DLTKG: Denoising Logic-based Temporal Knowledge Graph Reasoning

Anonymous ACL submission

Abstract

Temporal knowledge graph (TKG) reasoning, a central task in temporal knowledge representation, focuses on predicting future facts by leveraging historical temporal contexts. However, current approaches face two major challenges: limited generalization to unseen facts and insufficient interpretability of reasoning processes. To address these challenges, this paper proposes the **D**enoising **L**ogic-based **T**emporal **K**nowledge **G**raph (**DLTKG**) framework, which employs a denoising diffusion process to complete reasoning tasks by introducing a noise source and a historical condition-guiding mechanism. Specifically, DLTKG constructs fuzzy entity representations by treating historical facts as noise sources, thereby enhancing the semantic associations between entities and the generalization ability for unseen facts. Additionally, a condition-based guidance mechanism, rooted in the relationship evolutionary paths, is designed to improve the interpretability of the reasoning process. Furthermore, we introduce a fine-tuning strategy that optimizes the denoising process by leveraging shortest path information between head entity and candidate entities. Experimental results on three benchmark datasets demonstrate that DLTKG outperforms state-of-the-art methods across multiple evaluation metrics[1].

1 Introduction

The temporal knowledge graph (TKG) (Gottschalk and Elena, 2018; Zhao, 2021) is a dynamic multirelational graph structure represented in the form of quadruples (s, r, o, t), where s denotes the subject (*i.e.*, head entity), r denotes the relation, o denotes the object (*i.e.*, tail entity), and t denotes the timestamp. The reasoning tasks of TKGs are primarily divided into interpolation and extrapolation (Jin et al., 2020). The *interpolation* task (Xu et al., 2020; Xiong et al., 2024) involves inferring missing facts within a known time interval, while the *extrapolation* task (Xu et al., 2020; Sun et al., 2021; Liu et al., 2022) focuses on predicting future events. This study specifically addresses the extrapolation task of TKGs, as it can forecast future events and provide forward-looking insights for decision-making, offering substantial practical value in areas such as event prediction (Deng et al., 2020), risk prediction (Jhee et al., 2025), and trend analysis (Choudhury et al., 2020).

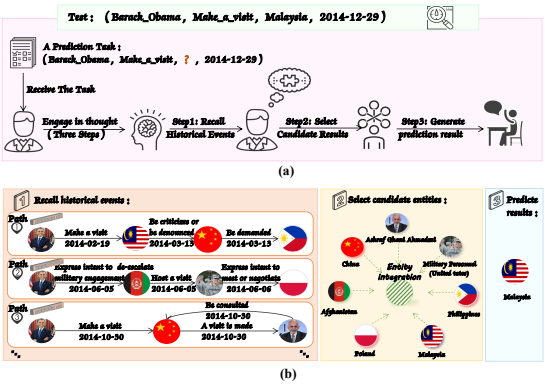


Figure 1: (a) The thought process involved in making predictions. (b) An example of reasoning used to answer the query in (a).

Recent studies (Hahamy et al., 2023; Kolibius et al., 2025) suggest that during narrative comprehension, humans activate neural representations of relevant historical events at event boundaries, facilitated by the hippocampus and default mode network. This enables memory integration and knowledge structure updates across time scales. As shown in Figure 1(a), humans follow a three-step process in prediction tasks: recalling historical events, filtering potential answers, and combining personal experience to form a prediction. Figure 1(b) illustrates TKG extrapolation, where the task is to predict whom Obama visited on December 29, 2014. The process begins with

[1]Code is available at https://anonymous.4open.science/r/DLTKG-7CCB/

recalling events related to "Make_a_visit" and "Barack_Obama", filtering outcomes like Malaysia, China, Poland, and concluding that Obama visited Malaysia.

TKG reasoning has seen notable progress in recent years (Trivedi et al., 2017; Wang et al., 2023; Chen and Chen, 2024), as will be discussed in Section 2. Motivated by the success of diffusion techniques in sequence-to-sequence tasks (Gong et al., 2023), DiffuTKG (Cai et al., 2024) introduces diffusion methods to TKG reasoning, achieving competitive results. However, existing diffusion models (Cai et al., 2024) lack interpretability, as random noise and denoising are disconnected from the target entities. To address this, we propose a noise-adding and denoising method based on historical evolutionary paths (HEPs), leveraging past entities to generate fuzzy entities, i.e., the memory fusion process. Denoising these fuzzy entities enhances the interpretability and generalization of the reasoning process.

Specifically, we propose three innovative strategies: (1) ***Sequence Learning Strategy***: This module extracts the HEPs of each relationship and utilizes them as denoising conditional guiding information. This design enables DLTKG to perform efficient logical reasoning based on HEPs. (2) ***Entity Fusion Strategy***: This strategy merges entities that have appeared in HEPs as memory information to obtain noisy fuzzy entities. Given the strong semantic and structural correlations between historical and target entities, key information about the target entity may be implicitly embedded within the historical entities. Consequently, generating noise through the fusion of historical entities is more justifiable. (3) ***Fine-tuning Strategy***: After the initial round of denoising, candidate entities are ranked from high to low based on their scores, and the top k entities are selected. The shortest paths between the query head entity and the top k candidate entities are then obtained, and these shortest paths are used as guiding information for further denoising. Empirical research on three benchmark datasets validates the effectiveness of DLTKG.

The main contributions are as follows:

- To the best of our knowledge, DLTKG is the first model to apply memory fusion strategy to diffusion-based temporal knowledge graph reasoning, aiming to enhance the logical interpretability of model through the noise-adding and denoising process.
- We propose a fuzzy entity noise addition strategy and introduce a historical condition guidance mechanism, aiming to enhance the correlation between historical events and thereby delve deeper into the potential information between entities and relationships.
- We propose a fine-tuning strategy that utilizes shortest path information between the head entity and candidate entities to optimize the denoising process, further enhancing the relevance between the query and historical knowledge.
- DLTKG significantly surpasses the existing diffusion-based models on three representative TKGR datasets, including ICEWS14, ICEWS05-15, YAGO, and achieves competitive performance with the other state-of-the-art baselines.

2 Related Work

2.1 Temporal Knowledge Graph Reasoning

Existing TKG extrapolation methods can be broadly categorized into four types: (1) **Embedding-based models** dynamically model temporal evolutionary patterns of entities and relationships using low-dimensional vectors, inferring missing facts through the similarity of historical embeddings. Representative models include CyGNet (Zhu et al., 2021), HIP (He et al., 2021), among others. (2) **Graph neural network-based models** focus on uncovering structural evolution in temporal knowledge graphs, predicting dynamic associations by aggregating neighborhood information through message passing, e.g., xERTE (Han et al., 2020) and SRPL (Li et al., 2024). (3) **Rule-based models** focus on inductively deriving interpretable logical rules from historical facts. TLogic (Liu et al., 2022) extracts interpretable temporal logic rules through temporal random walks. TempValid (Huang et al., 2024) models the temporal validity of rule confidence and designs learnable temporal functions. (4) **Language model-based models** treat entities and relationships as semantic symbols, using generative models to predict knowledge completion, such as CoH (Xia et al., 2024) and GenTKG (Liao et al., 2024).

