Computing, Performance and Communication Systems (2025)
Clausius Scientific Press, Canada
DOI: 10.23977/cpcs.2025.090107
ISSN 2371-8889 Vol. 9 Num. 1

Research on Temporal Knowledge Graph Reasoning Based on Trend and Dynamic Change Perception Enhancement

Hanzhang Liu

Nanjing University of Finance and Economics, Nanjing, China

Keywords: Temporal Knowledge Graphs; Trend Modeling; Dynamic Entity Representation; Contrastive Learning; Temporal Reasoning

Abstract: Temporal knowledge graphs (TKGs), as an effective means of modeling dynamic relationships among entities, have shown great potential in tasks such as event prediction in recent years. However, most existing reasoning approaches tend to overlook the diversity and complexity of historical information. In reality, reasoning at the current timestamp is often constrained by the limited scope of historical data and the influence of unobserved latent factors. There are three major limitations in current TKG reasoning methods: (1) the inability to effectively highlight the importance of historical snapshots relevant to the current query when integrating both local and global history; (2) the neglect of temporal trends inherent in the evolution of facts, resulting in insufficient modeling of evolutionary patterns; and (3) the lack of effective mechanisms for capturing abrupt, short-term changes in the temporal dimension of facts. To address these challenges, we propose a novel Trend- and Variation-aware Contrastive Learning Network. Specifically, we introduce a local-global contrastive learning mechanism to guide the model's focus toward historical information that is more relevant to the current query. We further design a trend-aware attention module to capture the regularities and temporal evolution patterns in long-term entity representations. Additionally, a time-aware convolution module is developed to perceive abrupt dynamic changes in entity states across consecutive time slices, enabling the model to better integrate this information with current context representations. Experimental results on four benchmark TKG datasets demonstrate that our model outperforms several state-of-the-art baseline models in prediction tasks, showcasing its superior generalization ability and effectiveness in modeling complex temporal evolution patterns.

1. Introduction

In numerous real-world applications such as event prediction, financial risk control, and social behavior modeling, understanding the dynamic evolution of entity relationships over time is of significant importance. Traditional Knowledge Graphs (KGs), as structured tools for representing entities and their relationships, demonstrate strong performance in static environments. However, real-world entity interactions often exhibit significant temporal evolution, which has gradually shifted research focus towards the construction and reasoning of Temporal Knowledge Graphs (TKGs).

TKGs model facts as quadruples (subject, relation, object, time) by incorporating a temporal dimension, thereby more closely capturing the dynamic evolution of the real world. TKG reasoning tasks are primarily divided into two categories: historical completion (interpolation) and future prediction(extrapolation).Comparatively, extrapolation tasks are more aligned with practical needs but face greater uncertainty and modeling challenges. To enhance models' ability to capture temporal evolution, numerous methods have been proposed in recent years. Early TKG reasoning methods, such as Know-Evolve [1] and DyRep [2], introduced event-driven dynamic representation update mechanisms capable of capturing the temporal dependencies of interactions between entities. These methods effectively simulate representation changes triggered by events by modeling the evolving trajectories of entity and relation embeddings as temporal point processes. However, they often focus on modeling event time distributions while lacking the ability to capture complex spatio-temporal dependencies within the graph structure.

