future internet

MDPI

Article

MSEN: A Multi-Scale Evolutionary Network for Modeling the Evolution of Temporal Knowledge Graphs

Yong Yu [1,2], Shudong Chen [1,2,*], Rong Du [1], Da Tong [1,2], Hao Xu [1,2] and Shuai Chen [1,2]

[1] Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100191, China; yuyong@ime.ac.cn (Y.Y.); durong@ime.ac.cn (R.D.); tongda@ime.ac.cn (D.T.); xuhao2022@ime.ac.cn (H.X.); chenshuai2022@ime.ac.cn (S.C.)
[2] University of Chinese Academy of Sciences, Beijing 100191, China
* Correspondence: chenshudong@ime.ac.cn

Abstract: Temporal knowledge graphs play an increasingly prominent role in scenarios such as social networks, finance, and smart cities. As such, research on temporal knowledge graphs continues to deepen. In particular, research on temporal knowledge graph reasoning holds great significance, as it can provide abundant knowledge for downstream tasks such as question answering and recommendation systems. Current reasoning research focuses primarily on interpolation and extrapolation. Extrapolation research aims to predict the likelihood of events occurring in future timestamps. Historical events are crucial for predicting future events. However, existing models struggle to fully capture the evolutionary characteristics of historical knowledge graphs. This paper proposes a multi-scale evolutionary network (MSEN) model that leverages Hierarchical Transfer aware Graph Neural Network (HT-GNN) in a local memory encoder to aggregate rich structural semantics from each timestamp's knowledge graph. It also utilizes Time Related Graph Neural Network (TR-GNN) in a global memory encoder to model temporal-semantic dependencies of entities across the global knowledge graph, mining global evolutionary patterns. The model integrates information from both encoders to generate entity embeddings for predicting future events. The proposed MSEN model demonstrates strong performance compared to several baselines on typical benchmark datasets. Results show MSEN achieves the highest prediction accuracy.

Keywords: knowledge graph; temporal knowledge graph; knowledge graph reasoning; evolution of temporal knowledge graphs

Citation: Yu, Y.; Chen, S.; Du, R.; Tong, D.; Xu, H.; Chen, S. MSEN: A Multi-Scale Evolutionary Network for Modeling the Evolution of Temporal Knowledge Graphs. *Future Internet* **2023**, *15*, 327. https://doi.org/10.3390/fi15100327

Academic Editors: Edson Talamini, Letícia De Oliveira and Filipe Portela

Received: 1 September 2023
Revised: 28 September 2023
Accepted: 28 September 2023
Published: 30 September 2023

1. Introduction

Temporal knowledge graphs (TKGs) are a type of dynamic knowledge structure in which knowledge may change over time. Compared with traditional static knowledge graphs, TKGs take into account the influence of the time dimension. TKGs with multiple timestamps can be represented as a series of knowledge graph (KG) snapshots, where each snapshot contains all facts within the same timestamp.

For static knowledge graph completion, methods such as DiSMult [1], ComplEx [2], RGCN [3], ConvE [4], and RotatE [5] have been proposed. TKG reasoning tasks can perform temporal reasoning and evolution analysis of knowledge, mainly divided into interpolation and extrapolation [6]. Interpolation is to complete missing events based on existing facts; extrapolation is to predict future events based on past events. Predicting future events based on the evolution of historical knowledge graphs is very important and challenging. Obtaining possible future events in a timely manner based on temporal relations can support applications such as financial risk control, disaster relief, etc.

This paper focuses on extrapolation research. The challenge of this task is how to better obtain relevant historical information that can reflect future behaviors. Currently, there are two methods that can be used for TKG extrapolation, namely, query-specific and

entire graph-based methods [7]. The former includes RE-NET [6], CyGNet [8], xERTE [9], TITer [10], CENET [11], etc., using high-frequency historical facts related to a query's topics and relations to predict future trend, ignoring structural dependencies within snapshot; the latter includes RE-GCN [12], CEN 7, TANGO [13], TiRGN [14], L2TKG [15], etc., using historical knowledge graph as input to obtain the evolutionary patterns of TKG.

The above two methods do not consider the historical features from different perspectives, ignoring potentially useful information. Relying solely on either local or global knowledge graphs may result in information loss, affecting prediction results. However, future events may depend on both local structural dependencies and global temporal patterns. Historical knowledge from varying perspectives likely influences future occurrences to differing degrees.

In this work, we propose a model to capture multi-scale evolutionary features of historical information, called Multi-Scale Evolutionary Network (MSEN). Typically, different features manifest at different scales, providing diverse semantic information. The local memory focuses on contextual knowledge, while the global memory recalls long-range patterns. By integrating both scales, MSEN can better model TKG evolution for reasoning.

The main contributions are:

(1) In the local memory encoder, a hierarchical transfer aware graph neural network (HT-GNN) is proposed to enriches structural semantics within each timestamp;
(2) In the global memory encoder, a time related graph neural network (TR-GNN) is proposed to extract periodic and non-periodic temporal patterns globally across timestamps;
(3) Through experiments on typical event-based datasets, the paper demonstrates the effectiveness of our proposed model MSEN.

2. Related Work

In this section, this paper reviews the existing methods for TKG Reasoning under the interpolation setting and the extrapolation setting, and summarizes the methods for event prediction.

2.1. TKG Reasoning under the Interpolation Setting

For the interpolation setting, models try to complete missing facts in the past timestamps. TTransE [16], TA-TransE [17], and TA-DistMult [17] embed temporal information into scoring functions and relations, respectively, to obtain the evolution information of facts; HyTE [18] proposes a time-aware model based on hyperplanes, and projects entities and subgraphs onto specific hyperplanes; TNTComplEx [19] introduces complex numbers to factorize a fourth-order tensor to generate timestamp embeddings. DE-SimplE [20] accomplishes the task of knowledge graph completion by making feature representations of temporal information; ChronoR [21] captures the rich interactions between time and multi-relational features in knowledge graphs by learning embeddings of entities, relations, and time; TempCaps [22] is a capsule network model that uses information retrieved through dynamic routing to construct entity embeddings. TKGC [23] maps entities and relations in TKG to multivariate Gaussian processes, thereby simulating overall and local trends. However, these models cannot obtain embeddings of future events.

2.2. TKG Reasoning under the Extrapolation Setting

Recently, several studies have attempted to infer future events, and existing methods can be divided into two categories: query-specific and entire graph-based methods [7] They all use historical information to predict future trends.