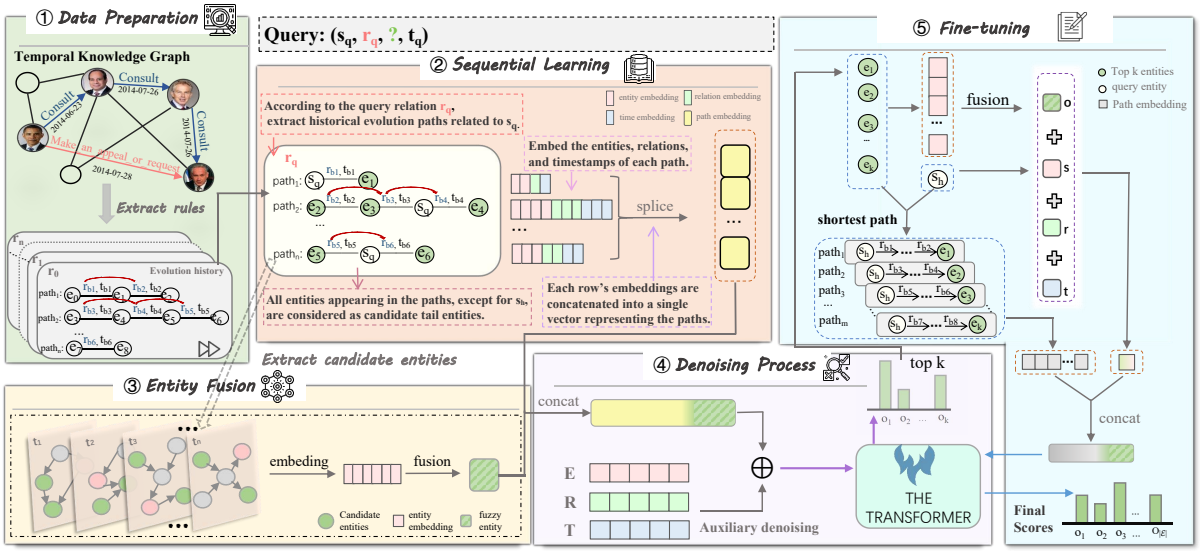


Figure 2: Overview of DLTKG structure. DLTKG mainly consists of five parts: (1) Data Processing, which is used to obtain the historical evolutionary paths (HEPs) of relationships; (2) Sequence Learning Module, which captures historical evolutionary information; (3) Entity Fusion Module, which generates fuzzy entity representations from noise sources; (4) Denoising Module, which cleans fuzzy entities using the HEPs; (5) Fine-tuning Module, which further optimizes the denoising process. The core components are (2), (3), and (5).

2.2 Diffusion Model

Diffusion models are a type of generative model that learn data distributions by gradually adding and removing noise, and they are commonly used for high-quality image and audio generation. Currently, some research has explored text diffusion models in discrete state spaces (Li et al., 2022a; Reid et al., 2023; Gong et al., 2023).

DiffuTKG (Cai et al., 2024) is the first model to introduce diffusion methods into TKG reasoning tasks, by introducing random noise to target entities and reconstructing the entities from it.

Unlike DiffuTKG, our approach: (1) constructs entity-related noise sources that are not random; (2) utilizes logical reasoning information about relationship evolution as conditional guidance for the denoising process. DLTKG enables collaborative modeling of semantic correlation and temporal dependence, leading to more accurate predictions.

3 Problem Formulation

A Temporal Knowledge Graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{Q})$ (Gu et al., 2022) is a directed multirelational graph where there are timestamped edges between entities, with $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ representing the sets of entities, relations, and timestamps, respectively. $\mathcal{Q} = \{(e_s, r, e_o, t) \mid e_s, e_o \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{T}\}$ is a set of quadruples in $\mathcal{G}$. The TKG is viewed as a series of snapshots arranged in ascending order of timestamps, denoted as $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{|\mathcal{T}|}\}$.

The problem addressed in this paper is Temporal Knowledge Graph Reasoning (TKGR) through extrapolation, which is formalized as *link prediction* aimed at inferring future quadruples. Formally, for a quadruple query $(s_q, r_q, ?, t_q)$, the goal of extrapolated TKGR is to predict the missing entity o_q, given the historical graph sequence $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{t_q-1}\}$ prior to the prediction time t_q.

4 Method

The DLTKG framework (Figure 2) consists of several key components: (1) *Sequence learning*, which helps denoise by extracting relationship evolutionary patterns as conditional information; (2) *Entity fusion strategy*, which integrates historical information related to the target entity to obtain a fuzzy entity, modeling the potential uncertainty between entities; (3) *Fine-tuning strategy*, which uses the top k candidate entities and the query head entity s_q to calculate shortest paths, providing conditional information for further training.

4.1 Sequence Learning

We frame the link prediction task as a sequence prediction problem, focusing on exploring HEPs of re-

lationships. Section 4.1.1 derives all possible HEPs using temporal walks. Section 4.1.2 constructs a path encoding representation. Section 4.1.3 filters the most relevant HEPs to the query relationship r_q using a relevance discrimination function.

4.1.1 Historical Evolution of Relationship Exploration

We extract temporal walks from the TKG $\mathcal{G}$ as follows: For a historical evolutionary path (HEP) of length ℓ, we sample a non-increasing random walk sequence of length $\ell+1$, where the additional step corresponds to querying events for relation r_q. The walk starts by randomly sampling an edge $(e_1, r_q, e_{\ell+1}, t_{\ell+1})$, then iteratively sampling adjacent edges until it reaches length $\ell+1$. In the final step, if an edge links back to the first entity e_1, we sample it; otherwise, we proceed to the next path.

For sampling steps $s \in \{2, 3, \ldots, \ell+1\}$, let (e_s, r, e_o, t) represent the edge sampled previously, and $\mathcal{N}_e(s, e_o, t)$ denote the set of feasible edges for the next transition. To satisfy the temporal constraints, we define $\mathcal{N}_e(s, e_o, t) :=$

$$\begin{cases} \{(e_o, r, e, \hat{t}) \mid (e_o, r, e, \hat{t}) \in \mathcal{G}, \hat{t} < t\} \\ \qquad\qquad \textit{if } s = 2, \\ \left\{(e_o, r, e, \hat{t}) \mid (e_o, r, e, \hat{t}) \in \tilde{\mathcal{G}}, \hat{t} \leq t\right\} \\ \qquad\qquad \textit{if } s \in \{3, \ldots, \ell\}, \\ \left\{(e_o, r, e_1, \hat{t}) \mid (e_o, r, e_1, \hat{t}) \in \tilde{\mathcal{G}}, \hat{t} \leq t\right\} \\ \qquad\qquad \textit{if } s = \ell+1, \end{cases} \tag{1}$$

where $\hat{\mathcal{G}} := \mathcal{G} \setminus \left\{\left(e_o, r^{-1}, e_s, t\right)\right\}$ excludes the inverse edges to avoid redundant rules. Remove the edges sampled in the first step, then arrange the remaining random walk sequence of length ℓ in reverse chronological order. This results in a HEP of r_q denoted as $\boldsymbol{p}_{r_q}^{\ell}$:

$$\begin{aligned} &((e_1, r^{-1}, e_2, t_1), \ldots, (e_\ell, r^{-1}, e_{\ell+1}, t_\ell)) \\ &\quad \textit{with} \quad t_\ell > t_{\ell-1} \ldots > t_1. \end{aligned} \tag{2}$$

