# DA-Net: Distributed Attention Network for Temporal Knowledge Graph Reasoning

Anonymous authors
Paper under double-blind review

## ABSTRACT

Predicting future events in dynamic knowledge graphs has attracted significant attention. Existing work models the historical information in a holistic way, which achieves satisfactory performance. However, in real-world scenarios, the influence of historical information on future events is changing over time. Therefore, it is difficult to distinguish the historical information of different roles by invariably embedding historical entities with simple vector stacking. Furthermore, it is laborious to explicitly learn a distributed representation of each historical repetitive fact at different timestamps. This poses a challenge to the widely adopted codec-based architectures. In this paper, we propose a novel model for predicting future events, namely Distributed Attention Network (DA-Net). Rather than obtaining the fixed representations of historical events, DA-Net attempts to learn the distributed attention of future events on repetitive facts at different historical timestamps inspired by human cognitive theory. In human cognitive theory, when humans make a decision, similar historical events are replayed during memory recall. Based on memory, the original intention is adjusted according to their recent knowledge developments, making the action more reasonable to the context. Experiments on four benchmark datasets demonstrate a substantial improvement of DA-Net on multiple evaluation metrics.

## KEYWORDS

Knowledge graphs, temporal reasoning, cognitive modeling

## 1 INTRODUCTION

Knowledge graphs (KGs) are generated by extracting facts and events from occurrences in the real world. Traditional KGs represent information in a static graph; however, most real-world facts are dynamic. Therefore, temporal knowledge graphs (TKGs), which represent each fact as a quadruple (*subject, predicate, object, timestamp*), have been proposed to address these limitations.

The reasoning over TKGs is to predict events (facts) at the different timestamps. Previous work has attempted to learn temporal historical information to predict future events, achieving superior performance. For example, CyGNet [33] uses the abstractive summarization copy mechanism to model the previous events, and RE-GCN [19] uses the recurrent relation-aware graph convolutional network (GCN) and static information to model the historical events. However, in existing work, historical information has been modeled holistically, ignoring the dynamic evolution of events at different timestamps. For example, the prediction query of a quadruple is $(s, p, ?, t_n)$, and historical repetitive facts are represented by the set $\{(s, p, o_j, t_i)|t_0 \leq t_i \leq t_{n-1}\}$, where $\{t_i\}$ represents the historical timestamps before $t_n$ and $\{o_j\}$ represents all historical events. Existing work has mostly aimed to learn the fixed representations for the historical repetitive entities $\{o_j\}$ and neglect the temporal evolution of events, i.e., the associated attention weight. However, events at various timestamps contribute differently to the reasoning, which we discuss in Section 4.2; this phenomenon is also referred to as the problem of time-variability [17]. Consider the query (*The man, Dinner at, ?, New Year's Eve*) shown in Figure 1. According to tradition, the historical repetitive locations (objectives) include 1) his home, 2) relatives' homes, and 3) restaurants. The choice of historical repetitive events (new year gatherings) is considerably influenced by event occurrences over time, as well as the surrounding context.

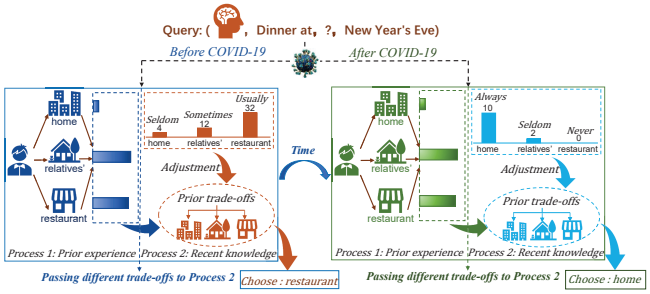


**Figure 1: An illustration of dual-process reasoning. The two boxes indicate human decision processes before and after the outbreak of COVID-19.**

However, existing work has faced challenges when treating the historical repetitive facts as dynamic distributions. First, existing work has obtained definite representations of historical entities, also known as the encoder in codec-based architectures; thus, the distributed features of historical entities $\{o_j\}$ at different timestamps are compressed into invariant vectors. Therefore, previous models have had difficulty in capturing the historical variations of repetitive events over time. Second, although historical entities $\{o_j\}$ at different timestamps play different roles in predicting future events, it is both time and space consuming to obtain a distributed representation at each historical timestamp. To solve these problems, we consider cognitive science theory.

How do humans make future decisions? According to dual-process theory [24] and studies on concentrated and distributed attention [3], in the first process, humans filter out appropriate judgments in their memory space based on prior experience, which is often derived from tradition and preserves distant features of historical facts; and then in the second process, humans use recent developments of their knowledge to adjust their judgments. For instance, as shown in Figure 1, when a man considers where to celebrate on New Year's Eve, he first recall what he usually did in previous years. In years before the outbreak of COVID-19, he likely visited

