

# Search from History and Reason for Future: Two-stage Reasoning on Temporal Knowledge Graphs

Zixuan Li<sup>1,2</sup>, Xiaolong Jin<sup>1,2</sup>, Saiping Guan<sup>1,2</sup>, Wei Li<sup>3</sup>, Jiafeng Guo<sup>1,2</sup>,  
Yuanzhuo Wang<sup>1,2</sup> and Xueqi Cheng<sup>1,2</sup>

<sup>1</sup>School of Computer Science and Technology, University of Chinese Academy of Sciences;

<sup>2</sup>CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences;

<sup>3</sup>Baidu Inc.

{lizixuan, jinxiaolong, guansaiping}@ict.ac.cn  
liwei85@baidu.com

## Abstract

Temporal Knowledge Graphs (TKGs) have been developed and used in many different areas. Reasoning on TKGs that predicts potential facts (events) in the future brings great challenges to existing models. When facing a prediction task, human beings usually search useful historical information (i.e., clues) in their memories and then reason for future meticulously. Inspired by this mechanism, we propose CluSTeR to predict future facts in a two-stage manner, Clue Searching and Temporal Reasoning, accordingly. Specifically, at the clue searching stage, CluSTeR learns a beam search policy via reinforcement learning (RL) to induce multiple clues from historical facts. At the temporal reasoning stage, it adopts a graph convolution network based sequence method to deduce answers from clues. Experiments on four datasets demonstrate the substantial advantages of CluSTeR compared with the state-of-the-art methods. Moreover, the clues found by CluSTeR further provide interpretability for the results.

## 1 Introduction

Temporal Knowledge Graphs (TKGs) (Boschee et al., 2015; Gottschalk and Demidova, 2018, 2019; Zhao, 2020) have emerged as a very active research area over the last few years. Each fact in TKGs has a timestamp indicating its time of occurrence. For example, the fact, (*COVID-19, New medical case occur, Shop, 2020-10-2*), indicates that a new medical case of COVID-19 occurred in a shop on 2020-10-2. In this paper, reasoning on TKGs aims to predict future facts (events) for timestamp  $t > t_T$ , where  $t_T$  is assumed to be the current timestamp (Jin et al., 2020). An example of the task is shown in Figure 1, which attempts to answer the query (*COVID-19, New medical case occur, ?, 2020-12-23*) with the given historical facts. Obviously, such a task may benefit many practical

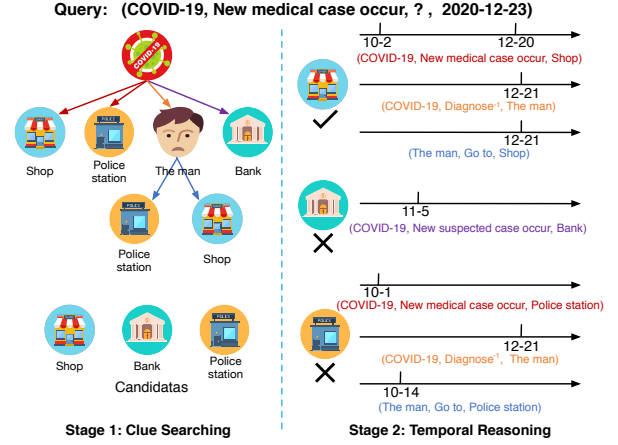


Figure 1: An illustration of the reasoning process inspired by human cognition. Different colors indicate different relations.  $r^{-1}$  is the inverse relation of  $r$ .

applications, such as, emerging events response (Muthiah et al., 2015; Phillips et al., 2017; Korkmaz et al., 2015), disaster relief (Signorini et al., 2011), and financial analysis (Bollen et al., 2011).

How do human beings predict future events? According to the dual process theory (Evans, 1984, 2003, 2008; Sloman, 1996), the first thing is to search the massive-capacity memories and find some related historical information (i.e., clues) intuitively. As shown in the left part of Figure 1, there are mainly three categories of clues vital to the query: 1) the 1-hop paths with the same relation to the query (thus called repetitive 1-hop paths), such as (*COVID-19, New medical case occur, Shop*); 2) the 1-hop paths with relations different from the query (called non-repetitive 1-hop paths), such as (*COVID-19, New suspected case occur, Bank*); and 3) the 2-hop paths, such as (*COVID-19, Diagnose<sup>-1</sup>, The man, Go to, Police station*). Human beings recall these clues from their memories and have some intuitive candidate answers for the query. Secondly, human beings get the accurate answer by diving deeper into the clues'

temporal information and performing a meticulous reasoning process. As shown in the right part of Figure 1, the man went to the police station more than two months earlier than the time when he was diagnosed with *COVID-19*, indicating that *Police station* is probably not the answer. Finally, human beings derive the answer, *Shop*.

Existing models mainly focus on the above second process but underestimate the first process. Some recent studies (Trivedi et al., 2017, 2018) learn the evolving embeddings of entities with all historical facts considered. However, only a few historical facts are useful for a specific prediction. Thus, some other studies (Jin et al., 2020, 2019; Zhu et al., 2020) mainly focus on encoding the 1-hop repetitive paths (repetitive facts) in the history. However, besides the 1-hop repetitive paths, there are massive other related information in the datasets. Taking the widely used dataset ICEWS18 (Jin et al., 2020) as an example, 41.2% of the training queries can get the answers through the 1-hop repetitive paths in the history. But, almost 64.6% of them can get the answers through 1-hop repetitive and non-repetitive paths, and 86.2% through the 1-hop and 2-hop paths.

Thus, we propose a new model called **CluSTeR**, consisting of two stages, **Clue Searching** (Stage 1) and **Temporal Reasoning** (Stage 2). At Stage 1, CluSTeR formalizes clue-searching as a Markov Decision Process (MDP) (Sutton and Barto, 2018) and learns a beam search policy to solve it. At Stage 2, CluSTeR reorganizes the clues found in Stage 1 into a series of graphs and then a Graph Convolution Network (GCN) and a Gated Recurrent Unit (GRU) are employed to deduce accurate answers from the graphs.

In general, this paper makes the following contributions:

- We formulate the TKG reasoning task from the view of human cognition and propose a two-stage model, CluSTeR, which is mainly composed of a RL-based clue searching stage and a GCN-based temporal reasoning stage.
- We advocate the importance of clue searching for the first time, and propose to learn a beam search policy via RL, which can find explicit and reliable clues for the fact to be predicted.
- Experiments demonstrate that CluSTeR achieves consistently and significantly better performance on popular TKGs and the clues

found by CluSTeR can provide interpretability for the reasoning results.