For each relation $r \in \mathcal{R}$, we draw $n \in \mathcal{N} = \{1, \ldots, N\}$ time walks from a pre-specified set of lengths $\mathcal{L}$. The set $\mathcal{W}_r^{\ell}$ denotes all evolving paths of length ℓ that are headed by relation r. All HEPs of relation r are contained in $\mathcal{W}_r := \bigcup_{\ell \in \mathcal{L}} \mathcal{W}_r^{\ell}$ and the complete set of learned HEPs is $\mathcal{W} := \bigcup_{r \in \mathcal{R}} \mathcal{W}_r$.

4.1.2 Sequence Prediction

Let $\mathcal{G}_{0:t_q-1}$ be the historical TKG snapshot, and $\boldsymbol{q}_t = (s_q, r_q, ?, t_q)$ be the query quadruple. Inspired by DiffuTKG (Cai et al., 2024), we reshape the task into a sequence prediction problem. The difference is that we predict the missing entities in $\boldsymbol{q}_t$ by observing the historical evolutionary patterns.

First, we extract the HEPs $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_i, \ldots, \mathcal{P}_{n-1}\}$ of the query relation r_q from $\mathcal{W}$, where each path contains at least one query subject s_q, and the length of each path is ℓ. Each path is represented as $\mathcal{P}_i = \{(e_0, r_0, e_1, t_0), \ldots, (e_{\ell-1}, r_{\ell-1}, e_\ell, t_{\ell-1})\}$. Additionally, let $\mathcal{P}_i = \left\{\mathcal{S}_e^i, \mathcal{S}_r^i, \mathcal{S}_t^i\right\}$, where $\mathcal{S}_e^i = \{e_0, \ldots, e_\ell\}$ represents the sequence of entities in the HEP p_i, $\mathcal{S}_r^i = \{r_0, \ldots, r_{\ell-1}\}$ represents the sequence of relations in the HEP, and $\mathcal{S}_t^i = \{t_0, \ldots, t_{\ell-1}\}$ represents the sequence of timestamps in the HEP.

Next, we obtain the representations of entities, relations, and time in each HEP as follows:

$$\mathbf{e} = \sum_{i=0}^{\ell} \mathbf{E}(e_i), \ \mathbf{t} = \sum_{k=0}^{\ell-1} \mathbf{T}(t_k), \ \mathbf{r} = \sum_{j=0}^{\ell-1} \mathbf{R}(r_j), \tag{3}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times d}$, $\mathbf{R} \in \mathbb{R}^{2|\mathcal{R}| \times d}$, $\mathbf{T} \in \mathbb{R}^{|\mathcal{T}| \times d}$, $\mathbf{e}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$, d represents the size of the hidden dimension. We combine entity, relationship, and timestamp embeddings along the HEPs to obtain the final evolutionary embeddings, as follows:

$$\mathbf{p}_i = \mathbf{e} + \mathbf{r} + \mathbf{t}. \tag{4}$$

The embedded representation of HEPs is denoted as $\mathbf{P} = [\mathbf{p}_0; \ldots; \mathbf{p}_{n-1}]$, where $\mathbf{P} \in \mathbb{R}^{n \times d}$, $[;]$ represents the concatenation operation.

4.1.3 Path Selection

We use the path relevance discrimination function $\Psi(\cdot)$ to filter the HEPs most relevant to the query. Given the query relation r_q, we obtain the embedded representation of HEPs, i.e. $\mathbf{P}$, and we apply the path relevance discrimination function:

$$\Psi(\mathbf{P}, r_q) = \|\mathbf{P} \circ \mathbf{E}(r_q)\|^2 > \lambda, \tag{5}$$

where $\circ$ denotes the Hadamard product operation, λ represents the adaptive threshold. The path embeddings that satisfy $\Psi(\mathbf{P}, r_q)$ = True form the condition-guided set $\tilde{\mathbf{P}}$, and the corresponding set of HEPs is $\tilde{\mathcal{P}}$.

4.2 Entity Fusion Strategy

After obtaining the HEPs related to the query relationship r_q and the query subject s_q, we extract the relevant candidate entities from the HEPs $\tilde{\mathcal{P}}$. The extracted entities are distinct and are represented as the set $\mathcal{C}$:

$$\mathcal{C} = \mathcal{S}_e^0 \cup \mathcal{S}_e^1 \cup \cdots \cup \mathcal{S}_e^{|\tilde{\mathcal{P}}|}. \tag{6}$$

Based on the Historical Recurrence Hypothesis (Trompf, 1979), we propose a fuzzy entity construction method based on maximum entropy fusion: by using a nonlinear aggregation function $\Phi(\cdot)$ to perform information fusion on the entity set $\mathcal{C}$. The specific process is as follows:

$$\Phi(\{\mathbf{e}_i\}) = \underset{\hat{\mathbf{e}}_1}{argmin} \left(\sum_{i=0}^{|\mathcal{C}|} \omega_i D_{KL}\left(f(\mathbf{e}_i) \parallel f(\hat{\mathbf{e}}_1)\right) + \lambda L(\hat{\mathbf{e}}_1)\right), \tag{7}$$

$$f(\mathbf{e}_i) = softmax\left(\mathbf{W}\mathbf{e}_i + \mathbf{b}\right), \tag{8}$$

$$w_i = \exp\left(-\gamma\left(t_q - t_i\right)\right), \tag{9}$$

where D_{KL} represents the KL divergence, $\mathbf{e}_i$, $\hat{\mathbf{e}}_1 \in \mathbb{R}^d$ denote the historical candidate entity embeddings and the fuzzy entity embeddings, respectively. The function $f(\cdot)$ is a probability mapping function, w_i is the time decay weight, and γ controls the decay rate. The variable t_i indicates the timestamp corresponding to the entity e_i, while $L(\cdot)$ is the L_2 regularization term. $\mathbf{W}$, $\mathbf{b}$, and λ are learnable parameters.

4.3 Auxiliary Denoising Strategy

During the denoising phase, DLTKG cleans the fuzzty entity $\hat{e}_1$ to obtain the target entity $\hat{e}_q$, using historical information as a condition. This approach relieves the need for additional classifier training. Following DiffusEQ (Gong et al., 2023), we use a Transformer architecture to model f_θ, where historical information is inherently considered during the cleaning process. The denoising process is as follows:

$$\hat{\mathbf{e}}_q = Transformer(\tilde{\mathbf{e}}), \tag{10}$$

$$\tilde{\mathbf{e}} = [\tilde{\mathbf{P}}; \hat{\mathbf{e}}_1] + \mathbf{E}(s_q) + \mathbf{R}(r_q) + \mathbf{T}(t_q), \tag{11}$$

where $\hat{\mathbf{e}}_q \in \mathbb{R}^{|\mathcal{E}|}$. We introduce the query subject s_q, the query relation r_q, and the query time t_q to strengthen the connection between the query problem and the target entity.