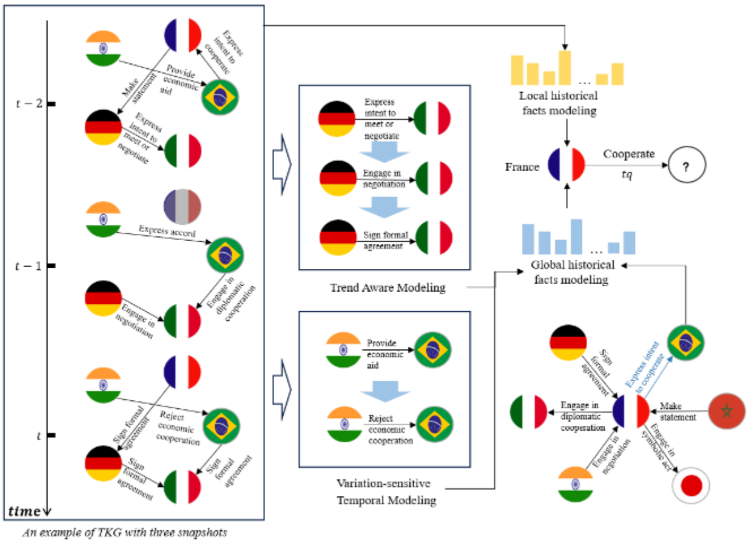

Fig. 1 An illustrative example that highlights the importance of capturing historical information related to queries. And in the process of historical information evolution, the significance of regular trend information and emergency event information. The blue arrows indicate the most important facts related to the query

To better integrate graph structure and temporal information, subsequent studies like RE-NET [3]and T-GCN [4] combined Graph Neural Networks (GNNs) with temporal modeling modules. These approaches encode historical snapshots at each timestep, enhancing the model's contextual understanding of entity representations. These methods significantly improve the modeling of long-term temporal dependencies and have achieved performance breakthroughs on multiple TKG benchmark datasets.

Furthermore, to address uncertainty in extrapolative prediction, some works have introduced contrastive learning and attention mechanisms, such as HiSMatch [5] and TiRGN [6]. By emphasizing historical segments relevant to the query and suppressing irrelevant information, these methods enhance the model's discriminative power and generalization ability. These advances progressively address key challenges in TKG reasoning, including temporal dependency modeling, graph structure integration, and noise suppression, laying the groundwork for building more robust and efficient TKG reasoning models. Nevertheless, existing research still possesses several limitations and areas for improvement:

1.1 Weak Extraction Capability for Query-Relevant Historical Information

Most Temporal Knowledge Graph (TKG) extrapolation methods assume chronologically closer KG snapshots are more relevant for prediction. HiSMatch [5] constructs historical structure graphs but rigidly handles chronological order, failing to dynamically capture shifting importance of distant snapshots. Crucially, query entities (e.g., France in query $(France, Cooperate, ?, t_q)$, may be absent in recent snapshots t_{q-1} rendering them minimally useful (Fig 1). Conversely, earlier snapshots containing related entities (e.g., China) may prove more predictive. Existing methods lack mechanisms to identify these fundamentally relevant historical patterns, impairing future fact prediction. Thus, filtering irrelevant KG snapshots based on query context is essential for enhancing TKG reasoning performance.

1.2 Lack of capturing regular trend information within historical information

Current models lack deep semantic mining in entity neighborhood networks, particularly in capturing historical trends for future prediction. LGTQ [7] captures multi-granularity semantics but is limited by fixed time windows and cannot adaptively filter key trend paths.

These approaches neglect explicit modeling of relationship evolution patterns, overlooking long-term semantic evolution paths. For instance, Figure 1 shows Germany-Italy relationships progressively evolving: from Express intent to meet → Engage in negotiation → Sign formal agreement.

Such sequences reveal temporal dependency structures with high predictive value. Modeling these historical trends enhances long-term dependency understanding and provides structured semantic guidance for future event reasoning.

1.3 Lack of effective modeling of abrupt short-term changes of facts along the time dimension

Real-world events often involve pivotal occurrences that are neither trend continuations nor regular patterns. They arise abruptly from latent/sudden factors, disrupting established paths and impacting fact distribution over time. Existing models struggle to capture such abrupt changes.

GHNN [8] models event self-excitation via Hawkes processes but lacks inhibitory/interrupted state modeling (e.g., India's abrupt support withdrawal from Brazil in Fig 1). CRNet [9] targets concurrent interactions but fails to recognize sudden event termination or discontinuous leaps.