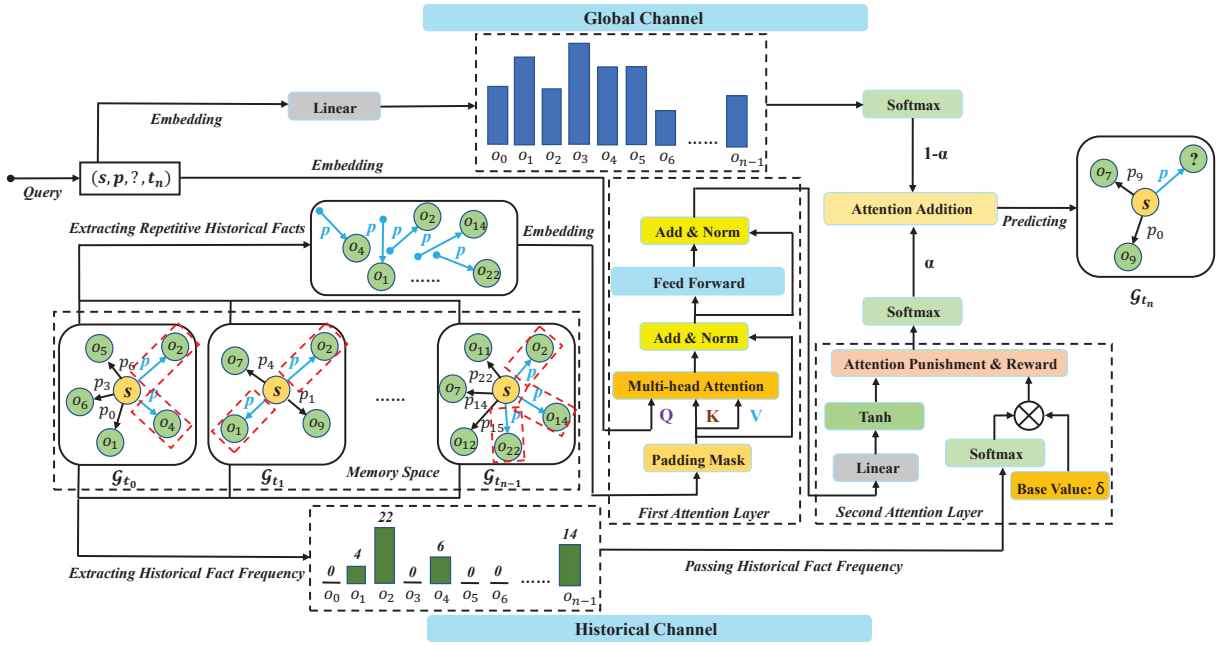


**Figure 2: The framework of our DA-Net model. The blue bars indicate the probability predicted by the global channel for a query $(s, p, ?, t_n)$. In the historical channel, the red dotted squares in the memory space represent historical predicate-object pairs. The green bars indicate the frequency of the historical facts. The two layers include distributed attention to learn prior experience and to utilize recent knowledge developments, respectively.**

their relatives or went to a restaurant, as informed by the automatic thinking phrase (Process I), because it is a tradition to have family reunions with senior family members at their homes or at pre-booked restaurants on New Year's Eve. However, such choices are sometimes impacted by unpredictable events, such as COVID-19. In this situation, in the second decision making process, the choice of having a family reunion must be adjusted, either by staying at home or having small group gatherings, reflecting the newly enforced COVID restrictions (e.g., restaurant capacity) and safety considerations. Therefore, as a result of this dual-process mechanism, humans' attention to historical events during different periods changes at various timestamps. To make better decisions, humans use distributed attention rather than concentrated attention to emphasize key information in their memory at different time periods.

In this paper, we simulate the abovementioned dual-process mechanism to model the distribution of historical information at different timestamps and propose a new method for temporal knowledge graph reasoning known as DA-Net (**D**istributed **A**ttention **Net**work). To address the first challenge, we design distributed attention mechanisms to learn the variable distributions of historical repetitive events by modeling the attention of each repetitive fact in different subgraphs rather than by learning only their representations. To address the second challenge, we develop innovative frameworks for learning distributed representations of historical information while consuming limited computational resources. As shown in Figure 2, in the first attention layer, we extract the historical repetitive facts of each event in the current subgraph, and then uniformly learn the attention of these repetitive facts. By training the model in chronological order beginning with the 1st timestamp, we ensure that the learned attention preserves traditional historical features. In the second attention layer, we consider the influence of unexpected emergencies on the prediction. At this stage, recent knowledge developments are critical for adjusting decisions. In Section 4.5, we prove the important role of shallow memory, and the changes of the rule-based statistical information in shallow memory adjust the decision-making process, which we refer to as "knowledge sensitivity" and our proposed DA-Net successfully captures it. We extract the historical frequency information of each fact in the current subgraph, which changes according to the humans' recent knowledge developments over time and dynamically assign attention rewards and punishments to the facts based on changes in their historical frequency. In Section 3.5, we demonstrate that the computational complexity of the proposed framework is linearly related to the size of the datasets.

The main contributions of this work are as follows:

- We demonstrate the time-variability problem during TKG reasoning at the data level for the first time, showing that the representations of different subgraphs at various historical timestamps play distinct roles in predicting future events.
- In contrast to conventional codec-based methods, we propose a novel network for predicting future events in TKGs that imitates human decision-making processes, modeling dynamic distributions of historical repetitive facts via distributed attention in a dual process.

- Based on cognitive theory modeling, we propose the concept of memory space and study the effect of memory space depth on model performance, proving that DA-Net successfully captures shallow memory features such as knowledge sensitivity.
- Extensive experiments on four public TKG datasets are conducted. The improvement on almost all evaluation metrics demonstrates the effectiveness of our method for predicting future events.

The remainder of this paper is organized as follows. Related work is introduced in Section 2, including existing static and dynamic reasoning methods for TKGs. The proposed model is detailed in Section 3. Besides, the experiments and analyses are presented in Section 4, followed by the conclusion in the final section.

## 2 RELATED WORK

Existing TKG reasoning approaches are mainly divided into two kinds by data modeling: static inference and dynamic inference.

### 2.1 Static Reasoning Methods

Before temporal dynamics are investigated, much research is conducted on static reasoning methods. Embedding-based models, such as TransE [4], RotatE [26], and ConvE [8], map predicates and entities to low-dimensional vector spaces. In addition, matrix decomposition-based methods, including DistMult [32] and TuckER [2], learn the embedding vectors of entities and predicates by outputting a core tensor. Relation path reasoning uses path information in graph structures to model complex relation (predicate) paths. Among them, the reinforcement learning-based reasoning methods express the process of finding paths between entities as sequential decisions, especially the Markov Decision Process (MDP), such as DeepPath [31] and MINERVA [6]. Reasoning methods based on graph neural networks, including R-GCN [23] and Comp-GCN [28], apply graph algorithms to knowledge graphs. However, these methods model knowledge graphs in a static manner, neglecting the dynamic evolution of the graph, which differs from real-world situations and leads to deviations in the predictions.

### 2.2 Dynamic Reasoning Methods

We focus on dynamic reasoning methods for TKGs, which is currently a popular research topic. TTransE [13] models the time-predicate sequence for inference. Deriving [15] embeds the time and predicates into low-dimensional vector space. HyTE [7] makes projection of predicates and entities onto the hyperplane of particular timestamps. TeMP [30] completes TKGs by simulating the information of the multi-hop structure and the temporal facts of neighboring timestamps. DySAT [22] calculates entity representations by combining the self-attention along two dimensions of the neighboring structure and the temporal variations. Recent work has focused on predicting future events in TKGs. RE-NET [14] models the occurrence of facts as a conditional probability distribution based on the subgraphs of previous time series. CyGNet [33] treats the historical entities that appear in previous timestamps as abstract summaries, and predicts future facts based on them. The HIP network [12] transmits historical information from the perspective of time, structure and repetition to make predictions. xERTE [10] generates query subgraphs with certain hop numbers by constructing inference graphs. CluSTeR [18] and TITer [25] both use reinforcement learning to determine evolutionary patterns in query paths. RE-GCN [19] learns the entity representations containing evolutionary information by modeling the sub-graph sequences of recent timestamps. TLogic [20] constrains the query path based on temporal logic rules extracted from temporal random walks. However, in the above work, the problem of time-variability during the temporal reasoning process is ignored. CEN [17] addresses this issue in an online learning setting; however, this method can fine-tune only representation vectors with finite lengths.