4.4 Fine-tuning Strategy

To enhance the ability of model to recognize low-discriminative entities, we employ a fine-tuning strategy. After the initial denoising, we select top k candidate entities based on their scores for further fine-tuning. Building on the query relation evolutionary path features used in the initial training, this phase further strengthens the semantic association between the query head entity and the target entity. Specifically, we introduce the multi-hop shortest paths between the head entity and each candidate entity as auxiliary guiding information. This process is described as follows:

$$\mathbf{p}^*_{s_q \to e_k} = \sum\nolimits_{x \in path(s_q, e_k)} \begin{cases} \mathbf{E}(x) & if\ x \in \mathcal{E}, \\ \mathbf{R}(x) & if\ x \in \mathcal{R}, \\ \mathbf{T}(x) & if\ x \in \mathcal{T}, \end{cases} \tag{12}$$

$$\tilde{\mathbf{P}}^* = [\mathbf{p}^*_0; \mathbf{p}^*_1; \ldots; \mathbf{p}^*_{m-1}], \tag{13}$$

where x represents the entity, relationship, or timestamp within the path, m represents the total number of shortest paths, $path(s_q, e_k)$ refers to the shortest path between the head and tail entity. To maintain consistency in the noise injection strategy, we perform feature fusion on the set of entities involved in the shortest paths, constructing a fuzzy entity representation denoted as $\hat{\mathbf{e}}_2$. Then, perform denoising according to the method in Section 4.3:

$$\mathbf{o} = Transformer(\mathbf{e}^*), \tag{14}$$

$$\mathbf{e}^* = [\tilde{\mathbf{P}}^*; \hat{\mathbf{e}}_2] + \mathbf{E}(s_q) + \mathbf{R}(r_q) + \mathbf{T}(t_q). \tag{15}$$

4.5 Train and Inference

We will perform a dot product operation between the predicted entities $\mathbf{o}$ and the embedding matrix $\mathbf{E}$ to obtain the distance between the vectors. A shorter distance indicates a higher predicted probability for that entity. The calculation process is as follows:

$$y = Softmax\left(\mathbf{o} \cdot (\mathbf{E})^T\right), \tag{16}$$

$$\mathcal{L}_{recon} = -\sum_{i \in \{1,2,\ldots,|\mathcal{E}|\}} g_i log(y_i), \tag{17}$$

where "$\cdot$" denotes the inner product operation, $(\cdot)^T$ denotes the matrix transpose operation, g_i denotes the unique heat coding of the i-th real object entity, and y_i is the predicted probability of the entity.

Additionally, we employ a regularization method based on uncertainty perception, as detailed in Cai

	N_{train}	N_{valid}	N_{test}	N_{ent}	N_{rel}	N_{time}	$Interval$
ICEWS14	74845	8514	7371	7128	230	365	24 hours
ICEWS05-15	368868	46302	46159	10488	251	4017	24 hours
YAGO	161540	19523	20026	10623	10	188	1 year

Table 1: Statistics of the datasets.

et al. (2024), which dynamically adjusts the regularization strength according to the prediction confidence and applies stronger constraints to high-uncertainty predictions.

$$Score\left(y, \mathbf{F}_{01}\right) = f\left(\sigma\left(f\left(y \otimes \mathbf{F}_{01}\right)\right)\right), \tag{18}$$

$$\begin{aligned}\mathcal{L}_{uncertainty} =& \mathbb{E}_{u \backsim P_{seen}} \\ & \left[-log \frac{exp^{-Score(u, \mathbf{F}_{01})/\tau}}{1 + exp^{-Score(u, \mathbf{F}_{01})/\tau}}\right] \\ & + \mathbb{E}_{v \backsim P_{unseen}} \\ & \left[-log \frac{1}{1 + exp^{-Score(v, \mathbf{F}_{01})/\tau}}\right],\end{aligned} \tag{19}$$

where $Score(y, \mathbf{F}_{01}) \in \mathbb{R}^{1\times 2}$ denotes the confidence score. $f(\cdot)$ represents the fully connected layer, and σ denotes the ReLU activation function. The binary vector $\mathbf{F}_{01} \in \mathbb{R}^{1\times d}$ denotes the occurrence of the fact before the timestamp t, where 0 means it has not occurred and 1 means it has occurred. τ is the temperature coefficient.

The overall training loss is:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{uncertainty}. \tag{20}$$

5 Experiments

5.1 Experimental Setup

Datasets We evaluate DLTKG on three widely used datasets: ICEWS14 (García-Durán et al., 2018), ICEWS05-15 (García-Durán et al., 2018), and YAGO (Mahdisoltani et al., 2013). ICEWS14 and ICEWS05-15 are derived from the Integrated Crisis Early Warning System (ICEWS) (Boschee et al., 2015), which records political events that occurred in 2014 and from 2005 to 2015, respectively. YAGO (Mahdisoltani et al., 2013) is a knowledge base that includes temporal information. The statistics of these datasets are shown in Table 1.

Evaluation Metrics During testing, we perform experiments under time-aware filtering settings (Dong et al., 2023; Zhang et al., 2023) to filter out other correct entities. To evaluate model performance, we adopt standard evaluation metrics from the field, including Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10, where higher metric values indicate better performance.

Baseline Methods We compare the performance of our model with five categories of state-of-the-art models. The **embedding-based models** include CyGNet (Zhu et al., 2021), HIP Network (He et al., 2021). The **graph neural network-based models** include RE-NET (Jin et al., 2020), xERTE (Han et al., 2020), REGCN (Li et al., 2021), ODE (Han et al., 2021), HiSMatch (Li et al., 2022b), RETIA (Liu et al., 2023), SRPL (Li et al., 2024). The **rule-based models** include TLogic (Liu et al., 2022), TR-Rules (Li et al., 2023), TempValid (Huang et al., 2024), ONSEP (Yu et al., 2024). The **language model-based methods** include ChapTER (Peng et al., 2024), STORE (Zhang et al., 2024), CoH (Xia et al., 2024), LLM-DA (Wang et al., 2024), GenTKG (Liao et al., 2024). The **diffusion-based model** DiffuTKG (Cai et al., 2024), which is the first model to introduce diffusion into TKG reasoning tasks.

We provide the implementation details of DLTKG in Appendix A and introduce each baseline model in detail in Appendix D.

5.2 Main Results

The comparative performance of various baseline models on the link prediction task is detailed in Table 2. DLTKG consistently outperforms the main *diffusion-based* baseline DiffuTKG across all datasets, with improvements of 19.00%, 16.60%, 21.52% and 23.76% in MRR, Hits@1, Hits@3, and Hits@10, respectively, on ICEWS14. This demonstrates the effectiveness of denoising training guided by the HEPs of relationships in TKGR.

