benchmarks, FC17, UMLS and WN18RR. The results demonstrate the superiority of GMH with significant improvements over baselines in mixed long-short distance reasoning scenarios and with competitive performances in short distance reasoning scenarios.

2 Related Work

In this section, we summarize the related work and discuss their connections to our model. Firstly, we introduce the two lines of work on the KG completion task: multi-hop reasoning and KG embedding.

The **multi-hop reasoning** task focuses on learning logical multi-hop paths reasoned from KG. The multi-hop reasoning models distill deep information from paths thereby generating further directly interpretable results. (Lao et al., 2011; Das et al., 2017; Jiang et al., 2017; Yin et al., 2018) predicted the missing relations of incomplete triples based on pre-computed paths. (Xiong et al., 2017) firstly adopted the RL framework to improve the reasoning performance. The task of finding a missing entity is orthogonal to the prediction of the missing relation in a complementary manner. (Das et al., 2018) used the history path to facilitate the search agent finding the missing entity and (Lin et al., 2018) optimized the reward function of RL framework based on (Das et al., 2018). (Lv et al., 2019) adopted the meta learning framework for multi-hop

3438

e_1 = Golden State Warriors e_2 = NBA e_3, e_3^3 = Houston Rockets e_3^1 = Mike D'Antoni e_3^2 = James Harden

r_1, r_3^2, r_4 = plays for r_2, r_3 = compete in r_3^1 = employ r_3^3 = self-loop   (e) Notation
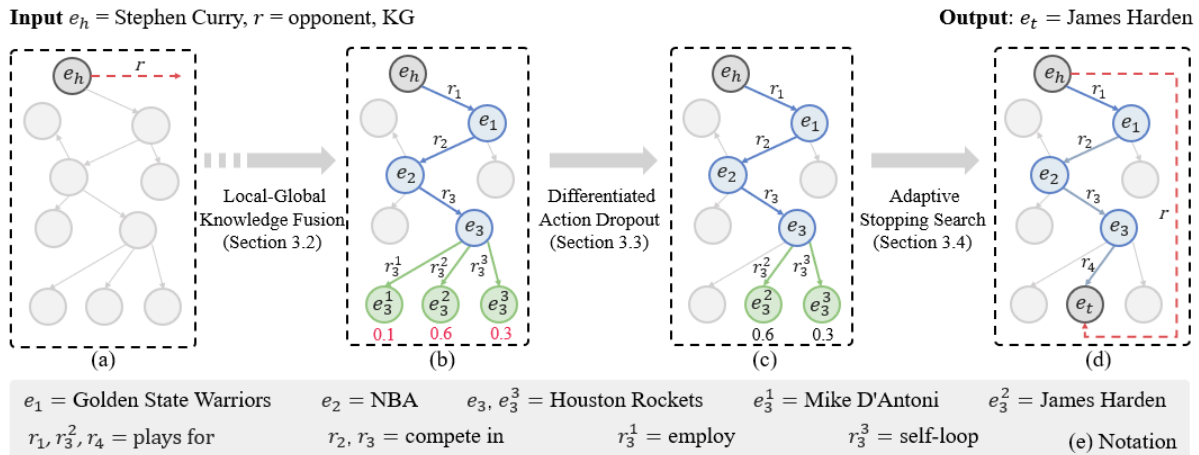
Figure 2: An illustration of the GMH Model. We reuse the example in Figure 1 (c) for explanation. The input includes the head entity and the relation of the incomplete triple *(Stephen Curry, opponent, ⋆)* with the background KG, and the output is the tail entity *James Harden*. The subgraph (a) is the initial state of the search process. The subgraphs (b-d) show the search process at step 4. Specifically, 1) we develop the local-global knowledge fusion module to estimate the possible paths, 2) the differentiated action dropout module to dilute the negative paths, and 3) the adaptive stopping search module to avoid over searching. Best viewed in color.

reasoning over few-shot relations. These works are conditioned in short distance scenarios, and tend to rapidly lose effectiveness as the distance increases. In contrast, we propose a general model which can be sufficiently utilized in both the short and long distance reasoning scenarios.