## 3 METHOD

In this section, we introduce the proposed DA-Net method. We first describe the notations and definitions. Then, we present the model architecture and the two channels of the model. In addition, we also discuss the training strategy and analyze the computational complexity.

### 3.1 Definitions and Model Architecture

*3.1.1 Notations and Definitions.* In a TKG, let $\mathcal{E}$ be the entity set, $\mathcal{R}$ be the predicate set, $\mathcal{T}$ be the timestamp set, $N$ be the size of $\mathcal{E}$, $P$ be the size of $\mathcal{R}$ and $T$ be the size of $\mathcal{T}$. We divide the TKG into a series of sequential subgraphs $\mathcal{G} = \{\mathcal{G}_0, \mathcal{G}_1, ..., \mathcal{G}_{T-1}\}$ to simulate the evolution over time. $\mathcal{G}$ is composed of the facts containing time information, such as $(s, p, o, t_n)$, where $\{s, o\} \in \mathcal{E}$, $p \in \mathcal{R}$, and $t_n \in \mathcal{T}$. $\mathbf{s}$, $\mathbf{p}$, $\mathbf{o}$, and $\mathbf{t_n}$ are the embedding representations of $s$, $p$, $o$, and $t_n$, respectively, and $d$ is the embedding dimension. Future event prediction on the TKG is to predict the missing object entity, $(s, p, ?, t_n)$, or the missing subject entity, $(?, p, o, t_n)$, according to previous temporal subgraphs $\{\mathcal{G}_t | t < t_n\}$ with historical information, where $t_n$ is a future timestamp.

As shown in Figure 2, for a query $(s, p, ?, t_n)$ at timestamp $t_n$, the memory space is defined as a sequence of multi-hot vectors generated according to temporal static subgraphs, $\{m_{t_i}^{(s,p)} \in \mathbb{R}^N | t_0 \leq t_i \leq t_{n-1}\}$. The value in the $i$-th dimension of $m_{t_i}^{(s,p)}$ is 1 if the fact $(s, p, o_i)$ occurred at timestamp $t_i$. To predict the future object entity in $(s, p, ?, t_n)$, the historical information extracted from the memory space is represented as:

$$\mathbf{M}_{t_n}^{(s,p)} = m_{t_0}^{(s,p)} + m_{t_1}^{(s,p)} + ... + m_{t_{n-1}}^{(s,p)}, \tag{1}$$

where $\mathbf{M}_{t_n}^{(s,p)}$ is an N-dimensional vector, with each dimension representing the occurrence frequency of the corresponding historical entity, thus imitating memory in the human brain. We assume that for all facts in the dataset, the memory space starts with the 1st timestamp.

*3.1.2 Model Architecture.* Figure 2 shows an outline of our proposed framework, which is composed of two channels. Specifically, the global channel is responsible for learning the global information according to the original query, which ensures that the event prediction model does not rely too much on historical information in the historical channel. The historical channel uses two attention layers to mimic how humans dynamically utilize information

in the memory space and assigns distributed attention to information at different timestamps. In the historical channel, the historical repetitive facts and their frequencies are firstly extracted for the first and second attention layers, respectively. This information is obtained from memory space, which consists of a sequence of temporal static subgraphs divided by timestamps. Then through the self-attention mechanism [29], the first attention layer (for prior experience) simulates the efficient manner in which humans learn the traditional attention weights of historical facts. The second attention layer (for recent knowledge development) assigns reward or punishment scores to the historical facts (including both repetitive and nonrepetitive facts) according to recent changes in their occurrence frequency. The final prediction is generated by combining the attention of these two channels.

## 3.2 Historical Channel

The motivation to introduce the historical channel is to imitate human judgement processing, which includes two steps. First, people recall similar historical facts from their memory and assign the original attention to them according to prior experience; then, humans use recent knowledge developments to adjust and select a proper decision. Similarly, in an event prediction task, if a person needs to determine the answer of an unknown query $(s, p, ?, t_n)$, he first searches his memory space for similar situations, that is, for historical repetitive facts, denoted by $\{(s, p, o_0, t_0)$,...,$(s, p, o_i, t_i)$,...,$(s, p, o_{n-1}, t_{n-1})\}$, where $t_i \in [t_0, t_{n-1}]$. After collecting historical repetitive facts, he will decide which historical repetitive fact is the most valuable for predicting future events. In the historical channel, we use two attention layers to simulate this process. The details are presented below.

*3.2.1 First Attention Layer.* We further represent historical repetitive facts as $\{(s, p, o_0), ..., (s, p, o_i), ..., (s, p, o_{n-1})\}$, where $\{(p, o_i)|i \in [0, n-1]\}$ is the set of historical predicate-object pairs. In practice, we calculate a batch of queries, and generate the matrix $Q$ by concatenating $\mathbf{s}$ and $\mathbf{p}$:

$$Q = \mathbf{W}_q[\mathbf{s}, \mathbf{p}], \tag{2}$$

where $\mathbf{W}_q \in \mathbb{R}^{d_q \times 2d}$, and $d_q$ is the embedding dimension of the matrix $Q$. Furthermore, $[\mathbf{s}, \mathbf{p}] \in \mathbb{R}^{2d \times 1 \times B}$, where $B$ is the number of queries in a batch. To the best of our knowledge, we are the first to model the historical repetitive facts of each query as a sequence. However, we encounter the issue of inconsistent sequence length in sequence batches. We solve this problem by using the padding mask [29]. Then we generate matrices $K$ and $V$ by concatenating $\mathbf{p}$ and $\mathbf{o_i}$:

$$K = \mathbf{W}_k[\mathbf{p}, \mathbf{o_i}], V = \mathbf{W}_v[\mathbf{p}, \mathbf{o_i}], \tag{3}$$

where, $\mathbf{W}_k \in \mathbb{R}^{d_k \times 2d}$ and $\mathbf{W}_v \in \mathbb{R}^{d_v \times 2d}$. In our model, we set $d_q = d_k = d_v = 64$. Furthermore, $[\mathbf{p}, \mathbf{o_i}] \in \mathbb{R}^{2d \times S \times B}$, where $B$ is the size of a batch, $S$ is the number of historical repetitive facts in each sequence. We define the self-attention as:

$$\text{Self\_Attention}(Q, K, V) = \text{softmax}\left(\frac{\mathbf{W}_q[\mathbf{s}, \mathbf{p}](\mathbf{W}_k[\mathbf{p}, \mathbf{o_i}])^T}{\sqrt{d_k}}\right)\mathbf{W}_v[\mathbf{p}, \mathbf{o_i}], \tag{4}$$

