Temporal Knowledge Graph Reasoning via Time-Distributed Representation Learning

Kangzheng Liu[†‡], Feng Zhao[†*], Guandong Xu[‡], Xianzhi Wang[‡], and Hai Jin[†]
[†]National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[‡]Data Science and Machine Intelligence Lab, University of Technology Sydney, Sydney, Australia
[†]{frankluis, zhaof, hjin}@hust.edu.cn, [‡]{guandong.xu, xianzhi.wang}@uts.edu.au

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00038

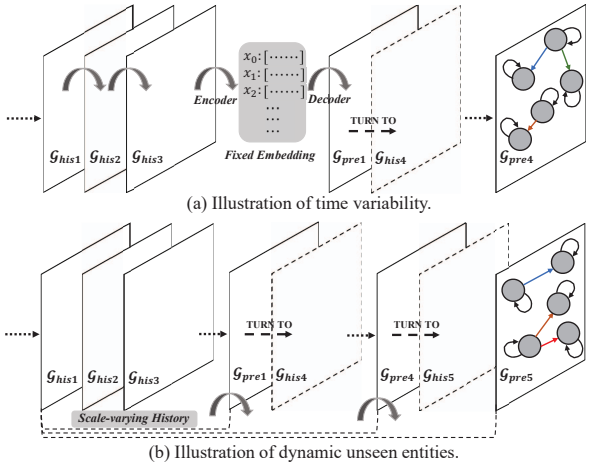
***Abstract*—*Temporal knowledge graph* (TKG) reasoning has attracted significant attention. Recent approaches for modeling historical information have led to great advances. However, the problems of time variability and unseen entities have become two major obstacles preventing further development. The time variability problem means that different historical timestamps play different roles in the inference process. Furthermore, in the context of time variability, the unseen entity problem means that a query cannot obtain a predicted entity that is unseen in the scale-varying history rather than in a fixed set, thus turning from static to dynamic. In this paper, we propose a novel method named DHU-NET for addressing the time variability challenge and the dynamic unseen entity challenge derived from it. With regard to the former concern, we propose a time-distributed representation learning method based on a *graph convolutional network* (GCN) and a self-attention mechanism, which learns the distributed representations of facts at different historical timestamps and comprehensively pays different levels of attention to the different timestamps. With regard to the latter issue, we extract the unseen entities from a global static KG based on a copy mechanism and bring them into consideration during the final prediction step. Experiments on six benchmark datasets demonstrate the substantial improvements achieved by DHU-NET in terms of multiple evaluation metrics. Our released codes are available at https://github.com/CGCL-codes/DHUNET.**

***Index Terms*—Temporal knowledge graph, Time-distributed representation learning, Graph convolutional network**

I. Introduction

Knowledge graphs (KGs) represent facts in triples (*subject*, *predicate*, *object*). However, real-world facts are often time-constrained. Thus, *temporal KGs* (TKGs) have been proposed to address this challenge and represent facts as quadruples (*subject*, *predicate*, *object*, *time*). For example, the quadruple (*Obama*, *Make a visit*, *China*, *2014-11-11*) means that Obama visited China on November 11th, 2014. Nevertheless, most TKGs in the real world are incomplete; thus, reasoning approaches based on TKGs mainly aim to predict missing subjects or objects.

A TKG is a series of sequential subgraphs, each of which is static at its corresponding timestamps. Compared to previous work [1]–[3] focusing on query (prediction) timestamp modeling, recent work [4]–[11] has achieved great improvements regarding the prediction of TKGs by modeling historical sequential information. The modeling objects of these methods mainly include the structural information contained in subgraphs and the sequential dependencies between subgraphs.



(a) Illustration of time variability.

(b) Illustration of dynamic unseen entities.

Fig. 1. Illustrations of the time variability challenge and the dynamic unseen entity challenge. The dotted arrow indicates the direction of time development.

However, modeling history inevitably presents the challenge of time variability. As shown in Figure 1(a), the traditional codec-based static modeling strategy tends to obtain fixed entity representations through an encoder from a scale-invariant history and then conducts reasoning on the query timestamp through a decoder. Nevertheless, in real scenarios, new historical information is constantly generated with the progression of time; that is, the previous prediction (query) timestamp (e.g., $\mathcal{G}_{pre3}$) is converted into a new historical timestamp (e.g., $\mathcal{G}_{his4}$). Thus, it is unreasonable to ignore the new historical information and use the previous fixed entity embeddings. Moreover, different from the sequential graph modeling approaches in previous work [4], [6], [8], the sequential subgraphs all reflect the historical states of the TKGs at a certain timestamp. Thus, for a query timestamp, the representation of each historical temporal subgraph (such as $\mathcal{G}_{his3}$, $\mathcal{G}_{his2}$, $\mathcal{G}_{his1}$, or the new historical subgraph $\mathcal{G}_{his4}$) plays a larger or smaller role in the reasoning process. Nonetheless,

*Corresponding author

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00038

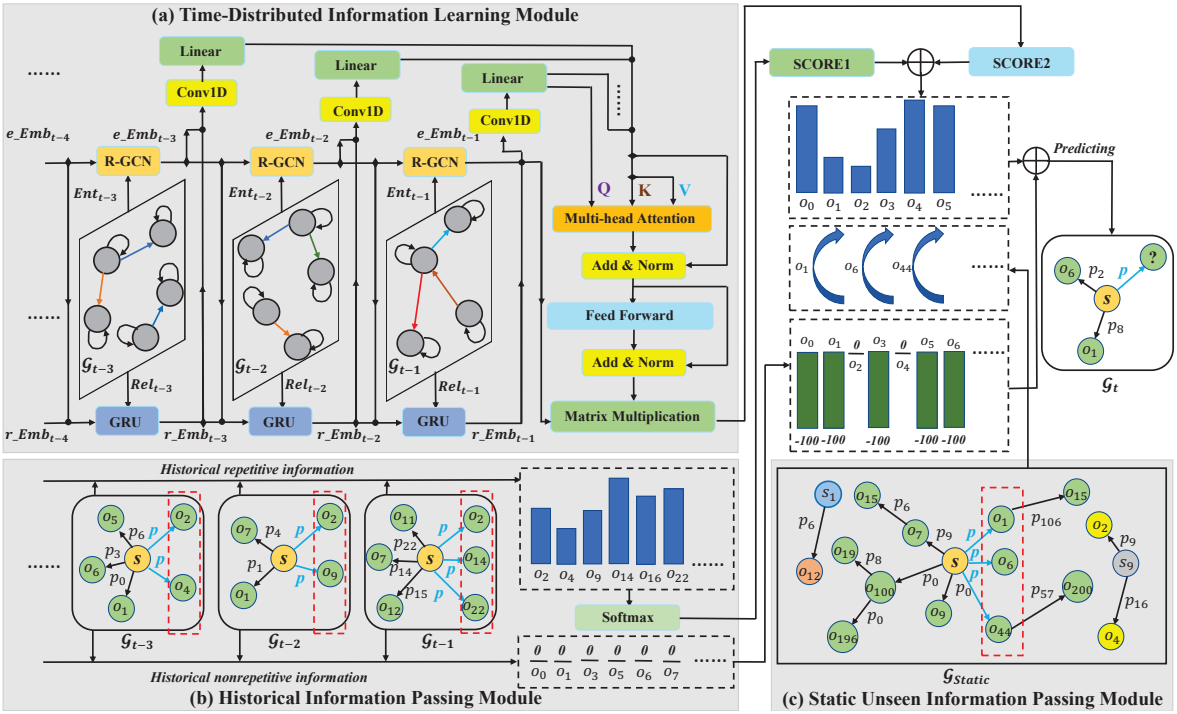


Fig. 2. The framework of DHU-NET. The boxes with solid lines indicate different types of KGs (i.e., sequential subgraphs at different timestamps or static graphs). The blue and green bars indicate the predicted positive and negative probabilities, respectively, for a query $(s, p, ?, t)$. Arrows indicate the direction of information transport, diamonds indicate the copy branches of a transportation vector, and dots indicate the concatenation of multiple vectors.