Compared to *embedding-based* and *language model-based* methods, our model DLTKG outperforms nearly all baselines, with an average MRR improvement of 20.38% on ICEWS14. Furthermore, it surpasses the LLM-based models STORE, CoH, and LLM-DA across all metrics, indicating that DLTKG is more effective at capturing abstract semantic relationships in TKGs.

When compared to *rule-based* models and *graph neural network-based* models, DLTKG significantly outperforms the second-best models on ICEWS14 and ICEWS05-15, with MRR improvement of 8.20% and 0.94%, respectively. Notably, DLTKG shows a more substantial performance gain on the ICEWS14 than on the ICEWS05-15

Method	ICEWS14				ICEWS05-15				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
CyGNet† (Zhu et al., 2021)	39.86	30.11	44.02	58.10	40.42	29.44	46.06	61.60	68.98	58.97	76.80	86.98
HIP Network (He et al., 2021)	50.57	45.73	54.28	61.65	-	-	-	-	67.55	66.32	68.49	70.37
RE-NET† (Jin et al., 2020)	38.48	28.52	42.85	58.10	44.56	34.16	50.06	64.51	66.93	58.59	71.48	86.84
xERTE (Han et al., 2020)	40.79	32.70	45.67	57.30	46.62	37.84	52.31	63.92	53.62	48.53	58.42	60.53
REGCN† (Li et al., 2021)	42.48	31.90	47.73	62.85	48.10	37.48	53.92	68.56	**82.30**	78.83	84.27	88.58
ODE (Han et al., 2021)	26.25	17.30	29.07	44.18	42.86	32.72	48.14	62.34	62.50	58.77	64.73	68.63
HiSMatch (Li et al., 2022b)	46.42	35.91	51.63	66.84	52.85	42.01	59.05	73.28	-	-	-	-
RETIA (Liu et al., 2023)	45.29	34.60	50.88	66.06	52.17	40.21	59.42	73.98	-	-	-	-
SRPL (Li et al., 2024)	56.19	50.12	59.02	67.43	-	-	-	-	-	-	-	-
TLogic (Liu et al., 2022)	42.53	33.20	47.61	60.29	46.94	36.16	53.24	67.21	78.76	74.31	83.38	83.72
TR-Rules* (Li et al., 2023)	43.32	33.96	48.55	61.17	45.91	36.22	51.60	65.57	-	-	-	-
TempValid (Huang et al., 2024)	45.78	35.50	51.34	65.06	50.31	39.46	56.71	70.55	79.72	74.64	84.78	85.73
ONSEP (Yu et al., 2024)	-	33.20	46.50	57.70	-	39.00	55.10	66.80	-	-	-	-
ChapTER (Peng et al., 2024)	33.80	-	38.00	52.70	33.10	-	36.90	52.50	-	-	-	-
STORE (Zhang et al., 2024)	48.77	36.53	55.58	71.91	49.74	38.52	55.91	71.14	64.65	51.94	71.50	83.10
CoH (Xia et al., 2024)	43.94	33.07	49.64	64.90	49.71	38.01	56.40	71.25	-	-	-	-
LLM-DA (Wang et al., 2024)	47.10	36.90	52.60	67.10	52.10	41.60	58.60	72.80	-	-	-	-
GenTKG (Liao et al., 2024)	-	36.85	47.95	53.50	-	-	-	-	-	**79.15**	83.00	84.25
DiffuTKG* (Cai et al., 2024)	45.39	35.88	50.12	63.56	50.74	39.73	55.51	73.17	80.98	74.25	85.63	89.41
DLTKG(Ours)	**64.39**	**52.48**	**71.64**	**87.32**	**53.79**	**42.05**	**59.43**	**76.35**	81.46	75.09	**86.57**	**90.74**
APG	8.20	2.36	12.62	15.41	0.94	0.04	0.01	2.37	-0.84	-4.06	0.94	1.33
RPG(%)	14.59	4.71	21.38	21.43	1.78	0.10	0.02	3.20	-1.02	-5.13	1.10	1.49

Table 2: Performance (%) comparison on temporal link prediction on three event-based TKG datasets (ICEWS14, ICEWS05-15, and YAGO). *APG* and *RPG* represent the absolute and relative performance gains of our model over the best-performing baselines, calculated as $APG = R_{ours} - R_{baseline}$ and $RPG = (R_{ours} - R_{baseline}) / R_{baseline}$, where R_{ours} and $R_{baseline}$ denote the results of our model and best-performing baselines, respectively. Best results are in bold, and the second best are underlined. The results marked with † are from Huang et al. (2024), marked with $*$ are from our reimplementation with default settings, and other results are retrieved from the original papers.

	$\alpha(e)$	$\alpha(r)$	$\lvert\mathcal{P}\rvert$
ICEWS14	48.5	90.22	13189
ICEWS05-15	57.96	92.03	28005
YAGO	35.09	55.00	1172

Table 3: Statistical data on three datasets. $\alpha(e)$ represents the percentage (%) of the target entity appearing historically in the test set. $\alpha(r)$ represents the proportion (%) of relationships with HEPs. $|\mathcal{P}|$ represents the total number of HEPs for each dataset.

Method	ICEWS14				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
DLTKG w/o fe	64.02	52.10	71.45	86.64	80.56	73.98	85.23	89.45
DLTKG w/o rh	63.97	52.12	71.31	86.50	80.93	74.82	85.02	90.23
DLTKG w/o ft	63.28	51.33	70.35	86.45	80.31	74.18	83.45	89.13
DLTKG	**64.39**	**52.48**	**71.64**	**87.32**	**81.46**	**75.09**	**86.57**	**90.74**

Table 4: Results (%) of the ablation studies on ICEWS14 and YAGO.

and YAGO. We attribute the primary reason to the differing proportions of entities, relations, and HEPs across the datasets, as shown in Table 3. The occurrence rate of target entities in the ICEWS14 is relatively low, leading to a greater distinction between fuzzy and target entities. By fusing historical information to obtain fuzzy entity representations, DLTKG effectively reconstructs the target entities. In ICEWS05-15, the abundance of high-quality facts at each timestamp complicates the ability of DLTKG to distinguish target entities from fuzzy ones. In the YAGO, only 55% of relationships have HEPs, resulting in fewer effective information for DLTKG. Experiments demonstrate the effectiveness of denoising fuzzy entities using HEPs and further enhance model accuracy through fine-tuning.

5.3 Ablation Study

To validate the effectiveness of various modules in DLTKG, we conduct ablation experiments on ICEWS14 and YAGO. (1) "$w/o\ fe$" indicates that we do not use fuzzy entities, but instead directly add random noise to the target entities. (2) "$w/o\ rh$" means that we remove the HEPs of relationships and do not use guiding information. (3) "$w/o\ ft$" indicates that we do not use the fine-tuning

Noise Sources	MRR	H@1	H@3	H@10
Random noise	64.02	52.10	71.45	86.64
Random entities	63.82	51.80	71.29	86.88
Relevant entities (Ours)	**64.15**	**52.20**	**71.50**	**87.09**

Table 5: Comparison results of denoising different noise sources on ICEWS14.

