The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Temporal Knowledge Graph Reasoning with Historical Contrastive Learning

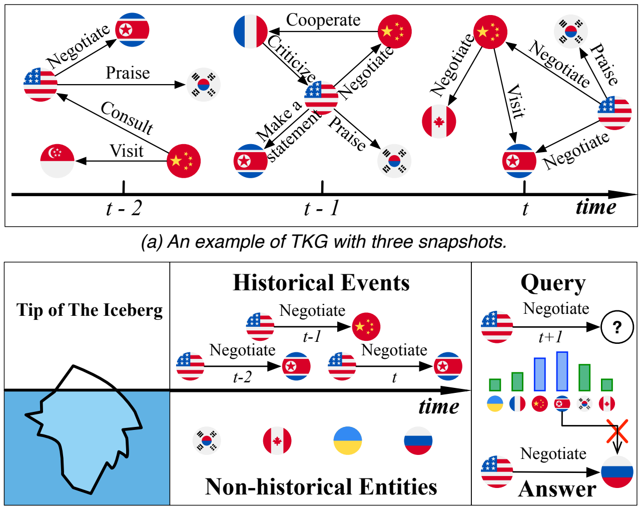
Yi Xu, Junjie Ou, Hui Xu, Luoyi Fu[*]
Department of Computer Science and Engineering
Shanghai Jiao Tong University
{yixu98, j_michael, xhui_1, yiluofu}@sjtu.edu.cn

Abstract

Temporal knowledge graph, serving as an effective way to store and model dynamic relations, shows promising prospects in event forecasting. However, most temporal knowledge graph reasoning methods are highly dependent on the recurrence or periodicity of events, which brings challenges to inferring future events related to entities that lack historical interaction. In fact, the current moment is often the combined effect of a small part of historical information and those unobserved underlying factors. To this end, we propose a new event forecasting model called **C**ontrastive **E**vent **Net**work (CENET), based on a novel training framework of historical contrastive learning. CENET learns both the historical and non-historical dependency to distinguish the most potential entities that can best match the given query. Simultaneously, it trains representations of queries to investigate whether the current moment depends more on historical or non-historical events by launching contrastive learning. The representations further help train a binary classifier whose output is a boolean mask to indicate related entities in the search space. During the inference process, CENET employs a mask-based strategy to generate the final results. We evaluate our proposed model on five benchmark graphs. The results demonstrate that CENET significantly outperforms all existing methods in most metrics, achieving at least 8.3% relative improvement of Hits@1 over previous state-of-the-art baselines on event-based datasets.

1 Introduction

Knowledge Graphs (KGs), serving as the collections of human knowledge, have revealed promising expectations in the field of natural language processing (Sun et al. 2020; Wang et al. 2021), recommendation system (Wang et al. 2019), and information retrieval (Liu et al. 2018), etc. A traditional KG is usually a static knowledge base that uses a graph-structured data topology to integrate facts (also called events) in the form of triples (s, p, o), where s and o denote subject and object entities respectively, and p as a relation type means predicate. In the real world, knowledge evolves continuously, inspiring the construction and application of the Temporal Knowledge Graphs (TKGs), where the fact has extended from a triple (s, p, o) to a quadruple with a timestamp t, i.e., (s, p, o, t). As a result, a TKG consists of multiple snapshots, and the facts in the same snapshot co-occur. Figure 1 (a) shows an example of TKG consisting of a series of international political events, where some events may occur repeatedly, and new events will also emerge.



(a) An example of TKG with three snapshots.

(b) Challenges of existing methods.

Figure 1: An example of TKG and challenges of existing methods.

TKGs provide new perspectives and insights for many downstream applications, e.g., policymaking (Deng, Rangwala, and Ning 2020), stock prediction (Feng et al. 2019), and dialogue systems (Jia et al. 2018), thus triggering intense interests in TKG reasoning. In this work, we focus on forecasting events (facts) in the future on TKGs, which is also called graph extrapolation. Our goal is to predict the missing entities of queries like $(s, p, ?, t)$ for a future timestamp t that has not been observed in the training set.

Many efforts (Garcia-Duran, Dumančić, and Niepert 2018; Jin et al. 2020) have been made toward modeling the structural and temporal characteristics of TKGs for future event prediction. Some mainstream examples (Jin et al. 2020; Li et al. 2021b) make reference to known events in history, which can easily predict repetitive or periodic events.

[*] Corresponding author.

However, in terms of the event-based TKG *Integrated Crisis Early Warning System*, new events that have never occurred before account for about 40% (Boschee et al. 2015). It is challenging to infer these new events because they have fewer temporal interaction traces during the whole timeline. For instance, the right part of Figure 1 (b) shows the query *(the United States, Negotiate, ?, t+1)* and its corresponding new events *(the United States, Negotiate, Russia, t+1)*, where most existing methods often obtain incorrect results over such query due to their focus on the high frequent recurring events. Additionally, during the inference process, existing methods rank the probability scores of overall candidate entities in the whole graph without any bias. We argue that the bias is necessary when approaching the missing entities of different events. For repetitive or periodic events, models are expected to prioritize a few frequently occurring entities, and for new events, models should pay more attention to entities with less historical interaction.