where $\frac{1}{\sqrt{d_k}}$ is the scaling factor, which is to deal with the effect when the softmax function reaches an area of a minimal gradient. $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are trainable parameters. To predict future events, matrices $\mathbf{W}_k$ and $\mathbf{W}_v$ assign unique coefficients to each historical repetitive fact. Thus, our model can assign different attention weights to various historical repetitive facts by learning separately. The introduction of multi-head attention allows the prediction to consider the importance of historical repetitive facts from multiple perspectives (subspaces). The heads denote the number of matrices $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$, and we use 8 heads in our model. Then we introduce a feed-forward network (FFN) with $d_{ff}$=2048 hidden units:

$$FFN(\mathbf{x}) = \mathbf{W}_1(RELU(\mathbf{W}_2\mathbf{x})), \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^{2d \times B}$ is the output of the multi-head attention operation, and $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times 2d}$, $\mathbf{W}_1 \in \mathbb{R}^{2d \times d_{ff}}$. After the layer of the FFN, we introduce residual connections [11] and layer normalization [1]. The output of the first attention layer is $\mathbf{y}$, with $\mathbf{y} \in \mathbb{R}^{2d \times B}$.

*3.2.2 Second Attention Layer.* The output $\mathbf{y}$ includes the attention information of historical repetitive facts in query $(s, p, ?, t_n)$. The timestamp of a future event is necessary for predicting future events; thus, we concatenate $\mathbf{y}$ and $\mathbf{t_n}$ and convert the result to an N-dimensional multi-hot vector through a linear layer:

$$s_t = tanh(\mathbf{W}_t[\mathbf{y}, \mathbf{t_n}] + \mathbf{b}_t), \tag{6}$$

where $\mathbf{W}_t \in \mathbb{R}^{N \times 3d}$, $\mathbf{b}_t \in \mathbb{R}^{N \times 1}$, and the $tanh$ layer allows $s_t$ range between $(-1, 1)$ (with a gap of 2). As shown in Figure 2, the attention punishment layer changes the index values of facts that have not occurred in history (corresponding to the dimensions with a value of zero in $\mathbf{M}_{t_n}^{(s,p)}$) to more negative numbers, denoted as $\overset{pu(s,p)}{\mathbf{M}}_{t_n}$. The attention reward layer presents the corresponding facts with rewards (denoted as $\overset{re(s,p)}{\mathbf{M}}_{t_n}$) based on the base value $\delta$ according to the frequency of the historical repetitive facts (denoted as $\overset{+(s,p)}{\mathbf{M}}_{t_n}$, corresponding to the values of nonzero dimensions in $\mathbf{M}_{t_n}^{(s,p)}$):

$$\overset{re(s,p)}{\mathbf{M}}_{t_n} = softmax(\overset{+(s,p)}{\mathbf{M}}_{t_n}) * \delta, \tag{7}$$

$$s_h = softmax(s_t + \overset{pu(s,p)}{\mathbf{M}}_{t_n} + \overset{re(s,p)}{\mathbf{M}}_{t_n}), \tag{8}$$

where the base value $\delta$ is chosen as the gap (i.e., 2) to ensure that both attention layers work. From the perspective of cognition, humans can either selectively use their memorized knowledge through experiential learning or adjust their learned preferences according to recent knowledge developments. This dual process is both objective and effective, achieving distributed attention to repetitive facts at different historical timestamps.

## 3.3 Global Channel

For the query $(s, p, ?, t_n)$, the global channel captures the original query information and generates a prediction of the object entity from a global perspective. The prediction of the global channel prevent one-sided judgements or over-reliance on historical information. The global channel first concatenates $\mathbf{s}$, $\mathbf{p}$ and $\mathbf{t_n}$ in the query and then converts the vector to size $N$ (the size of entity set $\mathcal{E}$).

Finally, we normalize the output multi-hot vector with a softmax function to obtain the result of the global channel:

$$s_g = softmax(\mathbf{W}_g[\mathbf{s}, \mathbf{p}, \mathbf{t_n}] + \mathbf{b}_g), \tag{9}$$

where $\mathbf{W}_g \in \mathbb{R}^{N\times 3d}$ and $\mathbf{b}_g \in \mathbb{R}^{N\times 1}$. The global channel outputs the entity corresponding to the maximum value in $s_g$.

## 3.4 Training Strategy

The final prediction of query $(s, p, ?, t_n)$ is obtained by combining the attention of the two channels:

$$\mathbf{p}(o|s, p, t_n) =$$
$$Attention\_Addition(s, p, t_n) = \alpha * s_h + (1 - \alpha) * s_g, \tag{10}$$

$$o = argmax_{o\in\mathcal{E}}(\mathbf{p}(o|s, p, t_n)), \tag{11}$$

where $0 \le \alpha \le 1$, and $Attention\_Addition(s, p, t_n)$ is an N-dimensional multi-hot vector, with each dimension indicating the probability of predicting the corresponding entity as the object.

To predict future events, our model first examines the historical repetitive facts in memory space, which increase with increasing time, and are passed to the validation and test sets. Then the global information and historical information are learned through the global and historical channels, respectively. We divide the data into batches according to timestamps to extract historical information from memory space. We treat the prediction process as a multiclass classification task with a classifier number $N$ and use the cross-entropy loss function for training:

$$\mathcal{L} = -\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{E}} o_i^t \ln \mathbf{p}\left(y_i^j \mid s, p, t_n\right), \tag{12}$$

where $o_i^t$ represents the i-th ground-truth object entity in the t-th timestamp subgraph $G_t$. $\mathbf{p}\left(y_i^j \mid s, p, t_n\right)$ denotes the probability that $o_i^t$ is the j-th object entity in the entity set $\mathcal{E}$.

## 3.5 Computational Complexity Analysis

The main calculation cost of our model is due to the multi-head attention operation in the first attention layer. We prove that the computational complexity of DA-Net is linearly related to the size of datasets by analyzing all the components of the model.