2.2.1. Query-Specific Models

The methods focus on obtaining historical information of a specific query. GHNN [24] predicts the occurrence of future events through obtaining the dynamic sequence of evolving graphs; xERTE [9] uses iterative sampling and attention propagation techniques to

extract closed subgraphs around specific queries. Both CluSTeR [25] and TITer [10] are based on reinforcement learning to discover paths for specific queries in the historical KG; RE-NET [6] can effectively extract sequential and structural information related to a specific query in the TKG to solve the entity prediction task for a given query, but cannot model long-term KG information; CyGNet [8] combines two reasoning modes, copy mode and generation mode, making predictions based on repeated entities in the historical vocabulary or the entire entity vocabulary, but ignores higher-order semantic dependencies between co-occurring entities; CENET [11] proposes a novel historical comparative learning approach for knowledge graph reasoning that uses temporal reasoning patterns mined from historical knowledge graph versions to make more accurate predictions.

2.2.2. Entire Graph-Based Models

The methods focus on obtaining the history of the latest KG of fixed length. Glean [26] introduces unstructured event description information to enrich entity representations, but description information is not available for all events in real applications; TANGO [13] extends neural ordinary differential equations to multi-relational graph convolutional networks to model structural information, but the model's utilization of long-distance information is still limited; RE-GCN [12] and CEN [7] both model the evolution of representations of all entities by capturing fixed-length historical information, but neither of these two models considers the evolutionary features of historical information from different perspectives; TiRGN [14] introduces a new recurrent graph network model that incorporates local and global historical patterns in temporal knowledge graphs to improve the accuracy of link prediction over time.

Some recent works have incorporated both local and global modeling for TKG reasoning. However, MSEN proposes new designs for each module and their integration, achieving better results. The hierarchical design in the local module enriches semantic learning, while the periodic and aperiodic temporal modeling in the global module improves temporal encoding.

3. The Proposed Method

In this section, this paper mainly introduces Temporal Knowledge Graph (TKG) Reasoning under the extrapolation setting and the proposed method. Table 1 shows some of the important symbols of the paper and the corresponding explanations.

Table 1. Important Symbols.

Symbol	Explanation
G_n	KG at timestamp n
E, R	The set of entities, relations
Γ_t	The set of facts at timestamp t
$h_{s,t}, h_{r,t}$	Embedding representation of entity s, relation r at timestamp t
$\left(h_{o,t}^{ent}\right)^l$	Embedding representation of entity o at l-th layer
H_t, R_t	The set of entity embedding, relation embedding at time t
T^p, T^{ap}	The periodic time vector, the non-periodic time vector
$h_{o,t}^{global}$, $h_{o,t}^{local}$	Global memory embeddings, local memory embeddings

3.1. Problem Formulation

A TKG $G = \{G_0, G_1, G_2, G_3, \cdots G_n\}$ is a KG sequence. In each KG, $G_t = \{E, R, \Gamma_t\}$ is a directed multi-relational graph, where E is the set of entities; R is the set of relations; Γ_t is the set of facts at timestamp t. A fact in Γ_t can be expressed as a quaternion (s,r,o,t), where $s, o \in E$ and $r \in R$. The goal of the TKG entity reasoning task is to complete tail entity prediction (s,r,?,t_m) or head entity prediction (?,r,o,t_m) based on the historical knowledge graph sequence $\{G_0, G_1, G_2, G_3, \cdots G_{t_m-1}\}$. For each quaternion (s,r,o,t), an inverse quaternion (o, r^{-1}, s, t) is added to the KG. Therefore, the prediction of head entity (?,r,o,t) can be transformed into the prediction of tail entity $(o, r^{-1}, ?, t)$.

3.2. Model Overview

In this section, a Multi-Scale Evolutionary Network (MSEN) is proposed to solve the problem of feature extraction at different scales. The overall framework of a MSEN is illustrated in Figure 1. There are three parts in our model: (1) A local memory encoder that applies a hierarchical transfer-aware graph neural network (HT-GNN) to obtain structural features within each timestamp's knowledge graph snapshot. The local encoder also performs temporal encoding of entities and relations using a gated recurrent unit (GRU); (2) a global memory encoder that employs a time-related graph neural network (TR-GNN) to mine temporal-semantic dependencies across the historical knowledge graph, yielding entity and relation embeddings; (3) a decoder that integrates the local and global encoder outputs and utilizes a scoring function to predict future entities.

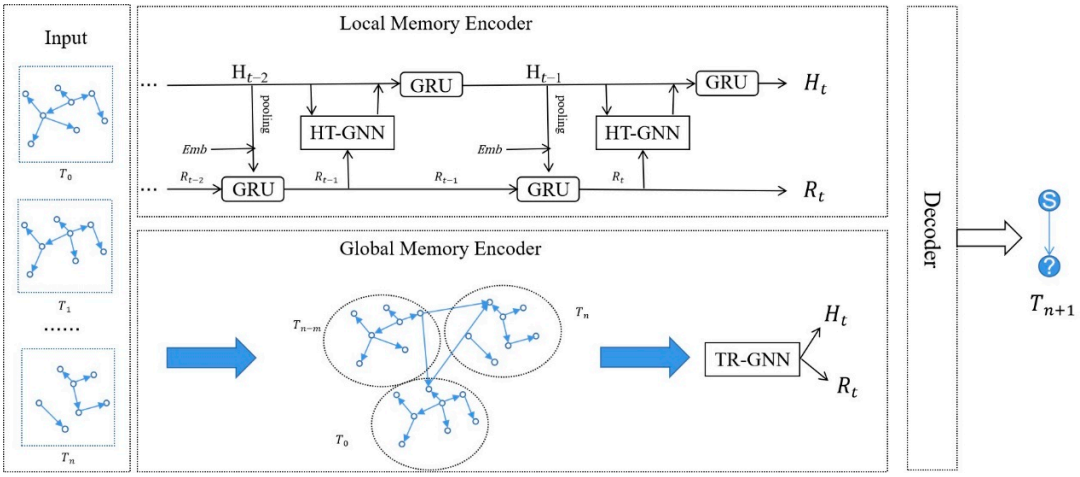

Figure 1. Illustration of the proposed MSEN model.

Specifically, the input is a series of temporal knowledge graph snapshots. The local encoder extracts per-timestamp features using HT-GNN and temporal dynamics via the GRU. The global encoder identifies long-range patterns using TR-GNN. Finally, the decoder combines local and global embeddings to predict future entities using the scoring function. This multi-scale approach allows MSEN to capture both contextual knowledge and global evolutions for enhanced temporal reasoning.