In this work, we will go beyond the limits of historical information and mine potential temporal patterns from the whole knowledge. To elaborate our design clearer, we call the past events associated with the entities in the current query $(s, p, ?, t)$ *historical events*, and others *non-historical events*. Their corresponding entities are called *historical* and *non-historical entities*, respectively. We will give formal definitions in Section 3.1. We intuitively consider that the events in TKG are not only related to their historical events but also indirectly related to unobserved underlying factors. The historical events we can see are only the tip of the iceberg. We propose a novel TKG reasoning model called CENET (**C**ontrastive **E**vent **Net**work) for event forecasting based on contrastive learning. Given a query $(s, p, ?, t)$ whose real object entity is o, CENET takes into account its historical and non-historical events and identify significant entities via contrastive learning. Specifically, a copy mechanism-based scoring strategy is first adopted to model the dependency of historical and non-historical events. In addition, all queries can be divided into two classes according to their real object entities: either the object entity o is a historical entity or a non-historical entity. Therefore, CENET naturally employs supervised contrastive learning to train representations of the two classes of queries, further helping train a classifier whose output is a boolean value to identify which kind of entities should receive more attention. During the inference, CENET combines the distribution from the historical and non-historical dependency, and further considers highly correlated entities with a mask-based strategy according to the classification results.

The contributions of our paper are summarized as follows:

- We propose a TKG model called CENET for event forecasting. CENET can predict not only repetitive and periodic events but also potential new events via joint investigation of both historical and non-historical information;
- To the best of our knowledge, CENET is the first model to apply contrastive learning to TKG reasoning, which trains contrastive representations of queries to identify highly correlated entities;
- We conduct experiments on five public benchmark graphs. The results demonstrate that CENET outperforms the state-of-the-art TKG models in the task of event forecasting.

2 Related Work

2.1 Temporal Knowledge Graph Reasoning

There are two different settings for TKG reasoning: interpolation and extrapolation (Jin et al. 2020). Given a TKG with timestamps ranging from t_0 to t_n, models with the interpolation setting aim to complete missing events that happened in the interval $[t_0, t_n]$, which is also called TKG completion. In contrast, the extrapolation setting aims to predict possible events after the given time t_n, i.e., inferring the entity o (or s) given query $q = (s, p, ?, t)$ (or $(?, p, o, t)$) where $t > t_n$.

Models in the former case such as HyTE (Dasgupta, Ray, and Talukdar 2018), TeMP (Wu et al. 2020), and ChronoR (Sadeghian et al. 2021) are designed to infer missing relations within the observed data. However, such models are not designed to predict future events that fall out of the specified time interval. In the latter case, various methods are designed for the purpose of future event prediction. Know-Evolve (Trivedi et al. 2017) is the first model to learn non-linearly evolving entity embeddings, yet unable to capture the long-term dependency. xERTE (Han et al. 2020) and TLogic (Liu et al. 2022) provide understandable evidence that can explain the forecast, but their application scenarios are limited. TANGO (Han et al. 2021) employs neural ordinary differential equations to model the TKGs. A copy-generation mechanism is adopted in CyGNet (Zhu et al. 2021) to identify high-frequency repetitive events. CluSTeR (Li et al. 2021a) is designed with reinforcement learning, yet constraining its applicability to event-based TKGs. There also emerge some models which try to adopt GNN (Kipf and Welling 2016) or RNN architecture to capture spatial temporal patterns. Typical examples include RE-NET (Jin et al. 2020), RE-GCN (Li et al. 2021b), HIP (He et al. 2021), and EvoKG (Park et al. 2022).

2.2 Contrastive Learning

Contrastive learning as a self-supervised learning paradigm focuses on distinguishing instances of different categories. In self-supervised contrastive learning, most methods (Chen et al. 2020) derive augmented examples from a randomly sampled minibatch of N examples, resulting in $2N$ samples to optimize the following loss function given a positive pair of examples (i, j). Equation 1 is the contrastive loss:

$$\mathcal{L}_{i,j} = -\log \frac{exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{k=1, k\neq i}^{2N} exp(\mathbf{z}_i \cdot \mathbf{z}_k / \tau)}, \quad (1)$$

where $\mathbf{z}_i$ is the projection embedding of sample i and $\tau \in \mathbb{R}^+$ denotes a temperature parameter helping the model learn from hard negatives. In the case of supervised learning, there is a work (Khosla et al. 2020) generalizing contrastive loss to an arbitrary number of positives, which separates the representations of different instances using ground truth labels. The obtained contrastive representations can promote the downstream classifier to achieve better performance compared with vanilla classification model.

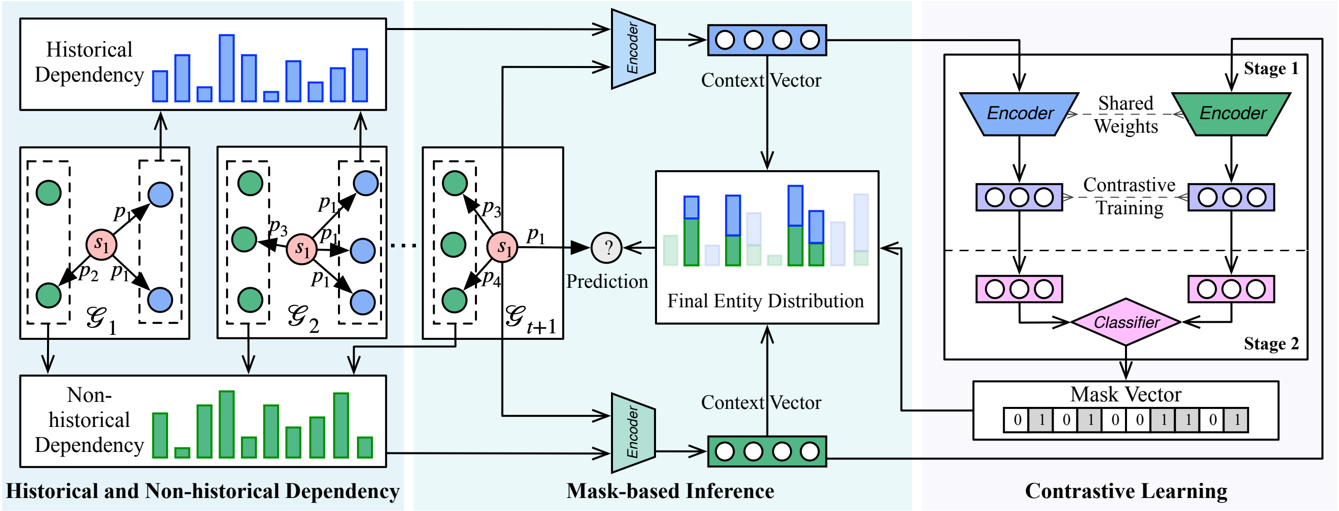


Figure 2: The overall architecture of CENET. The left part learns the distribution of entities from both historical and non-historical dependency. The right part illustrates the two stages of historical contrastive learning, which aims to identify highly correlated entities, and the output is a boolean mask vector. The middle part is the mask-based inference process that combines the distribution learned from the two kinds of dependency and the mask vector to generate the final results.