The **KG embedding** task is another line of work carried to alleviate the incompleteness of KG. Embedding-based models project KGs in the embedding space and estimate the likelihood of each triple using scoring functions. (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Ji et al., 2016) defined additive scoring functions based on the translation assumption. (Yang et al., 2015; Trouillon et al., 2016) defined multiplicative scoring functions based on linear map assumption. Moreover, recent models introduce special neural networks like neural tensor network (Socher et al., 2013), convolution neural network (Dettmers et al., 2018) and graph convolutional network (Nathani et al., 2019). Due to the neglection of deep information within multi-hop paths, the results of the embedding-based models lack interpretability, which is critical for KG-related applications. However, embedding-based models are less sensitive to the reasoning distance because they learn KG structure from the global perspective. Thus, we take advantage of this strength to learn the global knowledge from graph structure and retain the interpretability by reasoning from the history paths.

Secondly, we discuss the community research

on long distance reasoning scenarios. (Tuan et al., 2019) formed a transition matrix for reasoning over six-hop path in KG for the conversational reasoning task. It is however not suitable for large-scale KGs, because the matrix multiplication requires large calculation space. (Wang et al., 2019) proposed a long-term sequential pattern to encode long distance paths for the recommendation task. Because there is no real reasoning process for the long distance paths, it is not suitable for the KG completion. To summary, we are the first to study long distance reasoning scenarios in the KG completion. We propose a general model that tackles both short and long distance reasoning scenarios and works effectively on large-scale KGs.

3 Methodology

Figure 2 illustrates the entire process of the GMH model. The input involves the head entity and relation of the incomplete triple with the background KG. The output is the missing tail entity. We systematize the model in three steps: 1) the local-global knowledge fusion module to integrate knowledge of history paths and graph structure; 2) the differentiated action dropout module to diversify the reasoning paths; and 3) the adaptive stopping search module to formulate the optimal steps of searching. The local-global knowledge fusion and differentiated action dropout modules facilitate the agent to address the issue of *where to go*. The adaptive stopping search module controls

the search steps to resolve the issue of *when to stop*.

3.1 Preliminary

We formally represent a KG as a collection of triples $\mathcal{T} = \{(e_h, r, e_t) | e_h \in \mathcal{E}, e_t \in \mathcal{E}, r \in \mathcal{R}\}$, where e_h, r and e_t denote the head entity, relation and tail entity in one triple, $\mathcal{E}$ and $\mathcal{R}$ are the entity and relation sets, respectively. Each directed link in KG represents a valid triple (i.e., e_h and e_t are represented as the nodes and r as the labeled edge between them). For an incomplete triple, multi-hop reasoning can be perceived as searching a target tail entity e_t through limited steps in KG, starting from head entity e_h and based on the relation $r \in \mathcal{R}$. We use query q to represent (e_h, r) in the following sections. At step s, the search agent will transfer to the entity e_s updating the history path trajectory $\mathcal{H}_s = \{e_h, r_1, e_1, ..., r_s, e_s\}$, and the available action set $\mathcal{A}_s = \{(r_s^i, e_s^i) | (e_s, r_s^i, e_s^i) \in \mathcal{T}\}$. $\mathcal{A}_s$ consists of all outgoing relations and the associated entities of e_s. The agent will select one action from $\mathcal{A}_s$ to transfer to the next entity e_{s+1} through the correlated relation r_{s+1} at next step.

3.2 Local-Global Knowledge Fusion

In this module, we learn local knowledge lk_s and global knowledge gk_s to resolve the "where to go" issue, as shown in Figure 3. The local knowledge indicates that the agent makes decisions on the basis of the history path trajectory $\mathcal{H}_s$ at step s from a local perspective. The global knowledge is calculated through a pre-trained embedding-based models from a global perspective. We use an aggregate (abbr. AGG) block to aggregate lk_s and gk_s, which has two types: summation ($lk_s + gk_s$) and scalar product ($lk_s * gk_s$). The distribution $p(\mathcal{A}_s) \in \mathbb{R}^{|\mathcal{A}_s|}$ is calculated through the AGG block and represents the confidence score for each available entity in $\mathcal{A}_s$. The agent will select one action from $\mathcal{A}_s$ according to the distribution $p(\mathcal{A}_s)$ to transfer to the next entity.

Local Knowledge Learning

The local knowledge lk_s indicates from a local perspective that the agent makes decisions based on the history path trajectory $\mathcal{H}_s$ at step s. We adopt long short-term memory (LSTM) neural network and attention mechanism to encode the history path trajectory and yield the local knowledge.