3.3. Local Memory Encoder

The local memory encoder focuses on obtaining the memories of the m historical knowledge graphs adjacent to the query time. It aggregates each knowledge graph snapshot to capture structural dependencies within timestamps. Additionally, sequence neural networks are utilized to learn temporal features of entities and relations, enabling the model to better represent temporal dynamics across knowledge graphs.

3.3.1. HT-GNN

Obtaining informative embedding representations for each temporal knowledge graph is critical for effective local encoding. To better capture semantic information, we propose a novel graph neural network-based knowledge graph embedding method called Hierarchical Transfer-Aware Graph Neural Network (HT-GNN). As illustrated in Figure 2, HT-GNN performs sequential aggregation at each layer: entity aggregation, relation aggregation, convolution aggregation, and sum aggregation. The blue nodes serve as aggregation centers. Different aggregation results are merged as input to the next layer. Finally, the outputs from the multi-layer aggregations are combined as the overall model output.

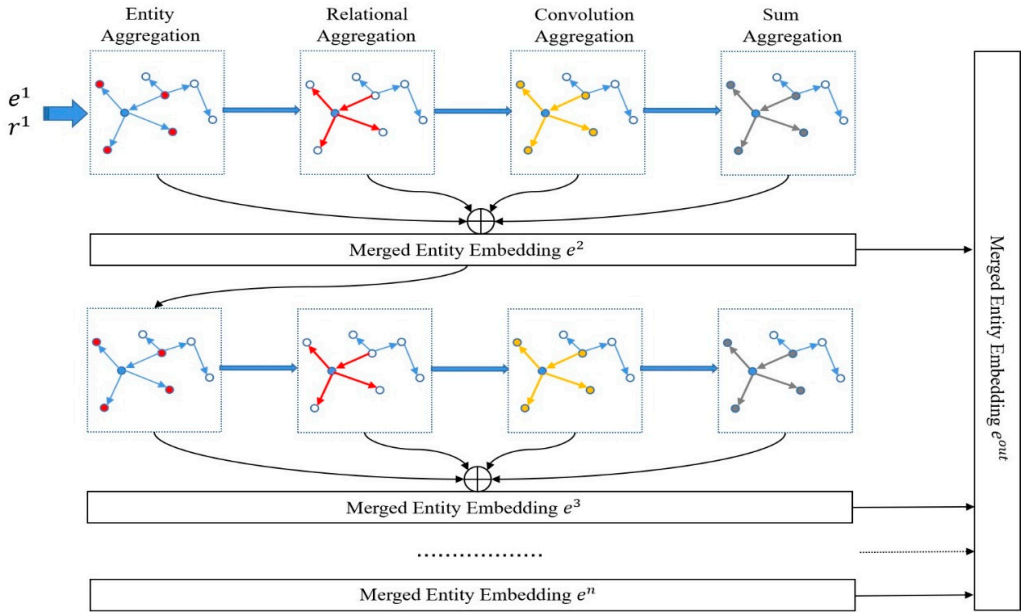

Figure 2. Architecture of the proposed HT-GNN model.

Specifically, entity aggregation allows nodes to incorporate semantics from adjacent entities based on co-occurrence, capturing semantic dependencies. The entity aggregation function and weight calculation are:

$$\left(h_{o,t}^{ent}\right)^l = \sigma\left(\sum_{(s,r)\in N_t} \alpha_{ent}^l w_{ent}^l h_{s,t}^l\right), \tag{1}$$

where $h_{o,t}^{ent}$ is embedding representation of entity o. $h_{s,t}^l$ is the embedding representation of entity s which is a neighboring node of entity o. w_{ent}^l is a learnable weight matrix. N_t represents the neighboring entities and relations of entity o. σ is a nonlinear activation function. α_{ent}^l is the aggregation weight, calculated as follows:

$$\alpha_{ent}^l = \frac{\exp\left(h\left(L\left(\left[h_{s,t}^l \middle\| h_{o,t}^l\right]\right)\right)\right)}{\sum_{(s,r)\in N_t} \exp\left(h\left(L\left(\left[h_{s,t}^l \middle\| h_{o,t}^l\right]\right)\right)\right)}, \tag{2}$$

where $h(\cdot)$ is the LeakyRelu activation function. $L(\cdot)$ represents the fully connected function. $\|$ represents the concatenation operation.

The result of entity aggregation is used as input to relation aggregation. The relation aggregation infuses relation semantics into the entity representations, thereby obtaining the co-occurrence of each node and relation in the knowledge graph. The relation aggregation function and corresponding weight calculation method used in the paper are as follows:

$$\left(h_{o,t}^{rel}\right)^l = \sigma\left(\sum_{(s,r)\in N_t} \alpha_{rel}^l w_{rel}^l h_{r,t}^l\right), \tag{3}$$

where $h_{r,t}^l$ is the embedding representation of the neighboring relation r of entity o. w_{rel}^l is a learnable weight matrix. α_{rel}^l is the relation aggregation weight, calculated as follows:

$$\alpha_{rel}^l = \frac{\exp\left(h\left(L\left(\left[h_{r,t}^l \middle\| h_{o,t}^l\right]\right)\right)\right)}{\sum_{(s,r)\in N_t} \exp\left(h\left(L\left(\left[h_{r,t}^l \middle\| h_{o,t}^l\right]\right)\right)\right)} \tag{4}$$

The result of relation aggregation is used as input to convolution aggregation. For convolution aggregation, it obtains the hidden relation between entity s, relation r, and

entity o, which enriches the semantic information of entities. The convolution aggregation function and weight calculation method used in the paper are as follows:

$$\left(h_{o,t}^{con}\right)^{l} = \sigma\left(\sum_{(s,r)\in N_t} \alpha_{con}^{l} w_{con}^{l} con(s,r,o)\right), \tag{5}$$

where $con(\cdot)$ is the convolution operation. w_{con}^{l} is a learnable weight matrix. α_{rel}^{l} is the aggregation weight, calculated as follows:

$$\alpha_{con}^{l} = \frac{\exp\left(h\left(L\left(\left[h_{s,t}^{l} \middle\| h_{r,t}^{l} \middle\| h_{o,t}^{l}\right]\right)\right)\right)}{\sum_{(s,r)\in N_t} \exp\left(h\left(L\left(\left[h_{s,t}^{l} \middle\| h_{r,t}^{l} \middle\| h_{o,t}^{l}\right]\right)\right)\right)}. \tag{6}$$