constrained by the codec-based framework, previously developed approaches compress time-distributed information into low-dimensional fixed embeddings, thus resulting in the loss of distributed information and greatly reducing the performance of these methods. For example, the occurring fact (*Man*, *Diagnose*$^{-1}$, *COVID-19*, *2019-12-30*) and the concomitant disease (*Man*, *Diagnose*$^{-1}$, *sepsis*, *2021-9-20*) are both likely to become mild diseases with improved treatment; that is, (*Man*, *Diagnose*$^{-1}$, *cough*, *2022-5-18*). Thus, the COVID-19, Sepsis, and Cough entities should have different representations in different periods so that they can highlight different reasonable features for reasoning at different timestamps. Therefore, the time variability challenge mainly includes two aspects: 1) the challenge of constantly emerging historical information and 2) the fact that different historical timestamps play different roles in temporal reasoning. CEN [11] tries to solve the first-level challenge through online training strategies but still fails to deal with the time variability challenge at the second level.

Modeling only historical information inevitably results in the unseen entity challenge. The static unseen entity challenge refers to a model's inability to obtain representations for entities that have not been present in the history at a fixed scale. Recent work [6], [9] limited the historical scale to the size of the training set, and this scale remained unchanged during inferential prediction. However, as shown in Figure 1(b), in the context of time variability, with the development of time, a constant change occurs from the previous prediction (query) timestamps (e.g., $\mathcal{G}_{pre1}$, $\mathcal{G}_{pre4}$) to the new historical timestamps (e.g., $\mathcal{G}_{his4}$, $\mathcal{G}_{his5}$). Therefore, for the query timestamps $\mathcal{G}_{pre1}$, $\mathcal{G}_{pre4}$, and $\mathcal{G}_{pre5}$ at different periods, their historical scales are constantly varying, resulting in some unseen entities becoming seen entities, and these newly generated seen entities can be accurately represented by modeling distributed historical information; thus, modeling them as invisible entities would greatly degrade the performance of the model. We refer to this problem as the dynamic unseen entity challenge.

In this paper, to address the time variability challenge and the dynamic unseen entity challenge, we propose a TKG reasoning method, namely, DHU-NET, which models evolutionary time-**D**istributed information, **H**istorical information, and global static **U**nseen information.

To comprehensively address the first challenge from the above two aspects, as Figure 2(a) shows, we design a time-distributed representation learning method based on a *graph convolutional network* (GCN) and a self-attention mechanism [12]. To accurately represent the different roles played by various timestamps, we need to simultaneously consider the structural information contained in the subgraphs and the time evolution information between the sequential subgraphs. Specifically, we model the entity sequence of the temporal subgraphs through a relational GCN (R-GCN) [13] to obtain a distributed representation of the entities at each historical timestamp, model the predicate sequence through a *gated recurrent unit* (GRU) [14] to obtain a distributed representation of the predicates at each historical timestamp and learn the

time dependencies between the subgraphs by sequentially passing the representation of the previous timestamp to the next timestamp. In addition, to consider the emerging historical information, we update the range of the historical subgraph sequence in time with the development of query timestamps. Finally, based on the self-attention mechanism [12], DHU-NET allocates appropriate attention to the historical subgraph representations at different timestamps through autonomous learning.

To address the second challenge, as Figure 2(c) shows, we first randomly initialize the global entities and obtain a distributed representation of the new historical entities by continuously modeling the latest historical timestamps. Then, we extract unseen entities from the global static KG based on a copy mechanism [5]. Although the historical scale is constantly changing, the global static KG always maintains a certain scale and preserves all the factual information.

In summary, the contributions of our work are as follows:

- Different from the traditional codec-based architecture, we propose a time-distributed representation learning method, which learns the distributed representations of entities and predicates at different historical timestamps and updates the historical modeling range with the evolution of time, thus effectively overcoming the time variability challenge during TKG reasoning.
- We first investigate the dynamic unseen entity challenge in TKG reasoning under the context of time variability. To address this challenge, we propose a static unseen information passing module; instead of being limited to a scale-varying history, this module extracts unseen entities from the global static KG through the copy mechanism and takes them into account during the prediction step.
- Extensive experiments on six public TKG datasets are conducted. The improvements achieved in terms of almost all evaluation metrics demonstrate the effectiveness of our method for TKG reasoning.

The remainder of this paper is organized as follows. Related work is introduced in Section II. The proposed model is detailed in Section III. In addition, the experimental analyses are presented in Section IV, followed by the conclusions in Section V.

II. Related Work

TKG reasoning methods can be divided into two categories according to whether they adapt to temporal dynamics: static inference methods and dynamic inference methods.

1) Static Reasoning Methods: Static modeling methods ignore the time information in TKGs, and they only deal with fact triples, excluding the time dimension. Translation-based methods sum the embeddings of the heads (subjects) and relations (predicates) to make them as close as possible to the vector of the tails (objects); such approaches include TransE [15], TransH [16], and so on. RotatE [17] maps entities and predicates to complex spaces and then defines each predicate as the rotation of the complex plane from one entity to another. In addition, ComplEx [18] performs eigenvalue decomposition on predicate matrices in a complex space, and DistMult [19] sets the predicate matrices to be decomposed as diagonal matrices. Convolutional network-based methods express the entities and predicates as two matrices and introduce a convolution kernel to perform the convolution operation; these approaches include ConvE [20], ConvKB [21], and Conv-TransE [22]. GCN-based approaches, which often act as encoders, learn the representations of the entities in KGs by aggregating the structural information of neighbors; representative techniques include R-GCN [13] and Comp-GCN [23].

2) Dynamic Reasoning Methods: Considering the temporal dynamics in TKGs, dynamic reasoning methods achieve better performance. TTransE [2] performs inference by adding time information to the embeddings of predicates. Deriving [24] predicts the time ranges of the unlabeled edges. HyTE [3] models time information as hyperplanes. TA-DistMult [1] also integrates the temporal information of facts into the embeddings of predicates. However, this method cannot capture the evolution patterns of sequential subgraphs by focusing on the current timestamp only; thus, some recent work models sequential history subgraphs and has been highly promoted. RE-NET [4] models the occurrence of facts as conditional probability based on historical sequence subgraphs with a certain length. CyGNet [5] suppresses the role of historical nonrepetitive facts in prediction based on a copy mechanism. xERTE [6] generates an inference graph with a certain number of hops. CluSTeR [7] uses reinforcement learning to extract the related subgraphs of queries, and then the related subgraph sequence is modeled by a GCN. RE-GCN [8] models the evolutionary patterns of historical subgraphs via a recurrent R-GCN and time gate units. TITer [9] uses reinforcement learning to determine the evolution pattern in the given query path. TLogic [10] constrains the query path based on the temporal logic rules extracted from a temporal random walk.