## 2 Related Work

**Static KG Reasoning.** Embedding based KG reasoning models (Bordes et al., 2013; Yang et al., 2014; Trouillon et al., 2016; Dettmers et al., 2018; Shang et al., 2019; Sun et al., 2018) have drawn increasing attention. All of them attend to learn the distributed embeddings for entities and relations in KGs. Among them, some works (Schlichtkrull et al., 2018; Shang et al., 2019; Ye et al., 2019; Vashishth et al., 2019) extend GCN to relation-aware GCN for the KGs.

However, embedding based models underestimate the symbolic compositionality of relations in KGs, which limits their usage in more complex reasoning tasks. Thus, some recent works (Xiong et al., 2017; Das et al., 2018; Lin et al., 2018; Chen et al., 2018; Wang et al., 2019; Li and Cheng, 2019) focus on multi-hop reasoning, which learns symbolic inference rules from relation paths. However, all the above methods cannot deal with the temporal dependencies among facts in TKGs.

**Temporal KG Reasoning.** Reasoning on temporal KG can broadly be categorized into two settings, interpolation (Sadeghian et al., 2016; García-Durán et al., 2018; Leblay and Chekol, 2018; Dasgupta et al., 2018; Wu et al., 2019; Xu et al., 2020; Goel et al., 2020; Wu et al., 2020; Han et al., 2020a; Jung et al., 2020) and extrapolation (Trivedi et al., 2017, 2018; Han et al., 2020b; Deng et al., 2020; Jin et al., 2019, 2020; Zhu et al., 2020; Li et al., 2021), as mentioned in Jin et al. (2020). Under the former setting, models attempt to infer missing facts at historical timestamps. While the latter setting, which this paper focuses on, attempts to predict facts in the future. Orthogonal to our work, Trivedi et al. (2017, 2018) estimate the conditional probability of observing a future fact via a temporal point process taking all historical facts into consideration. Although Han et al. (2020b) extends temporal point process to model concurrent facts, they are more capable of modeling TKGs with continuous time, where no events may occur at the same timestamp. Glean (Deng et al., 2020) incorporates a word graph constructed by the summary texts of events into TKG reasoning. The most related works are RE-NET (Jin et al., 2020) and CyGNet (Zhu et al., 2020). RE-NET uses a subgraph aggregator and GRU to model the subgraph sequence consist-

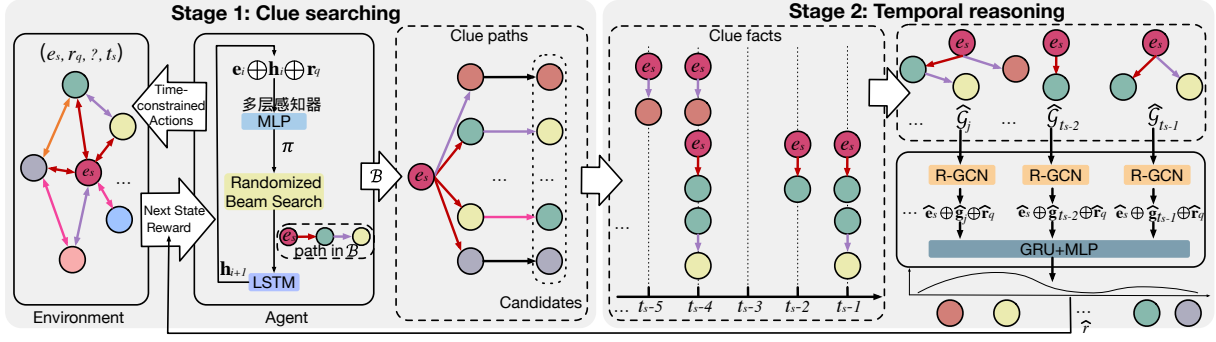


Figure 2: An illustrative diagram of the proposed CluSTeR model.

ing of 1-hop facts. CyGNet uses a sequential copy network to model repetitive facts. Both of them use heuristic strategies in the clue searching stage, which may lose lots of other informative historical facts or engage some noise. Although the above two models attempt to consider other information by pre-trained global embeddings or an extra generation model, they still mainly focus on modeling repetitive facts. Besides, all the models almost can not provide interpretability for the results.

### 3 The Proposed CluSTeR Model

We start with the notations, then introduce the model as well as its training procedure in detail.

#### 3.1 Notations

A TKG  $\mathcal{G}$  is a multi-relational directed graph with time-stamped edges between entities. A fact in  $\mathcal{G}$  can be formalized as a quadruple  $(e_s, r, e_o, t)$ . It describes that a fact of relation type  $r \in \mathcal{R}$  occurs between subject entity  $e_s \in \mathcal{E}$  and object entity  $e_o \in \mathcal{E}$  at timestamp  $t \in \mathcal{T}$ , where  $\mathcal{R}$ ,  $\mathcal{E}$  and  $\mathcal{T}$  denote the sets of relations, entities and timestamps, respectively. TKG reasoning aims to predict the missing object entity of  $(e_s, r_q, ?, t_s)$  or the missing subject entity of  $(?, r_q, e_o, t_s)$  given the set of historical facts before  $t_s$ , denoted as  $\mathcal{G}_{0:t_s-1}$ . Without loss of generality, in this paper, we predict the missing object entity in a fact, and the model can be easily extended to predicting the subject entity.

In this paper, a clue path is in the form of  $(e_s, r_1, e_1, \dots, r_k, e_k, \dots, r_I, e_I)$ , where  $e_k \in \mathcal{E}$ ,  $r_k \in \mathcal{R}$ ,  $k = 1, \dots, I$ ,  $I$  is the maximum step number and each hop in the path can be viewed as a triple  $(e_{k-1}, r_k, e_k)$ . Note that,  $e_0 = e_s$ . The clue facts are derived from the clue paths via mapping each hop  $(e_{k-1}, r_k, e_k)$  in the paths to corresponding facts  $(e_{k-1}, r_k, e_k, t_1), (e_{k-1}, r_k, e_k, t_2, \dots) \in \mathcal{G}_{0:t_s-1}$ .

#### 3.2 Model Overview

As illustrated in Figure 2, the model consists of two stages, clue searching and temporal reasoning. The two stages are coordinated to perform fast and slow thinking (Daniel, 2017), respectively, to solve the TKG reasoning task, inspired by human cognition. Specifically, Stage 1 mainly focuses on searching the clue paths of which the compositional semantic information relates to the given query with the time constraints. Then, the clue paths and the consequent candidate entities are provided for the reasoning in Stage 2, which mainly focuses on meticulously modeling the temporal information among clue facts and gets the final results. In the CluSTeR model, these two stages interact with each other in the training phase and decide the final answer jointly in the inference phase.