The result of convolution aggregation is used as input to sum aggregation. For sum aggregation, information is obtained through relational paths between entities, and sum-obtained information is similar to translation model TransE [27]. The sum aggregation function and corresponding weight calculation method used in the paper are as follows:

$$\left(h_{o,t}^{sum}\right)^{l} = \sigma\left(\sum_{(s,r)\in N_t} \alpha_{sum}^{l} w_{sum}^{l} \left(h_{s,t}^{l} + h_{r,t}^{l}\right)\right), \tag{7}$$

where $h_{s,t}^{l}$ and $h_{r,t}^{l}$ are the embedding representations of neighboring entity s and relation r of entity o. w_{sum}^{l} is a learnable weight matrix. α_{sum}^{l} is the relation aggregation weight, calculated as follows:

$$\alpha_{sum}^{l} = \frac{\exp\left(h\left(L\left(\left[h_{s,t}^{l} + h_{r,t}^{l} \middle\| h_{o,t}^{l}\right]\right)\right)\right)}{\sum_{(s,r)\in N_t} \exp\left(h\left(L\left(\left[h_{s,t}^{l} + h_{r,t}^{l} \middle\| h_{o,t}^{l}\right]\right)\right)\right)}. \tag{8}$$

Finally, the final entity embedding can be expressed as:

$$\left(h_{o,t}\right)^{l+1} = \left(h_{o,t}^{ent}\right)^{l} + \left(h_{o,t}^{rel}\right)^{l} + \left(h_{o,t}^{con}\right)^{l} + \left(h_{o,t}^{sum}\right)^{l}. \tag{9}$$

3.3.2. Local Memory Representation

In order to obtain the temporal features of entities and relations in adjacent historical knowledge graphs, following RE-GCN [12], GRU is adopted to update the representations of entities and relations. The update of entity embedding representations is as follows:

$$H_{t+1} = GRU\left(H_t, H_t^{HT-GNN}\right), \tag{10}$$

where H_{t+1} is the entity embedding at time t + 1. H_t^{HT-GNN} is the entity embedding obtained after aggregating the knowledge graph using HT-GNN.

Similarly, in order to obtain the temporal embedding representation of the relationship, the following method is used for calculation:

$$R_{t+1} = GRU\left(R_t, \acute{R}\right), \tag{11}$$

$$\acute{R} = pooling(H_t), \tag{12}$$

where R_{t+1} is the relation embedding at time t + 1. $pooling(\cdot)$ represents mean pooling operation to retain all entity embeddings related to the relationship r and average them.

3.4. Global Memory Encoder

To extract cross-timestamp entity dependencies, we construct a global knowledge graph by associating historical snapshots containing the same entities [28]. We introduce a Time-Related Graph Neural Network (TR-GNN) to effectively encode this global graph.

The global graph aggregates facts across timestamps, including periodic and non-periodic occurrences. To fully utilize this temporal information, we design separate periodic and aperiodic time vector representations:

$$T^{p} = f(w_1 t + \varphi_1), \tag{13}$$

$$T^{ap} = w_2 t + \varphi_2, \tag{14}$$

where T^{p} is the periodic time vector; T^{ap} is the non-periodic time vector. w and φ are learnable vectors, representing frequency and phase respectively. $f(\cdot)$ is a periodic activation function, which is sin function in the paper. t represents the time interval $\left|T_i - T_j\right|$ between two connected entities in the global knowledge graph.

To obtain the time-semantic dependencies between entities in the global knowledge graph, the paper designs TR-GNN to encode the knowledge graph:

$$(h_{o,t})^{l+1} = \sigma\left(\sum_{(s,r)\in N_t}(\alpha_1^l h_{s,t}^l + \alpha_2^l \left(h_{r,t}^l\right)')\right) + w_2 (h_{o,t})^l\Big), \tag{15}$$

$$\left(h_{r,t}^l\right)' = h_{r,t}^l + T^{p}, \tag{16}$$

where $(h_{o,t})^{l+1}$ is the encoded entity embedding. w_2 is a learnable weight parameter. α_1^l and α_2^l are weights for entities and relations respectively, calculated as follows:

$$\alpha_1^l = \frac{\exp\left(h\left(L\left(\left[h_{s,t}^l \middle\| h_{o,t}^l \| T^{ap}\right]\right)\right)\right)}{\sum_{(s,r)\in N_t}\exp\left(h\left(L\left(\left[h_{s,t}^l \middle\| h_{o,t}^l \| T^{ap}\right]\right)\right)\right)}, \tag{17}$$

$$\alpha_2^l = \frac{\exp\left(h\left(L\left(\left[h_{r,t}^l \middle\| h_{o,t}^l \| T^{ap}\right]\right)\right)\right)}{\sum_{(s,r)\in N_t}\exp\left(h\left(L\left(\left[h_{r,t}^l \middle\| h_{o,t}^l \| T^{ap}\right]\right)\right)\right)}. \tag{18}$$

Using the non-periodic time vector to compute weights for entities and relations better captures the temporal-semantic relationships between entities. Finally, the updated entity embeddings are nonlinearly transformed to obtain the embedding representations of each entity in the global knowledge graph.

3.5. Gating Integration

In order to better integrate the embeddings vectors obtained by the local memory encoder and the global memory encoder for reasoning, the paper applies a learnable gating function [29] to adaptively adjust the weights of local memory embeddings and global memory embeddings, finally obtaining a vector:

$$h_{o,t} = \lambda(g) \circledcirc h_{o,t}^{global} + (1 - \lambda(g)) \circledcirc h_{o,t}^{local}, \tag{19}$$

$$h_{r,t} = \lambda(g) \circledcirc h_{r,t}^{global} + (1 - \lambda(g)) \circledcirc h_{r,t}^{local}, \tag{20}$$

where $\lambda(\cdot)$ is the sigmoid function, aiming to restrict the range of each element to [0, 1]. g is the gate parameter to adjust the weights. $\circledcirc$ is element-wise multiplication.

3.6. Scoring Function and Loss Function

In this section, the paper mainly introduces the details of scoring function and the loss function. In order to obtain the probability of an event occurring at timestamp t + 1 in the future, ConvTransE [12] is utilized as the decoder to calculate the probability of the existence between entities at timestamp t + 1 under the relation r:

$$P_{t+1}(o|s,r) = \lambda(h_{o,t+1}\mathrm{ConvTranE}(h_{s,t+1}, h_{r,t+1})), \tag{21}$$

where $\lambda(\cdot)$ is the sigmoid function. $h_{s,t+1}$ and $h_{r,t+1}$ are the embedding representations of entities and relations at timestamp t + 1. $h_{o,t+1}$ is the embedding representations of entities after global and local fusion.