The history path trajectory $\mathcal{H}_s = (e_h, r_1, e_1, ..., r_s, e_s)$ consists of the sequence of entities and relations which the agent has selected over the last s steps. We adopt an embedding layer
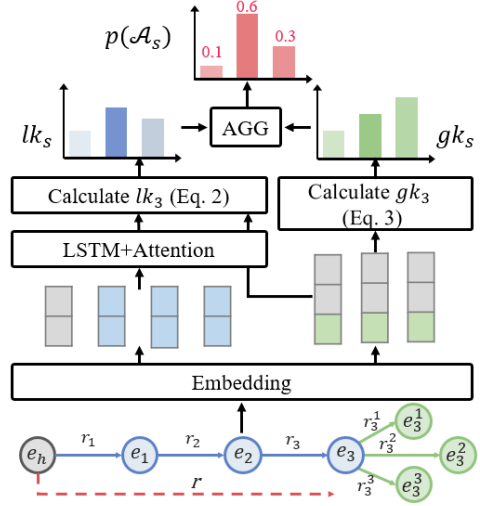

Figure 3: An illustration of the local-global knowledge fusion module. We reuse the search process in Figure 2 for detailed explanation. Best viewed in color.

to generate the embedding of entities and relations. The embedding of query is $\vec{q} = [\vec{e_h}; \vec{r}] \in \mathbb{R}^{2dim}$, i.e., the concatenation of the embeddings of the head entity $\vec{e_h} \in \mathbb{R}^{dim}$ and relation $\vec{r} \in \mathbb{R}^{dim}$, where dim is the dimension. We use an LSTM to encode the embedding of $\mathcal{H}_s$ to yield the hidden state embedding sequence $(\vec{h_0}, ..., \vec{h_s})$, where $\vec{h_s} = LSTM(\vec{h_{s-1}}, [\vec{r_s}, \vec{e_s}]) \in \mathbb{R}^{2dim}$ is the hidden state at step s, e_s is the current entity and r_s is the relation that connects e_{s-1} and e_s.

Prior works (Das et al., 2018; Lin et al., 2018) use only the current hidden state embedding (i.e., $\vec{h_s}$) to yield the next action and they neglect the differentiated importance between the hidden states over the last s steps. Therefore, the attention weight value calculated between the hidden state embedding sequence and the query embedding is introduced to optimize the local knowledge lk_s. Each weight value is derived by comparing the query $\vec{q}$ with each hidden state $\vec{h_i}$:

$$\alpha(\vec{q}, \vec{h_i}) = \frac{\exp(f(\vec{q}, \vec{h_i}))}{\sum_{j=0}^{s} \exp(f(\vec{q}, \vec{h_j}))}, \quad (1)$$

where i and j stand for the i-th and j-th hidden state candidate, respectively. Here, $f(\cdot)$ is represented as a query-based function: $f(v_q, h_m) = v_q^\top h_m$. Ultimately, local knowledge $lk_s \in \mathbb{R}^{|\mathcal{A}_s|}$, which reflects the influence of the history path trajectory on each element in $\mathcal{A}_s$, can be obtained:

$$\vec{lk_s} = \mathcal{A}_s \times \boldsymbol{W}_1 \delta_1(\boldsymbol{W}_2 \sum_{m=1}^{s} \alpha(\vec{q}, \vec{h_m})\vec{h_m}), \quad (2)$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the weights, and δ_1 is the activation function.

Global Knowledge Learning

Prior works (Das et al., 2018; Lin et al., 2018) use the local knowledge and neglect the long distance cases which requires higher decision accuracy of the agent. We introduce the global knowledge gk_s learnt from graph structure by a pre-trained embedding-based model.