#### 3.3 Stage 1: Clue Searching

The purpose of Stage 1 is to search and induce the clue paths related to the given query  $(e_s, r_q, ?, t_s)$  from history. The previous studies (Jin et al., 2019, 2020; Zhu et al., 2020) use heuristic strategies to extract 1-hop repetitive paths, losing lots of other informative clue paths. Besides, there are enormous facts in the history. Thus, a learnable and efficient clue searching strategy is of great necessity. Motivated by these observations, Stage 1 can be viewed as a sequential decision problem and solved by the RL system.

##### 3.3.1 The RL System

The RL system consists of two parts, the agent and the environment. We formulate the RL system as an MDP, which is a framework of learning from interactions between the agent and the environment to find  $B$  promising clue paths. Starting from  $e_s$ , the agent sequentially selects outgoing edges via randomized beam search strategy, and traverses to

new entities until it reaches the maximum step  $I$ . The MDP consists of the following parts:

**States.** Each state  $s_i = (e_i, t_i, e_s, r_q, t_s) \in \mathcal{S}$  is a tuple, where  $\mathcal{S}$  is the set of all the available states;  $e_i$  ( $e_0 = e_s$ ) is the entity where the agent visited at step  $i$ ; and  $t_i$  ( $t_0 = t_s$ ) is the timestamp of the action taken at the previous step. Note that,  $e_s$ ,  $r_q$ , and  $t_s$  are shared by all the states for the given query.

**Time-constrained Actions.** Compared to static KGs, the time dimension of TKGs leads to an explosively large action space. Besides, the human memories focus on the latest occurring events. Thus, we constrain the time interval between the timestamp of each fact and  $t_s$  to be no more than  $m$ . And the time interval between the timestamp of the previous action and each available action is no more than  $\Delta$ . Therefore, the set of the possible actions  $A_i \in \mathcal{A}$  ( $\mathcal{A}$  is the set of all available actions) at step  $i$  consists of the time-constrained outgoing edges of  $e_i$ ,

$$A_i = \{(r', e', t') | (e_i, r', e', t') \in \mathcal{G}_{0:t_s-1}, |t' - t_i| \leq \Delta, t_s - t' \leq m\}. \quad (1)$$

To give the agent an adaptive option to terminate, a self-loop edge is added to  $A_i$ .

**Transition.** A transition function  $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$  is deterministic under the situation of TKG and just updates the state to new entities incident to the actions selected by the agent.

**Rewards.** The agent only receives a terminal reward  $R$  at the end of search, which is the sum of two parts, binary reward and real value reward. The binary reward is set to 1 if the destination entity  $e_I$  is the correct target entity  $e_o$ , and 0 otherwise. Besides, the agent gets a real value reward  $\hat{r}$  from Stage 2 if  $e_I$  is the target entity, which will be introduced in Section 3.4.

### 3.3.2 Semantic Policy Network

Given the time-constrained action space, the compositional semantic information implied in the clue paths and the time information of the clue facts is vital for reasoning. However, considering that modeling the time information requires to dive deeply into the complex temporal patterns of facts and is not the emphasis of Stage 1. Thus, we design a semantic policy network which calculates the probability distribution over all the actions according to the current state  $s_i$  and search history  $h_i = (e_s, a_0, \dots, a_{i-1})$  without considering timestamps in Stage 1. Here,  $a_i = (r_{i+1}, e_{i+1}, t_{i+1})$  is

the action taken at step  $i = 0, \dots, I - 1$ . Note that,  $h_0$  is  $e_s$ . Actually, the search history without timestamps is a candidate clue path (a clue path at step  $i$ ) mentioned in Section 3.1.

The embedding of the action  $a_i$  is  $\mathbf{a}_i = \mathbf{r}_{i+1} \oplus \mathbf{e}_{i+1}$ , where  $\oplus$  is the concatenation operation;  $\mathbf{r}_{i+1}, \mathbf{e}_{i+1}$  are the embeddings of  $r_{i+1}$  and  $e_{i+1}$ , correspondingly. Then, a Long Short Term Memory network (LSTM) is applied to encode the candidate clue path  $h_i$  as a continuous vector  $\mathbf{h}_i$ ,

$$\mathbf{h}_i = LSTM(\mathbf{h}_{i-1}, \mathbf{a}_{i-1}), \quad (2)$$

where the initial hidden embedding  $\mathbf{h}_0$  equals to  $LSTM(\mathbf{0}, \mathbf{r}_{dummy} \oplus \mathbf{e}_s)$  and  $\mathbf{r}_{dummy}$  is the embedding of a special relation introduced to form a start action with  $e_s$ . For step  $i$ , the action space is encoded by stacking the embeddings of all the actions in  $A_i$ , which are denoted as  $\mathbf{A}_i \in \mathbb{R}^{|A_i| \times 2d}$ . Here,  $d$  is the dimension of entity embeddings and relation embeddings. Then, the policy network calculates the distribution  $\pi$  over all the actions by a Multi-Layer Perceptron (MLP) parameterized with  $\mathbf{W}_1$  and  $\mathbf{W}_2$  as follows:

$$\pi(a_i | s_i; \Theta) = \eta(\mathbf{A}_i \mathbf{W}_2 f(\mathbf{W}_1 [\mathbf{e}_i \oplus \mathbf{h}_i \oplus \mathbf{r}_q])), \quad (3)$$

where  $\eta(\cdot)$  is the softmax function,  $f(\cdot)$  is the ReLU function (Glorot et al., 2011) and  $\Theta$  is the set of all the learnable parameters in Stage 1.

### 3.3.3 Randomized Beam Search

In the scenario of TKGs, the occurrence of a fact may result from multiple factors. Thus, multiple clue paths are necessary for the prediction. Besides, the intuitive candidates from Stage 1 should recall the right answers as many as possible. Therefore, we adopt randomized beam search (Sutskever et al., 2014; Guu et al., 2017; Wu et al., 2018) as the action sampling strategy of the agent, which injects random noise to the beam search in order to increase the exploration ability of the agent.

