SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2025, Vol. 68, Iss. 10, 202101:1–202101:19
https://doi.org/10.1007/s11432-024-4329-3

Temporal knowledge graph diffusion model for open-world reasoning

Yuhan WANG[1], Xiang ZHAO[2*], Zhen TAN[1], Weidong XIAO[1] & Xueqi CHENG[3]

[1]*National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China*
[2]*Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China*
[3]*Institute of Computer Technology, Chinese Academy of Sciences, Beijing 100195, China*

Received 5 May 2024/Revised 27 July 2024/Accepted 2 September 2024/Published online 4 August 2025

Abstract In recent years, temporal knowledge graphs (TKGs) have emerged as a prominent research area in artificial intelligence and knowledge engineering, offering notable potential for various applications. However, as the scope of TKG applications continues to expand, distinct challenges arise in specific contexts, especially in open-world scenarios. Traditional representation and reasoning models struggle to capture the associations of unseen entities within the knowledge graph, making it difficult to accurately represent these entities and perform unseen entity prediction tasks in open-world settings. To address these challenges, this paper proposes a novel reasoning framework, the diffusion model-based unseen entity prediction method (DM-UEP). This innovative approach generates virtual representations for unseen entities using a diffusion model and establishes a two-phase reasoning framework. The framework includes a graph extension and a transformer with reference to human cognitive reasoning. In addition, we reconstruct three test datasets tailored specifically for unseen entity prediction by leveraging existing public datasets. These datasets demonstrate the superiority of the proposed DM-UEP method in tackling the specialized task of unseen entity prediction in open-world scenarios.

Keywords temporal knowledge graph, unseen entity prediction, open-world scenarios, diffusion model, knowledge graph reasoning

Citation Wang Y H, Zhao X, Tan Z, et al. Temporal knowledge graph diffusion model for open-world reasoning. Sci China Inf Sci, 2025, 68(10): 202101, https://doi.org/10.1007/s11432-024-4329-3

1 Introduction

The concept of the temporal knowledge graph (TKG) was first introduced in 2016, making it a relatively young research area with less than a decade of development [1]. As illustrated in Figure 1(a), an example of TKG about Real Madrid CF is shown. Temporal knowledge representation [2] and reasoning [3] have been recognized as the most central and pivotal research directions [4], driving rapid advancements and numerous remarkable research achievements. Within the field of TKG, temporal knowledge representation and reasoning occupy a paramount position, serving as the foundation for all other research directions. The quality of representation and reasoning substantially influences progress in other areas and determines the practical effectiveness of TKGs.

TKG representation and reasoning models can be divided into three core modules based on their characteristics and application scenarios: the base-level embedding module, the temporal information processing module, and the reasoning module. The base-level embedding module underpins the entire model structure, where the choice of embedding considerably influences the design of subsequent models and determines the overall capability for representation and reasoning. This module can be further categorized into two types based on semantic or structural correlation. The temporal information processing module is crucial for handling temporal data in the TKGs, capturing the evolution patterns of knowledge over time. The models ability to effectively mine and utilize historical features directly impacts the accuracy of TKG representation and the precision of subsequent reasoning. Current mainstream methods for temporal modeling include explicit approaches and implicit methods that extract historical information. The reasoning module serves as the interface for practical applications, where selecting the

* Corresponding author (email: xiangzhao@nudt.edu.cn)

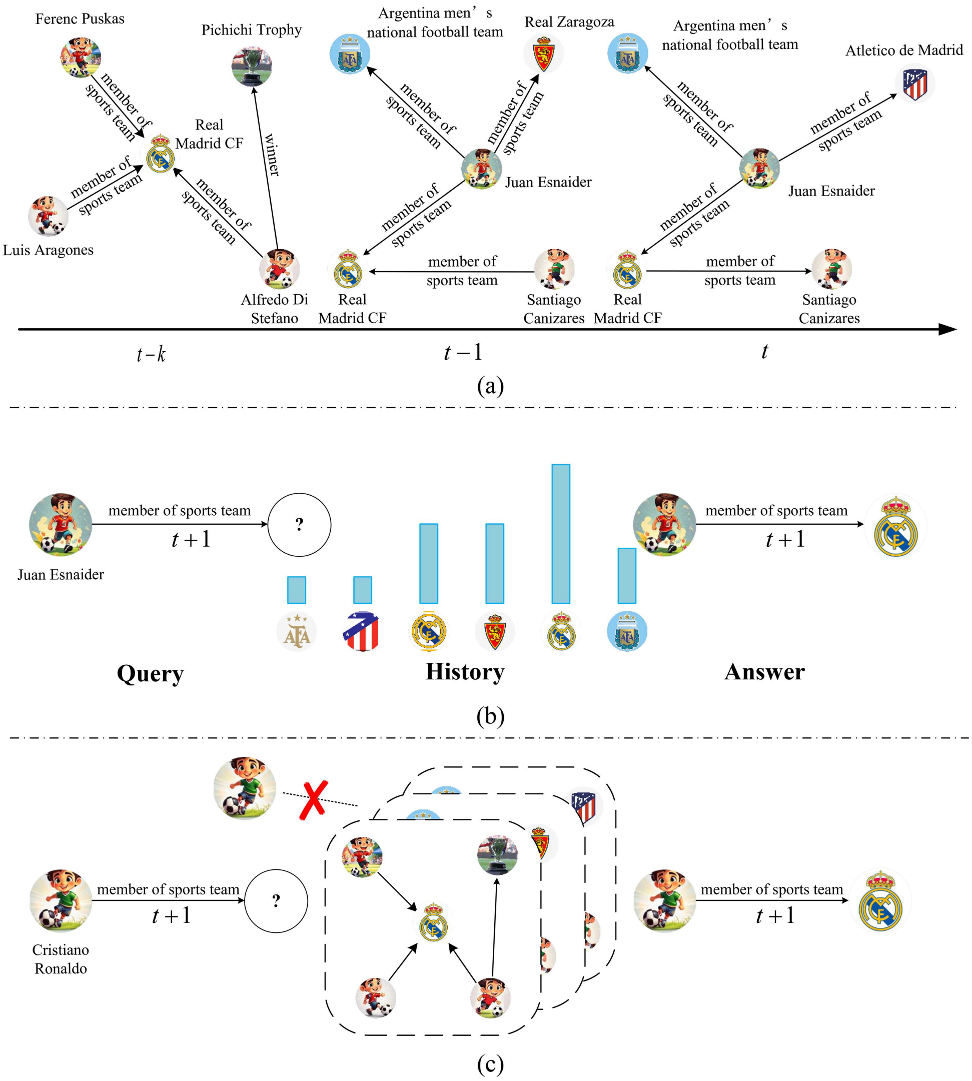

Figure 1 Examples of (a) temporal knowledge graphs, (b) conventional entity prediction, and (c) unseen entity prediction in open-world scenarios.

appropriate reasoning method based on the task can remarkably enhance overall model performance. Currently, reasoning methods are classified into four main types: semantic-based, structure-based, probabilistic computation, and hybrid approaches. However, existing temporal knowledge graph reasoning methods heavily rely on historical information, making it challenging to reason about unseen entities without such data. Therefore, research on TKG reasoning in open-world scenarios [5–7] remains scarce.

As TKGs are increasingly applied in real-world scenarios, the traditional closed-world assumption no longer meets current needs. Consequently, research on TKGs in open-world scenarios has become crucial. In open-world scenarios, reasoning tasks differ from those in closed-world settings [5]. For example, in the entity prediction task, conventional methods in closed-world scenarios rely on the historical information of the main entity in the query to predict its missing entities [8]. As shown in Figure 1(b), the answer to the query (Juan Esnaider, member of sports team, ?, $t + 1$) is derived from past historical information. However, in the unseen entity prediction task within open-world scenarios, the main entity in the query is an unseen entity with no direct connections to the original TKG. Current methods are unable to generate representations for such unseen entities, making it challenging to access relevant historical information for prediction reasoning. This poses a substantial challenge for reasoning tasks. As illustrated in Figure 1(c), the unseen entity Cristiano Ronaldo does not exist in the original TKG (i.e., Cristiano Ronaldo is absent from the training dataset). Traditional methods [9] are unable to generate a representation for Cristiano Ronaldo.

This paper addresses the challenge of open-world scenarios reasoning in TKGs by proposing a diffusion model-based unseen entity prediction (DM-UEP) method. DM-UEP uses a structure-correlation-based relational graph convolutional network (RGCN) [10] as the base-level embedding module and utilizes a hybrid reasoning approach that incorporates historical information extraction. In open-world scenarios, predicting triplets that include unseen entities poses a challenge since these entities lack prior embeddings derived from the existing knowledge graph. Early approaches used random or zero vectors as placeholders for unseen entity embeddings, but these methods proved insufficient for accurate predictions. Subsequently, researchers introduced entity descriptions and other attributes to strengthen the connection between unseen entities and the original knowledge graph, using these descriptions as surrogates for entity representation. Although this approach notably improved the accuracy of unseen entity tasks, it also introduced auxiliary information, limiting its application owing to dependencies on specific scenarios. Our method avoids introducing additional information and instead relies on the relations in queries to derive virtual representations of unseen entities. By treating the relations in all triplets as a unique attribute of the head entity, our diffusion model-based module [11] generates embeddings inversely, encapsulating key information about unseen entities. This approach enhances the effectiveness of unseen entity prediction tasks.

Additionally, inspired by the two-step reasoning process [12,13] humans use during inductive reasoning, which integrates pertinent historical information with immediate context for deduction, this study introduces a two-phase reasoning framework. This framework simulates human reasoning processes, thereby improving reasoning capabilities for temporal knowledge. Subsequently, the study refines existing public datasets for TKG reasoning tasks by meticulously filtering out noise, enabling the construction of new datasets specifically suited for predicting unseen entities in open-world scenarios. When evaluated against current advanced algorithms on these novel datasets, the performance of our methods highlights its cutting-edge capabilities.