Embedding-based models map the graph structure in continuous vector space by using a scoring function $\psi(e_h, r, e_t)$. We generate the new triple (e_h, r, e_s^i) by concatenating the head entity and relation with available entity $e_s^i \in \mathcal{E}_t^{\mathcal{A}}$, where $\mathcal{E}_t^{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}_s| \times dim}$ contains all available entities in $\mathcal{A}_s$. As we consider that the positive available entity is closer to the target tail entity in vector space, combining the positive available entity in $\mathcal{A}_s$ with the query will get a higher score than that using negative available entities. Formally, we adopt a pretrained embedding-based model to calculate these new triples to obtain the global knowledge gk_s:

$$\vec{gk}_s = [\psi(\vec{e}_h, \vec{r}, \vec{e}_s^{\,1}); ...; \psi(\vec{e}_h, \vec{r}, \vec{e}_s^{\,|\mathcal{A}_s|})]. \quad (3)$$

Concatenating each of new triples' scores gives the global knowledge $gk_s \in \mathbb{R}^{|\mathcal{A}_s|}$. The selection of scoring function $\psi(\cdot)$ is discussed in Section 4.3.

3.3 Differentiated Action Dropout

In the multi-hop reasoning task, it is important to enforce effective exploration of a diverse set of paths and dilute the impact of negative paths. (Lin et al., 2018) forced the agent to explore a diverse set of paths using action dropout technique which randomly masks some available actions in $\mathcal{A}_s$, i.e., blocking some outgoing edges of the agent. However, in the case of reasoning over long distances, the number of paths is much greater than that in the short distance scenarios because the search space grows exponentially. The random action dropout technique is inefficient because it cannot discriminate paths of different qualities. We then propose the differentiated action dropout (DAD) technique based on the global knowledge gk_s to mask available actions, since we believe that higher-scoring actions are more likely to exist in a high-quality path. In particular, the mask matrix $M_t \in \mathbb{R}^{|\mathcal{A}_s|}$ is sampled from the Bernoulli distribution:

$$\vec{M}_t \sim Bernoulli(sigmoid(\vec{gk}_s)). \quad (4)$$

The element in M_t is binary, where 1 indicates the action is reserved and 0 indicates abandonment. The fusion of local-global knowledge and differentiated action dropout modules helps the agent to tackle the key problem *where to go* jointly.

Algorithm 1: Training process of GMH

Input: The training samples set $\mathcal{D}_{train}$; background KG $(\mathcal{T}, \mathcal{E}, \mathcal{R})$; the maximum search step S; the maximum loop number N; the randomly initialized parameters θ

Output: The optimized parameters θ

1 **repeat**
2   Sample a triple (e_h, r, e_t) from $\mathcal{D}_{train}$;
3   **Initialize:** $s = 0; n = 0; e_s = e_h; \mathcal{H}_s = \{e_h\}$; $\mathcal{A}_s = \{(r_s^i, e_s^i)|(e_h, r_s^i, e_s^i) \in \mathcal{T}\}$;
4   **for** $s < S$ *and* $n < N$ **do**
5   Calculate lk_s (Eq. 2) and gk_s (Eq. 3);
6   Fuse lk_s and gk_s to yield the final score;
7   Dropout actions from $\mathcal{A}_s$ (Eq. 4);
8   Select the next entity e_{s+1} and the related relation r_{s+1};
9   **if** $r_{s+1} =$*self-loop* **then**
10            $n \leftarrow n + 1$;
11   **update** $His_{s+1} \leftarrow \mathcal{H}_s \cup \{r_{s+1}, e_{s+1}\}$; $\mathcal{A}_{s+1} \leftarrow \{(r_{s+1}^i, e_{s+1}^i)|(e_{s+1}, r_{s+1}^i, e_{s+1}^i) \in \mathcal{T}\}$;
12          $s \leftarrow s + 1$;
13      $\hat{e}_t = e_s$; calculate the reward $R(\hat{e}_t|e_h, r, e_t)$ and update θ (Eq. 6);
14 **until** *model converged*;

3.4 Adaptive Stopping Search

For the second key issue of *when to stop*, we devise the adaptive stopping search (ASS) module inspired by the early stopping strategy (Prechelt, 1997) which is used to avoid overfitting when training a learner with an iterative method. We add a self-loop action $(self\text{-}loop, e_s)$ to give the agent an option of not expanding from e_s. When the agent chooses the self-loop action for several times, we consider it means that the agent has found the target tail entity, thus it can choose to end early.

In this module, we devise a self-loop controller to avoid over searching and resource wasting. The self-loop controller has a dual judgment mechanism based on the the maximum search step S and the maximum loop number N. When the search step reaches the maximum S, or the agent selects the self-loop action for N consecutive times, the search process will be stopped. Using the ASS strategy improves our model's scalability on both short and long distances and effectively avoids wasting of resources caused by over-searching.