3 Method

As shown in Figure 2, CENET captures both the historical and non-historical dependency. Simultaneously, it utilizes contrastive learning to identify highly correlated entities. A mask-based inference process is further employed for reasoning performing. In the following parts, we will introduce our proposed method in detail.

3.1 Preliminaries

Let $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote a finite set of entities, relation types, and timestamps, respectively. A temporal knowledge graph $\mathcal{G}$ is a set of quadruples formalized as (s, p, o, t), where $s \in \mathcal{E}$ is a subject (head) entity, $o \in \mathcal{E}$ is an object (tail) entity, $p \in \mathcal{R}$ is the relation (predicate) occurring at timestamp t between s and o. $\mathcal{G}_t$ represents a TKG snapshot which is the set of quadruples occurring at time t. We use boldfaced **s**, **p**, **o** for the embedding vectors of s, p, and o respectively, the dimension of which is d. $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times d}$ is the embeddings of all entities, the row of which represents the embedding vector of an entity such as **s** and **o**. Similarly, $\mathbf{P} \in \mathbb{R}^{|\mathcal{R}| \times d}$ is the embeddings of all relation types.

Given a query $q = (s, p, ?, t)$, we define the set of *historical events* as $\mathcal{D}_t^{s,p}$ and the corresponding set of *historical entities* as $\mathcal{H}_t^{s,p}$ in the following equations:

$$\mathcal{D}_t^{s,p} = \bigcup_{k<t} \{(s, p, o, k) \in \mathcal{G}_k\}, \tag{2}$$

$$\mathcal{H}_t^{s,p} = \{o | (s, p, o, k) \in \mathcal{D}_t^{s,p}\}. \tag{3}$$

Naturally, entities not in $\mathcal{H}_t^{s,p}$ are called *non-historical entities*, and the set $\{(s, p, o', k) | o' \notin \mathcal{H}_t^{s,p}, k < t\}$ denotes the set of *non-historical events*, where some quadruples may not exist in $\mathcal{G}$. It is worth noting that we also use $\mathcal{D}_t^{s,p}$ to represent the set of historical events for a current event (s, p, o, t). If an event (s, p, o, t) itself does not exist in its corresponding $\mathcal{D}_t^{s,p}$, then it is a new event. Without loss of generality, we detail how CENET predicts object entities with a given query $q = (s, p, ?, t)$ in the following parts.

3.2 Historical and Non-historical Dependency

In most TKGs, although many events often show repeated occurrence pattern, new events may have no historical events to refer to. To this end, CENET takes not only historical but also non-historical entities into consideration. We first investigate the frequencies of historical entities for the given query $q = (s, p, ?, t)$ during data pre-processing. More specifically, we count the frequencies $\mathbf{F}_t^{s,p} \in \mathbb{R}^{|\mathcal{E}|}$ of all entities served as the objects associated with subject s and predicate p before time t, as shown in Equation 4:

$$\mathbf{F}_t^{s,p}(o) = \sum_{k<t} |\{o | (s, p, o, k) \in \mathcal{G}_k\}|. \tag{4}$$

Since we cannot count the frequencies of non-historical entities, CENET transforms $\mathbf{F}_t^{s,p}$ into $\mathbf{Z}_t^{s,p} \in \mathbb{R}^{|\mathcal{E}|}$ where the value of each slot is limited by a hyper-parameter λ:

$$\mathbf{Z}_t^{s,p}(o) = \lambda \cdot (\Phi_{\mathbf{F}_t^{s,p}(o)>0} - \Phi_{\mathbf{F}_t^{s,p}(o)=0}). \tag{5}$$

Φ_β is an indicator function that returns 1 if β is true and 0 otherwise. $\mathbf{Z}_t^{s,p}(o) > 0$ represents the quadruple (s, p, o, t_k) is a historical event bound to s, p, and t $(t_k < t)$, while $\mathbf{Z}_t^{s,p}(o) < 0$ indicates that the quadruple (s, p, o, t_k) is a non-historical event that does not exist in $\mathcal{G}$. Next, CENET learns the dependency from both the historical and non-historical events based on the input $\mathbf{Z}_t^{s,p}$. CENET adopts a copy mechanism based learning strategy (Gu et al. 2016) to capture different kinds of dependency from two aspects: one is the similarity score vector between query and the set of entities, the other is the query's corresponding frequency information with copy mechanism.