Following RE-GCN [12], the loss function during training is defined as:

$$L_e = -\sum_{t=0}^{T}\sum_{(s,r,0,t+1)\in G_{t+1}} \log P_{t+1}(o|s,r), \tag{22}$$

$$L_r = -\sum_{t=0}^{T}\sum_{(s,r,0,t+1)\in G_{t+1}} \log P_{t+1}(r|s,o). \tag{23}$$

The final loss is $L = \lambda L_e + (1-\lambda)L_r + \mu\|\Theta\|_2$, where L_e is the loss for predicting entities, L_r is the loss for predicting relations, λ is the hyperparameter balancing different tasks. $\|\Theta\|_2$ is L_2 norm, and μ is to control the penalty term. The overall training algorithm is shown in Algorithm 1.

Algorithm 1. Training of MSEN

Input: The train set: $\{G_0, G_1, G_2, G_3, \cdots G_{t_m-1}\}$; the number of epoch n.
Output: The minimum Loss on the train set.

1. $E, R, \Gamma_t \longleftarrow$ Init.normal_(), $loss_{min} = 0$ //Initialize the model' parameters
2. For each $i \in [1, T]$ do
   /* Local Memory Encoder */
3. $(h_{o,t})^{l+1} = (h_{o,t}^{ent})^l + \left(h_{o,t}^{rel}\right)^l + (h_{o,t}^{con})^l + (h_{o,t}^{sum})^l$ //Obtaining the semantic information using HT-GNN
4. $H_{t+1} = GRU\left(H_t, H_t^{HT-GNN}\right)$ //local memory representation
   /* Global Memory Encoder */
5. Construct a global graph G;
6. $(h_{o,t})^{l+1} = \sigma\left(\sum_{(s,r)\in N_t}(\alpha_1^l h_{s,t}^l + \alpha_2^l\left(h_{r,t}^l\right)\prime\right) + w_2(h_{o,t})^l\right)$ //Obtaining time-semantic dependencies using TR-GNN
7. $h_{o,t} = \lambda(g) \circledcirc h_{o,t}^{global} + (1-\lambda(g)) \circledcirc h_{o,t}^{local}$ //Gating Integration
8. $P_{t+1}(o|s,r) \longleftarrow$ ConvTranE()
9. Loss$\longleftarrow$Calculate the sum of L_e and L_r
10. Update parameters
11. End for
12. Return Loss

4. Experiments

In this chapter, the paper compares MSEN with several baselines on two datasets. Meanwhile, ablation experiments are conducted to analyze the ability of acquiring semantic information at different scales. Comparison experiments evaluate the effects of different GCNs. In addition, the paper also studies the parameter settings of the model.

4.1. Datasets

To evaluate the effect of the MSEN model, experiments are conducted on two typical event-based datasets, including: ICEWS14 [17], ICEWS18 [6]. Both datasets are from the event-based dataset Integrated Crisis Early Warning System [30], with the time interval of 24 h. The statistical details of the datasets are listed in Table 2.

Table 2. The statistical details of the datasets.

Datasets	Train	Valid	Test	Ent	Rel	Time Gap
ICEWS14	74,845	8514	7371	6869	230	24 h
ICEWS18	373,018	45,995	49,545	23,033	256	24 h

4.2. Baselines

The paper compares MSEN with two types of methods: static KG reasoning and TKG reasoning methods.

For static KG methods: DiSMult [1], ComplEx [2], RGCN [3], ConvE [4], and RotatE [5] are selected.

For TKG reasoning under the extrapolation setting: CyGNet [8], RE-NET [6], xERTE [9], RE-GCN [12], TITer [10], CENET [11], and TiRGN [14] are selected.

4.3. Evaluation Metrics

This paper adopts mean reciprocal rank (MRR) and Hit@k (k is 1 and 10) to evaluate the performance of models for entity prediction. For the fairness of the comparison, the ground truth is used when performing multi-step reasoning, and the experimental results are reported under the time-aware filtered setting. MRR is the average of the reciprocal ranks of all relevant candidate entities, measuring the rank of the highest-ranking relevant entity. Hit@k represents the proportion of cases where the true target entity is included in the top k predicted candidates, reflecting the model's ability to contain the target entity within the top k.

4.4. Implementation Details

In the experiments, the embedding dimension was fixed at 200 for all methods. The Adam optimizer was used for parameter training with a learning rate of 0.001. The hyper-parameter for balancing different tasks was set to 0.7. To ensure fairness of comparison with other models, for the local memory encoder, the number of adjacent historical knowledge graphs obtained was 3 and 6 on the ICEWS14 and ICEWS18 datasets respectively; for the global memory encoder, the number of adjacent historical knowledge graphs obtained was 10 and 10 on the ICEWS14 and ICEWS18 datasets, respectively. The number of layers in HT-GNN was 2 on both datasets. The number of layers in TR-GNN was 2 and 3 on the ICEWS14 and ICEWS18 datasets respectively. For the ConvTranE decoder, the number of channels used was 50 and the kernel size was 2×3.

5. Results

5.1. Performance Comparison

As shown in Tables 3 and 4, entity prediction experiments are conducted on facts at future timestamps. The tables list the predictive performance of MSEN and the baseline models on ICEWS14 and ICEWS18 datasets. The results of baseline models in the table are from [15].

Table 3. Performance on the ICEWS14 dataset in terms of MRR (%), Hit@1 (%), and Hit@10 (%).

Model	MRR	Hit@1	Hit@10
DistMult	25.31	17.93	42.22
ComplEx	32.33	23.21	52.37
RGCN	28.14	19.43	46.02
ConvE	30.93	21.74	50.18
RotatE	27.53	18.60	47.62
CyGNet	37.65	27.43	57.90
RE-NET	39.86	30.11	58.21
xERTE	40.79	32.70	57.30
TITer	41.73	32.74	58.44
RE-GCN*	41.99	32.93	61.92
CENET	41.30	32.58	58.22
TiRGN*	43.18	33.12	62.24
Our Method	**46.68**	**35.29**	**68.96**

The mark * in the table represents removing static information from models for fair comparison [15]. The best results are in bold.

Table 4. Performance on the ICEWS18 dataset in terms of MRR (%), Hit@1 (%), and Hit@10 (%).