3.5 Training

Following (Das et al., 2018), we frame the search process as a Markov Decision Process (MDP) on the KG and adopt the on-policy RL method to train the agent. We design a randomized history-dependent policy network $\pi =$

$(p(\mathcal{A}_1), ..., p(\mathcal{A}_s), ..., p(\mathcal{A}_S))$. The policy network is trained by maximizing the expected reward over all training samples $\mathcal{D}_{train}$:

$$J(\theta) = \mathbb{E}_{(e_h, r, e_t) \sim \mathcal{D}_{train}}$$
$$[\mathbb{E}_{A_1, ..., A_S \sim \pi}[R(\hat{e}_t | e_h, r, e_t)]]. \quad (5)$$

where θ denotes the set of parameters in GMH, $R(\cdot)$ is the reward function and $\hat{e}_t$ is the final entity chosen by the agent. If $\hat{e}_t = e_t$, then the terminal reward is assigned +1 and 0 otherwise.

The optimization is conducted using the REINFORCE algorithm (Williams, 1992) which iterates through all (e_h, r, e_t) triples in $\mathcal{D}_{train}$ and updates θ with the following stochastic gradient:

$$\nabla_\theta J(\theta) \approx \nabla_\theta \sum_s R(\hat{e}_t | e_h, r, e_t) \log \pi_\theta. \quad (6)$$

The training process is detailed in Algorithm 1. During a search process, for each search step, the agent takes three operations: local-global knowledge fusion (lines 5-6), differentiated action dropout (line 7) and adaptive stopping search (lines 8-10). After finding the tail entity, the reward is calculated and the parameters are updated (line 13). Finally, the optimized parameters are output.

4 Experiment

4.1 Setup

Dataset Existing popular benchmarks, such as UMLS (Stanley and Pedro, 2007) and WN18RR (Dettmers et al., 2018), focus on the multi-hop reasoning in short[1] distance scenarios. Thus, they are unsuitable for evaluating complex cases requiring both long and short distance learning. To this end, we adopt the large-scale dataset FC17 (Neelakantan et al., 2015) which contains triples based on Freebase (Bollacker et al., 2008) enriched with the information fetched from ClueWeb (Orr et al., 2013). Because the data with distance type larger than five is relatively small, we maintain the data with distance type between 2 and 5. The sample number of each distance type (2-5) is 63k, 53k, 11k, 5k, respectively. Note that, there are extra relations served in the background KG plus 46 relation types in the train/valid/test sets of FC17. We also evaluate our model on the other short distance datasets, i.e., UMLS and WN18RR. Table 1 summarizes the basic statistics of datasets.
Baselines We compare GMH with 1) the *embedding-based models* involving TransE (Bor-

Table 1: Statistics of datasets w.r.t. the number of entities and edges (the middle two columns) and the separation of the train/valid/test sets (the right three columns).

	entity	relation	train	valid	test
FC17	49k	6k	125k	4k	5k
UMLS	135	46	5k	652	661
WN18RR	41k	11	87k	3k	3k

des et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and ConvE (Dettmers et al., 2018); as well as 2) the *multi-hop reasoning models* involving MINERVA (Das et al., 2018) and MultiHop (Lin et al., 2018).
Implementation Details GMH is implemented on PyTorch and runs on a single TITAN XP. Following (Das et al., 2018), we augment KG with the reversed link (e_t, r^{-1}, e_h) for each triple. We exclude the triples from the training set if they occur in the validation or testing sets. For the baselines and GMH, we set the maximum search step S to five because the entity pair's distance is up to five in FC17. For the short distance datasets, UMLS and WN18RR, S is set to three. The maximum loop number N for all datasets is set to two. We employ softmax function as the activation function. All hyper-parameters are tuned on the validation set can be found in supplementary materials The pre-trained embedding-based model that we adopt is ConvE. We optimize all models with Adam (Kingma and Ba, 2015)[2].
Metrics We follow the evaluation protocol of (Lin et al., 2018) that records the rank of the available entities at final step in a decreasing order of confidence score for each query, and adopts mean reciprocal rank (MRR) and HITS@N to evaluate the results. All results given in our experiments are the mean and standard deviation values of three training repetitions.