Specifically, a beam contains  $B$  candidate clue paths at step  $i$ . For each candidate path, we append  $B$  most likely actions (according to Equation 3) to the end of the path, resulting in a new path pool with size  $B \times B$ . Then we either pick the highest-scoring paths with probability  $\mu$  or uniformly sample a random path with probability  $1 - \mu$  repeatedly for  $B$  times. The score of each candidate clue path at step  $i$  equals to  $\sum_{k=0}^i \log \pi(a_k | s_k; \Theta)$ . Note that, at the first step,  $B$  1-hop candidate paths starting from  $e_s$  are generated by choosing  $B$  paths via the above picking strategy.



### 3.4 Stage 2: Temporal Reasoning

To dive deeper into the temporal information among clue facts at different timestamps and the structural information among concurrent clue facts, Stage 2 reorganizes all clue facts into a sequence of graphs  $\hat{\mathcal{G}} = \{\hat{\mathcal{G}}_0, \dots, \hat{\mathcal{G}}_j, \dots, \hat{\mathcal{G}}_{t_s-1}\}$ , where each  $\hat{\mathcal{G}}_j$  is a multi-relational graph consisting of clue facts at timestamp  $j = 0, \dots, t_s - 1$ . We use an  $\omega$ -layer RGCN (Schlichtkrull et al., 2018) to model  $\hat{\mathcal{G}}_j$ ,

$$\hat{\mathbf{h}}_{o,j}^{l+1} = f\left(\frac{1}{d_o} \sum_{(s,r)|(s,r,o,j) \in \hat{\mathcal{G}}_j} \mathbf{W}_r^l \hat{\mathbf{h}}_{s,j}^l + \mathbf{W}_{loop}^l \hat{\mathbf{h}}_{o,j}^l\right), \quad (4)$$

where  $\hat{\mathbf{h}}_{o,j}^l$  and  $\hat{\mathbf{h}}_{s,j}^l$  denote the  $l^{th}$  layer embeddings of entities  $o$  and  $s$  in  $\hat{\mathcal{G}}_j$  at timestamp  $j$ , respectively;  $\mathbf{W}_r^l$  and  $\mathbf{W}_{loop}^l$  are the weight matrices for aggregating features from different relations and self-loop in the  $l^{th}$  layer;  $d_o$  is the in-degree of entity  $o$ ; the input embedding for each entity  $k$ ,  $\hat{\mathbf{h}}_{k,j}^{l=0}$  is set to  $\hat{\mathbf{e}}_k$ , which is different from that of Stage 1.

Then,  $\hat{\mathbf{g}}_j$ , the embedding of  $\hat{\mathcal{G}}_j$ , is calculated by the mean pooling operation of all entity embeddings calculated by Equation 4 in  $\hat{\mathcal{G}}_j$ . The concatenation of  $\hat{\mathbf{e}}_s$ ,  $\hat{\mathbf{g}}_j$  and  $\hat{\mathbf{r}}_q$  (the embedding of  $r_q$  in Stage 2) is fed into a GRU,

$$\mathbf{H}_j = GRU([\hat{\mathbf{e}}_s \oplus \hat{\mathbf{g}}_j \oplus \hat{\mathbf{r}}_q], \mathbf{H}_{j-1}). \quad (5)$$

The final output of GRU, denoted as  $\mathbf{H}_{t_s-1}$ , is fed into a MLP decoder parameterized with  $\mathbf{W}_{mlp}$  to get the final scores for all the entities, i.e.,

$$p(e|e_s, r_q, t_s) = \sigma(\mathbf{H}_{t_s-1}^T \cdot \mathbf{W}_{mlp}), \quad (6)$$

where  $\sigma$  is the sigmoid activation function.

Finally, we re-rank the candidate entities according to Equation 6. To give a positive feedback to the clue paths arriving at the answer, Stage 2 gives a beam-level reward which equals to the final score of  $e_I$  from Equation 6, i.e.,  $\hat{r} = p(e_I)$ , to Stage 1.

### 3.5 Training Strategy

For Stage 1, the beam search policy network is trained by maximizing the expected reward over all queries in the training set,

$$\mathcal{J}(\Theta) = \mathbb{E}_{(e_s, r_q, e_o, t_s) \in \mathcal{G}} [\mathbb{E}_{a_0, \dots, a_{I-1}} [R(e_I|e_s, r_q, t_s)]]. \quad (7)$$

The REINFORCE algorithm (Williams, 1992) is used to optimize Equation 7. For Stage 2, we

| Datasets        | ICE14  | ICE05-15 | ICE18   | GDEL      |
|-----------------|--------|----------|---------|-----------|
| # $\mathcal{E}$ | 6,869  | 10,094   | 23,033  | 7,691     |
| # $\mathcal{R}$ | 230    | 251      | 256     | 240       |
| # <i>Train</i>  | 74,845 | 368,868  | 373,018 | 1,734,399 |
| # <i>Valid</i>  | 8,514  | 46,302   | 45,995  | 238,765   |
| # <i>Test</i>   | 7,371  | 46,159   | 49,545  | 305,241   |
| Time gap        | 1 day  | 1 day    | 1 day   | 15 mins   |

Table 1: Statistics of the datasets.

define the objective function using cross-entropy:

$$\mathcal{J}(\Phi) = -\frac{1}{|\mathcal{G}|} \sum_{(e_s, r_q, e_o, t_s) \in \mathcal{G}} \log p(e_o|e_s, r_q, t_s), \quad (8)$$

where  $\Phi$  is the set of all the learnable parameters in Stage 2. The Adam (Kingma and Ba, 2014) optimizer is used to minimize Equation 8. As Stages 1 and Stage 2 are correlated mutually, they are trained jointly. Stage 1 is pre-trained with only binary reward before the joint training process starts. Then Stage 2 is trained with the parameters of Stage 1 frozen. At last, we jointly train the two stages. Such a training strategy is widely used by other RL studies (Bahdanau et al., 2016; Feng et al., 2018).

## 4 Experiment

We design experiments to answer the following questions: **Q1.** How does CluSTeR perform on the TKG reasoning task? **Q2.** How do the two stages contribute to the final results respectively? **Q3.** Which clues are found and used for reasoning? **Q4.** Can CluSTeR provide some interpretability for the results?

### 4.1 Experimental Setup

**Datasets and Metrics.** There are four typical TKGs commonly used in previous studies, namely, ICEWS14 (García-Durán et al., 2018), ICEWS05-15 (García-Durán et al., 2018), ICEWS18 (Jin et al., 2019) and GDEL (Jin et al., 2020). The first three datasets are from the Integrated Crisis Early Warning System (ICEWS) (Boschee et al., 2015) and the last one is from Global Database of Events, Language, and Tone (GDEL) (Leetaru and Schrodt, 2013). We evaluate CluSTeR on all these datasets. ICEWS14 and ICEWS05-15 are divided into training, validation, and test sets following the preprocessing on ICEWS18 in RE-NET (Jin et al., 2020). The details of the datasets are presented in Table 1.