However, all the above-mentioned methods ignore the time variability problem in TKG reasoning. CEN [11] only tries to address the challenge of newly emerging historical timestamps via an online training strategy; it still cannot effectively model time-distributed historical information. In addition, xERTE [6] and TITer [9] both try to address the challenge of static unseen entities, but they are not applicable in dynamic situations under time-variable backgrounds and cannot deal with newly emerging historical facts in time.

III. Methodology

In this section, we introduce the proposed DHU-NET method. We first describe the employed notations and definitions. Then, we present the model framework and the three modules of the model. In addition, we discuss the parameter learning strategy and the computational complexity.

A. Definitions and Overview

1) Notations and Definitions: In a TKG $\mathcal{G}$, we define the entity set as $\mathcal{E}$ with a size of N, the predicate set as $\mathcal{R}$ with a size of P, and the timestamp set as $\mathcal{T}$ with a size of T.

Thus, a TKG is composed of facts in the form of quadruples $(s,\ p,\ o,\ t)$, in which $\{s,\ o\} \in \mathcal{E}$, $p \in \mathcal{R}$, and $t \in \mathcal{T}$. To facilitate modeling, a TKG can be reorganized as a sequence of temporal static subgraphs $\mathcal{G} = \{\mathcal{G}_0,\ \mathcal{G}_1,\ \mathcal{G}_3,\ \mathcal{G}_4, ...,\ \mathcal{G}_{T-1}\}$, where each static subgraph $\mathcal{G}_t$ is composed of triples $(s,\ p,\ o)$ occurring at timestamp t. In addition, we define the embedding dimensionality as d. For each temporal subgraph $\mathcal{G}_t$, we define an entity embedding matrix as $\mathbf{E}_t$ and a predicate embedding matrix as $\mathbf{R}_t$. In particular, we randomly initialize the input embeddings of the entities and predicates for the first historical timestamp as $\mathbf{E}_{init}$ and $\mathbf{R}_{init}$, respectively.

TKG reasoning predicts a missing object entity, $(s,\ p,\ ?,\ t)$ or predicts a missing subject entity, $(?,\ p,\ o,\ t)$ at a specific timestamp t. For each fact $(s,\ p,\ o,\ t)$, we append the inverse quadruples $(o,\ p^{-1},\ s,\ t)$ into the TKGs; thus, the adjacent nodes can all be expressed as in-degree forms, which is conducive to aggregating their structural dependencies, and the predictions of missing subjects and missing objects can be combined into one task of predicting object entities.

2) Framework Overview: As shown in Figure 2, our proposed DHU-NET model is composed of three modules.

In the time-distributed information learning module, we model each temporal subgraph through an R-GCN [13] and a GRU [14], and we use the embedding output of the previous subgraph as the input of the next to model the sequential dependencies. Then, the entity embeddings and predicate embeddings of the corresponding knowledge subgraphs are obtained. Next, following Conv-TransE [22], we input the entity embeddings and predicate embeddings into a 1-dimensional convolutional network and make the convolution output of the latest subgraph exert self-attention on the convolution outputs of all historical subgraphs. By continuously modeling the newly emerging historical timestamps, DHU-NET can learn accurate distributed embeddings for the newly emerging seen entities. To consider the dynamic unseen entity problem from the global static point of view, we initialize all the entity embeddings in the first historical subgraph (including the trainable seen entities and untrainable dynamic unseen entities) so that the module can finally score all entities for queries.

In the historical information passing module, we assign positive values to the scores of repetitive facts based on historical frequency information, while giving lower negative scores to historical nonrepetitive facts, which contain unseen entities for each specific query.

In the static unseen information passing module, we extract unseen entities at each dynamic query (prediction) timestamp from the global static KG based on the copy mechanism [5]. Specifically, the scores of the dynamic unseen entities with low, negative scores in the historical information passing module are re-zeroed, and they become qualified to make inferential predictions according to the scores obtained by the time-distributed information learning module.

B. Time-Distributed Information Learning Module

The purpose of this module is to learn the distributed representations of k-length historical subgraphs and their different roles in the reasoning process. To obtain accurate representations of the subgraphs in each timestamp, we need to pay attention to the structural information contained in subgraphs and the sequential dependencies between subgraphs.

1) Time-Distributed Entity Embedding Representation: For a TKG $\mathcal{G}$, its temporal subgraphs $\{\mathcal{G}_i | 0 \leq i \leq T-1\}$ can be regarded as static multi-relational graphs. To aggregate the structural information in the entity embeddings, we use a 2-layer R-GCN to model each sequential subgraph $\mathcal{G}_i$. Following the processing strategy of RE-GCN [8], for each node (i.e., entity) in the graph, we aggregate its adjacent information through the message passing architecture. Specifically, each entity in the $(l-1)^{th}$ layer aggregates the adjacent predicate embeddings and entity embeddings according to the different contents of the in-degree edges (i.e., predicates) and then obtains the entity embeddings in the l^{th} layer. Two layers of aggregation are sufficient to capture enough structural information for the entity embeddings. With inverse quadruples, a temporal subgraph is actually an undirected graph, and each predicate of an entity can be regarded as an in-degree edge:

$$\mathbf{h}_o^{(l)} = f\left(\sum_{p\in\mathcal{R}} \frac{1}{c_{o,p}} \sum_{s\in\mathcal{E}_o^p} \mathrm{W}_p^{(l-1)}(\mathbf{h}_s^{(l-1)} + \mathbf{p}^{(l-1)}) + \mathrm{W}_0^{(l-1)}\mathbf{h}_o^{(l-1)}\right) \tag{1}$$

where $\mathbf{h}_o^{(l-1)} \in \mathbb{R}^{N\times d}$ indicates the embeddings of all entities in the $(l-1)^{th}$ layer of the R-GCN and $\mathbf{h}_o^{(l)} \in \mathbb{R}^{N\times d}$ indicates the embeddings of all entities in the l^{th} layer of the R-GCN. $\mathcal{E}_o^p$ indicates the set of entities that are adjacent to the currently computed node o via edge p. $\mathbf{h}_s^{(l-1)}$ and $\mathbf{p}^{(l-1)}$ indicate the embeddings of the adjacent entities and corresponding predicates in the $(l-1)^{th}$ layer of the R-GCN, respectively. $c_{o,p}$ is a normalization constant and represents the size of $\mathcal{E}_o^p$. $f(\cdot)$ represents the adopted reflected rectified linear unit ($RReLU$) activation function. $\mathrm{W}_p^{(l-1)}$ indicates predicate-specific parameters for aggregating the structural features according to different edges. In particular, $\mathrm{W}_0^{(l-1)}$ indicates the parameters for aggregating the self-loop features of all entities. Thus, for a temporal subgraph $\mathcal{G}_t$, through Eq. 1, we can obtain its entity embeddings $\mathbf{E}_t$ at timestamp t.