For historical dependency, CENET generates a latent context vector $\mathbf{H}_{his}^{s,p} \in \mathbb{R}^{|\mathcal{E}|}$ for query q, which scores the historical dependency of different object entities:

$$\mathbf{H}_{his}^{s,p} = \underbrace{tanh(\mathbf{W}_{his}(\mathbf{s} \oplus \mathbf{p}) + \mathbf{b}_{his})\mathbf{E}^{T}}_{\text{similarity score between } q \text{ and } \mathcal{E}} + \mathbf{Z}_t^{s,p}, \qquad (6)$$

where *tanh* is the activation function, $\oplus$ represents the concatenation operator, $\mathbf{W}_{his} \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_{his} \in \mathbb{R}^{d}$ are trainable parameters. We use a linear layer with *tanh* activation to aggregate the query's information. The output of the linear layer is then multiplied by $\mathbf{E}$ to obtain an $|\mathcal{E}|$-dimensional vector, where each element represents the **similarity** score between the corresponding entity $o' \in \mathcal{E}$ and the query q. Then, according to the copy mechanism, we add the copy-term $\mathbf{Z}_t^{s,p}$ to change the index scores of historical entities in $\mathbf{H}_{his}^{s,p}$ to higher values directly without contributing to the gradient update. Thus, $\mathbf{Z}_t^{s,p}$ makes $\mathbf{H}_{his}^{s,p}$ pay more attention to historical entities. Similarly, for non-historical dependency, the latent context vector $\mathbf{H}_{nhis}^{s,p}$ is defined as:

$$\mathbf{H}_{nhis}^{s,p} = tanh(\mathbf{W}_{nhis}(\mathbf{s} \oplus \mathbf{p}) + \mathbf{b}_{nhis})\mathbf{E}^{T} - \mathbf{Z}_t^{s,p}. \qquad (7)$$

Contrary to historical dependency (Equation 6), subtracting $\mathbf{Z}_t^{s,p}$ makes $\mathbf{H}_{nhis}^{s,p}$ focus on non-historical entities. The training objective of learning from both historical and non-historical events is to minimize the following loss $\mathcal{L}^{ce}$:

$$\mathcal{L}^{ce} = -\sum_{q} \log\{\frac{exp(\mathbf{H}_{his}^{s,p}(o_i))}{\sum\limits_{o_j \in \mathcal{E}} exp(\mathbf{H}_{his}^{s,p}(o_j))} + \frac{exp(\mathbf{H}_{nhis}^{s,p}(o_i))}{\sum\limits_{o_j \in \mathcal{E}} exp(\mathbf{H}_{nhis}^{s,p}(o_j))}\}, \qquad (8)$$

where o_i denotes the ground truth object entity of the given query q. The purpose of $\mathcal{L}^{ce}$ is to separate ground truth from others by comparing each scalar value in $\mathbf{H}_{his}^{s,p}$ and $\mathbf{H}_{nhis}^{s,p}$.

During the inference, CENET combines the softmax results of the above two latent context vectors as the predicted probabilities $\mathbf{P}_t^{s,p}$ over all object entities:

$$\mathbf{P}_t^{s,p} = \frac{1}{2}\{softmax(\mathbf{H}_{his}^{s,p}) + softmax(\mathbf{H}_{nhis}^{s,p})\}, \qquad (9)$$

where the entity with maximum value is the most likely entity the component predicts.

3.3 Historical Contrastive Learning

Clearly, the learning mechanism defined above well captures the historical and non-historical dependency for each query. However, many repetitive and periodic events are only associated with historical entities. Besides, for new events, existing models are likely to ignore those entities with less historical interaction and predict the wrong entities that frequently interact with other events. The proposed historical contrastive learning trains contrastive representations of queries to identify a small number of highly correlated entities at the query level.

Specifically, the training process of supervised contrastive learning (Khosla et al. 2020) consists of two stages. We first introduce I_q to indicate whether the missing object is in $\mathcal{H}_t^{s,p}$ for query q. In other words, if I_q is equal to 1, the missing object of the given query q is in $\mathcal{H}_t^{s,p}$, and 0 otherwise. The aim of the two stages is to train a binary classifier which infers the value of such boolean scalar for query q.

Stage 1: Learning Contrastive Representations. In the first stage, the model learns the contrastive representations of queries by minimizing supervised contrastive loss, which takes whether I_q is positive as the training criterion to separate representations of different queries as far as possible in semantic space. Let $\mathbf{v}_q$ be the embedding vector (representation) of the given query q:

$$\mathbf{v}_q = MLP(\mathbf{s} \oplus \mathbf{p} \oplus tanh(\mathbf{W}_F \mathbf{F}_t^{s,p})), \qquad (10)$$

where the query's information is encoded by an MLP to normalize and project the embedding onto the unit sphere for further contrastive training. Let M denote the minibatch, $Q(q)$ denote the set of queries in the M except q whose boolean labels are the same as I_q, given as:

$$Q(q) = \bigcup_{m \in M \setminus \{q\}} \{m | I_m = I_q\}. \qquad (11)$$

The detail of computing supervised contrastive loss $\mathcal{L}^{sup}$ in the first stage is as follows:

$$\mathcal{L}^{sup} = \sum_{q \in M} \frac{-1}{|Q(q)|} \sum_{k \in Q(q)} \log \frac{exp(\mathbf{v}_q \cdot \mathbf{v}_k / \tau)}{\sum\limits_{a \in M \setminus \{q\}} (\mathbf{v}_q \cdot \mathbf{v}_a / \tau)}, \qquad (12)$$

where, $\mathbf{W}_F \in \mathbb{R}^{d \times |\mathcal{E}|}$ is the trainable parameter, $\tau \in \mathbb{R}^{+}$ is the temperature parameter set to 0.1 in experiments as recommended in the previous work (Khosla et al. 2020). The objective of $\mathcal{L}^{sup}$ is to make the representations of the same category closer. It should be noted that the contrastive supervised loss $\mathcal{L}^{sup}$ and the previous cross-entropy-like loss $\mathcal{L}^{ce}$ are trained simultaneously.

Stage 2: Training Binary Classifier. When the training of the first stage is finished, CENET freezes the weights of corresponding parameters including $\mathbf{E}$, $\mathbf{P}$ and their encoders in the first stage. Then it feeds $\mathbf{v}_q$ to a linear layer to train a binary classifier with cross-entropy loss according to the ground truth I_q, which is trivial to mention. Now, the classifier can recognize whether the missing object entity of query q exists in the set of historical entities.