In the experiments, the widely used Mean Reciprocal Rank (MRR) and Hits@{1,10} are employed as the metrics. Without loss of generality, only the experimental results under the raw setting are

reported. The filtered setting is not suitable for the reasoning task under the exploration setting, as mentioned in (Han et al., 2020b; Ding et al., 2021; Jain et al., 2020). The reason is explained in terms of an example as follows: Given a test quadruple (Barack Obama, visit, ?, 2015-1-25) with the correct answer India. Assume there is a quadruple (Barack Obama, visit, Germany, 2013-1-18) in the training set. The filtered setting used in the previous studies ignores time information and considers (Barack Obama, visit, Germany, 2015-1-25) to be valid because (Barack Obama, visit, Germany, 2013-1-18) appears in the training set. It thus removes the quadruple from the corrupted ones. However, the fact (Barack Obama, visit, Germany) is temporally valid on 2013-1-18, instead of 2015-1-25. Therefore, to test the quadruple (Barack Obama, visit, ?, 2015-1-25), (Barack Obama, visit, Germany, 2015-1-18) should not be removed. In this way, the filtered setting wrongly removes quite a lot of quadruples and thus leads to over-optimistic experimental performance.

**Baselines.** The CluSTeR model is compared with two categories of models, i.e., models for static KG reasoning and models for TKG reasoning under the exploration setting. The typical static models DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), RGCN (Schlichtkrull et al., 2018), ConvE (Dettmers et al., 2018) and RotE (Sun et al., 2018) are selected with the temporal information of facts ignored. We also choose MINERVA (Das et al., 2018), the RL-based multi-hop reasoning model, as the baseline. For TKG models, the representative Know-evolve (Trivedi et al., 2017), DyRep (Trivedi et al., 2018), CyGNet (Zhu et al., 2020) and RE-NET (Jin et al., 2020) are selected. Besides, following RE-NET (Jin et al., 2020), we extend two models for temporal homogeneous graphs, GCRN (Seo et al., 2018) and EvolveGCN-O (Pareja et al., 2019), to RGCRN and EvolveRGCN by replacing GCN with RGCN. We use ConvE (Dettmers et al., 2018), a more stronger decoder to replace the MLP (Jin et al., 2020) for the two models. For Know-evolve and DyRep, RE-NET extends them to TKG reasoning task but does not release their codes. Thus, we only report the results from their papers. For other baselines, we reproduce all the results with the optimal parameters tuning on the validation set.

**Implementation Details.** In the experiments, the embedding dimension  $d$  for the two stages, is

set to 200. For Stage 1, we adopt an adaptive approach for selecting the time interval  $m$ . Specifically, for ICEWS14, ICEWS05-15, and GDEL,  $m$  is set to the last one timestamp the query pattern  $(e_s, r_q, ?)$  appearing in the dataset before  $t_s$ . And for ICEWS18,  $m$  is set to the last third timestamp.  $\Delta$  is set to 3 for all the datasets. We set the maximum step number  $I = 1, 2$  and find  $I = 1$  is better for all the datasets. The number of the LSTM layers is set to 2 and the dimension of the hidden layer of LSTM is set to 200 for all the datasets. The beam size is set to 32 for the three ICEWS datasets and 64 for GDEL.  $\mu$  is set to 0.3 for all the datasets. For Stage 2, the maximum sequence length of GRU is set to 10, the number of the GRU layers is set to 1 and the number of the RGCN layers is set to 2 for all the datasets. For each fact in  $\mathcal{G}_{0:t_s-1}$ , we add the corresponding inverse fact into  $\mathcal{G}_{0:t_s-1}$ . All the experiments are carried out on Tesla V100.

## 4.2 Results on TKG Reasoning

The results on TKG reasoning are presented in Table 2. CluSTeR consistently outperforms the baselines on all the ICEWS datasets, which convincingly verifies its effectiveness and answers Q1. Especially on ICEWS14, CluSTeR even achieves the improvements of 7.1% in MRR, 4.5% in Hits@1, and 13.7% in Hits@10 over the best baselines. Specifically, CluSTeR significantly outperforms the static models (i.e., those in the first block of Table 2) because it captures the temporal information of some important history. Moreover, CluSTeR drastically performs better than those temporal models. Compared with DyRep and Know-evolve that consider all the history, CluSTeR can focus on more vital clues. Different from RGCRN and EvolveRGCN which model all history from several latest timestamps, CluSTeR models a longer history after reducing all history to a few clues. CyGNet and RE-NET mainly focus on modeling the repetitive clues or all the 1-hop clues and show strong performance. CluSTeR also outperforms them on the three ICEWS datasets, because the RL-based Stage 1 can find more explicit and reliable clues.

The experimental results on GDEL demonstrate that the performance of static models and temporal ones are similarly poor, as compared with those of the other three datasets. We further analyze the GDEL dataset and find that a large number of its entities are abstract concepts which do not indicate a specific entity (e.g., PRESIDENT, POLICE