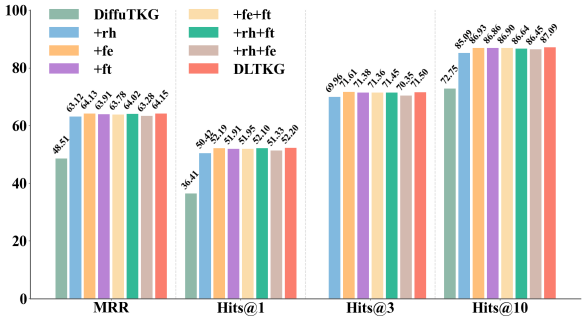


Figure 3: Generalization results on ICEWS14: DiffuTKG (Cai et al., 2024) is enhanced by our different strategy combinations.

strategy, but directly use the results from the initial training.

As shown in Table 4, we have the following observations: (1) Removing the entity fusion and HEPs leads to a decrease in all metrics. This indicates that treating related entities as noise sources and using evolutionary history for denoising can effectively capture the underlying patterns between events. (2) After removing the fine-tuning module, the MRR decreases by 1.11% and 1.15% for ICEWS14 and YAGO, confirming the effectiveness of the fine-tuning structure.

5.4 Analysis of Different Noise Sources

We believe future event prediction is strongly correlated with historical data. The process of denoising noise sources (entities) can be viewed as a search process for historical data. Table 5 compares results with random noise, fuzzy entities with random fusion, and fuzzy entities with historically relevant fusion. The model performs worse with random entities as noise sources than with random noise, as random entities interfere with the model's judgment. Additionally, correct entity information yields significantly better results than both incorrect and absent entity information.

5.5 Generalization Analysis

To validate the generalization of our proposed strategies in DLTKG, we conduct a comparative analysis with DiffuTKG (Cai et al., 2024) on the

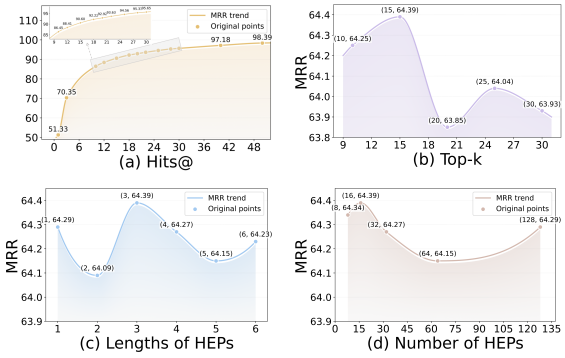


Figure 4: Performance of different parameters.

ICEWS14 dataset. Figure 3 shows that incorporating the entity fusion (fe), historical evolution (rh), and fine-tuning (ft) modules results in significant improvements in MRR, Hits@1, and Hits@10, with MRR increasing by 15.62%, 14.61%, and 15.40%, respectively. These results demonstrate the effectiveness of each strategy: entity fusion captures the relationships between entities, the historical evolution module models the trends of event development, and fine-tuning enhances reasoning performance.

5.6 Parameter Analysis and Case Study

We run our model with different important hyperparameters (*i.e.*, k, ℓ, and n) to explore the weight impacts. From the Figure 4, it can be observed that the influence of these parameters on model performance is relatively minimal. The detailed results are reported in Appendix B.

To facilitate the understanding of the modeling mechanism of DLTKG, we provide several case studies in Appendix C.

6 Conclusion

In this paper, we present DLTKG, a temporal knowledge graph reasoning model based on a denoising diffusion process for future fact prediction. We introduce an entity fusion strategy that aggregates past entities into fuzzy representations, reconstructed by a conditional denoising decoder. The fine-tuning phase further refines the model by incorporating the shortest paths between query head entity and candidate entities as additional conditions. Empirical results on benchmark datasets demonstrate that DLTKG outperforms existing methods, offering superior accuracy and generalization in temporal reasoning tasks.

Limitations

We demonstrate the effectiveness and generalizability of our DLTKG method through evaluations on multiple benchmarks. Nevertheless, DLTKG may still exhibit several limitations. On one hand, it uses a simple linear fusion method for generating fuzzy entities, and future work could explore more advanced strategies like weighted fusion. On the other hand, the one-step noise addition approach may be improved by investigating stepwise techniques to better capture entity uncertainties.

References

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data.

Yuxiang Cai, Qiao Liu, Yanglei Gan, Changlin Li, Xueyi Liu, Run Lin, Da Luo, and JiayeYang JiayeYang. 2024. Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5766–5778.

Qian Chen and Ling Chen. 2024. Decrl: A deep evolutionary clustering jointed temporal knowledge graph representation learning approach. *Advances in Neural Information Processing Systems*, 37:55204–55227.

Nazim Choudhury, Fahim Faisal, and Matloob Khushi. 2020. Mining temporal evolution of knowledge graphs and genealogical features for literature-based discovery prediction. *Journal of Informetrics*, 14(3):101057.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1585–1595.

Hao Dong, Zhiyuan Ning, Pengyang Wang, Ziyue Qiao, Pengfei Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Adaptive path-memory network for temporal knowledge graph reasoning. In *IJCAI*.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv e-prints*, pages arXiv–1809.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations (ICLR 2023)(01/05/2023-05/05/2023, Kigali, Rwanda)*.

Simon Gottschalk and Demidova Elena. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *European semantic web conference*, pages 272–287. Springer.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Yi Zheng, Baoxing Huai, and Nicholas Jing Yuan. 2022. Delving deep into regularity: A simple but effective method for chinese named entity recognition. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1863–1873.

Avital Hahamy, Haim Dubossarsky, and Timothy EJ Behrens. 2023. The human brain reactivates context-specific past information at event boundaries of naturalistic experiences. *Nature neuroscience*, 26(6):1080–1089.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International conference on learning representations*.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8352–8364.

Yongquan He, Peng Zhang, Luchen Liu, Qi Liang, Wenyuan Zhang, and Chuang Zhang. 2021. Hip network: Historical information passing network for extrapolation reasoning on temporal knowledge graph. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1915–1921.

Rikui Huang, Wei Wei, Xiaoye Qu, Shengzhe Zhang, Dangyang Chen, and Yu Cheng. 2024. Confidence is not timeless: Modeling temporal validity for rule-based temporal knowledge graph forecasting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10783–10794.

Jong Ho Jhee, Alberto Megina, Pacôme Constant Dit Beaufils, Matilde Karakachoff, Richard Redon, Alban Gaignard, and Adrien Coulet. 2025. Predicting clinical outcomes from patient care pathways represented with temporal knowledge graphs. *arXiv preprint arXiv:2502.21138*.

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

Luca D Kolibius, Sheena A Josselyn, and Simon Hanslmayr. 2025. On the origin of memory neurons in the human hippocampus. *Trends in Cognitive Sciences*.