For each query $(s, p, ?, t_n)$, there are $h$ heads in multi-head attention. In our model, $d_q{=}d_k{=}d_v{=}\frac{2d}{kh}$, where $k$ is used to explain the case where the dimension of the embedding vector is not an integer multiple of $h$, and $k$ and $h$ are constants. We represent the size of the entity set $\mathcal{E}$ as $N$ and the maximum sequence length of the historical repetitive facts as $n$. For a dataset with $D$ samples, similar to previous history-based models [14, 33], we adopt the idea of space for time and use the sparse matrix method to extract and store the historical repetitive facts. It processes all the facts in the dataset through a loop traversal with a computational complexity of $O(D)$. For each of the multiple heads, the computational complexity of the input linear mapping (Eq. 2) to $[s, p]$ is $O(d^2)$. Similarly, the computational complexity of the input linear mapping (Eq. 3) to $[p, o_i]$ is $O(nd^2)$. We also consider scaled dot-product attention (Eq. 4), which has a computational complexity of $O(nd^2)$. Finally, the computational complexity of feed-forward network (Eq. 5) is $O(d^2)$ ($d_{ff}$ is fixed in our model). Therefore, the computational complexity of the first attention layer is $O(nd^2)$. Similarly, the computational complexities of the second attention layer and the global channel are both $O(Nd^2)$. The total computational complexity of DA-Net is thus $O((N + n)d^2)$. Therefore, the computational complexity of the entire training and testing process is $O((N+n)d^2D)$. In summary, when $N$, $n$ and $d$ are fixed, the computational complexity of DA-Net is linearly associated with the scale of the data.

# 4 EXPERIMENTS

In this section, we evaluate the effectiveness of our proposed DA-Net model on four public TKG datasets.

## 4.1 Experimental Setup

*4.1.1 Datasets.* The TKG datasets for evaluation are WIKI [15], YAGO [21], GDELT [16] and ICEWS18 [5]. YAGO and WIKI are temporal subgraphs of YAGO3 and Wikipedia, respectively. ICEWS18 is extracted from temporal political events. GDELT comes from the news media on human societal scale behaviors. According to previous work [10, 12, 14, 17, 19, 20, 25, 33], the datasets are split into training/validation/test sets in proportions of 80%/10%/10%, respectively. Information on the datasets is detailed in Table 1.

*4.1.2 Baseline Methods.* Our proposed DA-Net method is compared with various static and dynamic TKG reasoning methods. The static methods include TransE [4], DistMult [32], ConvE [8], ComplEx [27], RotatE [26], R-GCN [23] and Comp-GCN [28]. Dynamic methods for TKG reasoning include TTransE [13], HyTE [7], TeMP [30], TA-DistMult [9] and DySAT [22]. And RE-NET [14], CyGNet [33] HIP network [12], xERTE [10], RE-GCN [19], TITer [25], TLogic [20] and CEN [17] predict future events based on the historical information and are similar work to ours. The baseline models are described in detail in Section 2.

*4.1.3 Evaluation Metrics.* We evaluate the effectiveness of our model with the link prediction task. For each query in the test set, we report the mean results of the two queries, $(s, p, ?, t_n)$ and $(?, p, o, t_n)$. We use conventional evaluation metrics, including the mean reciprocal rank (MRR), hits at 1 (Hits@1), hits at 3 (Hits@3) and hits at 10 (Hits@10), which all report the ranking of the missing ground-truth entity in the predicted results.

*4.1.4 Implementation Details.* We implement our DA-Net model in PyTorch and train the model on a GPU Tesla V100. We configure the model based on the MRR performance of the method on the validation set. In addition to the parameters given when introducing the model in Section 3, the $\alpha$ parameter for the attention addition is set to 0.5 for the YAGO and WIKI datasets, 0.8 for the ICEWS18 dataset, and 0.7 for the GDELT dataset. We use an AMSGrad optimizer to minimize the global loss with a 0.001 learning rate. The batch size is set to 1024 for all training datasets. The batch size of the testing datasets is set to 64 for YAGO and WIKI, 1024 for ICEWS18 and 512 for GDELT. We set the *n_layers* of the multi-head attention operation to 1, and the training epoch is limited to 30 for YAGO and WIKI, 6 for ICEWS18 and 2 for GDELT, which is sufficient for the task. For the static reasoning methods, the timestamp information is removed from all TKG datasets. For

R-GCN [23] and Comp-GCN [28], we use DistMult [32] as the decoder. We set the dimension of the embedding vectors to 200 to be consistent with the experimental settings in the HIP network [12]. Some of the baseline results are adopted from [12].

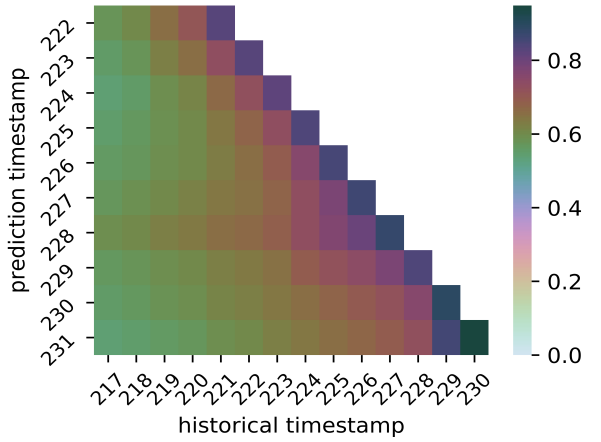


**Figure 3: Study on the time-variability problem on the WIKI dataset.**

**Table 1: Details of the TKG datasets.**

| #Datasets | #Entities | #Predicates | #Training | #Validation | #Test | #Granularity |
|---|---|---|---|---|---|---|
| YAGO | 10,623 | 10 | 161,540 | 19,523 | 20,026 | 1 year |
| WIKI | 12,554 | 24 | 539,286 | 67,538 | 63,110 | 1 year |
| ICEWS18 | 23,033 | 256 | 373,018 | 45,995 | 49,545 | 24 hours |
| GDELT | 7,691 | 240 | 1,734,399 | 238,765 | 305,241 | 15 mins |

For the similar baseline work xERTE [10], RE-GCN [19], TITer [25], TLogic [20] and CEN [17], we replicate the results on Tesla V100 using the default parameters in their open source codes and the same evaluation metrics as our model to ensure the consistency of the experimental settings. For CEN [17], we report its results under the online setting, which achieves the best results of it. For TITer [25] and xERTE [10], when we try to run on GDETT, the largest TKG dataset, their codes crash. TLogic [20], on the other hand, is only suitable for dealing with the ICEWS18 dataset, which provides the content references for entities, predicates, and timestamps. Therefore, we report only the results of the datasets that they are capable of processing, which is also consistent with the experiments reported in their papers.

## 4.2 On the Problem of Time-variability

In this section, we use the WIKI and YAGO datasets to study the time-variability problem in TKG reasoning. We show that different historical timestamps play various roles in predicting future events, which suggests that, under reasonable circumstances, future events should pay different attention to repetitive information at different historical timestamps.