Model	MRR	Hit@1	Hit@10
DistMult	16.59	10.01	31.69
ComplEx	18.84	11.41	25.78
RGCN	18.04	8.57	35.68
ConvE	24.28	15.61	44.59
RotatE	15.35	7.10	33.09
CyGNet	27.12	17.21	46.85
RE-NET	29.78	19.73	48.46
xERTE	29.31	21.03	46.48
TITer	29.98	22.05	44.83
RE-GCN*	30.13	19.11	48.86
CENET	29.65	19.98	48.23
TiRGN*	32.22	22.24	51.88
Our Method	**34.05**	**22.83**	**56.80**

The mark * in the table represents removing static information from models for fair comparison [15]. The best results are in bold.

In the experimental results, the MSEN model outperforms other baseline models. The results show the effectiveness of the MSEN model in predicting entities at future timestamps. Compared with TiRGN*, MSEN improves the MRR by 3.5 and 1.83 percentage points on ICEWS14 and ICEWS18, respectively.

Probing deeper, we evaluated recall capabilities against TiRGN* on ICEWS14. At recall@1, MSEN achieved 32.78%, while TiRGN* achieved 34.88%. For recall@10, the models achieved near identical results, with MSEN at 61.98% versus 62.21% for TiRGN*. While marginally behind in recall@k, MSEN still demonstrates competitive retrieval ability.

5.2. Ablation Study

The encoder of the MSEN model is mainly composed of two submodules: Local Memory Encoder (LME) and Global Memory Encoder (GME). To further analyze the effect of each module on the final entity prediction results, the paper reports the results of LME (HT-GNN) and GME (TR-GNN) on the ICEWS14 dataset using MRR and Hit@k (k = 1, 3, 10) metrics. LME (HT-GNN) represents removing the GME module from the MSEN model; GME (TR-GNN) represents removing the LME module from the MSEN model. As can be seen from Table 5, the performance of LME (HT-GNN) is worse than MSEN, because this part only extracts local features and the obtained embedding representations have less semantic information. The performance of GME (TR-GNN) has greatly improved compared to LME (HT-GNN), but integrating both using a gating function can further improve the accuracy of entity prediction.

Table 5. Performance of GNN-based model in LME and GME in terms of MRR (%), Hit@1 (%), Hit@3 (%), and Hit@10 (%).

Model	ICEWS14			
	MRR	Hit@1	Hit@3	Hit@10
LME (RGCN)	38.69	29.34	43.11	56.68
LME (KBGAT)	38.74	29.50	43.25	56.26
LME (CompGCN(add))	39.08	29.78	43.75	56.77
LME (HT-GNN)	39.91	30.45	44.48	57.91
GME (RGCN)	43.84	31.29	49.90	69.06
GME (KBGAT)	43.41	31.53	48.83	67.08
GME (CompGCN(add))	43.72	31.11	49.86	68.98
GME (HRGNN)	44.93	33.02	50.58	68.97
GME (TR-GNN)	45.24	33.36	50.88	69.22

To study the effects of HT-GNN and TR-GNN in LME and GME, respectively, the paper compares these two models with different types of GNN-based models on the ICEWS14 dataset using MRR and Hit@k (k = 1, 3, 10) metrics. In LME, RGCN [3], KBGAT [31], and CompGCN (add) [32] are used to replace the HT-GNN model; in GME, RGCN [3], KBGAT [31], CompGCN (add) [32], and HRGNN [28] are used to replace the TR-GNN model. The experimental results in Table 5 show that the MRR of HT-GNN is 0.83 percentage points higher than CompGCN (add); in GME, the MRR of TR-GNN is 0.31 percentage points higher than HRGNN. Overall, both can effectively improve the accuracy of entity prediction in LME and GME respectively. The improvement of HT-GNN may be due to the hierarchical transfer method that enriches the semantic information of entities in the final knowledge graph. The improvement of TR-GNN may be due to its full utilization of both periodic and non-periodic time in the global knowledge graph.

5.3. Sensitivity Analysis

The paper further studied the effect of the number of layers in TR-GNN and HT-GNN on entity prediction performance for GME and LME modules using MRR and Hit@k (k = 1, 3, 10) metrics, respectively. Figure 3 show the experimental results with different numbers of layers on the ICEWS14 dataset. The experimental results show that with just 1 layer, the TR-GNN module can already effectively extract temporal-semantic information between entities. As the number of layers increases, the extracted features may interfere with entity embeddings. The HT-GNN model can extract sufficient semantic information with 2 layers, thus achieving higher entity prediction accuracy. Through experiments, it was proven that the different number of layers in TR-GNN and HT-GNN can affect the extraction of semantic information in the knowledge graph, thus influencing the effect of entity prediction.

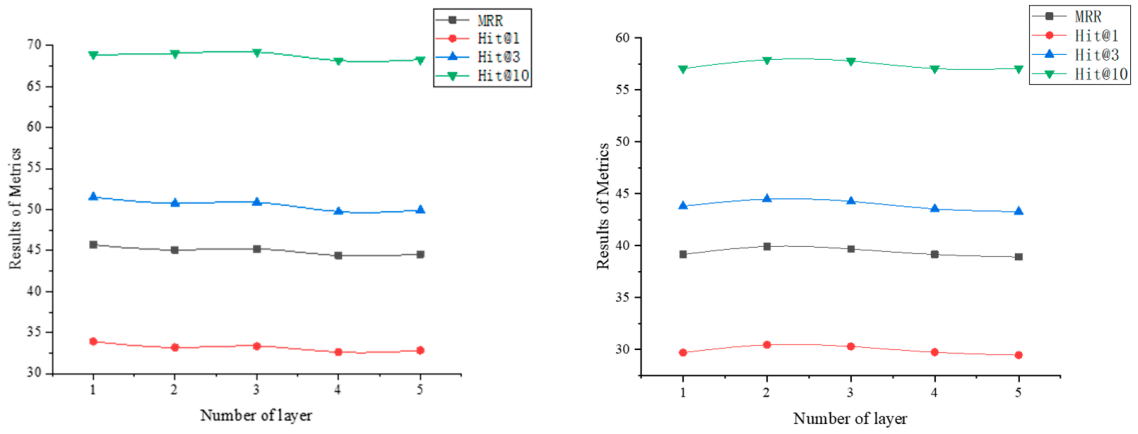

Figure 3. (**a**,**b**) are the effects of the number of layers on GME (TR-GNN) and LME (HT-GNN) in terms of MRR (%), Hit@1 (%), Hit@3 (%), and Hit@10 (%), respectively.