Ningyuan Li, E Haihong, Shi Li, Mingzhi Sun, Tianyu Yao, Meina Song, Yong Wang, and Haoran Luo. 2023. Tr-rules: Rule-based model for link forecasting on temporal knowledge graph considering temporal redundancy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7885–7894.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022a. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Xuefei Li, Huiwei Zhou, Weihong Yao, Wenchu Li, Yingyu Lin, and Lei Du. 2024. Sequential and repetitive pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14744–14754.

Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022b. Hismatch: Historical structure matching based temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7328–7338.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 408–417.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024. Gentkg: Generative forecasting on temporal knowledge graph with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4303–4317.

Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. 2023. Retia: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pages 1761–1774. IEEE.

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

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Miao Peng, Ben Liu, Wenjie Xu, Zihao Jiang, Jiahui Zhu, and Min Peng. 2024. Deja vu: Contrastive historical modeling with prefix-tuning for temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1178–1191.

Machel Reid, Vincent Josua Hellendoorn, and Graham Neubig. 2023. Diffuser: Diffusion via edit-based reconstruction. In *The Eleventh International Conference on Learning Representations*.

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

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. *arXiv preprint arXiv:1705.05742*.

Garry W Trompf. 1979. *The idea of historical recurrence in Western thought: from antiquity to the Reformation*, volume 1. Univ of California Press.

Jiapu Wang, Sun Kai, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410.

Jiapu Wang, Boyue Wang, Meikang Qiu, Shirui Pan, Bo Xiong, Heng Liu, Linhao Luo, Tengfei Liu, Yongli Hu, Baocai Yin, and 1 others. 2023. A survey on temporal knowledge graph completion: Taxonomy, progress, and prospects. *arXiv e-prints*, pages arXiv–2308.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16144–16159.

Siheng Xiong, Yuan Yang, Faramarz Fekri, and James Clayton Kerce. 2024. Tilp: Differentiable learning of temporal logical rules on knowledge graphs. *arXiv e-prints*, pages arXiv–2402.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593.

Xuanqing Yu, Wangtao Sun, Jingwei Li, Kang Liu, Chengbao Liu, and Jie Tan. 2024. Onsep: A novel online neural-symbolic framework for event prediction based on large language model. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6335–6350.

Mengqi Zhang, Yuwei Xia, Qiang Liu, Shu Wu, and Liang Wang. 2023. Learning latent relations for temporal knowledge graph reasoning. In *ACL*.

Tianli Zhang, Tongya Zheng, Zhenbang Xiao, Zulong Chen, Liangyue Li, Zunlei Feng, Dongxiang Zhang, and Mingli Song. 2024. Language models-enhanced semantic topology representation learning for temporal knowledge graph extrapolation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3227–3236.

Liang Zhao. 2021. Event prediction in the big data era: A systematic survey. *ACM Computing Surveys (CSUR)*, 54(5):1–37.

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

A Implementation Details

We compute the mean reciprocal rank (MRR) and hits@i for $i \in \{1, 3, 10\}$. For a rank $x \in \mathbb{N}$, the reciprocal rank is defined as $\frac{1}{x}$, and the MRR is the average of all reciprocal ranks of the correct query answers across all queries. The metric Hits@i indicates the proportion of queries for which the correct entity appears under the top i candidates.

We use the AdamW optimizer with a learning rate set to 0.001. The number of training epochs is set to 100, and if there is no improvement in the MRR on the validation set for 20 consecutive epochs, training will be terminated early. Additionally, the length of HEP is set to 3, the number of HEPs is set to 16, the number of candidate entities k in the fine-tuning module is set to 15, the total number of shortest paths is set to 128, and the number of random walk steps is set to 200. The hidden layer dimension size d for entities, relationships, and timestamps is fixed at 200 across all datasets.

B Parameter Sensitivity Analysis

Figure 4 in Section 5.6 and Figure 5 show the impact of different parameters on model performance. Overall, the influence of these parameters on model performance is relatively minimal.

For different lengths of HEPs, both shorter and longer values tend to degrade performance. This is attributed to the limited number of instances for shorter or longer HEPs in the ICEWS14. Specifically, when $\ell = 1$, the performance surpasses that of $\ell = 2$, as there are 2.5 times more HEPs for $\ell = 1$. Moreover, the number of historical iterations, denoted as n, also plays a crucial role in model performance. Excessive iterations introduce redundancy, which negatively impacts accuracy.

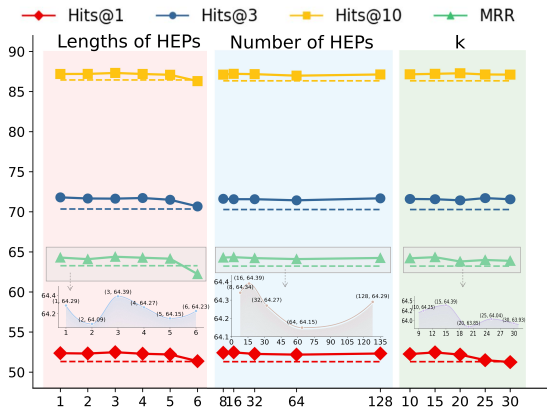


Figure 5: The impact of different parameters on all evaluation metrics. The pink blocks, blue blocks, and green blocks represent the changes in the indicators of HEPs length, the number of HEPs, and the number of candidate entities, respectively.

As such, we set $n = 16$ for optimal performance. Figure 4(a) presents the Hits@ metric results for ICEWS14 under optimal conditions, where the performance stabilizes within the range of [10, 30]. Consequently, we investigate the influence of different k values on model performance within this interval. As observed, when k exceeds 15, performance begins to decline, likely due to the excessive number of candidate entities, which may hinder the ability of model to make accurate predictions. Therefore, $k = 15$ is determined to be the optimal choice.

C Case Study

We present two queries in Table 6. For Query 1, as described in Section 4.1, we initially retrieve n HEPs related to *Use conventional military force*, all of which involve *Government* (*Nigeria*). The top three predicted entities all appeared in the historical data. We then fine-tune the model, leading to the final prediction of *Boko Haram*.

For Query 2, we utilize $r = 3$ HEPs. The combination of HEP guidance and the associations with historically relevant entities enables the model to effectively predict the target entity.

By incorporating the HEP conditional guidance mechanism, model effectively accounts for the influence of these conditions when denoising. The fuzzy entities, which contain historical information relevant to the target entity, are refined with the help of HEPs, thereby enhancing the interpretability of model.