In the process of reasoning, a boolean mask vector $\mathbf{B}_t^{s,p} \in \mathbb{R}^{|\mathcal{E}|}$ is generated to identify which kind of entities should be concerned according to the predicted $\hat{I}_q$ and whether $o \in \mathcal{H}_t^{s,p}$ is true:

$$\mathbf{B}_t^{s,p}(o) = \Phi_{o \in \mathcal{H}_t^{s,p} = \hat{I}_q}. \qquad (13)$$

The probabilities of entities in all positive positions ($\mathbf{B}_t^{s,p}(o) = 1$) will be further increased, and vice versa. In other words, if the missing object is predicted to be in $\mathcal{H}_t^{s,p}$, then entities in the historical set will receive more attention. Otherwise, those entities outside the historical set are more likely to be attended.

Algorithm 1: Learning algorithm of CENET

```
Input: Observed graph quadruples set $\mathcal{G}$, entity set $\mathcal{E}$, relation type set $\mathcal{R}$, hyper-paratermeter $\alpha$, and $\lambda$.
Output: A trained network.
Initiate parameters of network $Net$;
for each $(s, p, o, t)$ in $\mathcal{G}$ do
    Compute $\mathcal{H}_t^{s,p}$, $\mathbf{F}_t^{s,p}$, and $\mathbf{Z}_t^{s,p}$ for query $(s, p, ?, t)$ according to Eq.3, 4, and 5 respectively;
    Label $I_q$ for query $(s, p, ?, t)$ using $\mathcal{H}_t^{s,p}$;
end for
while loss does not converge do
    Compute $\mathbf{H}_{his}^{s,p}$ and $\mathbf{H}_{nhis}^{s,p}$ using $\mathbf{Z}_t^{s,p}$ according to Eq.6 and 7 for each $(s, p, o, t)$ in $\mathcal{G}$;
    Compute $\mathbf{v}_q$ using $\mathbf{F}_t^{s,p}$ according to Eq.10;
    Compute $\mathcal{L}^{ce}$ using $\mathbf{H}_{his}^{s,p}$ and $\mathbf{H}_{nhis}^{s,p}$ according to Eq.8;
    Compute $\mathcal{L}^{sup}$ using $\mathbf{v}_q$ according to Eq.12;
    $\mathcal{L} \leftarrow \alpha \cdot \mathcal{L}^{ce} + (1 - \alpha) \cdot \mathcal{L}^{sup}$;
    Optimize $Net$ according to $\mathcal{L}$;
end while
Freeze parameters of $Net$ except the classification layer in the second stage;
Train the classification layer in $Net$ according to $I_q$ and $\mathbf{v}_q$ with binary cross-entropy;
return $Net$;
```

3.4 Parameter Learning and Inference

We minimize the loss function in the first stage:

$$\mathcal{L} = \alpha \cdot \mathcal{L}^{ce} + (1 - \alpha) \cdot \mathcal{L}^{sup}, \qquad (14)$$

where α is a hyper-parameter ranging from 0 to 1 to balance different losses. As to the second stage, we choose binary cross-entropy with sigmoid activation to train the binary classifier. Taking the prediction of object entities as an example, the detailed training process of CENET is provided in Algorithm 1 (See Appendix 2 for the computational complexity). Such a training process is also used to predict the missing subject entities in the experiments.

As can be seen from Figure 2, the middle part illustrates the inference process that receives the distribution $\mathbf{P}_t^{s,p}$ and the mask vector $\mathbf{B}_t^{s,p}$ from both sides respectively. Then, CENET will choose the object with the highest probability as the final prediction $\hat{o}$:

$$\mathbf{P}(o|s, p, \mathbf{F}_t^{s,p}) = \mathbf{P}_t^{s,p}(o) \cdot \mathbf{B}_t^{s,p}(o), \qquad (15)$$

$$\hat{o} = argmax_{o \in \mathcal{E}} \mathbf{P}(o|s, p, \mathbf{F}_t^{s,p}). \qquad (16)$$

Additionally, it is possible that a poor classifier of the second stage of historical contrastive learning may deteriorate the performance when wrongly masking the expected object entities. Thus, there is a compromised substitution:

$$\mathbf{P}(o|s, p, \mathbf{F}_t^{s,p}) = \mathbf{P}_t^{s,p}(o) \cdot softmax(\mathbf{B}_t^{s,p})(o). \qquad (17)$$

We call the former version in Equation 15 *hard-mask*, the latter in Equation 17 *soft-mask*. The hard-mask can reduce the search space and the soft-mask can obtain a more convincing distribution which makes the model more conservative.

4 Experiments

This section conducts a series of experiments to validate the performance of CENET. We first present the experimental settings and then compare CENET with a wide selection of TKG models. After that, the ablation study is implemented to evaluate the effectiveness of various components. Finally, the analysis of hyper-parameter is discussed. All our datasets and codes are publicly available[1].

4.1 Experimental Settings

Datasets and Baselines We select five benchmark datasets, including three event-based TKGs and two public KGs. These two types of datasets are constructed in different ways. The former three event-based TKGs consist of *Integrated Crisis Early Warning System* (ICEWS18 (Boschee et al. 2015) and ICEWS14 (Trivedi et al. 2017)) and *Global Database of Events, Language, and Tone* (GDELT (Leetaru and Schrodt 2013)) where a single event may happen at any time. The last two public KGs (WIKI (Leblay and Chekol 2018) and YAGO (Mahdisoltani, Biega, and Suchanek 2014)) consist of temporally associated facts which last a long time and hardly occur in the future. Table 1 provides the statistics of these datasets.