5.4. Training Process Analysis

This paper studied the evolution of loss and evaluation metrics of the MSEN model during the training process. Figure 4 shows how the experimental results change with increasing epoch on the validation dataset of ICEWS14. The experimental results show that as the number of training epochs increases, the loss first gradually decreases, reaching the lowest point at the 10th epoch, and then slowly increases again; the values of the evaluation metrics MRR, Hit@1, Hit@3, and Hit@10 gradually increase, reaching a plateau after the 10th epoch. The results indicate that after 10 epochs of training on the ICEWS14 dataset, the model starts overfitting as evidenced by the increasing loss and stabilizing evaluation metrics. Therefore, 10 epoch is an appropriate number for training the MSEN model on the ICEWS14 dataset to achieve good performance without overfitting.

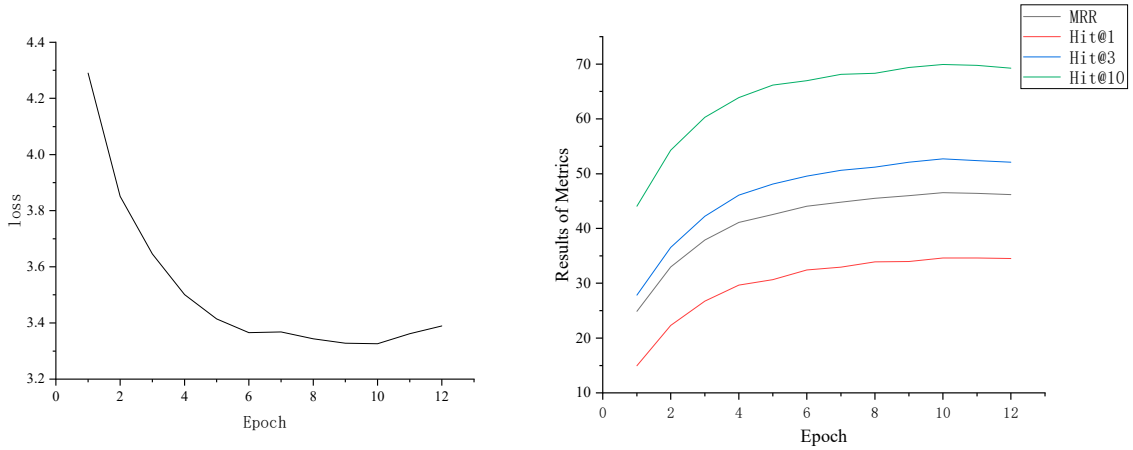

Figure 4. (**a**) is the change in loss as the training epoch increases; (**b**) is the change in metric results as the training epoch increases in terms of MRR (%), Hit@1 (%), Hit@3 (%), and Hit@10 (%).

5.5. Statistical Analysis

In order to demonstrate the reliability of the results, both permutation tests and multiple repeated experiments were conducted in this study.

For the permutation test, we first trained the proposed MSEN model on the original training set and recorded the model's performance (MRR) on the test set as the true observation. We then created perturbed training sets by replacing different proportions of head or tail entities in the original training set. The MSEN model was re-trained on these perturbed sets and achieved varying MRRs on the test set. The p-value was calculated using the following formula:

$$P = M/N \tag{24}$$

where M is the number of times the MRR obtained using the perturbed datasets was greater than the true observation, and N is the number of experiments conducted with the perturbed datasets.

As illustrated in Figure 5, the MSEN model achieved lower MRRs when trained on more heavily perturbed sets, with p-values below the 0.05 significance level. This suggests the model has genuinely captured meaningful patterns in the training data.

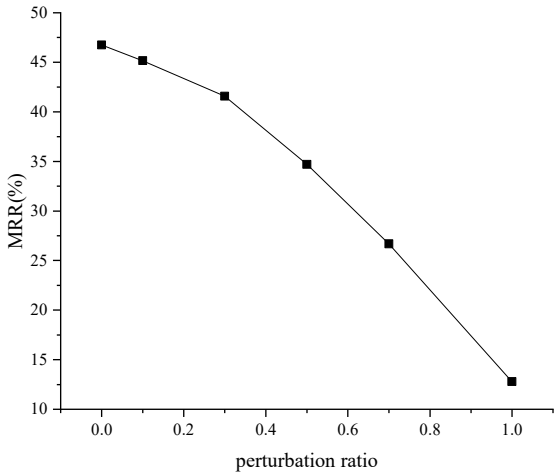

Figure 5. The change in MRR (%) as the perturbation ratio increases.

Additionally, the experiment was repeated three times using the original training set, yielding consistent MRRs of 46.68%, 46.75%, and 46.72%, with an average of 46.72%. This further demonstrates the reliability and reproducibility of the proposed model.

6. Discussion

The experimental results demonstrate that the proposed MSEN model achieves superior performance compared to several baselines on typical benchmark datasets. The better results validate the effectiveness of modeling multi-scale evolutionary information for predicting future events in TKGs.

The ablation studies in Section 5.2 provide insights into the contribution of each module in MSEN. Removing the global memory encoder leads to a significant performance drop, showing the importance of capturing long-distance temporal dependencies. The local memory encoder also contributes steady improvements by aggregating rich semantics within each timestamp. Using HT-GNN and TR-GNN as the graph neural network encoders also leads to better results compared to alternative GNN models, demonstrating their capabilities in encoding structural and temporal patterns from TKGs.

The multi-scale modeling in MSEN aligns with findings from previous works such as TiRGN [14] which show combining local and global patterns is beneficial. However, MSEN proposes new designs for each module, leading to better results. The hierarchical design in HT-GNN enriches semantic learning, while the periodic and aperiodic temporal modeling in TR-GNN improves temporal encoding.

7. Conclusions and Future work

In this work, we propose a new method Multi-Scale Evolutionary Network (MSEN) to solve temporal knowledge graph (TKG) reasoning under the extrapolation setting. First, a hierarchical transfer graph neural network (HT-GNN) is designed in the local memory encoder to acquire rich semantic information from the local knowledge graph, and sequence neural networks are used to learn temporal features of entities and relations. Second, in the global memory encoder, a time-related GNN (TR-GNN) is designed to make full use of temporal information in the global knowledge graph, obtaining periodic and aperiodic temporal-semantic dependencies in the global knowledge graph, providing different weights for entities and relations, improving performance of downstream tasks. Finally, embeddings from different scales are integrated to complete entity prediction. Experimental results on a typical benchmark demonstrate the effectiveness of MSEN in solving TKG reasoning problems.