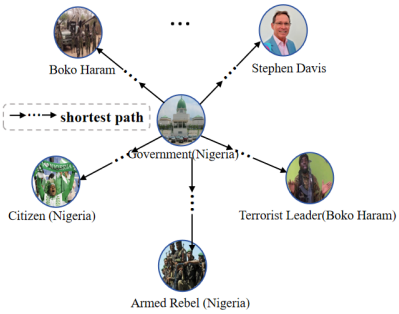
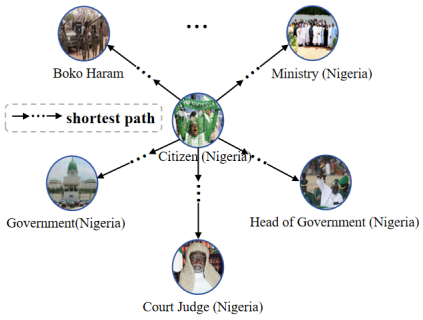
Query1	$(Government\,(Nigeria), Use\ conventional\ military\ force, ?, 2014-12-02)$	**Answer**	Boko Haram
HEP	p_1: $Boko\ Haram \xrightarrow[2014-05-19]{Threaten} Government\,(Nigeria) \xleftarrow[2014-05-22]{Appeal\ to\ engage\ in\ or\ accept\ mediation} Education\,(Nigeria)$ p_2: $Boko\ Haram \xrightarrow[2014-05-21]{Make\ statement} Government\,(Nigeria) \xleftarrow[2014-06-02]{Express\ intent\ to\ cooperate} Citizen\,(Nigeria)$ $\vdots$ p_n: $Government\,(Nigeria) \xrightarrow[2014-08-18]{Use\ conventional\ military\ force} Boko\ Haram \xleftarrow[2014-09-04]{Engage\ in\ negotiation} Stephen\ Davis$		
Entity fusion	$[Boko\ Haram; Stephen\ Davis; Citizen\,(Nigeria); Education\,(Nigeria); Government\,(Nigeria); \ldots; Muslim_(Nigeria)]$		
Initial prediction	1.$Stephen\ Davis$; 2.$Boko\ Haram$; 3.$Citizen\,(Nigeria)$; 4.$Terrorist\ Leader\,(Boko\ Haram)$; 5.$Armed\ Rebel\,(Nigeria)$		
Shortest paths			
Predicted results	1.$Boko\ Haram$; 2.$Stephen\ Davis$; 3.$Citizen\,(Nigeria)$; 4.$Terrorist\ Leader\,(Boko\ Haram)$; 5.$Militant\,(Nigeria)$		
Query2	$(Citizen\,(Nigeria), Make\ an\ appeal\ or\ request, ?, 2014-12-02)$	**Answer**	Government (Nigeria)
HEP	p_1: $Kashim\ Shettima \xrightarrow[2014-05-23]{Make\ statement} Head\ of\ Government\,(Nigeria) \xrightarrow[2014-09-30]{Praise\ or\ endorse} Government\,(Nigeria)$ $\xleftarrow[2014-09-30]{Make\ an\ appeal\ or\ request} Head\ of\ Government\,(Nigeria)$ p_2: $Citizen\,(Nigeria) \xleftarrow[2014-08-04]{Investigate} Court\ Judge\,(Nigeria) \xrightarrow[2014-08-11]{Praise\ or\ endorse} Ministry\,(Nigeria)$ $\xrightarrow[2014-08-11]{Make\ statement} Court\ Judge\,(Nigeria)$ $\vdots$ p_n: $Citizen\,(Nigeria) \xrightarrow[2014-09-22]{Make\ an\ appeal\ or\ request} Government\,(Nigeria) \xrightarrow[2014-09-23]{Engage\ in\ negotiation} Boko\ Haram$ $\xleftarrow[2014-09-24]{Make\ statement} Government\,(Nigeria)$		
Entity fusion	$[Citizen\,(Nigeria); Boko\ Haram; Ministry\,(Nigeria); Head\ of\ Government\,(Nigeria); \ldots; Court\ Judge\,(Nigeria)]$		
Initial prediction	1.$Boko\ Haram$; 2.$Ministry\,(Nigeria)$; 3.$Head\ of\ Government\,(Nigeria)$; 4.$Government\,(Nigeria)$; 5.$Court\ Judge\,(Nigeria)$;		
Shortest paths			
Predicted results	1.$Government\,(Nigeria)$; 2.$Boko\ Haram$; 3.$Ministry\,(Nigeria)$; 4.$Head\ of\ Government\,(Nigeria)$; 5.$Court\ Judge\,(Nigeria)$		

Table 6: Two case studies. We report the prediction results of DLTKG. The green font indicates the correct answers, while the orange font represents entities that appear in HEP.

D Baselines

The comparison of TKG reasoning models with our work is presented as follows:

CyGNet (Zhu et al., 2021) introduces time-aware replication generation, combining new facts with repeated pattern recognition to improve prediction accuracy.

HIP Network (He et al., 2021) integrates temporal, structure and repetitive patterns, dynamically updates relationships, and optimizes multi-dimensional score prediction.

RE-NET (Jin et al., 2020) combines event encoding and neighbor aggregation using an autore-

gressive architecture to sequentially reason about future facts.

xERTE (Han et al., 2020) is based on query subgraphs and integrates temporal attention and reverse update, taking into account both accuracy and interpretability.

REGCN (Li et al., 2021) combines relation-aware convolution with gated recurrence to dynamically model entity relations and fuse static attributes.

ODE (Han et al., 2021) extends multi-relation graph convolution to continuous time, integrates temporal structures, and models the dynamic formation and resolution of relationships.

HiSMatch (Li et al., 2022b) regards temporal knowledge graph reasoning as structural matching, integrating dual encoders with entity prior information.

RETIA (Liu et al., 2023) addresses the issues of relationship modeling and overfitting through dual hyper-relation subgraphs and dual interaction modules.

SRPL (Li et al., 2024) combines dependency-aware sequences with time intervals to guide repetitive pattern learning and to capture both temporal proximity dependencies and irregular intervals.

TLogic (Liu et al., 2022) extracts logical rules based on temporal random walks, taking into account both temporal consistency and inductiveness, and supports rule migration and cross-set prediction.

TR-Rules (Li et al., 2023) improves confidence accuracy and introduces non-circular rules to enhance rule diversity , as well as model interpretability and predictive capability.

TempValid (Huang et al., 2024) dynamically models rule confidence based on time functions and combines adversarial and time-aware negative sampling to improve learning efficiency.

ONSEP (Yu et al., 2024) integrates dynamic causal rule mining and dual history enhanced generation.

ChapTER (Peng et al., 2024) integrates contrastive learning and prefix tuning, and uses virtual time prefixes to achieve low-parameter fine-tuning and multi-scenario adaptation.

STORE (Zhang et al., 2024) combines time-aware semantic sampling and virtual tokens, using multi-head attention to jointly optimize both semantic and topological representations of temporal knowledge graphs.

CoH (Xia et al., 2024) leverages higher-order historical information to enhance temporal reasoning capabilities in large language models.

LLM-DA (Wang et al., 2024) utilizes large language models to extract temporal rules, enabling dynamic adaptation to ever-changing knowledge.

GenTKG (Liao et al., 2024) integrates temporal logic rule retrieval and few-shot instruction fine-tuning, connecting temporal knowledge graphs with large language models.

DiffuTKG (Cai et al., 2024) frames temporal knowledge graph reasoning as a denoising process for future fact sequences. It restores target facts using conditional sequence encoding and a Transformer-based denoiser, while applying uncertainty regularization to reduce prediction bias and handle rare or unseen facts..