For a query $(s, p, ?, t_n)$ with a missing object entity $o$, the ground-truth attention on historical repetitive facts can be represented by the set $\{(s, p, o, t_i)|t_i \in [t_0, t_{n-1}]\}$. We use different timestamps as research objects and design a metric $e_i^j$ for representing the ground-truth attention of a certain timestamp on its historical timestamps:

$$e_i^j = \frac{h_i^j}{p_j}, \tag{13}$$

where $h_i^j$ represents the ground-truth attention of all facts at the

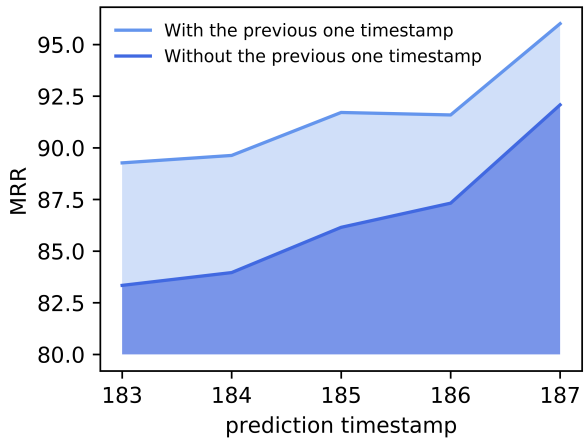


**Figure 4: The influence of the time-variability problem on future event prediction with the YAGO dataset.**

j-th prediction timestamp on the i-th historical timestamp, and $p_j$ represents the total number of facts at the j-th prediction timestamp. Therefore, $e_i^j$ indicates the overall attention of all facts at a given prediction timestamp on their historical timestamps. As observed in Figure 3, the vertical axis represents the prediction timestamps and the horizontal axis represents the historical timestamps, where each value represents the index of one year in the WIKI dataset. For each prediction timestamp in the test set, we determine the ground-truth attention on all its historical timestamps starting with the 217th timestamp. Finally, the attention of the 10 prediction timestamps (between 222 and 231) on their historical timestamps is represented as a heat map based on the calculation results of Eq. 13. It can be observed that the attention of the facts at the prediction timestamps on the historical information dynamically evolves over time. Interestingly, for all prediction timestamps, their ground-truth attention to the historical information decreases with increasing historical distance, and the historical information closer to the prediction timestamps makes more sense.

The intuitive consequence of the time-variability problem is that the prediction timestamp cannot capture and utilize new historical information in time, e.g., the prediction timestamp's previous timestamp. Each prediction timestamp in the YAGO dataset (except the 188th timestamp, which is removed because it contains only one fact) is used as a separate study object. For cases both with and without the previous timestamp, we test the individual performance of each prediction timestamp. Due to space, we report only the most representative MRR metric. As shown in Figure 4, when DA-Net ignores the time-variability problem and does

**Table 2: Performance (in percentage) on four datasets. The best results are bolded, and the second-best results are underlined.**

| | WIKI | | | | YAGO | | | | GDELT | | | | ICEWS18 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | 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 |
| TransE | 46.68 | 36.19 | 49.71 | 51.71 | 48.97 | 46.23 | 62.45 | 66.05 | 16.05 | 0.00 | 26.10 | 42.29 | 17.56 | 2.48 | 26.95 | 43.87 |
| DistMult | 46.12 | 37.24 | 49.81 | 51.38 | 59.47 | 52.97 | 60.91 | 65.26 | 18.71 | 11.59 | 20.05 | 32.55 | 22.16 | 12.13 | 26.00 | 42.18 |
| ComplEx | 47.84 | 38.15 | 50.08 | 51.39 | 61.29 | 54.88 | 62.28 | 66.82 | 22.77 | 15.77 | 24.05 | 36.33 | 30.09 | 21.88 | 34.15 | 45.96 |
| ConvE | 47.57 | 38.76 | 50.10 | 50.53 | 62.32 | 56.19 | 63.97 | 65.60 | 35.99 | 27.05 | 39.32 | 49.44 | 36.67 | 28.51 | 39.80 | 50.69 |
| RotatE | 50.67 | 40.88 | 50.71 | 50.88 | 65.09 | 57.13 | 65.67 | 66.16 | 22.33 | 16.68 | 23.89 | 32.29 | 23.10 | 14.33 | 27.61 | 38.72 |
| R-GCN | 37.57 | 28.51 | 39.66 | 41.90 | 41.30 | 32.56 | 44.44 | 52.68 | 23.31 | 17.24 | 24.96 | 34.36 | 23.19 | 16.36 | 25.34 | 36.48 |
| Comp-GCN | 37.64 | 28.33 | 39.87 | 42.03 | 41.42 | 32.63 | 44.59 | 52.81 | 23.46 | 16.65 | 25.54 | 34.58 | 23.31 | 16.52 | 25.37 | 36.61 |
| TTransE | 31.74 | 22.57 | 36.25 | 43.45 | 32.57 | 27.94 | 43.39 | 53.37 | 5.52 | 0.47 | 5.01 | 15.27 | 8.36 | 1.94 | 8.71 | 21.93 |
| HyTE | 43.02 | 34.29 | 45.12 | 49.49 | 23.16 | 12.85 | 45.74 | 51.94 | 6.37 | 0.00 | 6.72 | 18.63 | 7.31 | 3.10 | 7.50 | 14.95 |
| TA-DistMult | 48.09 | 38.71 | 49.51 | 51.70 | 61.72 | 52.98 | 63.32 | 65.19 | 29.35 | 22.11 | 31.56 | 41.39 | 28.53 | 20.30 | 31.57 | 44.96 |
| DySAT | 31.82 | 22.07 | 26.59 | 35.59 | 43.43 | 31.87 | 43.67 | 46.49 | 23.34 | 14.96 | 22.57 | 27.83 | 19.95 | 14.42 | 23.67 | 26.67 |
| TeMP | 49.61 | 46.96 | 50.24 | 51.81 | 62.25 | 55.39 | 64.63 | 66.12 | 37.56 | 29.82 | 40.15 | 48.60 | 40.48 | 33.97 | 42.63 | 52.38 |
| RE-NET | 51.97 | 48.01 | 52.07 | 53.91 | 65.16 | 63.29 | 65.63 | 68.08 | 40.12 | 32.43 | 43.40 | 53.80 | 42.93 | 36.19 | 45.47 | 55.80 |
| CyGNet | 52.60 | 50.48 | 53.26 | 55.82 | 66.58 | 64.26 | 67.98 | 70.16 | 51.06 | 44.66 | 54.74 | 61.32 | 47.83 | 42.02 | 50.71 | 57.72 |
| HIP network | 54.71 | 53.82 | 54.73 | 56.46 | 67.55 | 66.32 | 68.49 | 70.37 | 52.76 | 46.35 | 55.31 | 61.87 | 48.37 | 43.51 | 51.32 | 58.49 |
| xERTE | 77.47 | 76.01 | 78.79 | 79.54 | 88.75 | 87.88 | 89.30 | 90.38 | - | - | - | - | 36.47 | 29.60 | 40.26 | 50.41 |
| RE-GCN | 81.07 | 78.84 | 82.36 | 84.95 | 83.27 | 80.20 | 84.94 | 89.00 | 39.72 | 31.93 | 43.14 | 53.46 | 45.67 | 37.62 | 49.19 | 61.18 |
| TITer | 74.89 | 74.05 | 74.71 | 76.57 | 90.48 | **90.25** | 90.46 | 90.81 | - | - | - | - | 37.00 | 31.14 | 39.05 | 47.96 |
| TLogic | - | - | - | - | - | - | - | - | - | - | - | - | 37.35 | 29.57 | 40.56 | 53.02 |
| CEN | 83.11 | 81.20 | 84.15 | 86.46 | 85.84 | 83.55 | 87.11 | 90.02 | 43.54 | 36.51 | 46.13 | 56.88 | 45.09 | 37.85 | 47.92 | 59.12 |
| **DA-Net** | **84.13** | **81.66** | **86.46** | **87.37** | **91.59** | 90.07 | **92.94** | **93.43** | **58.47** | **51.89** | **62.32** | **69.82** | **51.92** | **45.55** | **55.70** | **62.62** |