4.2 Multi-hop Reasoning

Table 2 shows the results obtained on FC17 and two short distance datasets, UMLS and WN18RR based on MRR (%) and HITS@N (%) measures. On the FC17 dataset, GMH achieves 23.75% MRR score surpassing the second-best model MultiHop with 3.47% improvement based on the MRR metric. This includes 3.43% improvement on short distance samples and 4.45% improvement on long distance

---

[1]Considering that the current multi-hop reasoning works are concentrated on two or three-hop paths, we regard the path hop number greater than three as a long distance scenario.

[2]We will release the processed dataset and source code, after the paper is published. The description of datasets and other details can be found in supplementary materials.

Table 2: MRR (%) and HITS@N (%) scores (± standard deviation) for multi-hop reasoning task on FC17, UMLS and WN18RR (pairwise t-test at 5% significance level). Higher values mean better performances and the best solution is marked in bold for each case.

	FC17				UMLS			WN18RR		
	MRR	MRR(≤ 3)	MRR(≥ 4)	HITS@N @1	MRR	HITS@N @1	@10	MRR	HITS@N @1	@10
TransE	11.91 ±0.20	12.38 ±0.16	9.37 ±0.28	6.90 ±0.13	86.3	85.9	88.2	40.2	39.9	43.2
Distmult	12.99 ±0.67	13.57 ±0.42	10.63 ±0.43	8.53 ±0.71	86.8	82.1	96.7	46.2	43.1	52.4
ComplEx	14.73 ±0.32	15.53 ±0.37	10.91 ±0.54	9.68 ±0.53	93.4	89.0	99.2	43.7	41.8	48.0
ConvE	18.98 ±0.63	19.75 ±0.63	15.31 ±0.68	10.43 ±0.85	95.7	93.2	99.4	44.9	40.3	54.0
MINERVA	18.70 ±0.42	19.92±0.36	13.84±0.45	9.08 ±0.94	82.5	72.8	96.8	46.3	41.3	51.3
MultiHop	20.28 ±0.71	21.63 ±0.65	14.07±0.76	10.53 ±0.84	94.0	90.2	99.2	**47.2**	43.7	54.2
GMH	**23.75**±0.52	**25.06**±0.52	**18.52**±0.56	**12.98**±0.76	**96.2**	**93.9**	**99.9**	46.5	**45.3**	**55.8**

Table 3: MRR (%) scores (± standard deviation) for long distance multi-hop reasoning task (pairwise t-test at 5% significance level). The testing samples are divided into four types according to the distance.

	Distance Type			
	4	5	6	7
ConvE	16.43±0.59	10.45±0.53	13.64±0.63	9.38±0.97
MINERVA	17.60±0.73	10.90±0.61	10.65±0.88	5.09±0.87
MultiHop	17.61±0.62	12.58±0.89	12.99±0.85	5.68±0.95
GMH	20.53±0.56	14.62±0.85	14.12±0.65	9.74±0.83

samples. We observe that multi-hop reasoning models outperform most embedding-based models, but their performance declines when the distance increases. We assume this may be attributed to the significantly increasing difficulty of building long paths when predicting long distance relations. The embedding-based models appear to be less sensitive to the distance variations, but they neglect the deep information existing in multi-hop paths, which limits the interpretative ability of predicting results. We further evaluate the short-distance reasoning performance on UMLS and WN18RR. The results of the baselines are cited from (Lin et al., 2018). GMH performs comparably well in reasoning in the short distance scenarios, yet its effectiveness in the long-short compound reasoning or long distance reasoning scenarios is more obvious. For the WN18RR dataset, GMH performs weaker than MultiHop. We speculate that this is because the number of relations in WN18RR is extremely smaller than the number of entities, which will make it difficult to accurately learn the relation embeddings. Choosing a superior pre-trained embedding-based model is critical for our model.