To model the chronological dependencies between temporal subgraphs, we use the embedding output of the previous subgraph as the input of the R-GCN model at the next timestamp:

$$\mathbf{h}_{o,t}^1 = \mathbf{E}_{t-1} \tag{2}$$

where $\mathbf{E}_{t-1} \in \mathbb{R}^{N\times d}$ represents the output entity embeddings of the $(t-1)^{th}$ temporal subgraph (to be specific, the output of the 2nd layer of the $(t-1)^{th}$ R-GCN). $\mathbf{h}_{o,t}^1$ represents the input entity embeddings for the 1st layer of the t^{th} R-GCN (i.e., the t^{th} temporal subgraph). Thus, for the queries at the t^{th} timestamp, we can obtain the distributed representations of the temporal subgraph sequence with a historical length of k as $\{\mathbf{E}_{t-k},\ ...,\ \mathbf{E}_{t-2},\ \mathbf{E}_{t-1}\}$.

2) Time-Distributed Predicate Embedding Representation: To aggregate the structural information when learning predicate embeddings, we splice the mean value of the entity embeddings of the previous temporal subgraph associated with a predicate with the edge embeddings of the initial subgraph as the input of a GRU cell:

$$\mathbf{R}_{input}^{t} = [\mathbf{p}; mean(\mathbf{E}_{t-1}, E_{p}^{t-1})] \tag{3}$$

where $\mathbf{R}_{input}^{t} \in \mathbb{R}^{2P\times 2d}$ represents the input matrices of the GRU cell at the t^{th} timestamp. $\mathbf{p} \in \mathbb{R}^{2P\times d}$ indicates the predicate embeddings in the input matrices of the first historical subgraph $\mathbf{R}_{init}$. $\mathbf{E}_{t-1} \in \mathbb{R}^{N\times d}$ represents the entity embeddings of the temporal subgraph at the $(t-1)^{th}$ timestamp. E_{p}^{t-1} is a set that records the related entities of specific predicates at the $(t-1)^{th}$ timestamp.

Then, we take the hidden unit output of the GRU for the previous temporal subgraph as the hidden unit input of the GRU for the next temporal subgraph and further model the chronological dependencies of the predicates:

$$\mathbf{R}_{hidden}^{t} = GRU(\mathbf{R}_{input}^{t}, \mathbf{R}_{hidden}^{t-1}) \tag{4}$$

where $\mathbf{R}_{hidden}^{t-1} \in \mathbb{R}^{2P\times d}$ represents the predicate embeddings (including the inverse predicates) at the $(t-1)^{th}$ timestamp and can also be expressed as $\mathbf{R}_{t-1}$. $\mathbf{R}_{hidden}^{t} \in \mathbb{R}^{2P\times d}$ represents the predicate embeddings at the t^{th} timestamp and can also be expressed as $\mathbf{R}_{t}$. Specifically, for k-length historical subgraphs, the hidden unit input for the GRU cell at the 1st historical timestamp $\mathbf{R}_{hidden}^{t-k}$ is $\mathbf{R}_{init}$.

3) Time-Distributed Attention: After obtaining the distributed embeddings of the temporal subgraphs with k-length histories, we learn the different roles of the various timestamps for prediction through the self-attention mechanism [12]. For a query $(s,\ p,\ ?,\ t)$, we first obtain the embeddings $\mathbf{s}$ and $\mathbf{p}$ of its subject and predicate at the t^{th} timestamp via the embedding matrices $\mathbf{E}_t$ and $\mathbf{R}_t$, respectively. Then, following Conv-TransE [22], we splice $\mathbf{s}$ and $\mathbf{p}$ and input them into a 1-dimensional convolutional network; thus, the number of channels in the input is 2. We set the size of the convolution kernel to 3×2 and the number of channels to $c = 50$. To ensure that the shape of the output tensor is consistent with that of the input, we set the padding to 1:

$$\mathbf{x}_{temp} = Conv1\mathrm{D}([\mathbf{s}; \mathbf{p}]) \tag{5}$$

where $[\mathbf{s}; \mathbf{p}] \in \mathbb{R}^{2\times d}$ and $\mathbf{x}_{temp} \in \mathbb{R}^{c\cdot d}$. As shown in Figure 2(a), we convert the dimensionality of $\mathbf{x}_{temp}$ to d via a linear function:

$$\mathbf{y}_{\tau} = \mathrm{W}_1\mathbf{x}_{temp} + b_1 \tag{6}$$

where $\mathbf{y}_{\tau} \in \mathbb{R}^{d}$ represents the intermediate output at a specific historical timestamp τ for the prediction task. $\mathrm{W}_1 \in \mathbb{R}^{d\times c\cdot d}$ and $b_1 \in \mathbb{R}^{d}$ represent the learnable parameters. Thus, for the query $(s,\ p,\ ?,\ t)$, in the k-length historical subgraph sequence, the predicted distributed intermediate results are expressed as $\{\mathbf{y}_{t-k}, \cdots, \mathbf{y}_{t-2}, \mathbf{y}_{t-1}\}$. Considering that the latest historical timestamp plays the largest role in prediction (we prove this in Section IV-D2), we make $\mathbf{y}_{t-1}$ exert learnable attention on all historical timestamps, including the attention to itself. Then, we generate a query vector $\mathbf{Q}$ via $\mathbf{y}_{t-1}$ and key value vectors $\mathbf{K}$ and $\mathbf{V}$ via $\{\mathbf{y}_{t-k}, \cdots, \mathbf{y}_{t-2}, \mathbf{y}_{t-1}\}$:

$$\mathbf{Q} = \mathrm{W}_q\mathbf{y}_{t-1},\ \mathbf{K} = \mathrm{W}_k[\mathbf{y}_{t-k}; \cdots; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}] \tag{7}$$

$$\mathbf{V} = \mathrm{W}_v[\mathbf{y}_{t-k}; \cdots; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}] \tag{8}$$

where $\mathrm{W}_q \in \mathbb{R}^{d_q\times d}$, $\mathrm{W}_k \in \mathbb{R}^{d_k\times d}$, $\mathrm{W}_v \in \mathbb{R}^{d_v\times d}$, $\mathbf{y}_{t-1} \in \mathbb{R}^{d}$ and $[\mathbf{y}_{t-k}; \cdots; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}] \in \mathbb{R}^{k\times d}$. In practice, we set $d_q = d_k = d_v = 64$. Furthermore, the self-attention operation can be represented as follow:

$$\mathrm{Self\_Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Softmax(\frac{\mathbf{Q}\mathbf{K}^{T}}{\sqrt{d_k}}\mathbf{V}) \tag{9}$$

where $\mathbf{Q} \in \mathbb{R}^{d_q}$, $\mathbf{K} \in \mathbb{R}^{d_k\times k}$, and $\mathbf{V} \in \mathbb{R}^{d_v\times k}$. $\sqrt{d_k}$ is a problem-specific scaling factor used to prevent the vanishing gradient problem. W_q, W_k, and W_v are learnable parameters that assign learnable attention weights to each historical timestamp. In practice, we introduce multi-head attention and use 8 heads in our model. Then we use a *feedforward network* (FFN) to introduce deep semantic information:

$$FFN(\mathbf{z}) = \mathrm{W}_2(ReLU(\mathrm{W}_3\mathbf{z})) \tag{10}$$

where $\mathbf{z} \in \mathbb{R}^{d}$ is the output of the multi-head attention mechanism. $\mathrm{W}_2 \in \mathbb{R}^{d_{ff}\times d}$ and $\mathrm{W}_3 \in \mathbb{R}^{d\times d_{ff}}$ are parameters, where d_{ff} is the number of hidden units in the FFN. In practice, we set $d_{ff} = 1024$. In addition, we impose residual connections [25] and layer normalization [26] on the outputs of the multi-head attention mechanism and the FFN, respectively. Then, for the query $(s,\ p,\ ?,\ t)$, following Conv-TransE [22], we perform matrix multiplication on the output $\mathbf{g}$ and the entity embeddings $\mathbf{E}_{t-1}$ at the latest historical timestamp:

$$\mathcal{S}_D = \mathrm{mm}(\mathbf{g}, \mathbf{E}_{t-1}) \tag{11}$$

where $\mathbf{g} \in \mathbb{R}^{d}$ and $\mathbf{E}_{t-1} \in \mathbb{R}^{d\times N}$. Thus, $\mathcal{S}_D \in \mathbb{R}^{N}$ is the score finally obtained by this module.

C. Historical Information Passing Module

This module is built to complement the features of distant histories in terms of frequency.

As shown in Figure 2(b), for a query $(s,\ p,\ ?,\ t)$, the historical repetitive information is represented as the set $\{o_i|(s,p,o_i) \in \mathcal{G}_{\tau}\ and\ \tau < t\}$, and the historical nonrepetitive information is represented as the set $\{o_i|(s,p,o_i) \notin \mathcal{G}_{\tau}\ and\ \tau < t\}$. We use a sequence of sparse matrices with sizes of $N \cdot P \times N$ to store the repetitive and nonrepetitive patterns of each temporal subgraph. Each row in the matrix is represented as a multi-hot vector $\{v_i^{(s,p)} \in \mathbb{R}^{N}|0 \le i \le T-1\}$. If the fact $(s,\ p,\ o_j)$ occurred at the i^{th} historical timestamp, the value in the j^{th} dimension of $v_i^{(s,p)}$ is 1; otherwise, it is 0. Thus, the extracted historical frequency information is represented as follow:

$$\mathrm{V}_t^{(s,p)} = v_0^{(s,p)} + v_1^{(s,p)} + \cdots + v_{t-1}^{(s,p)} \tag{12}$$

where $\mathrm{V}_t^{(s,p)}$ is an N-dimensional vector, with each dimension representing the occurrence frequency of the corresponding

historical entity. For each query $(s,\ p,\ ?,\ t)$, we extract its complete historical frequency information to capture distant historical features. The historical information passing module assigns a lower negative score to a historical nonrepetitive entity, represented as $\overset{-}{\mathrm{V}}_t^{(s,p)}$, and assigns a positive score for a historical repetitive entity according to its frequency statistics:

$$\overset{+}{\mathrm{V}}_t^{(s,p)} = Softmax(\mathrm{V}_t^{(s,p)}) \cdot \delta \tag{13}$$

where $Softmax(\cdot)$ represents the activation function. δ is a problem-specific constant to adjust the role of the historical repetitive information, and we set $\delta = 0.5$. The output score of the historical information passing module is as follow:

$$\mathcal{S}_H = \overset{-}{\mathrm{V}}_t^{(s,p)} + \overset{+}{\mathrm{V}}_t^{(s,p)} \tag{14}$$

D. Static Unseen Information Passing Module

The historical information passing module dynamically assigns unseen entities lower negative scores according to their historical frequency statistics over time. As shown in Figure 2(c), the global static KG consists of all the facts in TKGs without their time information; this is represented as $\mathcal{G}_{static} = \{(s,p,o)|(s,p,o,t) \in \mathcal{G}_t \ and \ t \in [0, T-1]\}$. Although the scale of the unseen entities changes dynamically with time, the global static KG always contains all the factual information at different timestamps.

Similar to the processing strategy in Section III-C, we extract the factual distribution in $\mathcal{G}_{static}$ with a sparse matrix of size $N \cdot P \times N$. A query $(s,\ p,\ ?,\ t)$ corresponds to the $(s \cdot p)^{th}$ row of the sparse matrix, which is represented as a multi-hot vector $\mathrm{S}^{(s,p)} \in \mathbb{R}^N$ that records the occurrence entities in $\mathcal{G}_{static}$. Similarly, the module directly assigns lower negative scores to entities that are not present in $\mathcal{G}_{static}$:

$$\mathcal{S}_U = zero(\mathrm{S}^{(s,p)}) + (-\mathrm{C}) \tag{15}$$

where C is a constant with a large value; 200 is sufficient for the DHU-NET model. Specifically, as shown in Figure 2, if the query's ground-truth entity o does not exist in the history, different from the record in $\mathrm{V}_t^{(s,p)}$, it has a dimension value of 1 in $\mathrm{S}^{(s,p)}$ instead. Then, the entity can avoid obtaining a lower negative score in the historical information passing module, which allows DHU-NET to take the dynamic unseen entities into account during reasoning and finally obtain their accurate representations via continuously updated k-length history subgraph modeling in the time-distributed information learning module.

E. Parameter Learning

For a query, we combine the scores of the three modules during the final prediction. Thus, the time-distributed information (including structural information and sequential dependencies), the historical repetitive information, and the global static unseen information of the TKGs are simultaneously considered:

$$\mathbf{p}(o|s,p,t) = \mathcal{S}_{DHU} = \mathcal{S}_D + \mathcal{S}_H + \mathcal{S}_U \tag{16}$$

where $\mathcal{S}_{DHU}$ is an N-dimensional multi-hot vector, each dimension of which indicates the probability of predicting the corresponding entity as the object. The prediction process can be seen as an N-label classification problem, and we use the cross-entropy loss for this task:

$$\mathcal{L} = -\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{E}} o_i^t \ln \mathbf{p}_t\left(y_j^t \mid s,p,\mathbf{E}_t,\mathbf{R}_t\right) \tag{17}$$

where o_i^t indicates the i^{th} ground-truth object entity in the t^{th} temporal subgraph $\mathcal{G}_t$. $\mathbf{p}_t\left(y_j^t \mid s,p,\mathbf{E}_t,\mathbf{R}_t\right)$ denotes the probability of predicting the j^{th} entity as the object at the t^{th} timestamp.

F. Computational Complexity Analysis

To demonstrate the efficiency of our proposed DHU-NET, we analyze the computational complexity of its three modules. For the time-distributed information learning module, the time complexity of the entity representation, predicate representation, and time-distributed attention is $O(kN)$, $O(kPn)$, and $O(k^2d)$, respectively, where n is the maximum number of entities associated with a particular edge (predicate). For the historical information passing module, its time complexity is $O(T)$. For the static unseen information passing module, its time complexity is $O(1)$. Thus, the computational complexity of the DHU-NET model is $O(k(N + Pn + kd) + T)$.

IV. Experiments

In this section, we evaluate the performance of our proposed DHU-NET model on six popular TKG datasets.