**Table 3: Ablation study on the ICEWS18 dataset.**

| Evaluation Metrics | MRR | Hits@1 | Hits@3 | Hits@10 |
|---|---|---|---|---|
| Global channel only | 34.41 | 25.78 | 38.23 | 50.76 |
| Global channel and first layer of attention | 39.71 | 32.80 | 42.48 | 52.90 |
| Global channel and second layer of attention | 41.62 | 34.57 | 44.55 | 55.08 |
| Historical channel only | 47.23 | 44.36 | 49.84 | 51.03 |
| **DA-Net** | **51.92** | **45.55** | **55.70** | **62.62** |

not pay attention to the new historical timestamp of each prediction timestamp (corresponding to the dark blue area in Figure 4), the performance is considerably lower than the performance of the complete DA-Net model (corresponding to the light blue area in Figure 4). However, as shown in Figure 3, in addition to the above-mentioned problem, the time-variability problem also includes the different roles of various historical timestamps on future event prediction, which is ignored by CEN [17]. Therefore, to address the time-variability problem, DA-Net adopts distributed attention instead of the traditional codec-based framework to model the distribution of historical information.

## 4.3 Results of Reasoning on TKGs

In this section, we compare DA-Net with static and dynamic inference methods based on link prediction tasks in TKGs.

As shown in Table 2, in the TKG reasoning task, dynamic methods generally perform better than static methods, with the exception of HyTE [7] and TTransE [13]. We believe that this result occurs because these models focus on the embedding representation of temporal information while ignoring the temporal evolution. In terms of the similar work, such as RE-NET [14], CyGNet [33], HIP network [12], xERTE [10], RE-GCN [19], TLogic [20] and CEN [17], our proposed DA-Net model has a considerable improvement over all baselines on all evaluation TKG datasets and all evaluation metrics. This improvement is because all of these models, with the exception of CEN [17], ignore the problem of time-variability in TKG reasoning. Therefore, these models treat repetitive information at different historical timestamps equally, and it is nearly impossible for these models to utilize distributed information at various timestamps, which inevitably degrades the performance of these models.

However, although CEN [17] proposes an online learning strategy to address the challenge of time-variability, it is still limited by its codec-based framework and must constantly fine-tune representation vectors with finite lengths, which compresses the distributed information of each historical timestamp into a finite vector and inevitably results in representation limitations and the loss of distributed information. It is observed that the performance of our proposed DA-Net model on the YAGO dataset is inferior to that of TITer [25] under the evaluation metric Hits@1. As mentioned in CEN [17], TITer [25] retrieves answers through an explicit path, which usually results in a high Hits@1 metric. We also observe that for large datasets, such as GDELT and ICEWS18, the performance of some recently proposed models [10, 17, 19, 20, 25] is far inferior to that of DA-Net, CyGNet [33] and HIP network [12], because DA-Net, similar to CyGNet [33] and HIP network [12], utilizes the frequency statistics of the repetitive facts. In the second attention layer, DA-Net not only uses these frequency statistics but also models and captures the changes in these statistics that impact the prediction of future events; thus, DA-Net also performs better than CyGNet [33] and the HIP network [12]. We believe that this is an advantage of modeling based on human cognition.

## 4.4 Ablation Study

We perform an ablation study on the ICEWS18 dataset. The two attention layers in the historical channel are removed from the

model both separately and simultaneously. Moreover, we evaluate the performance of the global channel. The MRR and Hit@1/2/3 metrics are used for evaluation.

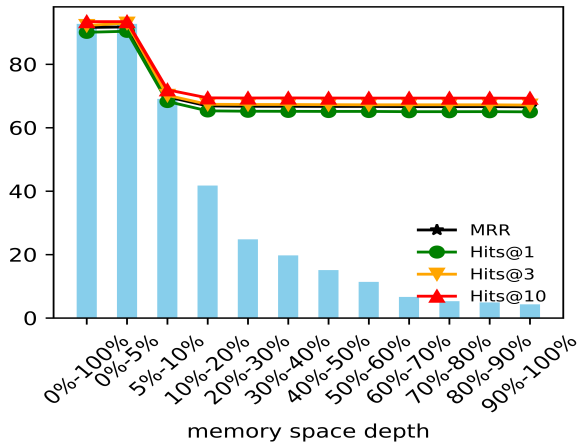


**Figure 5: Study on the depth of memory space with the YAGO dataset.**

As indicated in Table 3, the model performance decreases significantly when only the global channel is adopted because the model ignores the historical information. In the second row of Table 3, we use the global channel and the first attention layer for the prediction. The results are better than when only the global channel is adopted. This result shows the effect of the prior experience learned in the first attention layer; however, it is obviously of limited use because this traditional attention cannot cope with the development of time and the evolution of events. We then adopt the global channel with only the second attention layer in the historical channel.In this setting, the results are significantly better than when only the global channel is adopted. In addition, the results indicate that recent knowledge developments (changes in the frequency statistics of historical repetitive facts) learned by the second attention layer contribute more to the prediction. This result shows that there is both variant and invariant information in the data. Because the first and second attention layers (corresponding to the second and third rows in Table 3) both perform better than the global channel, DA-Net successfully captures the invariant historical information, and especially the changing distributed information of event evolution.