| Model       | ICE14       |             |             | ICEWS05-15  |             |             | ICE18       |             |             | GDELT       |             |             |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|             | MRR         | H@1         | H@10        | MRR         | H@1         | H@10        | MRR         | H@1         | H@10        | MRR         | H@1         | H@10        |
| DistMult    | 24.9        | 17.3        | 40.2        | 16.4        | 9.8         | 29.9        | 17.5        | 10.1        | 32.6        | 15.6        | 9.3         | 28.0        |
| ComplEx     | 31.9        | 22.2        | 50.7        | 23.1        | 14.5        | 40.6        | 18.8        | 11.1        | 26.8        | 12.3        | 8.0         | 20.6        |
| RGCN        | 27.1        | 18.4        | 44.2        | 27.3        | 19.1        | 43.6        | 17.0        | 8.7         | 34.0        | 10.9        | 4.6         | 22.6        |
| ConvE       | 30.9        | 21.7        | 50.1        | 25.2        | 16.0        | 44.4        | 24.8        | 15.1        | 44.9        | 17.3        | 10.4        | 31.3        |
| RotatE      | 27.5        | 18.0        | 47.2        | 19.9        | 10.9        | 38.7        | 15.5        | 7.0         | 33.9        | 5.3         | 1.2         | 12.5        |
| MINERVA     | 33.2        | 25.7        | 48.3        | 30.7        | 25.8        | 39.9        | 21.0        | 15.3        | 33.0        | 12.1        | 10.0        | 16.7        |
| Know-Evolve | —           | —           | —           | —           | —           | —           | 7.4         | 3.3         | 14.8        | 15.9        | 11.7        | 22.3        |
| DyRep       | —           | —           | —           | —           | —           | —           | 7.8         | 3.6         | 16.3        | 16.3        | <b>11.8</b> | 23.9        |
| RGCRN       | 36.9        | 27.0        | 56.1        | 39.4        | 28.7        | 60.4        | 26.2        | 16.4        | 45.8        | 17.7        | 10.9        | 30.9        |
| EvolveRGCN  | 37.1        | 27.0        | 57.0        | 40.7        | 30.3        | 61.3        | 23.6        | 36.3        | 50.4        | 17.4        | 11.0        | 29.9        |
| CyGNet      | 36.5        | 27.4        | 54.4        | 37.4        | 27.5        | 56.1        | 26.8        | 17.1        | 45.7        | 18.0        | 10.9        | 31.6        |
| RE-NET      | 38.9        | 29.3        | 57.5        | 41.7        | 31.1        | 62.0        | 28.4        | 18.4        | 47.9        | <b>19.0</b> | 11.6        | <b>33.5</b> |
| CluSTeR     | <b>46.0</b> | <b>33.8</b> | <b>71.2</b> | <b>44.6</b> | <b>34.9</b> | <b>63.0</b> | <b>32.3</b> | <b>20.6</b> | <b>55.9</b> | 18.3        | 11.6        | 31.9        |

Table 2: Experimental results on TKG reasoning (in percentage) compared with static models (the top part) and temporal models (the bottom part).

| Model               | ICE14       | ICE05-15    | ICE18       | GDELT       |
|---------------------|-------------|-------------|-------------|-------------|
| Stage 1 ( $I = 2$ ) | 43.1        | 43.3        | 27.6        | 15.3        |
| Stage 1 ( $I = 1$ ) | 44.1        | 46.0        | 30.3        | 17.6        |
| Stage 2             | 41.5        | 45.0        | 30.1        | <b>19.6</b> |
| CluSTeR             | <b>46.8</b> | <b>46.9</b> | <b>33.1</b> | 18.7        |

Table 3: Results (in percentage) by different variants of CluSTeR on all the datasets.

and GOVERNMENT). Among the top 50 frequent entities, 28 are abstract concepts and 43.72% corresponding events involve abstract concepts. Those abstract concepts make future prediction under the raw setting almost impossible, since we cannot predict a president’s activities without knowing which country he belongs to.

### 4.3 Ablation Study

To answer Q2, i.e., how the two stages contribute to the final results, we report the MRR results of the variants of CluSTeR on the validation set of all the datasets in Table 3. The first two lines of Table 3 show the results only using Stage 1, where the maximum step  $I$  is set to 1 and 2, respectively. Following Lin et al. (2018), the score of the target entity is set to the highest score among the paths when more than one path leads to it. It can be observed that the results decrease when only using Stage 1, because the temporal information among facts is ignored. The third line shows the results only using Stage 2 with extracted 1-hop repetitive clues as the inputs. The results decrease on all the ICEWS datasets when only using Stage 2, demonstrating that only repetitive clues are not enough for the prediction. For GDELT, only Stage 2 achieves the best results, which demonstrates that only using repetitive clues is effective enough for it. It is

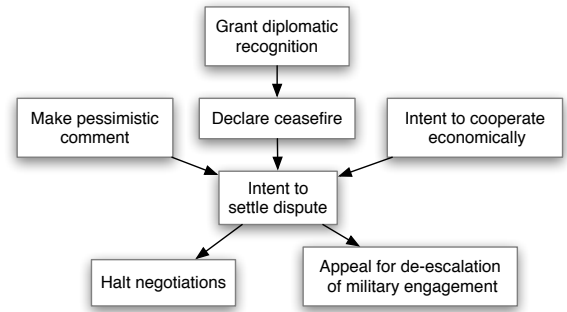


Figure 3: A clue graph constructed by Stage 1.

because that only using the most straightforward repetitive clues in Stage 2 can alleviate the influence of noise produced by abstract concepts. It also matches our observations mentioned in Section 4.2.

From the first two lines of Table 3, it can be seen that the performance of Stage 1 decreases when  $I$  is set to 2. To further analyze the reason, we extract paths from ICEWS18 without considering timestamps via AMIE+ (Galárraga et al., 2015), a widely used and accurate approach to extract logic rules (paths) in static KGs. We check the top fifty paths manually and present the top five convincing paths in Table 4. It can be seen that there are no strong dependencies between the query relations and the 2-hop paths. Thus, in this situation, longer paths bring exponential noise clues, which pull down the precision. We do experiments on all the datasets from ICEWS and GDELT and find the same conclusion. We leave it for future work to construct a more complex dataset for verifying the effectiveness of multi-hop clue paths.

| Query relations                  | 2-hop paths   | Scores |
|----------------------------------|---|--------|
| (A, Declare ceasefire, C)        | (A, Intent to cooperate, B, Intent to meet, C)              | 0.4071 |
| (A, Intent to settle dispute, C) | (A, Consult, B, Intent to diplomatic cooperation, C)        | 0.3843 |
| (A, Intent to settle dispute, C) | (A, Intent to diplomatic cooperation, B, Intent to meet, C) | 0.3725 |
| (A, Halt negotiations, C)        | (A, Engage in negotiation, B, Intent to meet, C)            | 0.3717 |
| (A, Accuse of crime, C)          | (A, Accuse, B, Criticize or denounce, C)                    | 0.3256 |

Table 4: The top five convincing 2-hop paths extracted by AMIE+ from ICEWS18.

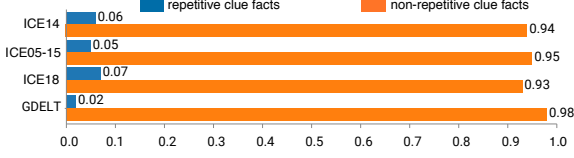


Figure 4: Statistic of categories of clue facts in Stage 2.