Dataset	Entities	Relation	Training	Validation	Test
ICEWS18	23,033	256	373,018	45,995	49,545
ICEWS14	12,498	260	323,895	-	341,409
GDELT	7,691	240	1,734,399	238,765	305,241
WIKI	12,554	24	539,286	67,538	63,110
YAGO	10,623	10	161,540	19,523	20,026

Table 1: Statistics of the datasets.

CENET is compared with 15 up-to-date knowledge graph reasoning models, including static and temporal approaches. Static methods include TransE (Bordes et al. 2013), DistMult (Yang et al. 2015), ComplEx (Trouillon et al. 2016), R-GCN (Schlichtkrull et al. 2018), and ConvE (Dettmers et al. 2018). Temporal models include TeMP (Wu et al. 2020), RE-NET (Jin et al. 2020), xERTE (Han et al. 2020), TLogic (Liu et al. 2022), RE-GCN (Li et al. 2021b), TANGO-TuckER (Han et al. 2021), TANGO-Distmult (Han et al. 2021), CyGNet (Zhu et al. 2021), EvoKG (Park et al. 2022), and HIP (He et al. 2021).

Training Settings and Evaluation Metrics All datasets except ICEWS14 are split into training set (80%), validation set (10%), and testing set (10%). The original ICEWS14 is not provided with a validation set. We report a widely used filtered version (Jin et al. 2020; Han et al. 2020; Zhu et al. 2021; He et al. 2021) of Mean Reciprocal Ranks (MRR) and Hits@1/3/10 (the proportion of correct predictions ranked within top 1/3/10). As to model configurations, we set the batch size to 1024, embedding dimension to 200, learning rate to 0.001, and use Adam optimizer. The training epoch for $\mathcal{L}$ is limited to 30, and the epoch for the second stage

[1] https://github.com/xyjigsaw/CENET

Method	ICEWS18				ICEWS14				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	17.56	2.48	26.95	43.87	18.65	1.12	31.34	47.07	16.05	0.00	26.10	42.29
DistMult	22.16	12.13	26.00	42.18	19.06	10.09	22.00	36.41	18.71	11.59	20.05	32.55
ComplEx	30.09	21.88	34.15	45.96	24.47	16.13	27.49	41.09	22.77	15.77	24.05	36.33
R-GCN	23.19	16.36	25.34	36.48	26.31	18.23	30.43	45.34	23.31	17.24	24.96	34.36
ConvE	36.67	28.51	39.80	50.69	40.73	33.20	43.92	54.35	35.99	27.05	39.32	49.44
TeMP	40.48	33.97	42.63	52.38	43.13	35.67	45.79	56.12	37.56	29.82	40.15	48.60
RE-NET	42.93	36.19	45.47	55.80	45.71	38.42	49.06	59.12	40.12	32.43	43.40	53.80
xERTE	36.95	30.71	40.38	49.76	32.92	26.44	36.58	46.05	≫ 1 day			
TLogic	37.52	30.09	40.87	52.27	38.19	32.23	41.05	49.58	22.73	17.65	24.66	32.59
RE-GCN	32.78	24.99	35.54	48.01	32.37	24.43	35.05	48.12	29.46	21.74	32.01	43.62
TANGO-TuckER	44.56	37.87	47.46	57.06	46.42	38.94	50.25	59.80	38.00	28.02	43.91	53.70
TANGO-Distmult	44.00	38.64	45.78	54.27	46.68	41.20	48.64	57.05	41.16	35.11	43.02	52.58
CyGNet	46.69	40.58	49.82	57.14	48.63	41.77	52.50	60.29	50.29	44.53	54.69	60.99
EvoKG	29.67	12.92	33.08	58.32	18.30	6.30	19.43	39.37	11.29	2.93	10.84	25.44
HIP	<u>48.37</u>	<u>43.51</u>	<u>51.32</u>	<u>58.49</u>	<u>50.57</u>	<u>45.73</u>	**54.28**	**61.65**	<u>52.76</u>	<u>46.35</u>	<u>55.31</u>	<u>61.87</u>
CENET	**51.06**	**47.10**	**51.92**	**58.82**	**53.35**	**49.61**	54.07	60.62	**58.48**	**55.99**	**58.63**	**62.96**

Table 2: Experimental results of temporal link prediction on three event-based TKGs. ≫ *1 day* means running time is more than 1 day. The best results are boldfaced, and the results of previous SOTAs are underlined.

of contrastive learning is limited to 20. The value of hyperparameter α is set to 0.2, and λ is set to 2. For the settings of baselines, we use their recommended configurations.

4.2 Results

Results on Event-based TKGs Table 2 presents the MRR and Hits@1/3/10 results of link (event) prediction on three event-based TKGs. Our proposed CENET outperforms other baselines in most cases. It can be observed that many static models are inferior to temporal models because static models do not consider temporal information and their dependency between different snapshots. In the case of temporal models, TeMP is designed to complete missing links (graph interpolation) rather than predict new events, and it thus shows worse performance than extrapolation models. Although xERTE provides a certain degree of predictive explainability, it is computationally inefficient to handle large-scale datasets such as GDELT, whose training set contains more than 1 million samples. In terms of Hits@10, CENET is on par with HIP on these three event-based datasets. Nevertheless, the results of Hits@1 improve the most in our model. CENET achieves up to **8.25%, 8.48%, and 20.80%** improvements of Hits@1 on ICEWS18, ICEWS14, and GDELT respectively. The main reason is that there exist a large proportion of new events without historical events in event-based datasets. CENET learns the historical and non-historical dependency of new events simultaneously, which mines those unobserved underlying factors. In contrast, models including TANGO and HIP perform well in terms of Hits@10 but cannot predict the correct entities exactly, making Hits@1 much lower than ours.