The main contributions of our research are as follows.

- We incorporate the diffusion model, which enhances the performance of predicting unseen entities in open-world scenarios of TKGs by generating virtual representations of unseen entities through the relations present in queries.
- We propose a two-phase reasoning framework, which amalgamates information from adjacent time points based on graph extension and extracts key features while also learning specific behavioral patterns from historical information using a transformer. This combination emulates the two-step cognitive reasoning process of humans, thereby bolstering the reasoning capabilities of TKGs.
- In relation to tasks associated with TKG, we refine existing public datasets and construct new datasets tailored for predicting unseen entities in open-world scenarios, validating the superior performance of our method against contemporary advanced algorithms.

2 Related work

Current research in TKG modeling is segmented into three core modules. The first is the base-level embedding module, where graph convolution techniques, such as RGCN [10], are widely applied to capture the structural information of graphs. Next is the temporal information processing module, which addresses the temporal dynamics of entities and relations. Finally, the reasoning module leverages the semantic and structural features of TKGs to better predict missing entities [14].

2.1 Base-level embedding module

Given that a knowledge graph is a specialized graph structure, representation methods designed for general graphs can often be adapted for knowledge graphs. Schlichtkrull, in collaboration with Kipf, the author of GCN [15], introduced RGCN [10], a representation learning model for knowledge graphs that extends the GCN framework to model various relations within the graph. Consequently, in TKGs, graph convolution has become a preferred choice for base-level embedding representation. It effectively captures the structural information of the graph in the initial stages, laying a solid foundation for subsequent representation. In [16], the RE-NET model, which is similar to recurrent neural networks (RNNs), employs RGCN as the base-level embedding to aggregate neighborhood information for entities at each timestamp. It also uses an RNN-based event encoder to model the sequence of events and capture features influenced by temporal information. Ref. [17] proposed the HGLS model, which employs a hierarchically structured relation-type graph neural network, HRGNN, to encode the global graph structure. At the subgraph level, RGCN is used to model the static graph at a single timestamp. At the whole graph level, temporal effects are incorporated as edge weights, and attention mechanisms [18] are employed to model the global features of entities.

2.2 Temporal information processing module

Historical information of entities and relations represents another critical aspect of temporal data, indirectly reflecting the evolution and impact of time. The historical information typically manifests in three forms: the historical information of an entity itself, entity-relation history, which indicates the presence of an entity-relation pair across different timestamps along with all associated entities, and the history of knowledge (facts), which refers to the presence of a particular triplet on the graph at various timestamps. Models typically target one or more types of historical information for extraction.

The earliest model to incorporate a historical information extraction module is Know-Evolve, proposed in [19]. This model converts TKGs into a sequence of static graphs constructed at various timestamps arranged in chronological order. Know-Evolve is designed to encode this sequence by introducing two custom RNNs capable of updating representations with new triplets at successive timestamps. These RNNs update the representations of entities by leveraging their historical information. Specifically, the representation of an entity at a given time point (t) is computed based on its representations at the previous two timestamps (t' and t'').

Subsequently, RE-GCN [9] was introduced, which featured evolutionary units for modeling historical information. These units employ gated recurrent unit (GRU) [20] components to characterize the evolution of entities and relations. The representation of a relation is aggregated from the representations of all associated entities.

In [21], CENET was proposed as another model that explicitly utilizes historical information. CENET incorporates contrastive learning into TKG reasoning. It initially makes predictive judgments about whether a missing entity is historical or nonhistorical. Based on these judgments, it focuses on making final predictions within these categories (historical/nonhistorical entities). Furthermore, CENET considers historical and nonhistorical information during the representation process, modeling entity relations accordingly. It introduces distinct evaluation functions tailored to scenarios involving historical and nonhistorical information, enabling a more nuanced approach to knowledge graph reasoning.

2.3 Reasoning module

Hybrid reasoning, which has emerged in models over the past three years, represents a sophisticated reasoning process that concurrently considers semantic and structural features.

In [22], EvoExplore is introduced as a model that simultaneously models local and global structures of TKGs. For local structures, the model employs a semantically-based method to reason about entities, while for global structures, it draws inspiration from the concept of communities in social networks. By mining the community structures within TKGs, EvoExplore effectively captures the global features of the graphs. This dual consideration of semantics and structure allows EvoExplore to better differentiate between time-sensitive and time-insensitive entities compared to previous models like HyTE [23] and TeMP [24]. In [21], CENET applies structural information to make predictive judgments about whether a missing entity is historical while leveraging semantic information for the final entity prediction.

2.4 TKG reasoning in open-world scenarios

As TKGs evolve over time, unseen entities continually emerge, posing challenges for accurate and effective modeling owing to their absence in the traditional knowledge graph. This directly impacts the reasoning capabilities of the model.

Unlike other reasoning models, TITer [25] has been recognized for its ability to perform predictive reasoning about unseen entities. The absence of direct connections between unseen entities and the existing graph makes their precise representation challenging, complicating reasoning and prediction based on the graph information. TITer addresses this challenge by leveraging the relations in the predicted triplets to preliminarily determine the type of the unseen entity. It then employs an inductive mean algorithm, which uses the representations of already trained entities of the same type to initialize values

Table 1 Notations with descriptions.

Notation	Description
G	The whole temporal knowledge graph (TKG)
G_t	The static slice of TKG at time t
G'_t	The enriched graph of G_t
E	The set of entities in TKG
R	The set of relations in TKG
T	The set of time in TKG
E_r	The set of entities whose unique attribute is r
(s, r, o, τ)	A fact knowledge in TKG
$\mathbf{s}, \mathbf{o}, \mathbf{r}$	The embeddings of subject/object entities and relations
q	A forward noise-adding process in the diffusion model
K	The round of noise-adding process
$\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^k$	A group of temporary embeddings in the noise-adding process
β	A hyperparameter determining the variance of the Gaussian distribution
ω	The weight of the adding edge
η	The exponential decay function
σ, ς	The hyperparameters in η
N_{s,t_i}	The neighbor nodes set of entity o_{t_i}
$f(\cdot)$	The RReLU activation function
$\mathbf{s}_t^l$	The embedding of s_t at the l-th layer of RGCN in the wide-level
$\mathbf{s}_t^L/\mathbf{r}_t^L$	The embedding of s_t/r_t computed by RGCN in the wide-level
N_r^t	The set of subject entities associated with r at time t
$E_{his}^{s,r,\tau}$	The set of history entities related by r and s before time τ
$Q_\tau^{s,r}$	The set of fact knowledge related to s and r before time τ
$\mathbf{F}$	The evaluation function matrix computed by similarity between the entity and relation embeddings
$\mathbf{E}$	The representation matrix computed by the wide-level
$\mathbf{W}_{his}^s, \mathbf{b}_{his}^s$	The trainable hyperparameters in $\mathbf{F}$
ζ	The hyperparameter in the whole loss function
$P_\tau^{s,r}(o)$	The probability of fact knowledge (s, r, o, τ) authenticity

for unseen entities. This method remarkably strengthens the connection between unseen entities and the TKGs, enhancing the predictive accuracy of these unseen entities.

Notably, our task does not rely on information beyond the entities and relations in the TKGs, such as entity types or descriptive information. Therefore, the comparative models selected for this study also use representation and reasoning methods that do not depend on additional information. Meanwhile, methods based on large language models [26], which incorporate pretraining on large-scale corpora, have access to more semantic information compared to other methods. As a result, such methods are not considered comparative models in our study.

3 Concepts and definitions of TKG

The researchers initiated a focused exploration into the impact of temporal information on knowledge graphs, culminating in the introduction of the TKG concept [1]. Simply put, a TKG extends the traditional knowledge graph triplet by incorporating temporal information. As shown in Figure 1(a), the relation “member of sports team” between the entities “Juan Esnaider” and “Real Madrid CF” at the timestamp of 1994, can be represented as (Juan Esnaider, Make a visit, France, [1994]).

The formal definitions of a TKG are as follows.

Definition 1 (TKG)**.** A TKG is defined as a directed graph with timestamps $G = (E, R, T)$. E represents a set of entities, and the TKG contains $|E|$ different entities. R represents a set of relations, and the TKG contains $|R|$ different relations. T represents a set of timestamps, and the TKG contains $|T|$ different timestamps. Concurrently, a TKG is also a collection of static knowledge graphs, each associated with a specific timestamp, i.e., $G = G_1, G_2, \ldots, G_T$. The important notations are in Table 1.