#### 4.4 Detail Analysis

To answer Q3, we show some non-repetitive clues found in Stage 1 in Figure 3. We use (relation in 1-hop non-repetitive clue path, query relation) pairs on ICEWS18 to construct a clue graph. Arrows begin with the relations in the clue paths and end with the query relations. It is interesting to find that CluSTeR can actually find some causal relations. Moreover, compared to the 2-hop clue paths shown in Table 4, the 1-hop clue paths are more informative. It also gives explanations to the outperformance of the 1-hop clue paths.

Besides, we illustrate the statistics of clue facts used during Stage 2 in Figure 4. The proportion of the repetitive clue facts is less than 7% and the proportion of the non-repetitive clue facts is more than 93% on the datasets. The abundant of the non-repetitive clue facts used in Stage 2 also explains the outperformance of CluSTeR to a certain degree.

#### 4.5 Case Study

To answer Q4, we show how CluSTeR conducts reasoning and explains the results for the given two queries from the test set of ICEWS14 in Figure 5. For the first query: (*Congress (United States), Impose sanctions, ?, 334*<sup>1</sup>), we choose the top three candidates in Stage 1 and demonstrate some clue paths of the three entities in the left top part of Figure 5. The clue paths like (*Congress (United States), Criticize or denounce*<sup>-1</sup>, *China*), (*Congress (United States), Engage in negotiation*<sup>-1</sup>, *Iran*) give the evidence for candidate entities *China* and *Iran*, correspondingly. In Stage 1, CluSTeR has an intuitive candidate set including *China*, *Iran* and *France*. The score of *China* (-2.69) and *Iran* (-2.71) are similar but the

wrong answer, *China*, has a higher score than the right one, *Iran*. It is because Stage 1 does not take the temporal information into consideration. However, the score gap is obvious between *Iran* and *France*, which shows that Stage 1 can measure the qualities of different clue paths and distinguish the semantic-related entities from the others. In Stage 2, CluSTeR reorganizes the clue facts by their timestamps, as shown in the right top part of Figure 5. (*Congress (United State), Engage in negotiation*<sup>-1</sup>, *Iran*, 323) and (*Congress (United State), Make a visit, China*, 227) make *Iran* the more possible answer. For the second query: (*China, Express intent to settle dispute, ?, 364*), clue paths in the left bottom of Figure 5 are all associated with the query. Stage 1 induces all entities to only two entities through these clue paths but misleads to the wrong answer, *Iran*. Actually, even a human may give the wrong answer with only fast thinking. After diving into the temporal information of clue facts and conduct slow thinking, some causal information and period information can be captured by Stage 2. Although *Sign formal agreement* is associated with *Express intent to settle dispute*, it can not be the reason for the latter. Moreover, from the subgraph sequence in the right bottom part of Figure 5, it can be seen that the cooperation period between *China* and *Japen* just begins at 363, but the cooperation period between *China* and *Iran* has been going on for several days. (*China, Express intent to settle dispute, ?, 364*) is more likely to be an antecedent event to the cooperation period and the answer is *Japen*.

Above all, for each fact to be predicted, CluSTeR can provide the clues for each candidate entity, which presents the insight and provides interpretability for the reasoning results. It is similar to the natural thinking pattern of human, in which only explicit and reliable clues are needed.

#### 4.6 Performance under the Time-aware Filtered Setting

As mentioned in Section 4.1, the widely adopted filtered setting in the existing studies is not suitable

<sup>1</sup>Here, 334 represents the 334th day in the year 2014.



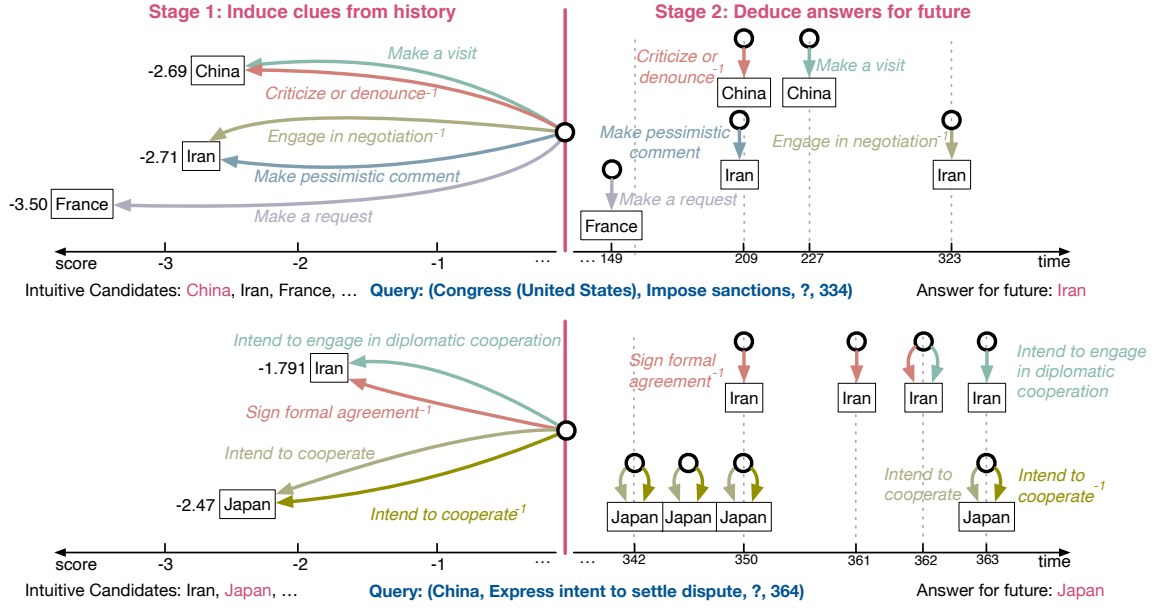


Figure 5: Two cases to illustrate how CluSTeR conducts reasoning and explains the results. Each black circle represents a query entity.

| Model    | ICE14 |      |      | ICEWS05-15 |      |      | ICE18 |      |      | GDELT |      |      |
|----------|-------|------|------|------------|------|------|-------|------|------|-------|------|------|
|          | MRR   | H@1  | H@10 | MRR        | H@1  | H@10 | MRR   | H@1  | H@10 | MRR   | H@1  | H@10 |
| raw      | 46.0  | 33.8 | 71.2 | 44.6       | 34.9 | 63.0 | 32.3  | 20.6 | 55.9 | 18.3  | 11.6 | 31.9 |
| filtered | 47.1  | 35.0 | 72.0 | 45.4       | 34.3 | 67.7 | 34.5  | 22.9 | 57.7 | 18.5  | 12.1 | 32.1 |

Table 5: Experimental results under the raw setting and the (time-aware) filter setting.

for the temporal reasoning task addressed in this paper. The essential problem of the above filtered setting is that it ignores the time information of a fact. Therefore, we also adopt an improved filtered setting where the time information is also considered, thus called time-aware filtered setting (Han et al., 2020b; Han et al.). Specifically, only the facts occur at the predicted time are filtered. The results are in Table 5. It can be seen that the experimental results under the time-aware filtered setting are close to those under the raw setting. This is because that only a very small number of facts are removed under this filtered setting. The results also show the convincing of the raw setting.