Results on Public KGs CENET also outperforms the baselines in all metrics on WIKI and YAGO. As can be seen from Table 3, CENET significantly achieves the improvements up to **23.68% (MRR), 25.77% (Hits@1), and 7.08% (Hits@3)** over SOTA on public KGs. This is because the recurrence rates in these two datasets are imbalanced (Zhu et al. 2021), and our model can easily handle such data. In terms of the WIKI dataset, 62.3% object entities associated with their corresponding facts (grouped by *(subject, relation)* tuples) have appeared repeatedly at least once in history. In contrast, the recurrence rate of subject entities (grouped by *(object, relation)* tuples) is 23.4%, which hinders many models learning from the historical information when inferring subject entities. CENET can effectively alleviate the problem of the imbalanced recurrence rate because the concurrent learning of historical and non-historical dependency can complement each other to generate the en-

Method	WIKI			YAGO		
	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
TransE	46.68	36.19	49.71	48.97	46.23	62.45
DistMult	46.12	37.24	49.81	59.47	52.97	60.91
ComplEx	47.84	38.15	50.08	61.29	54.88	62.28
R-GCN	37.57	28.15	39.66	41.30	32.56	44.44
ConvE	47.57	38.76	50.10	62.32	56.19	63.97
TeMP	49.61	46.96	50.24	62.25	55.39	64.63
RE-NET	51.97	48.01	52.07	65.16	63.29	65.63
xERTE	≫ 1 day			58.75	58.46	58.85
TLogic	<u>57.73</u>	<u>57.43</u>	57.88	1.29	0.49	0.85
RE-GCN	44.86	39.82	46.75	65.69	59.98	68.70
TANGO-TuckER	53.28	52.21	53.61	67.21	65.56	67.59
TANGO-Distmult	54.05	51.52	53.84	<u>68.34</u>	<u>67.05</u>	68.39
CyGNet	45.50	50.48	50.79	63.47	64.26	65.71
EvoKG	50.66	12.21	<u>63.84</u>	55.11	54.37	<u>81.38</u>
HIP	54.71	53.82	54.73	67.55	66.32	68.49
CENET	**68.39**	**68.33**	**68.36**	**84.13**	**84.03**	**84.23**

Table 3: Experimental results of temporal link prediction on two public KGs. See Appendix for more results.

Method	ICEWS18				YAGO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
CENET-his	50.65	**47.15**	51.23	57.42	71.64	70.24	71.81	74.39
CENET-nhis	31.75	24.22	34.01	46.69	61.73	59.64	62.50	65.38
CENET-$\mathcal{L}^{ce}$ (w/o-stage-1)	50.59	46.47	51.58	58.58	75.25	73.96	75.55	77.52
CENET-w/o-stage-2	50.32	46.30	51.29	58.16	77.53	76.12	78.04	79.84
CENET-w/o-CL	49.98	45.89	50.74	57.81	73.29	71.87	73.64	75.90
CENET-random-mask	26.80	24.42	27.47	31.60	39.07	38.31	39.28	40.41
CENET-hard-mask	49.66	46.69	50.78	55.75	**84.13**	**84.03**	**84.23**	**84.24**
CENET-soft-mask	**51.06**	47.10	**51.92**	**58.82**	80.03	79.09	80.30	81.57
CENET-GT-mask	52.75	48.21	53.97	61.84	84.73	84.31	84.76	85.34

Table 4: Ablation study of CENET on ICEWS18 and YAGO.

tity distribution. Also, the probability of selecting unrelated entities is greatly reduced on account of the binary classifier regardless of the imbalanced recurrence rate.

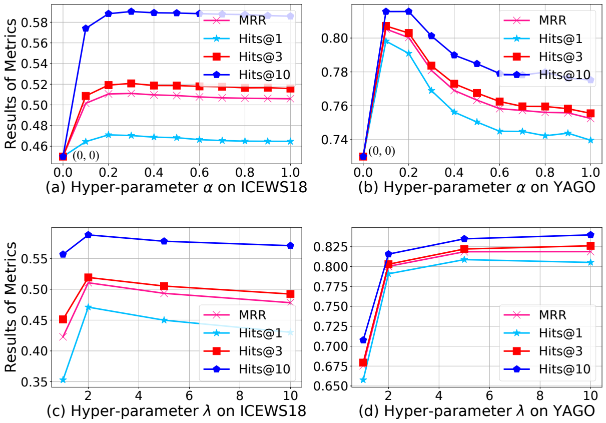
4.3 Ablation Study

We choose ICEWS18 and YAGO to investigate the effectiveness of the historical/non-historical dependency, contrastive learning, and the mask-based inference. Table 4 shows the results of ablation.

CENET-his only considers the historical dependency while CENET-nhis keeps the non-historical dependency. Both of them employ the contrastive learning. The performance of CENET-his is better than CENET-nhis since most events can be traced to their historical events especially in event-based TKGs. Still, for CENET-nhis, it also works on event prediction to a certain extent. Thus, it is necessary to consider both dependencies at the same time. We remove $\mathcal{L}^{sup}$ and only retain $\mathcal{L}^{ce}$ as the variant CENET-$\mathcal{L}^{ce}$. In the case of ICEWS18, the $\mathcal{L}^{ce}$ is capable of achieving high results close to the proposed CENET, while the results in YAGO have **dropped about 7%**. Such results verify the positive influence of the stage 1 in the historical contrastive learning. CENET-w/o-stage-2 is another variant that minimizes $\mathcal{L}^{ce}$ and $\mathcal{L}^{sup}$ without training the binary classifier, which naturally discards the mask-based inference. Such changes cause **1.7% and 3.8% drop** in terms of Hits@1 on ICEWS18 and YAGO respectively. CENET-w/o-CL removing the historical contrastive learning has worse performance than the above two variants. These results prove the significance of our proposed historical contrastive learning. As to the mask strategy. The mask vector is a randomly generated boolean vector in CENET-random-mask. CENET-hard-mask and CENET-soft-mask are our proposed two ways to tackle the mask vector. We use the ground truth in the testing set to generate a mask vector represented by CENET-GT-mask to explore the upper bound of CENET. We can see that untrained model with randomly generated mask vector is counterproductive to the prediction.