A representative challenge is exemplified by events exhibiting abrupt discontinuities. For instance, consider a scenario where India supports Brazil at one timestep but terminates this support abruptly in the subsequent time step. Such rapid reversals or temporal mutations violate common stationarity and trend assumptions. Effectively capturing these discontinuous changes is crucial, as it would reduce future misjudgments and enhance model adaptability, while enabling the simulation of complex phenomena like "intention reversal" and policy shocks for Temporal Knowledge Graph prediction.

To address this, we propose the Trend and Variation-aware Contrastive Learning Network (TVCL-Net) (Fig 2). It integrates local/global historical information via contrastive learning, enhancing key information discrimination to reduce noise interference. During encoding, TVCL-Net simultaneously models long-term trends and abrupt change dynamics, significantly improving robustness to complex temporal patterns. Key contributions include:

(1) Local-global query module integrating global/local contexts via contrastive learning, enhancing identification of query-relevant history.

(2) Global trend-aware encoder mining underlying regular/progressive trends, strengthening entity temporal continuity.

(3) Local variation-aware encoder recognizing mutational events, maintaining robustness in volatile extrapolation tasks.

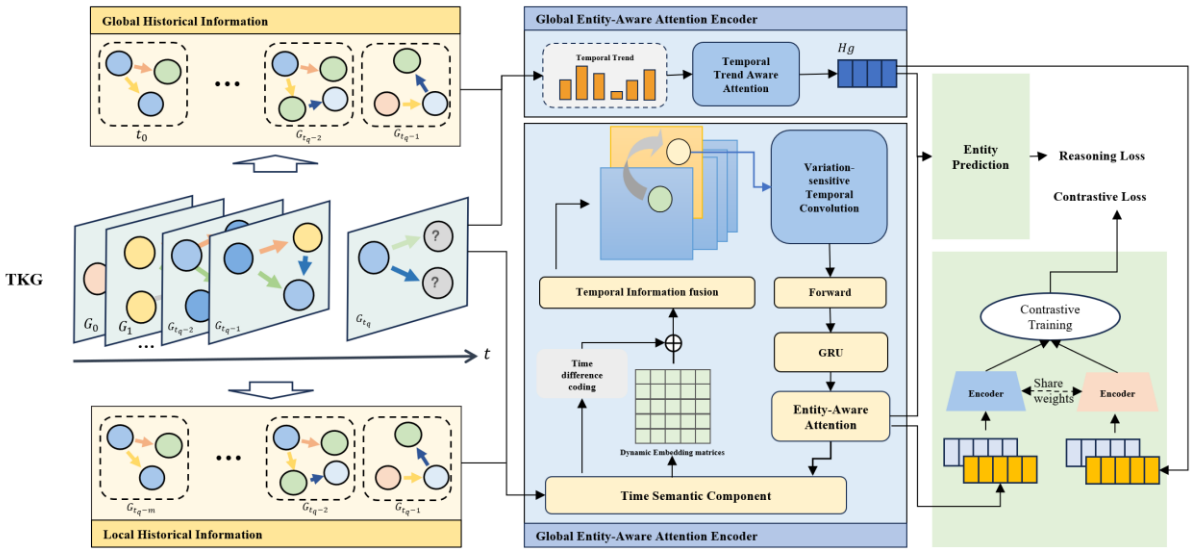

Fig. 2 Overall Model Architecture

The remainder of this paper is organized as follows: Section 2 reviews existing work related to this study; Section 3 introduces symbol definitions, the problem statement, and the detailed design of the TVCL-Net model; Section 4 presents the experimental setup and performance analysis on multiple standard datasets; Section 5 summarizes the research work and outlines future research directions.