The knowledge in each TKG can be represented as (s, r, o, τ), where $s, o \in E$ indicate the Subject

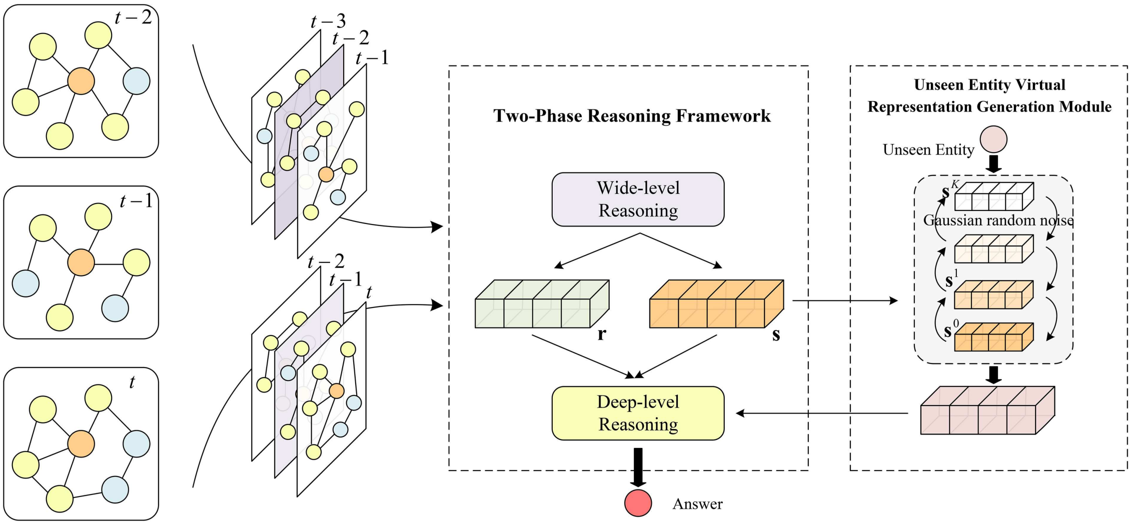

Figure 2 Illustration of our DM-UEP method. The unseen entity virtual representation generation module represents the unseen entity through the diffusion model. The two-phase reasoning framework is utilized to predict entities based on the extended graph.

entity and Object entity, $r \in R$ represents the relation, and $\tau \in T$ reflects the timestamp.

Reasoning in TKG, compared to traditional static knowledge graph reasoning, incorporates the additional temporal information, thereby broadening the types of reasoning possible. Beyond the basic tasks of predicting subject (or object) entities and relations, TKG reasoning expands to include time prediction and unseen entity prediction. The definitions of several reasoning tasks in TKGs $G = (E, R, T)$ are as follows.

Definition 2 (Conventional entity prediction)**.** For the TKGs $G = (E, R, T)$, given the entity $s \in E$ (or $o \in E$) and a relation $r \in R$, the process of solving $o \in E$(or $s \in E$) such that the resultant (s, r, o, τ) is true is termed as entity prediction.

Definition 3 (Unseen entity prediction)**.** For the TKGs $G = (E, R, T)$, given the entity $s \notin E$ and a relation $r \in R$, the process of solving $o \in E$ such that the resultant (s, r, o, τ) is true is termed as unseen entity prediction.

4 DM-UEP method

The DM-UEP method is mainly based on a diffusion model to generate virtual representations of unseen entities. As illustrated in Figure 2, it also constructs a two-phase reasoning framework that simulates the two-step cognitive reasoning process of humans, integrating RGCN with transformers [27] to achieve embedded representations of knowledge for predicting unseen entities. The details of each aspect of this method are presented below.

4.1 Unseen entities virtual representation generation module

Due to their absence in the original knowledge graph, unseen entities lack precise embedded representations. To address this, we introduce the diffusion model, which generates embedded representations in reverse, using relations in triplets as special attributes of entities, as illustrated in Figure 3. Initially, for triplets (s, r, o) in the training set, the relation r is regarded as a unique attribute of the entity s. And E_r is the set of entities whose unique attribute is r. Subsequently, in the diffusion model, a forward noise-adding process q is constructed for the entities in the training set. This process involves adding a series of (K rounds) Gaussian noise to the embedded representations of existing entities, gradually transforming these representations $\mathbf{s}^0$ of the entity s into pure Gaussian noise $\mathbf{s}^K$, while also obtaining a series of embeddings $\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^3, \ldots, \mathbf{s}^K$ in the process. Each round of noise addition in the forward process q is only related to the embedding after the previous round of noise addition, making it a Markov process. The noise addition operation in process q is as follows:

$$q(\mathbf{s}^k|\mathbf{s}^{k-1}) = \mathcal{N}(\mathbf{s}^k; \sqrt{1-\beta_k}\mathbf{s}^{k-1}, \beta_k\mathbf{I}), \tag{1}$$

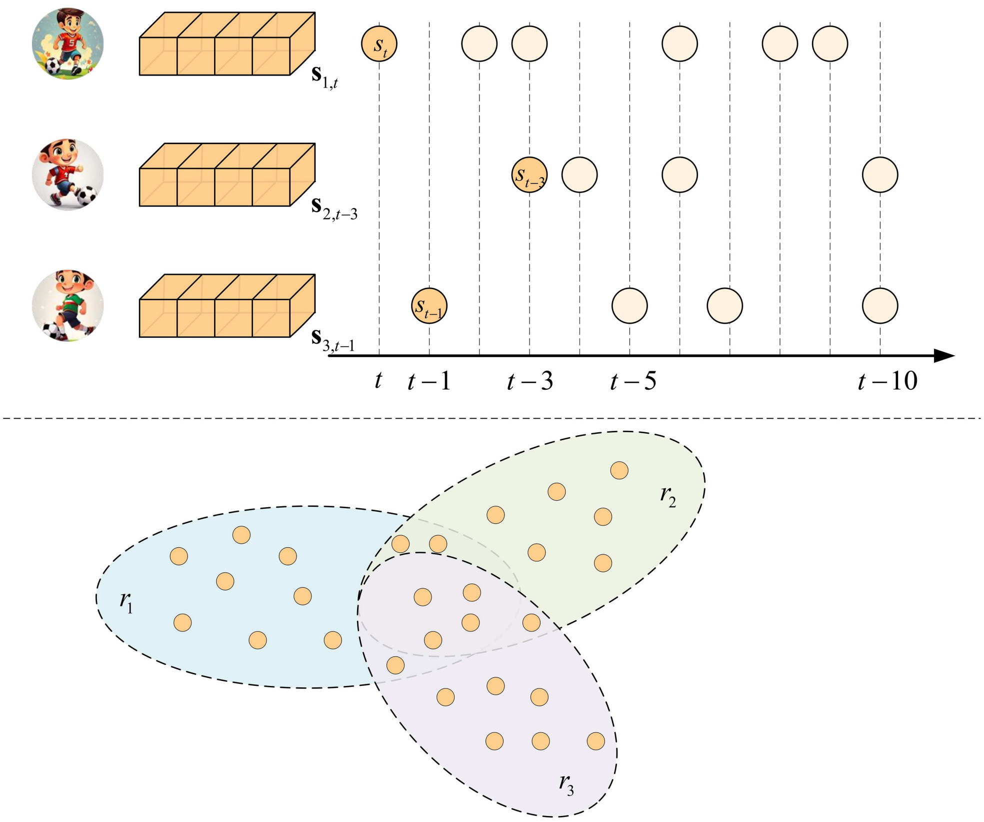

Figure 3 Overview of the unseen entity virtual representation generation module (DM module). For each entity-relation pair, we train the hyperparameters based on their representations at the latest time, while treating the relation as the category attribute of this entity.

$$q(\mathbf{s}^{1:K}|\mathbf{s}^0) = \prod_{k=1}^{K} q(\mathbf{s}^k|\mathbf{s}^{k-1}), \tag{2}$$

where $\{\beta_k \in (0,1)\}_{k=1}^K$ is a hyperparameter determining the variance of the Gaussian distribution used in the noise addition process. It can be found from the above equation that $\mathbf{s}^k$ is closer to pure noise as the number of rounds increases. Based on (1), the expression of $q(\mathbf{s}^k|\mathbf{s}^0)$ can be obtained as follows:

$$q(\mathbf{s}^k|\mathbf{s}^0) = \mathcal{N}(\mathbf{s}^k; \sqrt{\bar{\alpha}_k}\mathbf{s}^0, (1-\bar{\alpha}_k)\mathbf{I}), \tag{3}$$

where $\bar{\alpha}_k = \prod_{i=1}^k \alpha_i, \alpha_i = 1 - \beta_i$.

In contrast to the forward noise addition process, the reverse process in diffusion models represents the denoising reasoning process, which progressively reconstructs the original image from the noise. We can deduce through (1)–(3) using Bayes' theorem that

$$q(\mathbf{s}^{k-1}|\mathbf{s}^k\mathbf{s}^0) = \mathcal{N}\left(\mathbf{s}^{k-1}; \frac{1}{\sqrt{\alpha_k}}\mathbf{s}^k - \frac{\beta_k}{\sqrt{\alpha_k(1-\bar{\alpha}_k)}}Z, \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}\beta_k\right), Z \sim N(0, I). \tag{4}$$

In the reverse process, obtaining $\mathbf{s}^0$ requires knowledge of $q(\mathbf{s}^{k-1}|\mathbf{s}^k)$, which is challenging to ascertain. Therefore, we constructed a neural network $p_\theta(\mathbf{s}^{k-1}|\mathbf{s}^k)$ to approximate it and subsequently used $q(\mathbf{s}^{k-1}|\mathbf{s}^k\mathbf{s}^0)$ to guide its training.