A. Experimental Setup

1) Datasets: We use six well-known TKG datasets for evaluation, namely, YAGO [27], WIKI [24], ICEWS14 [1], ICEWS18 [4], ICEWS05-15 [1], and GDELT [28]. YAGO and WIKI are temporal subgraphs extracted from YAGO3 and Wikipedia, respectively. ICEWS14, ICEWS18, and ICEWS05-15 are extracted from the Integrated Crisis Early Warning System [29]. GDELT records news media information about human societal behaviors. Following previous work [4], [8], [11], the datasets are divided into training/validation/test sets at a ratio of 80%/10%/10%. The detailed statistics of the datasets are presented in Table III.

2) Baseline Methods: We compare the performance of our proposed DHU-NET model with that of multiple static and dynamic reasoning methods. The static methods include DistMult [19], ConvE [20], ComplEx [18], Conv-TransE [22], RotatE [17], and R-GCN [13]. The dynamic reasoning methods include TTransE [2], HyTE [3], and TA-DistMult [1]. Some dynamic methods are available for modeling historical information and obtaining great results, including RE-NET [4], CyGNet [5], xERTE [6], CluSTeR [7], RE-GCN [8], TITer [9], TLogic [10], and CEN [11]. The baseline models are detailed in Section II.

3) Evaluation Metrics: In the experiment, we use the link prediction task to evaluate the effectiveness of the DHU-NET model. We use classic evaluation metrics, including the *mean reciprocal rank* (MRR) and the hits at 1/3/10 (Hits@1/3/10), which all represent the rankings of missing ground-truth

TABLE I
PERFORMANCE (IN PERCENTAGES) ACHIEVED ON ICEWS14, ICEWS05-15, ICEWS18, AND GDELT DATASETS IN TERMS OF RAW METRICS

Method	ICEWS14				ICEWS05-15				ICEWS18				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
DistMult	20.32	6.13	27.59	46.61	19.91	5.63	27.22	47.33	13.86	5.61	15.22	31.26	8.61	3.91	8.27	17.04
ConvE	30.30	21.30	34.42	47.89	31.40	21.56	35.70	50.96	22.81	13.63	25.83	41.43	18.37	11.29	19.36	32.13
ComplEx	22.61	9.88	28.93	47.57	20.26	6.66	26.43	47.31	15.45	8.04	17.19	30.73	9.84	5.17	9.58	18.23
Conv-TransE	31.50	22.46	34.98	50.03	30.28	20.79	33.80	49.95	23.22	14.26	26.13	41.34	19.07	11.85	20.32	33.14
RotatE	25.71	16.41	29.01	45.16	19.01	10.42	21.35	36.92	14.53	6.47	15.78	31.86	3.62	0.52	2.26	8.37
R-GCN	28.03	19.42	31.95	44.83	27.13	18.83	30.41	43.16	15.05	8.13	16.49	29.00	12.17	7.40	12.37	20.63
TTransE	12.86	3.14	15.72	33.65	16.53	5.51	20.77	39.26	8.44	1.85	8.95	22.38	5.53	0.46	4.97	15.37
HyTE	16.78	2.13	24.84	43.94	16.05	6.53	20.20	34.72	7.41	3.10	7.33	16.01	6.69	0.01	7.57	19.06
TA-DistMult	26.22	16.83	29.72	45.23	27.51	17.57	31.46	47.32	16.42	8.60	18.13	32.51	10.34	4.44	10.44	21.63
RE-NET	35.77	25.99	40.10	54.87	36.86	26.24	41.85	57.60	26.17	16.43	29.89	44.37	19.60	12.03	20.56	33.89
CyGNet	34.68	25.35	38.88	53.16	35.46	25.44	40.20	54.47	24.98	15.54	28.58	43.54	18.05	11.13	19.11	31.50
xERTE	32.23	24.29	36.41	48.76	38.07	28.45	43.92	57.62	27.98	19.26	32.43	46.00	17.69	11.81	19.68	30.10
CluSTeR	46.00	33.80	-	71.20	44.60	34.90	-	63.00	32.30	20.60	-	55.90	18.30	11.60	-	31.90
RE-GCN	41.25	30.46	46.26	62.05	45.61	34.43	51.85	66.64	30.79	20.06	35.22	51.77	19.37	12.02	20.74	33.64
TITer	40.90	31.77	45.84	57.67	46.62	36.46	52.29	65.23	28.44	20.06	32.07	44.33	-	-	-	-
TLogic	41.80	31.93	47.23	60.53	45.99	34.49	52.89	67.39	28.41	18.74	32.71	47.97	-	-	-	-
CEN	41.64	31.22	46.55	61.59	49.57	37.86	56.42	71.32	29.70	19.38	33.91	49.90	21.22	13.19	23.04	36.95
DHU-NET	**62.01**	**47.74**	**71.71**	**88.39**	**56.52**	**42.88**	**64.89**	**82.38**	**47.44**	**33.17**	**54.75**	**76.17**	**27.06**	**16.51**	**29.52**	**48.80**

TABLE II
PERFORMANCE (IN PERCENTAGES) ACHIEVED ON YAGO AND WIKI DATASETS IN TERMS OF RAW METRICS

Method	YAGO			WIKI		
	MRR	Hits@3	Hits@10	MRR	Hits@3	Hits@10
DistMult	44.05	49.70	59.94	27.96	32.45	39.51
ConvE	41.22	47.03	59.90	26.03	30.51	39.18
ComplEx	44.09	49.57	59.64	27.69	31.99	38.61
Conv-TransE	46.67	52.22	62.52	30.89	34.30	41.45
RotatE	42.08	46.77	59.39	26.08	31.63	38.51
R-GCN	20.25	24.01	37.30	13.96	15.75	22.05
TTransE	26.10	36.28	47.73	20.66	23.88	33.04
HyTE	14.42	39.73	46.98	25.40	29.16	37.54
TA-DistMult	44.98	50.64	61.11	26.44	31.36	38.97
RE-NET	46.81	52.71	61.93	30.87	33.55	41.27
CyGNet	46.72	52.48	61.52	30.77	33.83	41.19
xERTE	64.29	74.50	87.38	52.85	60.96	71.89
RE-GCN	62.50	70.24	81.55	50.99	57.34	68.50
TITer	64.97	74.80	87.44	**57.36**	63.80	72.52
CEN	63.39	71.68	83.16	51.98	58.96	70.61
DHU-NET	**67.36**	**76.52**	**92.63**	56.18	**64.57**	**82.73**

TABLE III
DETAILS OF THE TKG DATASETS

#Datasets	#Entities	#Predicates	#Training	#Validation	#Test	#Granularity
ICEWS14	6,869	230	74,845	8,514	7,371	24 hours
ICEWS18	23,033	256	373,018	45,995	49,545	24 hours
ICEWS05-15	10,094	251	368,868	46,302	46,159	24 hours
YAGO	10,623	10	161,540	19,523	20,026	1 year
WIKI	12,554	24	539,286	67,538	63,110	1 year
GDELT	7,691	240	1,734,399	238,765	305,241	15 mins

entities in the prediction results. For a query, we report the mean results of the subject entity prediction and object entity prediction tasks. For the YAGO and WIKI datasets, following the prior work related to RE-GCN [8], we only report the MRR, Hits@3, and Hits@10 results.