2. Related Work

2.1 Static Graph Reasoning

Static Knowledge Graph (SKG) reasoning, a core topic in knowledge representation, encompasses four main approaches: Rule-based, Matrix decomposition-based, Large Language Model (LLM)-based, and Graph Neural Network (GNN)-based methods. Among these, GNNs excel in structure-aware modeling, CNNs in feature abstraction, while matrix factorization and rule-based methods maintain advantages in interpretability and efficiency. Recent work increasingly integrates multiple approaches for enhanced generalization.

Rule-based methods include AMIE [10], RLvLR [11], etc., which reveal co-occurrence or dependency patterns between items based on their occurrence in transactions or events. Matrix factorization methods include TransE [12], RESCAL [13], etc., which employ tensor factorization to model multi-relational data in knowledge graphs, capturing interactions between latent factors of each entity and relation for reasoning. Among GNN-based methods, R-GCN [14], etc., learn embeddings of nodes, edges, and relations by receiving adjacency information to learn representations for each node, updating nodes by receiving messages from neighbors via message passing. Unlike GCN, R-GCN can learn multi-relations, yielding better predictive power. LLM-based methods like KBERT [15] convert KGs into sequences and utilize pre-trained LLMs for reasoning, with the core task being the understanding, completion, or inference of entities and relations within the KG. Static graph reasoning methods demonstrate good performance and high efficiency when handling static data, but they do not consider temporal information, making dynamic fact evolution modeling difficult.

2.2 Dynamic Graph Reasoning

Dynamic Knowledge Graph Reasoning focuses on the temporal evolution of graph structures. Unlike static graphs, evolving nodes and edges render traditional embedding methods inadequate for capturing semantic shifts. Researchers address this via two primary frameworks: snapshot-based and event-based reasoning. Early methods include EvolveGCN [16], which updates representations through time-evolving graph convolutional kernels, and DySAT [17], modeling multi-snapshot graphs with structural and temporal attention. Event-driven models like TGN [18] encode individual interactions, balancing fine-grained temporal modeling with contextual understanding. Recent work includes APAN [19], enhancing cross-temporal node fusion via an awareness mechanism to improve perception of evolution trends, and HIPNetwork [20], propagating historical information across temporal, structural, and repetitiveness dimensions to model event evolution, intra-timestep interactions, and known patterns. HIPNetwork specifically updates relation representations using multiple scoring functions to boost accuracy.

2.3 Temporal Graph Reasoning

In recent years, driven by the rapidly growing demand for modeling time-sensitive information, Temporal Graph Reasoning (TGR) has emerged as a significant branch within Graph Neural Network (GNN) research. This task emphasizes the explicit modeling of temporal factors within graph structures to effectively capture the rich semantic information arising from the evolution of nodes, edges, and their relationships over time.

Early representative methods include TGAT [21], which introduced a temporal encoder to achieve context modeling based on event sequences by combining structural neighbors and temporal neighbors. TGN [18] further incorporated a memory module and message-passing mechanism, successfully enabling real-time online updates for event-stream graphs and significantly enhancing modeling capabilities for large-scale dynamic graphs. Subsequently, RE-GCN [22] fused graph convolution with temporal gating mechanisms, effectively modeling the evolution of entities and relations across timestamps by leveraging local historical dependencies, thereby capturing long-term dependencies between entities.

More recent advancements include CAWN [23], which proposed an attention mechanism centered on causal neighbor sampling to better model temporal paths of influence propagation. Furthermore, multimodal temporal graph models (e.g., DyRep [2]) have begun integrating user behavior, content information, and structural events, promoting the widespread application of temporal graphs in domains such as recommendation systems, financial risk control, and social computing.

However, when processing Knowledge Graph (KG) snapshots, the aforementioned methods generally lack the capability to model the importance of historical facts relevant to specific queries. This makes it difficult for them to discriminate the varying importance of different KG snapshots for predicting a given query. Additionally, these approaches remain insufficient in effectively capturing trend features and abrupt events within historical information. They struggle to extract stable, regular patterns from the event evolution process while also failing to adequately explore and learn the significance of abrupt events.