In [28], researchers have substantiated that L_{VLB} is the variational upper bound of $-\log p_\theta(\mathbf{s}^0)$ and $-E_{q(\mathbf{s}^0)} \log p_\theta(\mathbf{s}^0)$, i.e.,

$$\begin{aligned} &L_{VLB} \geqslant -\log p_\theta(\mathbf{s}^0), \\ &L_{VLB} \geqslant -E_{q(\mathbf{s}^0)} \log p_\theta(\mathbf{s}^0). \end{aligned} \tag{5}$$

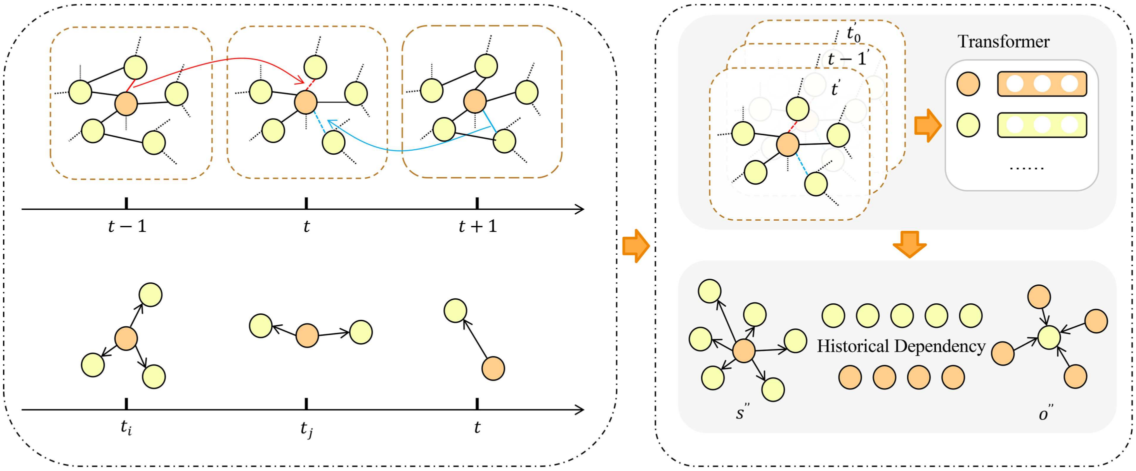

Figure 4 Overview of the two-phase reasoning framework. In the wide-level reasoning, it uses RGCN to represent entities and relations, followed by employing a Transformer to represent the updates of entities over temporal evolution. In the deep-level reasoning, it constructs an evaluation function based on dependencies of historical information.

$L_{VLB}(r,s)$ can be simplified to

$$L_{VLB}(r,s) = D_{\mathrm{KL}}(q(\mathbf{s}^K|\mathbf{s}^0)||p_\theta(\mathbf{s}^K)) + \sum_{k=2}^{K} D_{\mathrm{KL}}(q(\mathbf{s}^{K-1}|\mathbf{s}^k\mathbf{s}^0)||p_\theta(\mathbf{s}^{K-1}|\mathbf{s}^K)) - \log p_\theta(\mathbf{s}^0|\mathbf{s}^1). \tag{6}$$

Therefore, the training process of $p_\theta(\mathbf{s}^{k-1}|\mathbf{s}^k)$ aims to minimize the KL divergence between $p_\theta(\mathbf{s}^{k-1}|\mathbf{s}^k)$ and $q(\mathbf{s}^{k-1}|\mathbf{s}^k\mathbf{s}^0)$. We have established a loss function L_{DM} for the diffusion model component as

$$L_{\mathrm{DM}} = \sum_{r\in R, s\in E_r} L_{VLB}(r,s). \tag{7}$$

We applied the diffusion process to entities sharing a specific attribute, enabling the training of reverse-generative neural networks tailored to each entity category. When an unseen entity emerges, its relation in the context of the query, considered its unique attribute, is used to randomly generate Gaussian noise. This noise aids in reverse engineering a virtual representation of the unseen entity. Such a virtual representation effectively captures the essential characteristics of the unseen entity, thereby enhancing its interconnectedness with the existing knowledge graph. This methodology significantly improves the predictive accuracy for queries involving unseen entities.

4.2 Two-phase reasoning framework

Humans reason and predict events by organizing historical information and integrating it with the current situational context. As illustrated in Figure 4, our study emulates this process through a two-phase framework, allowing the model to both aggregate historical information and draw insights from adjacent time periods.

The model explores aggregated entity history information at a wide-level to aggregate adjacent temporal information and at a deep-level to explore aggregated entity history information.

4.2.1 *Wide-level reasoning*

At each moment, entities are not only associated with their current neighbors but are also influenced by neighboring entities from past and future timestamps. These influences, however, diminish as the time gap widens. Our method, therefore, attempts to capture the associative information from nearby moments while considering the relations of current entities, thus enhancing the perception of the current scenarios in the model and maximizing the acquisition of immediate information.

We aggregated static graph information from k time intervals adjacent to the current moment t. Considering the decay of information over time, the method assigns adjusted weights to new edges (triplets) added at the time t. The historical information decay is represented by the decay coefficient from the Hawkes process [29]. The weight representation ω_t for new edges incorporated into the graph at time t from adjacent time points t' is defined as follows:

$$\omega_t = \eta_r(|t - t'|), \tag{8}$$

where r represents the relation denoted by the new edge, and the decay function $\eta_r(\cdot)$ is an exponential decay function,

$$\eta_r(t) = \sigma_r \mathrm{e}^{-\varsigma_r t}, \tag{9}$$

where σ_r and ς_r are learnable parameters of the decay function.

Through this approach, the method successfully expands the original static graph G_i atlas of each moment into an enriched static graph G'_i, incorporating knowledge information from the adjacent time scenes of the current moment.

For these enriched static graphs $G' = (G'_1, G'_2, \ldots, G'_n)$, RGCN is used to generate representations of entities and relations in each graph,

$$\mathbf{s}_{t_i}^{l+1} = f\left(\frac{1}{|N_{s,t_i}|}\sum_{o_{t_i} \in N_{s,t_i}} \mathbf{W}_1^l(\mathbf{o}_{t_i}^l + \mathbf{r}_{t_i}) + \mathbf{W}_2^l \mathbf{s}_{t_i}^l\right), \tag{10}$$

where N_{s,t_i} denotes the neighbor nodes set of entity s_{t_i} in the static graph G'_{t_i}, $f(\cdot)$ is the RReLU activation function, and $\mathbf{W}_1^l$ and $\mathbf{W}_2^l$ are trainable weight matrices of the l-th layer.

This process via RGCN yields representations of entities in each expanded static graph. However, given the temporal continuity in TKGs, entities' representations in individual static graphs are not isolated. They are interrelated across the temporal sequence. Previous research [9] introduced GRU to depict the influence between static graphs over time. Our work employs a Transformer model to describe these temporal associations, capturing underlying features more effectively than GRU [30, 31]. The entity and relation representations are updated using the Transformer as follows:

$$\begin{aligned}\mathbf{s}_{t+1}^w &= \mathrm{Transformer}_E(\mathbf{s}_t^w, \mathbf{s}_t^L),\\ \mathbf{r}_{t+1}^w &= \mathrm{Transformer}_R(\mathbf{r}_t^w, \mathbf{r}_t^L),\end{aligned} \tag{11}$$

where $\mathbf{s}_t^L$ denotes the entity representation at time t, $\mathbf{r}_t^L$ denotes the relation representation at time t, computed by aggregating the representations of entities interacting with relation r at time t,

$$\mathbf{r}_t^L = \mathrm{Meanpooling}(\mathbf{s}_t^L) \oplus \mathbf{r}_t, s_t \in N_{r,t}, \tag{12}$$

the operator Meanpooling$(\cdot)$ acts on the entity set $N_{r,t}$ associated with r at time t, and $\oplus$ is concatenation operator. All relations share the same parameters for $\mathrm{Transformer}_R$.

4.2.2 *Deep-level reasoning*

Historical information is the most important clue and basis in entity prediction. Therefore, the method of using historical information plays a crucial role in our framework. To effectively utilize this data, we developed a deep-level structure designed to aggregate and apply historical information.

For a given knowledge (s, r, o, τ), we identify and collect all historically associated entities for the entities s and o based on the relations r up to time τ, thereby forming the sets $E_{his}^{s,r,\tau}$ and $E_{his}^{o,r,\tau}$ as follows:

$$\begin{aligned}E_{his}^{s,r,\tau} &= \{o|(s, r, o, \tau) \in Q_\tau^{s,r}\},\\ E_{his}^{o,r,\tau} &= \{s|(s, r, o, \tau) \in Q_\tau^{o,r}\},\end{aligned} \tag{13}$$

where $Q_\tau^{s,r} = \bigcup_{t<\tau} \{(s, r, o, t) \in G_t\}$ and $Q_\tau^{o,r} = \bigcup_{t<\tau} \{(s, r, o, t) \in G_t\}$ represent sets of all knowledge associated with entities s and o prior to the time τ.

Based on the approach proposed by [21], we constructed an evaluation function matrix $\mathbf{F} \in \mathbf{R}^{|E|}$ based on the representational similarity of entities and relations in each knowledge tuple,

$$\begin{aligned}\mathbf{F}_{his}^{s,r,\tau} &= \tanh(\mathbf{W}_{his}^s(\mathbf{s}_\tau^w \oplus \mathbf{r}_\tau^w) + \mathbf{b}_{his}^s)\mathbf{E}_\tau^T,\\ \mathbf{F}_{his}^{o,r,\tau} &= \tanh(\mathbf{W}_{his}^o(\mathbf{r}_\tau^w \oplus \mathbf{o}_\tau^w) + \mathbf{b}_{his}^o)\mathbf{E}_\tau^T,\end{aligned} \tag{14}$$

Algorithm 1 Learning algorithm of DM-UEP.

```
Require: Training fact knowledge quadruples set $G$, entity set $E$, relation set $R$, hyperparameters.
Ensure: A trained network.
  Initiate parameters of network $Net$;
  Compute new edge weight $\omega_t$ according to (8) and (9);
  Generate extended graph $G'$ based on $G, \omega_t, k$.
  for $l$ in 1 to $L$ do
      Compute $\mathbf{s}_t^L, \mathbf{r}_t^L$ according to (10) and (12);
  end for
  for $t$ in 1 to $T$ do
      Compute $\mathbf{s}_t^w, \mathbf{r}_t^w$ according to (11);
  end for
  for each $(s, r, o, \tau)$ in $G'$ do
      Compute $E_{his}^{s,r,\tau}, E_{his}^{o,r,\tau}$ according to (13);
      Compute $\mathbf{F}_{his}^{s,r,\tau}, \mathbf{F}_{his}^{o,r,\tau}$ according to (14);
  end for
  for each $r$ in $R$ do
      for each $s$ in $E_r$ do
          Find $\max \tau$ s.t. $(s, r, \tau) \in G$;
          $\mathbf{s}_0 \leftarrow \mathbf{s}_\tau^w$;
          Compute $L_{VLB}(r, s)$ according to (1)–(6);
      end for
  end for
  while loss does not converge do
      Compute $L_{Two}$ according to (15);
      Compute $L_{\mathrm{DM}}$ according to (7);
      $L \leftarrow \zeta L_{\mathrm{DM}} + L_{Two} + \zeta' ||\Theta||_2$;
      Optimize $Net$ according to $L$;
  end while
  return $Net$.
```

where tanh is the activation function, $\oplus$ is the concatenation operator, and $\mathbf{W}_{his}^s, \mathbf{W}_{his}^o \in R^{d\times 2d}$ and $\mathbf{b}_{his}^s, \mathbf{b}_{his}^o \in R^d$ are trainable parameters. $\mathbf{s}_\tau^w, \mathbf{r}_\tau^w$ denotes the entity and relation representation at time t through the wide-level process. $\mathbf{E}_\tau$ is the matrix of entity representations obtained at time τ through the wide-level process.