During the evaluation, although the time-aware filtered setting is widely used, its rationality is seriously affected by the time granularity of TKGs. For example, in the ICEWS series of datasets with a time granularity of 1 day (24 hours), the one-to-many relational query (*Police (Australia)*, *Investigate*, *?*, *2014-12-01*) has a limited number of conflicting entities (*women (Australia)* and *citizen (Australia)*). However, for the YAGO or WIKI dataset with a granularity of 1 year, the one-to-many or many-to-one relational queries might lead to more conflicting entities, such as the query (*Ignác Gyulay*, *hasWonPrice*, *?*, *1814*). General Ignác Gyulay had won the prizes of the Order of Leopold (Austria), the Military Order of Maria Theresa, the Order of Saint Stephen of Hungary, and the Order of the Red Eagle in 1814. The time-aware filtered setting filters all conflicting entities and only keeps the ground-truth entity, thereby avoiding the need to evaluate the processing capabilities of methods for one-to-many or many-to-one queries. Especially for datasets with larger time granularities, the queries have more candidate answers, such as the four awards won by General Ignác Gyulay. Through the above processing approach, the time-aware filtered setting yields better results. Nevertheless, effectively distinguishing and predicting the conflicting facts of a query should be one of the abilities of a TKG reasoning model and thus should be considered in the model performance evaluation. Without loss of generality, we do not perform the filtered operation and report the results obtained under the raw setting instead.

4) Implementation Details: We implemented our DHU-NET model in PyTorch and trained the model on a GPU (Tesla V100). We configured the parameters based on the MRR performance achieved by the model on the validation set and set the training epoch for all datasets to 250 to ensure model convergence. In addition to the parameters introduced in Section III, the historical length k of the modeling sequence is set to 3 for all the datasets. For the R-GCN, we set the dropout rate of each layer to 0.2. For Conv1D, the number of kernels is set to 50, and the dropout rate is also set to 0.2. For the multi-head attention mechanism, the number of layers is set to 1, which is sufficient for the current task. We use the Adam optimizer for parameter learning, and the learning rate is set to 0.001. The batch size is set to the size of each timestamp for both training and testing. For the static reasoning methods, the time dimension is removed from all the TKG datasets. We

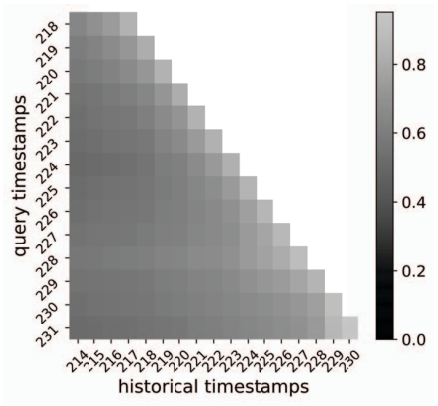


Fig. 3. Statistics regarding the ground-truth facts in the WIKI dataset at different historical timestamps

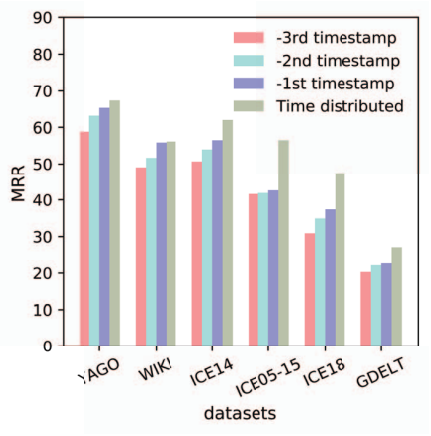


Fig. 4. Study on the roles of different historical timestamps in all the datasets

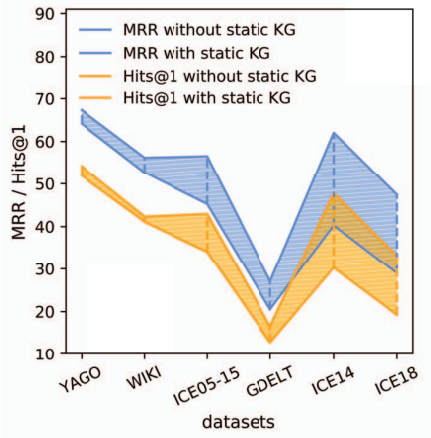


Fig. 5. Study on the global static unseen information in all the datasets

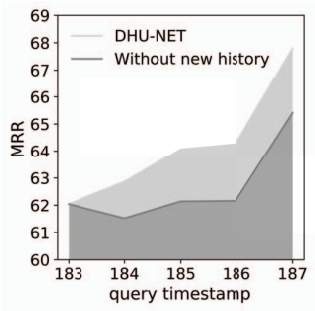


Fig. 6. Study on the newly emerging historical information in the YAGO dataset

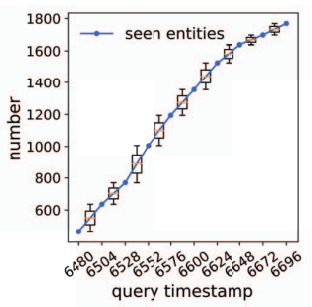


Fig. 7. Statistics on the seen entities in different query timestamp periods of the ICEWS18 dataset

TABLE IV
ABLATION STUDY RESULTS (IN PERCENTAGES) OBTAINED ON YAGO, WIKI, ICEWS14, AND ICEWS18 DATASETS

Datasets	YAGO	WIKI	ICEWS14	ICEWS18
Time **D**istributed information learning module	60.76	49.73	34.75	25.53
Historical information passing module	61.75	43.93	35.42	24.89
Static **U**nseen information passing module	58.37	45.20	56.54	39.85
DHU-NET	**67.36**	**56.18**	**62.00**	**47.44**

TABLE V
STATISTICS (IN PERCENTAGES) OF THE DYNAMIC UNSEEN ENTITIES IN ALL THE DATASETS

Datasets	YAGO	WIKI	ICEWS14	ICEWS05-15	ICEWS18	GDELT
Whole set	9.96*	4.78	55.38	36.14	**56.84**	42.59
Test set	7.27*	12.96	47.63	31.61	**49.57**	35.07

set the embedding dimensionality to 200 to be consistent with the experimental settings in RE-GCN [8]. Some of the baseline results are adopted from [8].

For the important xERTE [6], RE-GCN [8], TITer [9], TLogic [10], and CEN [11] baseline works, we use their default parameters and replicate the results obtained under raw settings with their open codes. For CluSTeR [7], we report the results presented in their paper because the model is not open source. For CEN [11], we report the results obtained under the online setting. For TITer [9], their codes crash when running on the largest GDELT dataset. TLogic [10] cannot process datasets other than the ICEWS series because it needs the content references of entities and predicates. Moreover, for the ICEWS14 and ICEWS05-15 datasets, since we adopt a different dataset splitting strategy than that of TLogic [10], we reorganize these datasets into the input format of the TLogic model for a consistent experimental configuration.

B. Results of TKG Reasoning

In this section, we compare the performance of DHU-NET with that of static and dynamic reasoning methods based on the TKG link prediction task.

Table I and Table II show the best results in boldface and the second-best results as underlined. The performance of DHU-NET is much better than that of the static reasoning methods because the static models completely ignore the time information of TKGs. However, some dynamic methods, such as TTransE [2] and HyTE [3], perform even worse than the static methods because they only focus on the embedding of time information while ignoring the evolutionary patterns of historical subgraphs. Compared with the RE-NET [4], CyGNet [5], xERTE [6], CluSTeR [7], RE-GCN [8], TITer [9], and TLogic [10] history-based methods, DHU-NET still performs better because these approaches all ignore the time variability problem of temporal reasoning.