4.4 Hyper-parameter Analysis

There are two unexplored hyper-parameters α and λ in CENET. We adjust the values of α and λ respectively to observe the performance change of CENET on ICEWS18 and YAGO. The results are shown in Figure 3. The hyper-parameter α aims at balancing the contribution of $\mathcal{L}^{ce}$ and $\mathcal{L}^{sup}$. Due to the difference of characteristics between event-based TKGs and public KGs, the hyper-parameter α ranging from 0 to 1 leads to different results on these two kinds of datasets. Specifically, $\mathcal{L}^{ce}$ contributes more to event-based TKGs, while $\mathcal{L}^{sup}$ is more friendly to public KGs. Considering that if we remove $\mathcal{L}^{ce}$ i.e. set α to 0, then we cannot obtain the final probability $\mathbf{P}(o|s,p,\mathbf{F}_t^{s,p})$ (and $\mathbf{P}(s|o,p,\mathbf{F}_t^{o,p})$) for inference. To this end, we set α to 0.2. With regard to the hyper-parameter λ, we first fix the value of hyper-parameter α, then the λ is analyzed. We can see that the higher the value of λ, the better the result on YAGO, whereas the worse the result on ICEWS18. Therefore, λ is set to 2.



(a) Hyper-parameter α on ICEWS18 (b) Hyper-parameter α on YAGO

(c) Hyper-parameter λ on ICEWS18 (d) Hyper-parameter λ on YAGO

Figure 3: Results of hyper-parameters α and λ of CENET on ICEWS18 and YAGO.

5 Conclusion and Future Work

In this paper, we propose a novel temporal knowledge graph representation learning model, Contrastive Event Network (CENET), for event forecasting. The key idea of CENET is to learn a convincing distribution of the whole entity set and identify significant entities from both historical and non-historical dependency in the framework of contrastive learning. The experimental results present that CENET outperforms all existing methods in most metrics significantly, especially for Hits@1. Promising future work includes exploring the ability of contrastive learning in knowledge graph, such as finding more reasonable contrastive pairs.

Acknowledgments

This work was supported by NSF China (No. 62020106005, 42050105, 62061146002, 61960206002), 100-Talents Program of Xinhua News Agency, Shanghai Pilot Program for Basic Research - Shanghai Jiao Tong University, and the Program of Shanghai Academic/Technology Research Leader under Grant No. 18XD1401800.

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; and Ward, M. 2015. ICEWS coded event data. *Harvard Dataverse*, 12.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Dasgupta, S. S.; Ray, S. N.; and Talukdar, P. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP)*, 2001–2011.

Deng, S.; Rangwala, H.; and Ning, Y. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1585–1595.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; and Chua, T.-S. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2): 1–30.

Garcia-Duran, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4816–4821.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640.

Han, Z.; Chen, P.; Ma, Y.; and Tresp, V. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Han, Z.; Ding, Z.; Ma, Y.; Gu, Y.; and Tresp, V. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8352–8364.

He, Y.; Zhang, P.; Liu, L.; Liang, Q.; Zhang, W.; and Zhang, C. 2021. HIP Network: Historical Information Passing Network for Extrapolation Reasoning on Temporal Knowledge Graph. In *IJCAI*.

Jia, Z.; Abujabal, A.; Saha Roy, R.; Strötgen, J.; and Weikum, G. 2018. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 1807–1810.

Jin, W.; Qu, M.; Jin, X.; and Ren, X. 2020. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6669–6683.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907.

Leblay, J.; and Chekol, M. W. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, 1771–1776.

Leetaru, K.; and Schrodt, P. A. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2. Citeseer.

Li, Z.; Jin, X.; Guan, S.; Li, W.; Guo, J.; Wang, Y.; and Cheng, X. 2021a. 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.

Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; and Cheng, X. 2021b. 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.

Liu, Y.; Ma, Y.; Hildebrandt, M.; Joblin, M.; and Tresp, V. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4120–4127.

Liu, Z.; Xiong, C.; Sun, M.; and Liu, Z. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2395–2405.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference.

Park, N.; Liu, F.; Mehta, P.; Cristofor, D.; Faloutsos, C.; and Dong, Y. 2022. EvoKG: Jointly Modeling Event Time and

Network Structure for Reasoning over Temporal Knowledge Graphs. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 794–803.

Sadeghian, A.; Armandpour, M.; Colas, A.; and Wang, D. Z. 2021. ChronoR: rotation based temporal knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6471–6479.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Sun, T.; Shao, Y.; Qiu, X.; Guo, Q.; Hu, Y.; Huang, X.-J.; and Zhang, Z. 2020. CoLAKE: Contextualized Language and Knowledge Embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3660–3670.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, 3462–3471. PMLR.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, 2071–2080. PMLR.

Wang, Q.; Li, M.; Wang, X.; Parulian, N.; Han, G.; Ma, J.; Tu, J.; Lin, Y.; Zhang, R. H.; Liu, W.; et al. 2021. COVID-19 Literature Knowledge Graph Construction and Drug Repurposing Report Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*.

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 950–958.

Wu, J.; Cao, M.; Cheung, J. C. K.; and Hamilton, W. L. 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.

Yang, B.; Yih, S. W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations*.

Zhu, C.; Chen, M.; Fan, C.; Cheng, G.; and Zhang, Y. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4732–4740.