The loss function L_{Two} for the deep-level component is designed as per (14) as

$$L_{Two} = -\sum_{q=(s,r,o,\tau)\in Q} \log\left(\frac{\exp(\mathbf{F}_{his}^{s,r,\tau}(o_i))}{\sum_{o_j\in E}\exp(\mathbf{F}_{his}^{s,r,\tau}(o_j))} + \frac{\exp(\mathbf{F}_{his}^{o,r,\tau}(s_i))}{\sum_{s_j\in E}\exp(\mathbf{F}_{his}^{o,r,\tau}(s_j))}\right), \tag{15}$$

where Q represents all quadruples in the training set, $o_i \in E_{his}^{s,r,\tau}$ and $s_i \in E_{his}^{o,r,\tau}$ denote the historically associated entities for the entities s and o.

4.3 Parameter learning and reasoning optimization strategy

The overarching objective of parameter learning in our model is to minimize the aggregate loss function L,

$$L = \zeta L_{\mathrm{DM}} + L_{Two} + \zeta' ||\Theta||_2, \tag{16}$$

where ζ indicates the hyperparameter used to balance the two types of loss functions. ζ' is to control the regularization strength.

During reasoning, for a given query (s, r, o, τ), where $s \notin E$ is the unseen entity and $o \in E$ is the entity to be predicted. For the unseen entity s, DM-UEP prioritizes the generation of a virtual representation for $\mathbf{s}$. This representation, along with the relation $\mathbf{r}$ and the time τ, is fed into the two-phase framework. The model is used to calculate probability values for all candidate entities, denoted by

$$P(o|s, r, \tau) = P_\tau^{s,r}(o) = \mathrm{softmax}(\mathbf{F}_{his}^{s,r,\tau}). \tag{17}$$

Based on the probability values, all candidate entities are ranked, and the predictive performance of the model is evaluated using the mean reciprocal rank (MRR) and hit rate (Hits) metrics, applied in descending order. The detailed training process of DM-UEP is provided in Algorithm 1.

Table 2 Details of three open-world datasets for unseen entity prediction and six filtered datasets for conventional entity prediction.

Dataset	Entities	Relations	Training set	Original test set	New test set	Time interval
ICEWS18-filter	21085	251	373018	49545	47590	24 h
ICEWS14-filter	9722	248	74845	7371	7242	24 h
ICEWS05-15-filter	9332	248	368868	46159	41581	24 h
WIKI-filter	9009	24	539286	63110	36031	1 year
YAGO-filter	9736	10	161540	20026	18417	1 year
GDELT-filter	7146	238	1734399	305241	304477	15 min
ICEWS18-OW	21085	251	373018	49545	1728	24 h
ICEWS05-15-OW	9332	248	368868	46159	5169	24 h
WIKI-OW	9009	24	539286	63110	27045	1 year

5 Experiments

This section outlines a series of experiments conducted to test the proposed DM-UEP model. Due to the limitations of current public datasets in fully accommodating the task of unseen entity prediction in open-world scenarios, we initially constructed three datasets specifically tailored for predicting unseen entities based on existing datasets. Simultaneously, we refined the original datasets to better suit the evaluation of entity prediction in standard scenarios. All our datasets are publicly available for research purposes. Additionally, we conducted an in-depth analysis of DM-UEP by exploring four key questions designed to elucidate the capabilities of DM-UEP and highlight its superiority in unseen entity prediction.

- Q1: How does DM-UEP perform compared with state-of-the-art TKG representation and reasoning methods on the unseen entity and conventional entity prediction tasks?
- Q2: How does every module affect the performance of DM-UEP?
- Q3: How do loss function modules affect the performance of DM-UEP?
- Q4: How sensitive is DM-UEP with different hyper-parameters?

5.1 Experimental setup

5.1.1 *Datasets*

Commonly used public datasets for TKG tasks include ICEWS05-15 [32], ICEWS14 [33], ICEWS18 [34], GDELT [35], WIKI [36], and YAGO [32]. The test datasets of these datasets are somewhat disordered, containing queries related to unseen entities as well as queries where the predicted entity is unseen, leading to biased test results in existing models. Therefore, we extracted queries related to unseen entities from the valid and test datasets of the original datasets to construct new test datasets. The datasets derived from ICEWS18, ICEWS05-15, and WIKI, in particular, offer extensive test data conducive to evaluating tasks in open-world scenarios involving unseen entity prediction. Specifically, construct the original valid and test datasets quadruplets whose subject entities are unseen and object entities are original into new test datasets $\star$-OW. Detailed statistics of these three datasets are shown in Table 2.

In addressing conventional prediction tasks, the test datasets, including queries related to unseen entities, can adversely affect the accuracy of entity prediction. To facilitate a more effective evaluation of the performance of the model in entity prediction, we filtered six existing public datasets $\star$-filter (subject and object entities in quadruplets are original). This process involved the extraction and refinement of datasets to better suit the evaluation needs of standard scenarios. The detailed statistics of these refined datasets, post-filtering, are presented in Table 2.

The filtered test dataset eliminates noise knowledge that interferes with entity prediction tasks, making it more suitable for evaluating conventional entity prediction tasks and better distinguishing the advantages and disadvantages of different representation inference models.

5.1.2 *Baseline models*

For the unseen entity prediction task, we chose HGLS, the latest model in the direction of time series graph representation in recent years, RE-GCN, a representative model from [9], and TITer, which has a certain ability to predict unseen entities, as the benchmark models to compare with our method.

HGLS encodes the global graph structure through the designed hierarchical relational graph neural network HRGNN. At the sub layer level, RGCN is used to model static graphs under the same timestamp; At the whole graph level, the temporal influence is transformed into edge weights, and then attention mechanism is used to model the full graph features of the entity. HGLS models short-term and long-term impacts separately, and the semantics of relationships tend to stabilize in long-term changes. Therefore, the static representation of relationships is used as their long-term representation; and they use GRU to capture short-term changes in entity relationships.

RE-GCN, as an earlier representative paper in the time series research of the Li team, models entities and relationships based on the time line static atlas sequence, while focusing on the impact of historical information on current facts. Therefore, relying on GCN to capture the structural information characteristics of the atlas, an evolutionary unit is designed to model the historical information, in which the gate cycle component and GRU component are used to represent the evolution of entities and relationships, and entity static attributes (common sense knowledge) are also used to constrain entity representation.

TITer is currently one of the few models with the ability to predict unseen entities. It uses the form of a map snapshot sequence to arrange the time series knowledge map in chronological order to form a time series map chain. In order to capture relevant historical information from the atlas on the historical timestamp, TITer connects the atlas entities on different timestamps across time according to the relation. In this way, the search space for path search can be expanded, cross time search can be realized, and historical information can be directly reflected in cross time connection. At the same time, TITer preliminarily judges the type of unseen entity through the relation in the triplet to be predicted, and designs an induced mean (IM) algorithm, which uses the trained entity representation of the same type to set the initial value of the unseen entity, gradually updates the representation of the unseen entity through the co-occurrence relation between the unseen entity at different times and the relation category, and finally further generates the representation of the unseen entity through its category.

For the conventional entity prediction task, we compare our newly proposed DM-UEP model against a diverse array of state-of-the-art TKG models, including RE-NET [16], xERTE [37], CyGNet [38], RE-GCN [9], TITer [25], CEN [33], CENET [21], TECHS [39], and HGLS [17].

5.1.3 *Evaluation setting and metrics*

For the evaluation index, we still choose to use the general evaluation method of temporal knowledge graph representation and reasoning. That is, MRR and Hits@(1, 10). MRR calculates the average reciprocal of the ranks of positive fact candidates across all queries. Hits@K measures the proportion of instances where the positive fact candidates are ranked within the top K positions. These metrics are standard benchmarks for assessing the performance of extrapolation reasoning.

5.1.4 *Model configurations*

For all datasets, we maintain an embedding size of 200, aligning with the baseline methodology established in CyGNet. We employ a transformer with 8 encoders. The base embedding utilized is RGCN, with a Dropout setting of 0.2 to mitigate overfitting. The model parameters are optimized using the Adam optimizer, with a learning rate set to 0.001. The balance hyperparameter of loss is adjusted to 0.3, and the hyperparameter k for the graph extension time interval is established at 3. All experimental trials are executed on GeForce GTX 3090*2. Baseline results are obtained from testing on reconstructed data using codes publicly shared in previous studies. For models without available codes, we recreate the code based on descriptions in the respective papers and conduct comparative trials.