## 5 Conclusions

In this paper, we proposed a two-stage model from the view of human cognition, named CluSTeR, for TKG reasoning. CluSTeR consists of a RL-based clue searching stage (Stage 1) and a GCN-based temporal reasoning stage (Stage 2). In Stage 1, CluSTeR finds reliable clue paths from history and generates intuitive candidate entities via RL. With the found clue paths as input, Stage 2 reorganizes

the clue facts derived from the clue paths into a sequence of graphs and performs deduction on them to get the answers. By the two stages, the model demonstrates substantial advantages on TKG reasoning. Finally, it should be mentioned that, although the four TKGs adopted in the experiments were created based on the events in the real world, the motivation of this paper is to propose this TKG reasoning model only for scientific research.

## Acknowledgment

We gratefully acknowledge the help and assistance from Long Bai, Yunqi Qiu, Bing Li and Bingbing Xu. Moreover, the work is supported by the National Key Research and Development Program of China under grant 2016YFB1000902, the National Natural Science Foundation of China under grants U1911401, 62002341, 61772501, U1836206 and 61722211, the GFKJ Innovation Program, Beijing Academy of Artificial Intelligence under grant BAAI2019ZD0306, and the Lenovo-CAS Joint Lab Youth Scientist Project.

## References

- Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.
- Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.
- Elizabeth Boschee, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icwes coded event data. *Harvard Data-verse*, 12.
- Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. 2018. Variational knowledge graph reasoning. In *Proceedings of NAACL-HLT*, pages 1823–1832.
- Kahneman Daniel. 2017. Thinking, fast and slow.
- 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 *International Conference on Learning Representations*.
- Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011.
- Songgaogun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Zifeng Ding, Zhen Han, Yunpu Ma, and Volker Tresp. 2021. Temporal knowledge graph forecasting with neural ode. *arXiv preprint arXiv:2101.05151*.
- Jonathan St BT Evans. 1984. Heuristic and analytic processes in reasoning. *British Journal of Psychology*, 75(4):451–468.
- Jonathan St BT Evans. 2003. In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences*, 7(10):454–459.
- Jonathan St BT Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annu. Rev. Psychol.*, 59:255–278.
- Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.
- Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995.
- Simon Gottschalk and Elena Demidova. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *European Semantic Web Conference*, pages 272–287. Springer.
- Simon Gottschalk and Elena Demidova. 2019. Eventkg—the hub of event knowledge on the web—and biographical timeline generation. *Semantic Web*, (Preprint):1–32.
- Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926*.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020a. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7301–7316.
- Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020b. Graph hawkes neural network for forecasting on temporal knowledge graphs. *8th Automated Knowledge Base Construction (AKBC)*.

- Prachi Jain, Sushant Rathi, Soumen Chakrabarti, et al. 2020. Temporal knowledge base completion: New algorithms and evaluation protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3733–3747.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *EMNLP*.
- Woojeong Jin, Changlin Zhang, Pedro Szekely, and Xiang Ren. 2019. Recurrent event network for reasoning over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*.
- Jaehun Jung, Jinhong Jung, and U Kang. 2020. T-gap: Learning to walk across time for temporal knowledge graph completion. *arXiv preprint arXiv:2012.10595*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Gizem Korkmaz, Jose Cadena, Chris J Kuhlman, Achla Marathe, Anil Vullikanti, and Naren Ramakrishnan. 2015. Combining heterogeneous data sources for civil unrest forecasting. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 258–265.
- Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776. International World Wide Web Conferences Steering Committee.
- Kalev Leetaru and Philip A Schrod. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.
- Ruiping Li and Xiang Cheng. 2019. Divine: A generative adversarial imitation learning framework for knowledge graph reasoning. 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)*, pages 2642–2651.
- Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. *arXiv preprint arXiv:2104.10353*.
- Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. *arXiv preprint arXiv:1808.10568*.
- Sathappan Muthiah, Bert Huang, Jaime Arredondo, David Mares, Lise Getoor, Graham Katz, and Naren Ramakrishnan. 2015. Planned protest modeling in news and social media. In *Twenty-Seventh IAAI Conference*. Citeseer.
- Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E Leiserson. 2019. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *arXiv preprint arXiv:1902.10191*.
- Lawrence Phillips, Chase Dowling, Kyle Shaffer, Nathan Hodas, and Svitlana Volkova. 2017. Using social media to predict the future: a systematic literature review. *arXiv preprint arXiv:1706.06134*.
- Ali Sadeghian, Miguel Rodriguez, Daisy Zhe Wang, and Anthony Colas. 2016. Temporal reasoning over event knowledge graphs. In *Workshop on Knowledge Base Construction, Reasoning and Mining*.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.
- Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer.
- Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3060–3067.
- Alessio Signorini, Alberto Maria Segre, and Philip M Polgreen. 2011. The use of twitter to track levels of disease activity and public concern in the us during the influenza a h1n1 pandemic. *PloS one*, 6(5):e19467.
- Steven A Sloman. 1996. The empirical case for two systems of reasoning. *Psychological bulletin*, 119(1):3.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.
- Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3462–3471.

- Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2018. Dyrep: Learning representations over dynamic graphs.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.
- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*.
- Heng Wang, Shuangyin Li, Rong Pan, and Mingzhi Mao. 2019. Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning. 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)*, pages 2623–2631.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5730–5746.
- Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1808.08866*.
- Tianxing Wu, Arijit Khan, Huan Gao, and Cheng Li. 2019. Efficiently embedding dynamic knowledge graphs. *arXiv*, pages arXiv–1910.
- Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. *arXiv preprint arXiv:1707.06690*.
- Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. 2020. Temporal knowledge graph completion based on time series gaussian embedding. In *International Semantic Web Conference*, pages 654–671. Springer.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.
- Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4135–4141.
- Liang Zhao. 2020. Event prediction in big data era: A systematic survey. *arXiv preprint arXiv:2007.09815*.
- Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2020. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *arXiv preprint arXiv:2012.08492*.