Although CEN [11] tries to address the challenge of newly emerging items through the online training strategy, it ignores the different roles that different historical timestamps play in the prediction process. DHU-NET performs better than methods that addressing the challenge of unseen entities, such as xERTE [6] and TITer [9]. This is because both of these competitors believe that history is constant and ignore the time variability feature of histories, thus failing to solve the problem regarding dynamic unseen entities. We note that DHU-NET

performs slightly worse than TITer [9] in terms of the MRR metric, as described in CEN [11], because TITer [9] adopts reinforcement learning and ranks missing ground-truth entities based on candidate sets that are much smaller than the size of the entity set, which likely results in better performance.

C. Ablation Study

In this section, we conduct an ablation test based on the YAGO, WIKI, ICEWS14, and ICEWS18 datasets. We only report the most representative MRR metric.

As shown in Table IV, similarly, the best results are bolded, and the second-best results are underlined. It can be observed that the evolutionary time-distributed structural information and the global static unseen information play certain roles in the performance of DHU-NET. The module with the most critical role differs for different datasets. For the YAGO dataset with more historical repetitive information, the frequency statistics for repetitive entities throughout history are prominent; thus, the historical information passing module tends to contribute more functionality. However, when less repetitive information and unseen information are involved (such as in the WIKI dataset), the role of the time-distributed information learning module is highlighted. When the dataset contains more historical unseen entities (such as the ICEW14 and ICEWS18 datasets), the static unseen information passing module plays a more important role. In particular, the performance of the complete DHU-NET model is better than that of any single module, which demonstrates the effectiveness of considering the time-distributed information, the historical repetitive information, and the global static unseen information simultaneously for TKG reasoning.

D. On the Time Variability

In this section, we study the ability of DHU-NET to resolve the time variability problem in terms of two aspects.

1) On the Newly Emerging Historical Information: As Figure 6 shows, we use each query timestamp (except for the 188^{th} timestamp with only one fact) of the YAGO dataset as a separate research object and report the MRR metrics yielded by DHU-NET with and without newly emerging historical information. The dark gray area only uses the training set and validation set as a fixed historical scope. Instead, the light gray area represents the proposed DHU-NET. We do not add static unseen information to intuitively reflect the effect of the new historical information. At the 183^{th} timestamp next to the validation set, the historical ranges of the two comparison objects are consistent, as is the corresponding MRR metric. With the development of time, previous models are limited to a fixed historical range because they ignore the time variability phenomena. However, DHU-NET continuously updates the latest historical range and models the emerging historical information in time, thereby achieving better performance for subsequent query (prediction) timestamps.

2) On the Distinct Roles of Historical Timestamps: We define the numbers of ground-truth facts contained in different historical timestamps as the ground-truth effects of these historical timestamps on a query. As Figure 3 shows, we calculate statistics regarding the ground-truth effects of different historical timestamps on the query timestamps, where a number along the horizontal and vertical axes represents a year. It is observed that different historical timestamps play different roles in the prediction.

As shown in Figure 4, we further study the distinct roles of the historical timestamps on all six datasets. Because the historical length of the DHU-NET model is set to 3, we report the MRR metrics for three separate latest historical timestamps and the time-distributed strategy. In addition, -1st, -2nd, and -3rd represent the last, second, and third historical timestamps, respectively. It is observed that the contribution of the historical timestamps to the prediction process decreases as the distance from the query timestamp increases. Accordingly, the latest historical timestamps are the most valuable, which also demonstrates the importance of keeping the newly emerging historical information in mind. However, the performance of the time-distributed strategy is better than that of any single historical timestamp. This proves that although the latest historical timestamps are generally more powerful, different historical timestamps still hold different pieces of information that are useful for prediction, and the time-distributed strategy successfully captures their distributed features.

E. On the Global Static Unseen Information

In this section, we study the proposed DHU-NET model's ability to resolve dynamic unseen entity issues by capturing the global static unseen information.

We first study the scales of the seen entities for different query timestamps in the ICEWS18 dataset. We start with the first query (prediction) timestamp and count the number of seen entities associated with the queries. If a query fact appears in the scale-varying history, it can obtain useful information to complete the missing subject or object. Then, the subject and object entities are seen in the current time period. As Figure 7 shows, we use the horizontal axis to represent a certain time period (each number represents 24 hours a day), and the vertical axis to represent the number of seen entities. Thus, the seen entity set accumulates and increases over time. Therefore, setting the unseen entity set to a fixed value will result in inaccurate representations of the seen entities.

We count the proportions of queries facing the dynamic unseen entity dilemma on all datasets. As shown in Table V, the largest value is bolded, the second-largest value is underlined, the smallest value is starred, and the 0^{th} timestamp is not considered when calculating the statistics of the whole dataset. Then as shown in Figure 5, we sort the datasets from small to large in terms of their dynamic unseen entity scales and compare the performances achieved by the DHU-NET model with and without the static unseen information passing module. DHU-NET yields improvements on all datasets due to the resolution of the dynamic unseen entity issue. In particular, the extents to which different datasets are improved correspond to the scales of the dynamic unseen entities. In addition, on the largest GDELT dataset, although sizable dynamic unseen

entities remain, the improvement achieved by the model is still limited due to the difficulty inherent in handling this dataset.

V. Conclusions

In this paper, we propose DHU-NET to address the challenges of time variability and dynamic unseen entities in TKG reasoning. DHU-NET comprehensively models the distributed roles of each historical timestamp via time-distributed representation learning. Besides, it accounts for predictions by extracting dynamic unseen entities from the global static KG. Our extensive experiments demonstrate its significant improvement over baseline models.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0108501, National Natural Science Foundation of China under Grants No.62072203, and Australian Research Council Under Grants DP22010371, LE220100078.

References

[1] Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821, 2018.

[2] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. Towards time-aware knowledge graph completion. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724, 2016.

[3] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011, 2018.

[4] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, 2020.

[5] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4732–4740, 2021.

[6] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proceedings of International Conference on Learning Representations*, 2020.

[7] Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4732–4743, 2021.

[8] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417, 2021.

[9] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319, 2021.

[10] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4120–4127, 2022.

[11] Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. Complex evolutional pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 290–296, 2022.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceeding of Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[13] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, pages 593–607. Springer, 2018.

[14] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 2787–2795, 2013.

[16] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[17] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of International Conference on Learning Representations*, 2018.

[18] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of International Conference on Machine Learning*, pages 2071–2080. PMLR, 2016.

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

[20] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 1811–1818, 2018.

[21] Tu Dinh Nguyen Dai Quoc Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–333, 2018.

[22] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 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, 2019.

[23] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *Proceedings of International Conference on Learning Representations*, 2019.

[24] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *Proceedings of Companion Proceedings of the The Web Conference*, pages 1771–1776, 2018.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[26] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[27] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2014.

[28] Kalev Leetaru and Philip A. Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

[29] Elizabeth Boschee, Jennifer Lautenschlager, Sean OBrien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data. *Harvard Dataverse*, 12, 2015.