5.2 Results comparison (Q1)

5.2.1 *Unseen entity prediction task*

The results of the unseen entity prediction task are shown in Table 3. We test and compare our model with the three benchmark models on the newly constructed unseen entity prediction datasets and the filtered conventional entity prediction datasets.

Through the contrast experiment of unseen entity prediction in Table 3, we can find that our model has significantly improved the effect on the unseen entity prediction task. Among them, RE-GCN and HGLS have no targeted design for unseen entities, so compared with the conventional entity prediction, their effects are significantly reduced. When designing, TITer has made a targeted design for unseen entities,

Table 3 Performance (%) for the unseen entity prediction task on ICEWS18-OW, ICEWS05-15-OW, and WIKI-OW datasets. The best performance is in bold, and the second best is underlined.

Model	ICEWS18-OW		ICEWS05-15-OW		WIKI-OW	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
RE-GCN	16.55	14.69	16.46	13.27	43.36	40.67
TITer	21.56	19.80	18.73	17.06	56.91	54.82
HGLS	18.72	16.67	21.56	19.79	52.12	51.34
DM-UEP	**26.55**	**23.26**	**30.27**	**28.79**	**62.31**	**61.24**

Table 4 Performance (%) for the conventional entity prediction task on ICEWS14, ICEWS05-15, ICEWS18, and GDELT datasets. The best performance is in bold, and the second best is underlined.

Model	ICEWS14-filter			ICEWS05-15-filter			ICEWS18-filter			GDELT-filter		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
RE-NET	40.56	31.35	59.01	44.77	34.78	64.72	30.62	20.91	50.67	20.45	14.23	35.86
xERTE	41.68	33.41	58.34	47.21	38.58	65.82	30.22	22.24	47.93	20.98	13.12	36.14
CyGNet	38.85	29.78	59.19	43.25	31.12	64.61	28.36	18.34	48.67	22.31	14.45	37.88
RE-GCN	43.26	33.59	66.59	50.52	39.26	72.08	33.28	22.88	53.76	21.03	13.56	36.68
TITer	42.98	33.86	49.68	48.86	39.3	69.04	30.26	23.23	46.10	19.23	11.89	35.56
CEN	43.89	34.12	67.31	–	–	–	32.51	24.01	53.59	–	–	–
CENET	43.24	33.98	68.21	51.23	40.36	73.26	34.29	25.25	54.23	20.34	12.23	35.26
TECHS	45.23	36.78	69.89	52.23	41.25	74.55	35.58	26.23	54.65	–	–	–
HGLS	**48.21**	**37.03**	75.67	51.16	40.65	73.98	34.86	25.81	54.86	**23.43**	**14.86**	38.16
DM-UEP	48.02	36.86	**76.86**	**53.36**	**42.35**	**74.92**	**36.13**	**28.13**	**56.50**	23.21	14.32	**38.56**

so its performance in unseen entity prediction tasks is better than the current latest model HGLS. At the same time, our model has the most obvious improvement over other models in terms of WIKI data. This is because in the original dataset, unseen entities in the WIKI dataset account for a large proportion, and their distribution is more inclined to the unseen entity task. Due to the targeted design of our model, the model has better performance on it.

5.2.2 *Conventional entity prediction task*

The results of the conventional entity prediction task are shown in Tables 4 and 5. We test and compare our model with the nine benchmark models on the newly constructed filtered conventional entity prediction datasets.

According to the results of the conventional entity prediction task on the filtered data of each model in Table 4, compared with the results on the original data, the test results of the model have generally improved after removing the noise data. HGLS is still the best performing model among the comparison models so far. Since our model has no unseen entities in the conventional prediction, the DM module does not play a role. At this time, the two-phase module we built plays a major role. Through comparative experiments, we can find that our model human reasoning thinking framework still maintains a high performance, ranking the best and second best in all test indicators. Our method is affected by the DM module during the training process and will slightly affect the performance of the method in the conventional entity prediction task. So the performance of our method is slightly improved in conventional prediction, combined with the results in Table 3, we can intuitively show the excellent performance of the DM module in unseen entity prediction. On the dataset ICEWS14, due to the smaller amount of training data compared with other datasets, our method cannot aggregate information well by the graph expansion module and capture more historical information by the transformer module. Therefore, on the dataset ICEWS14, the performance of our method is slightly inferior to other methods.

5.3 Ablation study on different modules (Q2)

To clarify the contributions of key modules within our model towards DM-UEP, we conducted ablation experiments for both unseen entity prediction and conventional entity prediction tasks, utilizing MRR (mean reciprocal rank) as our evaluation metric.

Table 5 Performance (%) for the conventional entity prediction task on YAGO and WIKI datasets. The best performance is in bold, and the second best is underlined.

Model	YAGO-filter			WIKI-filter		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
RE-NET	69.35	61.45	88.67	69.24	61.23	84.26
xERTE	86.13	81.46	90.43	83.26	71.12	89.32
CyGNet	70.56	59.45	87.56	69.41	57.12	88.65
RE-GCN	84.32	80.15	90.42	89.23	87.61	90.12
TITer	88.64	81.32	91.13	87.68	85.12	89.89
CEN	–	–	–	**89.93**	87.05	90.91
TECHS	72.35	65.34	83.54	40.12	34.35	48.68
DM-UEP	**88.93**	**82.25**	**91.26**	89.68	**87.43**	**93.15**

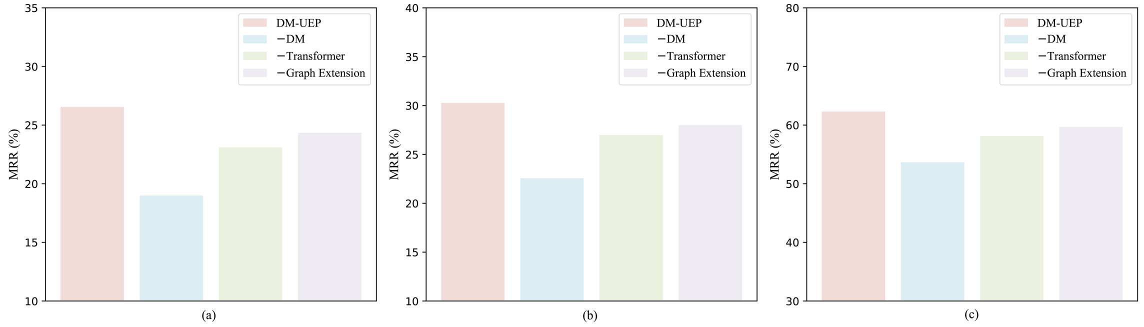

Figure 5 Impact of the main modules on MRR (%) in the unseen entity prediction task. By removing the three main modules (DM, Transformer, and Graph Extension), the unseen entity prediction performance of DM-UEP shows varying degrees of decline. The results emphasize the effectiveness of these three modules. (a) ICEWS18; (b) ICEWS05-15; (c) WIKI.

5.3.1 *Unseen entity prediction task*

Ablation experiments are carried out on three datasets: ICEWS18, ICEWS05-15, and WIKI. We closely examine the influence of three critical modules: the diffusion model (DM), Transformer, and Graph Extension. The results, as presented in Figure 5, include a comparative analysis of four variables: (1) the full DM-UEP model, (2) DM-UEP without DM (-DM), (3) DM-UEP without Transformer (-Transformer), and (4) DM-UEP without Graph Extension (-Graph Extension).

Overall, the ablation of these modules results in a significant reduction in MRR across all three TKG datasets. Particularly, the '-DM' variant leads to a noticeable decline in results, affirming the critical role our proposed diffusion model plays in the unseen entity prediction task.

The comparison with '-Transformer' indicates that Transformer can effectively capture the evolving characteristics of entities over time, enhancing the ability of our method to predict unseen entities. Furthermore, through the comparison experiment with '-Graph Extension', our designed Graph Extension mechanism can effectively aggregate relevant information at the current moment, providing effective assistance for the unseen entities prediction.

5.3.2 *Conventional entity prediction task*

For the conventional prediction task, we conduct ablation experiments on six filtered datasets. We analyze the impact of three key components: temporal evolution modeling, graph extension, and evaluation function components. The corresponding results are shown in Figure 6, which includes a comparative analysis of five variables: (1) the full DM-UEP model, (2) DM-UEP without Transformer (-Transformer), (3) DM-UEP replacing Transformer with GRU (-Transformer+GRU), (4) DM-UEP without Graph Extension (-Graph Extension), and (5) DM-UEP's evaluation function without $\mathbf{F}_{his}^{(o,r,\tau)}$(-$\mathbf{F}_{his}^{(o,r,\tau)}$).