The model in the fourth row of the table uses only the historical channel for prediction, and its results are slightly worse than those of DA-Net in terms of the MRR, Hits@1 and Hits@3. However, in terms of the Hits@10 metric, the performance when only the historical channel is adopted is worse than the performance of the second and third rows in Table 3. This result demonstrates the importance of the global channel in the DA-Net model. When all the components are adopted, it observes a substantial improvement over any single component. This is because DA-Net includes and models the global information in the global channel, the invariant historical information in the first attention layer and the variant historical information in the second attention layer. Therefore, through the ablation test, we can conclude that each model component contributes to DA-Net, and DA-Net successfully learns the original attention on repetitive facts as prior experience and the frequency changes of historical facts as recent knowledge developments in the novel historical channel.

## 4.5 On the Depth of Memory Space

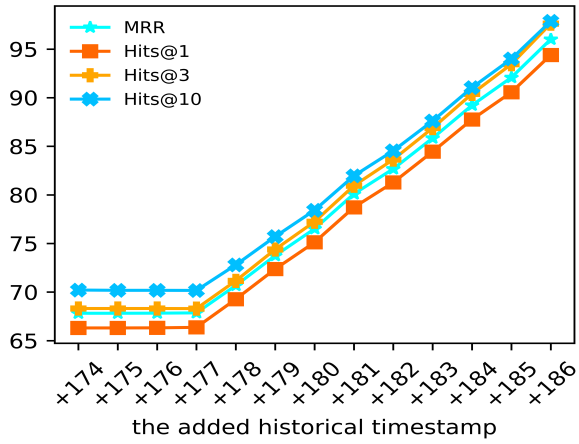


**Figure 6: Study on the knowledge sensitivity of shallow memory on the YAGO dataset.**

In this section, we use the YAGO dataset to extend the research object to the entire memory space and focus on the second attention layer. By investigating the contribution of memory space segments at different depths to the prediction, we demonstrate the critical role of shallow memory in predicting future events, and our proposed DA-Net successfully captures knowledge sensitivity.

*The vital effect of shallow memory.* As shown in Figure 5, for the test sets of the YAGO dataset, we split the memory space into 11 segments according to depths. For the 189 timestamps (from 0 to 188), the memory space segments {100%-90%, 90%-80%, 80%-70%, 70%-60%, 60%-50%, 50%-40%, 40%-30%, 30%-20%, 20%-10%, 10%-5%, 5%-0%} represent the timestamp ranges {[0, 19), [19, 38), [38, 56), [56, 75), [75, 94), [94, 113], [113, 132), [132, 150), [150, 169), [169, 179), [179, 188)} from deep to shallow, respectively. We report the MRR metric and the Hits@1/3/10 metrics of our DA-Net model for both the complete memory space and the different memory segments. As observed in Figure 5, the most significant contribution to the prediction is concentrated in the 5% short-term memory, which has almost the same effect as extracting the historical information across the entire memory space. With increasing memory depth, the prediction performance decreases. In particular, when the segment range of the memory space increases from 5% to 20%, the performance decrease sharply. The contribution of the memory space at a depth of greater than 20% remains stable and at a low level. For the 20026 nonrepetitive triples in the test set, we count the number of their historical repetitive facts in each segment of the memory space. The final result is represented as a percentage.

The blue bars in Figure 5 show that 93% of the test facts are repeated in shallow memory (0-5%), which is equivalent to the effect of the complete memory space. This result proves that for the facts of prediction timestamps, the repetitive historical information gathers in shallow memory space, and decreases with increasing depth.

*Capturing the knowledge sensitivity of shallow memory.* Knowledge sensitivity indicates that the change in frequency-based statistical information in shallow memory reflects the recent knowledge developments, which adjust the prediction of future events. Because most repetitive facts are concentrated in shallow memory, their frequency statistics accumulate from the shallow memory boundary and remain at a similar level. Therefore, the frequency changes in shallow memory at each timestamp have distinct impacts on the overall number of repetitive facts. Thus, shallow memory possesses the feature of knowledge sensitivity. We use the 187th timestamp in the YAGO dataset as the test object and successively add new historical timestamps to the DA-Net model starting with the 174th timestamp, which is sufficient for demonstrating the effectiveness of DA-Net. We report how performance changes as a result of the recent new knowledge developments. As shown in Figure 6, DA-Net is not affected by the addition of new historical information until the 178th timestamp. We note that the shallow memory starts at the 179th timestamp, and the performance of DA-Net improves with the addition of new historical information. This result demonstrates that DA-Net is sensitive to recent knowledge developments when faced with new historical information over time.

## 5 CONCLUSIONS

In this paper, to address the time-variability problem in temporal knowledge graph reasoning, we propose DA-Net. Inspired by dual-process theory in cognitive science, DA-Net assigns distributed attention to historical information at different timestamps through dual layers of attention, and models the dynamic distribution of repetitive facts. In the first attention layer (Process I), DA-Net assigns traditional attention to repetitive facts based on their history of distant dependencies. In the second attention layer (Process II), DA-Net adjusts the original attention based on recent knowledge developments (changes in the historical frequency statistics). A large number of experiments demonstrate that DA-Net achieves a qualitative improvement over baseline methods.

## REFERENCES


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

[2] Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5185–5194.

[3] Jacob Beck and Bruce Ambler. 1973. The effects of concentrated and distributed attention on peripheral acuity. *Perception & Psychophysics* 14, 2 (1973), 225–230.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[5] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS coded event data. *Harvard Dataverse* 12 (2015).

[6] 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*.

[7] 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*. 2001–2011.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

[9] Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] Yongquan He, Peng Zhang, Luchen Liu, Qi Liang, Wenyuan Zhang, and Chuang Zhang. 2021. HIP Network: Historical Information Passing Network for Extrapolation Reasoning on Temporal Knowledge Graph. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 1915–1921. https://doi.org/10.24963/ijcai.2021/264

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

[14] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6669–6683.

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

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

[17] Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022. Complex Evolutional Pattern Learning for Temporal Knowledge Graph Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

[18] Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021. 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)*. 4732–4743.

[19] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. 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*. 408–417.

[20] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. TLogic: Temporal Logical Rules for Explainable Link Forecasting on Temporal Knowledge Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[21] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference.

[22] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 519–527.

[23] 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*. Springer, 593–607.

[24] Steven A Sloman. 1996. The empirical case for two systems of reasoning. *Psychological bulletin* 119, 1 (1996), 3.

[25] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 8306–8319.

[26] 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*.

[27] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International*

*conference on machine learning*. PMLR, 2071–2080.

[28] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[30] 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)*. 5730–5746.

[31] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 564–573.

[32] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

[33] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4732–4740.