2.4 Contrastive Learning

Contrastive Learning, as a self-supervised learning paradigm, has achieved remarkable success in fields like computer vision and recommendation systems in recent years and is gradually being applied to KG-related tasks. It aims to enhance the discriminative ability of entity and relation representations by constructing positive and negative triple pairs, while learning high-quality representations by comparing similarities and differences between samples. The method guides model training through constructed "positive- negative sample pairs," making positive samples closer in the representation space while distancing negative samples. In self-supervised contrastive learning, augmented instances are typically obtained by augmenting the original samples and randomly drawing a mini-batch of size N instances. Given a positive instance pair (i, j), the original and augmented instances are used to optimize the following loss function. The contrastive loss is expressed as:

$$\mathcal{L}_{i,j} = -log \frac{\exp(X_i \cdot X_j/\tau)}{\sum_{K=1, K\neq i}^{2N} \exp(X_i \cdot X_j/\tau)}, \tag{1}$$

Where $2N$ is the sum of the number of original and augmented instances, X_i is the projected embedding of instance i, τ is a temperature parameter that helps the model learn from hard negative samples, and • denotes the dot product operation used to calculate similarity between instances under different views. However, its application in Temporal Knowledge Graph (TKG) reasoning remains relatively limited. For example, the recently proposed CENET [24] is a single-view historical contrastive learning method designed to enhance the representation ability of entities with sparse historical interactions through a contrastive strategy, thereby improving the model's ability to predict their future behavior.

3. Research

3.1 Notations and Definitions

A Temporal Knowledge Graph (TKG)(TKG)G is formally a sequence of KG snapshots, e.g $G = \{G_1, G_2, G_3, \dots, G_{|\tau|}\}$,Each KG snapshot $G_t = (E, R, F_t)$ is essentially a directed multi-relational graph at time t, A fact or event is represented as a quadruple (e_s, r, e_o, t),where the subject entity $e_s \in E$,the object entity $e_o \in E$ and the relation $r \in R$ connect them at time $t \in T$.

The TKG extrapolation task involves predicting the missing object entity or subject entity given a query $(e_q, r_q, ?, t_q)$ or $(?, r_q, e_q, t_q)$.Without loss of generality, $(e_o, r^{-1}, e_s, t)$is added to the TKG dataset Therefore, the TKG extrapolation task can be simplified to object entity prediction.

3.2 Model Architecture

We first elaborate on the overall framework of the model, as shown in Figure 3. The model mainly consists of two key modules: Global Historical Information Modeling and Local Historical Information Modeling. In the global path, we design a Global Trend-Aware Encoder to mine structural patterns with temporal trends in historical evolution.

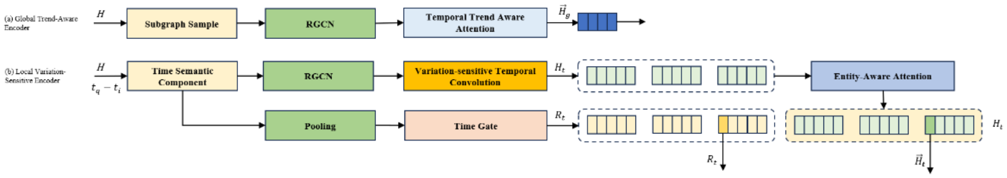

Fig. 3 Local and global coding

This module integrates information from multiple temporal snapshots combined with a trend attention mechanism to extract node representations with global evolutionary significance.

In the local path, we propose a Local Variation-Sensitive Encoder to capture fine-grained changes evolving over time within the entity's neighborhood. This module introduces Time

Difference Coding and Dynamic Embedding Matrices and achieves local updating and aggregation of node representations through structure-aware convolution and entity-aware attention mechanisms.

As shown in Figure 3, the model further introduces a Local-Global Query Contrast Module to guide better semantic alignment and fusion of global and local historical representations, thereby enhancing the model's robust modeling capability for sudden events and complex evolutions. Finally, the model combines local and global embedded representations to complete entity prediction and reasoning tasks.