Overall, the ablation of these modules leads to a significant decrease in the conventional entity prediction performance across six filtered datasets. Among them, the Transformer module has the greatest impact. Additionally, comparison experiments with GRU replacements demonstrate the importance of the temporal processing module, and also show that Transformer outperforms GRU in handling the effects of temporal evolution on entity representation update. Similar to the experiments on unseen entity prediction, the Graph Extension mechanism also plays a crucial role in the conventional entity prediction

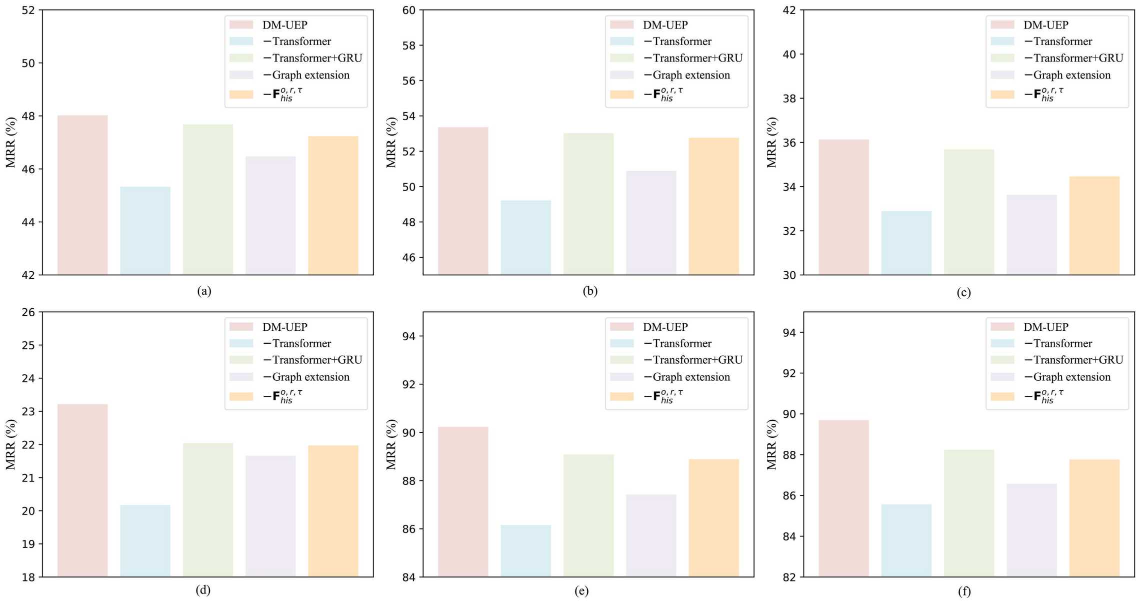

Figure 6 Impact of the main modules on MRR (%) in the conventional entity prediction task. By removing the three main modules (Transformer, Graph Extension, and $\mathbf{F}_{his}^{(o,r,\tau)}$), the conventional entity prediction performance of DM-UEP shows varying degrees of decline. The results emphasize the effectiveness of these three modules. (a) ICEWS14; (b) ICEWS05-15; (c) ICEWS18; (d) GDELT; (e) YAGO; (f) WIKI.

task.

In our method, although the head entity prediction task is not conducted in the experiments, $\mathbf{F}_{his}^{(o,r,\tau)}$ is still added during the construction of the evaluation function. The comparison with '-$\mathbf{F}_{his}^{(o,r,\tau)}$' shows that adding $\mathbf{F}_{his}^{(o,r,\tau)}$ indeed improves the performance of the tail entity prediction task. This might be because some knowledge in the graph is bidirectional, and adding $\mathbf{F}_{his}^{(o,r,\tau)}$ can better enhance the training effect of 1-to-N instances between head and tail entities, ultimately improving the overall tail entity prediction performance.

The ablation experiments conducted validate the effectiveness of the proposed modules for virtual representation generation of unseen entities, temporal processing with Transformer, and Graph Extension, in unseen entity prediction and conventional entity prediction tasks. The integration of these modules has significantly enhanced the prediction reasoning performance of DM-UEP, confirming their importance in both types of tasks.

5.4 Ablation study on loss function and evaluation function (Q3)

To analyze the impact of the evaluation and loss functions designed in the DM-UEP inference process on the final entity prediction tasks, we conduct detailed ablation studies to explore the effectiveness of the proposed diffusion model and two-phase framework. These ablation comparison results are shown in Figure 7, where the impact of key components within the loss and evaluation functions on the overall performance of DM-UEP is clearly displayed.

The analysis clearly shows that the absence of loss (L_{Two}) leads to a significant decline in model performance. This outcome highlights the substantial impact of entity and relation representations on the results of the unseen entity prediction task. Similarly, the ablation of loss ($L_{\rm DM}$) also results in decreased model performance, corroborating the findings presented in Figure 5 and emphasizing the importance of the unseen entity virtual representation generation module DM which we designed. The removal of L_{Two} ($\mathbf{F}_{his}^{(o,r,\tau)}$) from the evaluation function, as corroborated by the results in Figure 6, once again demonstrates that adding $\mathbf{F}_{his}^{(o,r,\tau)}$ can further improve the entity prediction effect.

From these ablation studies, it is evident that both parts L_{Two} and $L_{\rm DM}$ in the loss function of our method design hold significant value, and their combination leads to better outcomes. Our method, DM-UEP, can effectively generate a virtual representation of unseen entities in the unseen entity prediction task, thereby enhancing the temporal knowledge graph reasoning capabilities in open-world scenarios.

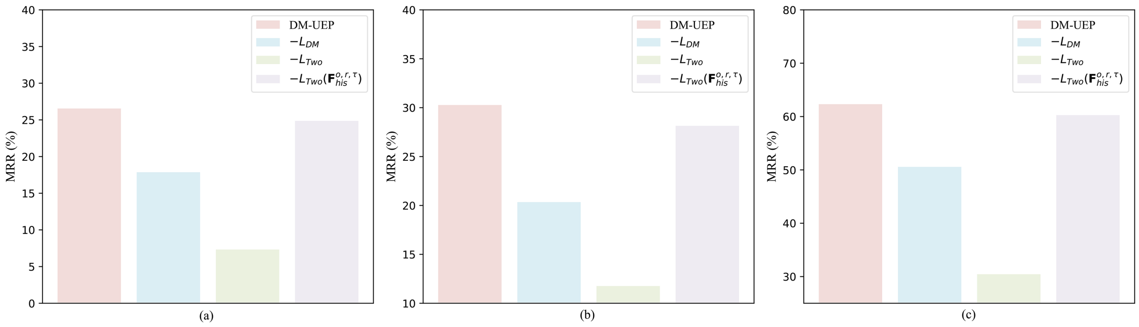

Figure 7 Impact of loss functions and evaluation metrics on MRR (%). Through ablation experiments, it has been proven that the loss generated by the unseen entity representation module and the loss generated by the entity-relation representation play a crucial role in the task of unseen entity prediction. Meanwhile, the comparative results also illustrate the effectiveness of $\mathbf{F}_{his}^{(o,r,\tau)}$ in the evaluation metrics. (a) ICEWS18; (b) ICEWS05-15; (c) WIKI.

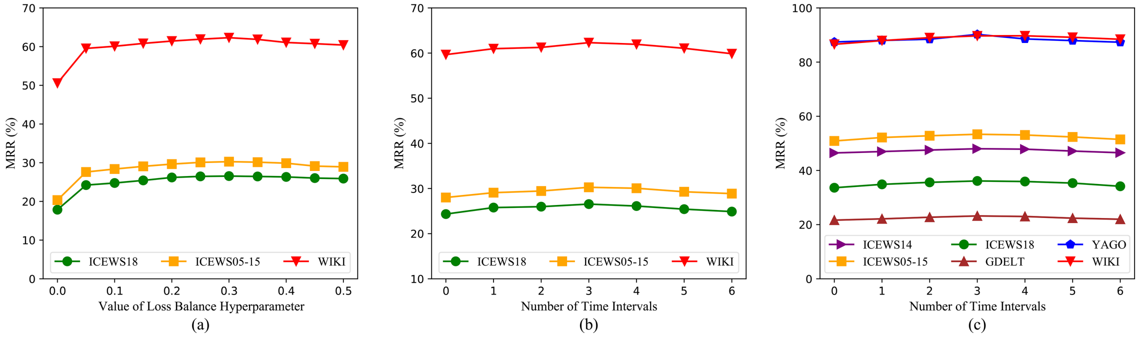

Figure 8 Impact of (a) loss balance hyperparameter ζ, (b) hyperparameter k in new entity prediction, and (c) hyperparameter k in conventional entity prediction variations on DM-UEP. When the hyperparameter ζ is set to 0.3, our method achieves the best performance. Setting the graph expansion time interval value to 3 yields the optimal result for the model.

5.5 Hyperparameter study (Q4)

In this analysis, we delve into the performance of DM-UEP under various hyperparameters.

5.5.1 *Balance hyperparameter ζ of loss*

As shown in Figure 8(a), the performance of our method improves as the initial proportion of the DM module loss function increases, but further increases in weight lead to a gradual decline in method performance. This trend indicates that the DM module plays a significant role in unseen entity prediction, but the modeling and representation of entities and relations are the core of the entire method. A DM loss proportion of around 0.3 allows the method to achieve its optimal state.

5.5.2 *Hyperparameter k in unseen and conventional entity prediction*

As Figures 8(b) and (c) demonstrate, the time interval value k for graph extension, shows that aggregating information from three time intervals before and after the current moment can optimize subsequent prediction performance. Aggregating information beyond three time intervals leads to a gradual decrease in model performance. This is because information from overly long time intervals can introduce a large amount of data, the impact of which on the current entity significantly fades as the interval lengthens, resulting in a lot of redundant information. Moreover, graph extension aims to capture the contextual information of an entity at the current moment, emulating the second step of human reasoning and playing a critical role in the wide level of our two-phase reasoning framework. Thus, overly long intervals diminish the importance of current scenario information, while overly short intervals reduce the aggregation of current scenario information, leading to a loss of relevant contextual information for the entity. Therefore, based on the results depicted, setting the time interval for graph extension around three achieves better outcomes, enhancing the entity prediction capability of our method significantly.

Table 6 Runtime (s) comparison with RE-GCN and TITer.

	Conventional entity prediction task on ICEWS14			Unseen entity prediction task on ICEWS18		
	TITer	RE-GCN	DM-UEP	TITer	DM-UEP	RE-GCN
Runtime	82.74	11.29	10.67	552.02	106.88	46.77

Table 7 Performance on the relation prediction task. The best results are in bold.