Multi-Hop Reasoning in long distance scenarios As we noticed in Table 2, GMH achieves new state-of-the-art results on FC17 dataset which contains both short distance and long distance types. We further evaluate its performance in terms of reasoning on the relations in longer distances, which have been rarely examined by the existing works. Therefore, we extract the relations from FC17 whose distances span from 4 to 7 and in this way we construct a subdataset, called FC17-8, which contains eight query relation types . Table 3 displays the results of reasoning on the four distance types based on the MRR metric. Compared with GMH and the multi-hop reasoning models, the embedding-based model seems less sensitive to the distance variations, while its reasoning performance is inferior to the compared models on all distance types. GMH consistently yields the best performance on the long distance reasoning scenarios. We observe that all the models perform better on the even distance type (4 and 6) than odd distance type (5 and 7). There are two possible reasons: 1) there is an imbalance between the difficulty and the number of different distance types; 2) the models are better at reasoning on symmetric paths like the four-hop path *Stephen Curry* $\xrightarrow{plays\ for}$ *Golden State Warriors* $\xrightarrow{compete\ in}$ *NBA* $\xleftarrow{compete\ in}$ *Houston Rockets* $\xleftarrow{plays\ for}$ *James Harden.*

In addition to the superior reasoning capability of GMH as demonstrated in Table 2 and Table 3, other promising potentials pave the way for GMH in advanced applications. First, GMH is explainable because it considers the path information, which is beyond the scope of the existing embedding-based models. Second, the global

Table 4: Analysis of GMH on FC17: GKL, LKL, DAD, and ASS represent the global knowledge learning, local knowledge learning, differentiated action dropout, and adaptive stopping search respectively.

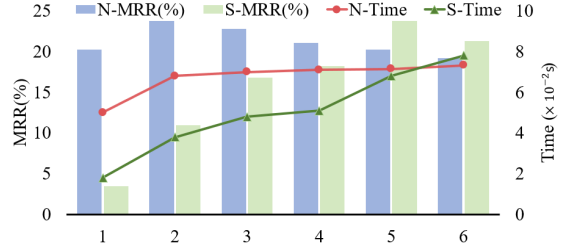|	MRR	HITS@N	
		@1	@10
GKL	13.28	6.47	13.39
LKL	18.33	9.86	25.17
LKL+GKL	22.38	11.65	28.05
LKL+GKL+DAD	23.25	12.10	28.29
GMH (LKL+GKL+DAD+ASS)	23.75	12.98	29.86

Figure 4: MRR (%) scores and running time of GMH over different (a) maximum search step S and (b) maximum loop number N on FC17. Best viewed in color.

knowledge learnt from graph structure, which has been overlooked by the existing multi-hop reasoning models, is incorporated in GMH.

4.3 Analysis of GMH

In this section, we conducted an extensive analysis of GMH from two aspects: 1) modules (c.f., Table 4); 2) hyper-parameters (c.f., Figure 4); and 3) scoring functions and aggregators (c.f., Figure 5).

Local Knowledge vs. Global Knowledge We fuse two components (i.e., the local knowledge lk_s and the global knowledge gk_s) to enable the search agent to find the target tail entity. Thus, an extensive experiment is conducted to test the contributions of lk_s and gk_s in the multi-hop reasoning task. The top three lines of Table 4 reveal that fusing lk_s and gk_s achieves the best results under different evaluation metrics. Removing either knowledge yields a significant performance drop. Concretely, removing the local knowledge causes a 9.10% MRR degradation, and removing the global knowledge results in a 4.05% MRR degradation. This suggests that the local knowledge may be more beneficial for the search agent than the global knowledge, and using only the local knowledge to find a path in KG may be ineffective in the training process. Still we argue that the importance of the global knowledge should not be neglected, especially when it is combined with the local knowledge to handle the "where to go" issue.

Performance w.r.t. Differentiated Action Dropout The differentiated action dropout module is adopted to increase the diversity of search paths in the training stage. The fourth line of Table 4 shows the validity of this module. We also test the effect of randomly action dropout (22.15% under MRR), and there is a gap with our model. This illustrates that the reason why the differentiated action dropout performs well is because the mask operation is based on the global knowledge rather

than on random strategy.

Performance w.r.t. Adaptive Stopping Search As mentioned before, we have devised the adaptive stopping search module to avoid wasting of resources caused by over-searching, i.e., the "when to stop" issue. As can be seen from the bottom two rows of Table 4, ASS also has a slight effect on the performance. This is because the module can partially prevent the search agent from continuing to search when the target tail entity has been found.