While MSEN achieves good results, there remain several promising directions for future investigation: (1) Model Compression: The current MSEN model entails relatively high computational complexity. To enable real-time inference and deployment under resource constraints, we can explore model compression techniques such as pruning and knowledge distillation. By simplifying the model while retaining predictive power, we can enhance its applicability to real-world systems. (2) Robustness to Noisy Data: Historical knowledge graphs often suffer from incomplete, biased, or noisy data. The robustness of MSEN's predictions rely on the quality of the historical inputs. Advanced data filtering, denoising, and imputation methods can be studied to handle imperfect historical graphs. Techniques such as outlier detection and data estimation can help improve the model's resilience against noise and missing facts. (3) Advanced GNN Architectures: Further improving the local and global graph neural network encoders with more sophisticated architectures such as graph attention networks is a promising direction. This can potentially capture more complex structural, semantic, and temporal patterns.

In the future, we will aim to enhance MSEN along these directions to make broader impacts in real-world applications of temporal knowledge graph reasoning.

Author Contributions: Conceptualization, S.C. (Shudong Chen) and Y.Y.; methodology, Y.Y. and R.D.; software, Y.Y. and D.T.; validation, Y.Y., H.X. and D.T.; formal analysis, Y.Y.; investigation, H.X.; writing—original draft preparation, Y.Y.; writing—review and editing, S.C. (Shuai Chen). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The ICEWS14 and ICEWS18 datasets used in this study are derived from the Integrated Crisis Early Warning System event dataset, which is available from the ICEWS project website (https://www.andybeger.com/icews/ (accessed on 27 September 2023)). The specific dataset versions utilized can be accessed through the reference links provided (references [6,17]).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
2. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the 33rd International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016.
3. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018, Proceedings 15*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 593–607.
4. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 1811–1818.
5. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
6. Jin, W.; Qu, M.; Jin, X.; Ren, X. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6669–6683.
7. Li, Z.; Guan, S.; Jin, X.; Peng, W.; Lyu, Y.; Zhu, Y.; Bai, L.; Li, W.; Guo, J.; Cheng, X. Complex Evolutional Pattern Learning for Temporal Knowledge Graph Reasoning. *arXiv* **2022**, arXiv:2203.07782. [CrossRef]
8. Zhu, C.; Chen, M.; Fan, C.; Cheng, G.; Zhang, Y. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. *arXiv* **2020**, arXiv:2012.08492. [CrossRef]
9. Han, Z.; Chen, P.; Ma, Y.; Tresp, V. Explainable Subgraph Reasoning for Forecasting on Temporal Knowledge Graphs. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021.
10. Sun, H.; Zhong, J.; Ma, Y.; Han, Z.; He, K. TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting. *arXiv* **2021**, arXiv:2109.04101. [CrossRef]
11. Xu, Y.; Ou, J.; Xu, H.; Fu, L. Temporal knowledge graph reasoning with historical contrastive learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
12. Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; Cheng, X. 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, Montreal, QC, Canada, 11–15 July 2021.
13. Han, Z.; Ding, Z.; Ma, Y.; Gu, Y.; Tresp, V. 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, Online and Punta Cana, Dominican Republic, 7–11 November 2021.
14. Li, Y.; Sun, S.; Zhao, J. TiRGN: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23–29 July 2022; pp. 2152–2158.
15. Zhang, M.; Xia, Y.; Liu, Q.; Wu, S.; Wang, L. Learning Latent Relations for Temporal Knowledge Graph Reasoning. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023.
16. Leblay, J.; Chekol, M.W. Deriving Validity Time in Knowledge Graph. In Proceedings of the Companion of the the Web Conference, Lyon, France, 23–27 April 2018; pp. 1771–1776.
17. García-Durán, A.; Dumančić, S.; Niepert, M. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4816–4821.
18. Dasgupta, S.S.; Ray, S.N.; Talukdar, P. HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2001–2011.
19. Lacroix, T.; Obozinski, G.; Usunier, N. Tensor Decompositions for Temporal Knowledge Base Completion. *arXiv* **2020**, arXiv:2004.04926. [CrossRef]

20. Goel, R.; Kazemi, S.M.; Brubaker, M.; Poupart, P. Diachronic Embedding for Temporal Knowledge Graph Completion. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 3988–3995.
21. Sadeghian, A.; Armandpour, M.; Colas, A.; Wang, D.Z. ChronoR: Rotation Based Temporal Knowledge Graph Embedding. *arXiv* **2021**, arXiv:2103.10379. [CrossRef]
22. Fu, G.; Meng, Z.; Han, Z.; Ding, Z.; Ma, Y.; Schubert, M.; Tresp, V.; Wattenhofer, R. TempCaps: A Capsule Network-based Embedding Model for Temporal Knowledge Graph Completion. In Proceedings of the Sixth Workshop on Structured Prediction for NLP, SPNLP@ACL 2022, Dublin, Ireland, 27 May 2022; pp. 22–31.
23. Linhai, Z.; Deyu, Z. Temporal Knowledge Graph Completion with Approximated Gaussian Process Embedding. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 12–17.
24. Han, Z.; Wang, Y.; Ma, Y.; Günnemann, S.; Tresp, V. Graph Hawkes Network for Reasoning on Temporal Knowledge Graphs. *arXiv* **2020**, arXiv:2003.13432. [CrossRef]
25. Li, Z.; Jin, X.; Guan, S.; Li, W.; Guo, J.; Wang, Y.; Cheng, X. Search from History and Reason for Future: Two-stage Reasoning on Temporal Knowledge Graphs. *arXiv* **2021**, arXiv:2106.00327. [CrossRef]
26. Deng, S.; Rangwala, H.; Ning, Y. Dynamic Knowledge Graph based Multi-Event Forecasting. In Proceedings of the KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, Virtual Event, CA, USA, 6–10 July 2020. [CrossRef]
27. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 2787–2795.
28. Zhang, M.; Xia, Y.; Liu, Q.; Wu, S.; Wang, L. Learning long-and short-term representations for temporal knowledge graph reasoning. In Proceedings of the ACM Web Conference, Austin, TX, USA, 30 April–4 May 2023; pp. 2412–2422.
29. Hu, L.; Yang, T.; Zhang, L.; Zhong, W.; Tang, D.; Shi, C.; Duan, N.; Zhou, M. Compare to The Knowledge: Graph Neural Fake News Detection with External Knowledge. 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), Virtual, 1–6 August 2021; pp. 754–763.
30. Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; Ward, M.; Natarajan, D.; Bagozzi, B.; Berger, D.; Carley, K.M.; et al. ICEWS Coded Event Data. Harvard Dataverse 2015. Available online: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075 (accessed on 27 September 2023).
31. Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. *arXiv* **2019**, arXiv:1906.01195. [CrossRef]
32. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-based Multi-Relational Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.