	ICEWS18-OW	ICEWS05-15-OW
RE-GCN	20.18	21.77
DM-UEP	**31.43**	**33.72**

5.6 Comparison on prediction time

To study the efficiency of DM-UEP, we compare its prediction runtime with two baseline models, RE-GCN and TITer, on the conventional entity prediction task and the unseen entity prediction task. For a fair comparison, the experiments are conducted in the same environment (RTX1080Ti) and on the same test datasets.

In the conventional entity prediction task, we compare the prediction time of the three models using ICEWS14. As can be seen from the results in Table 6, DM-UEP is slightly more efficient than RE-GCN, and TITer is significantly less efficient than both. This is because DM-UEP does not need to generate virtual representations for entities on a conventional entity prediction task, and the Transformer handles temporal information more efficiently.

In the unseen entity prediction task, we compare the prediction time of the three models using ICEWS18. As shown in Table 6, DM-UEP's prediction efficiency is five times higher than TITer's. This is because DM-UEP generates virtual representations through a diffusion model, which not only produces more accurate new entity representations but also significantly reduces processing time compared with TITer's path reasoning. However, due to the additional step of generating virtual representations, DM-UEP's prediction speed on the unseen entity prediction task is only half that of RE-GCN (though DM-UEP's prediction performance is 1.6 times better than RE-GCN's).

5.7 Scalability study

To study the scalability of DM-UEP, we compare its performance with the baseline model RE-GCN in relation prediction experiments on Open-World knowledge graphs using the ICEWS18-OW and ICEWS05-15-OW datasets. Unlike conventional relation prediction experiments, in Open-World knowledge graph relation prediction, the head entities of the quadruplets are unseen, meaning s is an unseen entity in $(s, ?, o, \tau)$.

As shown in the results in Table 7, our model significantly outperforms the baseline model RE-GCN in relation prediction on both datasets. The comparative experiments demonstrate that our model is not only effective in entity prediction tasks in open-world scenarios but also excels in relation prediction. This is because the unseen entities' virtual representations generated by the diffusion model effectively simulate the characteristics of unseen entities. Our model, DM-UEP, exhibits good scalability.

5.8 Case study

To understand how our model generates virtual representations for unseen entities in open-world scenarios, we visualized two test quadruples as shown in Table 8. These cases demonstrate that the virtual representations generated by our model, DM-UEP, are capable of reflecting the connections between unseen entities, queries, and the original knowledge graph to a certain extent. By calculating the distances between the virtual representations of unseen entities and the representations of other entities, we visualize the entities with the highest similarities. It can be observed that these entities share similar features with the unseen entities, indirectly proving the feasibility of generating virtual representations through the diffusion model. In the example, Cristiano Ronaldo and Romain Edouard are the unseen entities not present in the training set. Through the virtual representations generated and the tail entity predictions ranked by the subsequent two-phase framework, it can be seen that the final answers, Real Madrid CF and International Master, are ranked first respectively, indicating that our model is somewhat effective in predicting unseen entities.

Table 8 Case study. Bold entities in the query indicate the unseen entities.

Query (new entity prediction)	Similar entities computed by virtual representation	Top 5 of prediction results	Answer
(**Cristiano Ronaldo**, member of sports team, ?, 2015)	Santiago Canizares, Alfredo Di Stefano, Juan Esnaider, Luis Aragones, Ferenc Puskas	Real Madrid CF, Argentina men's national football team, Portugal men's national football team, Manchester United F.C., Real Zaragoza	Real Madrid CF
(**Romain Edouard**, title of chess person, ?, 2007)	Wolfgang Unzicker, Erich Eliskases, Ludek Pachman, Arluro Pomar, Mark Taimanov	International Master, Grandmaster, Johannes Hendrikus Donner, Johan Hellsten, Christian Bauer	International Master

6 Conclusion

This study provides a comprehensive analysis of TKGs, focusing specifically on addressing the challenges encountered in open-world scenarios. Our introduction of the innovative DM-UEP method represents a considerable advancement in predicting unseen entities. This method enhances the reasoning capabilities of TKGs, demonstrating notable effectiveness in complex scenarios. Furthermore, we have reconstructed three datasets suitable for the unseen entity prediction task and six datasets suitable for the conventional entity prediction task, all based on six existing public datasets.

However, the datasets we constructed still fall short of fully capturing the diversity of real-world scenarios. This limitation prevents us from thoroughly testing and demonstrating the model's effectiveness in real-world applications. In the future, it will be essential to develop more suitable real-world datasets for open-world scenarios. Such datasets would enable more robust testing of the model performance and better equip it to handle more complex open-world challenges.

Acknowledgements This work was partially supported by National Key R&D Program of China (Grant No. 2022YFB3103600), National Natural Science Foundation of China (Grant Nos. U23A20296, 62272469, 72371245), and Science and Technology Innovation Program of Hunan Province (Grant No. 2023RC1007).

References

1 Jiang T, Liu T, Ge T, et al. Encoding temporal information for time-aware link prediction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 2350–2354

2 Jiang T, Liu T, Ge T, et al. Towards time-aware knowledge graph completion. In: Proceedings of the 26th International Conference on Computational Linguistics, Osaka, 2016. 1715–1724

3 Han Z, Ma Y, Wang Y, et al. Graph Hawkes neural network for forecasting on temporal knowledge graphs. In: Proceedings of Conference on Automated Knowledge Base Construction, 2020

4 Zhang Y J, Li P F, Zhu Q M. Event temporal relation classification method based on self-attention mechanism. Comput Sci, 2019, 46: 244–248

5 Wang Y, Xiao W, Tan Z, et al. Caps-OWKG: a capsule network model for open-world knowledge graph. Int J Mach Learn Cyber, 2021, 12: 1627–1637

6 Shi B, Weninger T. Open-world knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, 2018. 1957–1964

7 Shah H, Villmow J, Ulges A, et al. An open-world extension to knowledge graph completion models. In: Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, 2019. 3044–3051

8 Guan S, Cheng X, Bai L, et al. What is event knowledge graph: a survey. IEEE Trans Knowl Data Eng, 2022, 35: 7569–7589

9 Li Z, Jin X, Li W, et al. 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, 2021. 408–417

10 Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks. In: Proceedings of the Semantic Web: 15th International Conference, Heraklion, 2018. 593–607

11 Luo C. Understanding diffusion models: a unified perspective. 2022. ArXiv:2208.11970

12 Sloman S A. The empirical case for two systems of reasoning. Psychol Bull, 1996, 119: 3–22

13 Li Z, Jin X, Guan S, et al. 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, 2021. 4732–4743

14 Xu C, Nayyeri M, Chen Y Y, et al. Geometric algebra based embeddings for static and temporal knowledge graph completion. IEEE Trans Knowl Data Eng, 2022, 35: 4838–4851

15 Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, 7370–7377

16 Jin W, Qu M, Jin X, et al. Recurrent event network: autoregressive structure inference over temporal knowledge graphs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 6669–6683

17 Zhang M, Xia Y, Liu Q, et al. Learning long-and short-term representations for temporal knowledge graph reasoning. In: Proceedings of the ACM Web Conference, Austin, 2023. 2412–2422

18 Jin J. Wan H, Lin Y. Knowledge graph representation learning fused with entity category information. Comput Engin, 2021, 47: 77–83

19 Trivedi R, Dai H, Wang Y, et al. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, 2017. 3462–3471

20 Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural network. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, 2011. 315–323

21 Xu Y, Ou J, Xu H, et al. Temporal knowledge graph reasoning with historical contrastive learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Washington, 2023. 4765–4773

22 Zhang J, Liang S, Sheng Y, et al. Temporal knowledge graph representation learning with local and global evolutions. Knowledge-Based Syst, 2022, 251: 109234

23 Dasgupta S S, Ray S N, Talukdar P. Hyte: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 2001–2011

24 Wu J, Cao M, Cheung J C K, et al. Temp: temporal message passing for temporal knowledge graph completion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 5730–5746

25 Sun H, Zhong J, Ma Y, et al. TimeTraveler: reinforcement learning for temporal knowledge graph forecasting. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, 2021. 8306–8319

26 Wang J, Kai S, Luo L, et al. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. In: Proceedings of Advances in Neural Information Processing Systems, 2025. 37: 8384–8410

27 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 30: 5998–6008

28 Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of the International Conference on Machine Learning, Lile, 2015. 2256–2265

29 Hawkes A G. Spectra of some self-exciting and mutually exciting point processes. Biometrika, 1971, 58: 83–90

30 Xiong R, Yang Y, He D, et al. On layer normalization in the transformer architecture. In: Proceedings of the International Conference on Machine Learning, 2020. 10524–10533

31 Tay Y, Dehghani M, Bahri D, et al. Efficient transformers: a survey. ACM Comput Surv, 2022, 55: 1–28

32 García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 4816–4821

33 Li Z, Guan S, Jin X, et al. Complex evolutional pattern learning for temporal knowledge graph reasoning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, 2022. 290–296

34 Ward M D, Beger A, Cutler J, et al. Comparing GDELT and ICEWS event data. Analysis, 2013, 21: 267–297

35 Zhang F, Zhang Z, Ao X, et al. Along the time: timeline-traced embedding for temporal knowledge graph completion. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, 2022, 2529–2538

36 Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Commun ACM, 2014, 57: 78–85

37 Han Z, Chen P, Ma Y, et al. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In: Proceedings of the International Conference on Learning Representations, 2020

38 Zhu C, Chen M, Fan C, et al. Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 4732–4740

39 Lin Q, Liu J, Mao R, et al. TECHS: temporal logical graph networks for explainable extrapolation reasoning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, 2023. 1281–1293