Maximum Search Step As shown in Figure 4, GMH achieves best performance at $S = 5$. Using a large S will cause wasting of resources, while if using a small S, it will affect the performance on the long distance reasoning samples. Meanwhile, the running time rises sharply when increasing S. Therefore, the introduction of adaptive stopping search module is necessary and rational.

Maximum Loop Number We divide the self-loop action into two types: positive and negative. The positive self-loop action means the agent arrives at the target tail entity, while the negative self-loop action means the current entity is not the target. See Figure 4, a small N may cause the agent to misrecognize negative actions as positive actions, while a large N may lead to lose the advantage of reducing time consumption. Compared with not using the adaptive stopping search module (i.e., $N = 1$), using it has resulted in a significant improvement with the optimal number of 2.

Scoring Function Types The pre-trained embedding-based model that we adopt is ConvE. For more extensive ablation analysis, we have conducted the experiments by incorporating effective embedding-based models (i.e., TransE, DistMult, ComplEx, and ConvE). As shown in Figure 5(a), ConvE has a superb ability to learn the global semantic representation than other embedding-based models.

Aggregator Types We next investigate the perfor-

3444

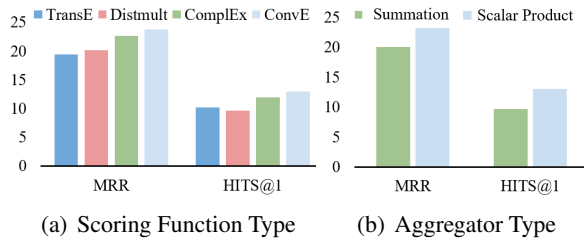(a) Scoring Function Type   (b) Aggregator Type

Figure 5: MRR (%) and HITS@N (%) scores comparison of GMH over different scoring functions and aggregators on FC17. Best viewed in color.

mance of our model w.r.t different aggregator types. We adopt two types of aggregators: summation and scalar product, to fuse the local knowledge lk_s and global knowledge gk_s. We can see from Figure 5(b) that the scalar product outperforms the summation. The advantage of the scalar product aggregator is that the multiplication operation can increase the discrimination between available actions.

5 Conclusions

We have studied the multi-hop reasoning task in long distance scenarios and proposed a general model which could tackle both short and long distance reasoning scenarios. Extensive experiments showed the effectiveness of our model on three benchmarks. We will further consider the feasibility of applying our model to complex real-world datasets with more long distance reasoning scenarios and more relation types. Besides, we have noticed that there are other "interference" in long distance reasoning. For example, noise from the KG itself, i.e., the fact that it lacks validity. These noises can gradually accumulate during long distance reasoning and affect the result confidence. We leave the further investigation to future work.

6 Acknowledgements

References

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th Conference on Neural Information Processing Systems (NeurIPS'13)*, pages 2787–2795.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*, pages 132–141.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 1811–1818.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM'19)*, pages 105–113.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 985–991.

Xiaotian Jiang, Quan Wang, Baoyuan Qi, Yongqin Qiu, Peng Li, and Bin Wang. 2017. Attentive path combination for knowledge graph completion. In *Proceedings of The 9th Asian Conference on Machine Learning (ACML'17)*, pages 590–605.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, pages 529–539.

Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2020. Conversational recommendation: Formulation, methods, and evaluation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'20)*, pages 2425–2428.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, pages 3243–3253.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 2181–2187.

Xin Lv, Yuxian Gu, Xu Han, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2019. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. 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'19)*, pages 3376–3381.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, pages 4710–4723.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL'15)*, pages 156–166.

Dave Orr, Amarnag Subramanya, Evgeniy Gabrilovich, and Michael Ringgaard. 2013. 11 billion clues in 800 million documents: A web research corpus annotated with freebase concepts.

Lutz Prechelt. 1997. Early stopping - but when? *Neural Networks: Tricks of the Trade*, 1524(2):55–69.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems ((NeurIPS'13))*, page 926–934.

Kok Stanley and Domingos Pedro. 2007. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, pages 433–440.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*, pages 697–706.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning (ICML'16)*, pages 2071–2080.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. DyKgChat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. 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'19)*, pages 1855–1865.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*, pages 5329–5336.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, pages 1112–1119.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, pages 564–573.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*.

Wenpeng Yin, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2018. Recurrent one-hop predictions for reasoning over knowledge graphs. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING'18)*, pages 2369–2378.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, pages 6069–6076.