3.3 Global Trend-Aware Encoder

The Global-Aware Attention Encoding module captures long-term historical patterns under-represented in local KG snapshots. Unlike CyGNet [25] and TiRGN [6] that rely solely on single-hop query entities to learn repetitive fact patterns, we incorporate candidate multi-hop facts and multidimensional dynamics (e.g., abrupt node changes) to enhance entity prediction.

For instance, while country diplomatic relations show regularity, involved nations vary temporally. Capturing such multi-hop facts improves relationship prediction. We thus construct semantically rich historical query subgraphs via multi-level sampling. Specifically, given a sequence of historical KG snapshots $G_{<t_q} = \left\{G_{1,}G_2, \dots, G_{t_q-1}\right\}$,we first sample single-hop historical facts containing the current query entity s denoted as G'_{g1}. We then further sample single-hop historical facts associated with the target object entities linked to the query entity-relation pair, denoted as G'_{g2}.Finally, we combine these two sets of historical facts to obtain a historical query subgraph closely related to the current query, i.e., $G'_g = G'_{g1} \cup G'_{g2}$.

After obtaining the historical query subgraph, we model its rich structural and semantic features to update the global entity representations. In the implementation, we combine the Relational Graph Convolutional Network (RGCN) with trend-aware convolution to perform message aggregation on the historical query subgraph. Since the historical query subgraph does not explicitly incorporate temporal information, we directly use randomly initialized entity and relation embeddings as input to the RGCN module. The entity representations are updated using
the following formula:

$$h_g^{l+1} = RGCN_{Global}\left(h_{g,e_s}^l, r^l, h_{g,e_o}^l\right). \tag{2}$$

Here, h_g^{l+1} represents the output embeddings of entities in the historical query subgraph at the l-th layer of the RGCN network. For conciseness, we denote the final layer output of RGCN as H_g^{Agg}, To enhance the final entity representations using global historical information, we further design and incorporate a global trend-aware attention module. This module captures latent long-term trend patterns in historical facts, with the formal representation:

$$\vec{h}_g^{e_q} = Trend\left(G_{<t_q}\right). \tag{3}$$

$\vec{h}_g^{e_q}$ represents the final entity embedding obtained through global entity-aware encoding, and $Trend$ denotes our proposed global trend-aware attention mechanism, which we will elaborate below.

We design a context-aware global trend attention mechanism to capture temporal trend patterns in KG history. Specifically, we propose a convolutional multi-head self-attention variant (improving convolutional self-attention [26]) with shared node-wise parameters, extracting inter-entity evolution trends efficiently. Unlike standard self-attention, we replace Query/Key linear projections with 1D convolutions. This incorporates local historical context, enhancing perception of fine-grained local variation trends.

Formally, we define tensor $X \in \mathbb{R}^{B \times N_S \times V_n \times D}$, where N_s represents the number of historical timesteps, V_n denotes the number of entity nodes, and D indicates the embedding dimension. The trend-aware multi-head self-attention is defined as:

$$TrendAttetion(Q,K,V) = \oplus\,(Trhead_1, \dots, Trhead_h)W^o$$
$$Trend_j(Q,K,V) = Attention\left(Q\Delta\gamma_j^Q, K\Delta\gamma_j^K, VW_j^Q\right). \quad (4)$$

Here, Q, K, V are derived from tensor X through normalization and causal convolution, Δ represents the convolution operation, γ_j^Q and γ_j^K are convolution kernel parameters. This approach effectively captures temporal trend characteristics exhibited during entity evolution. At the l-th encoding layer, given input $X^{(l-1)}$, the updated node features after temporal trend-aware multi-head self-attention processing are represented as:

$$\mathcal{Z}^{(l-1)} = \left(Z_{t-T_h+1}^{(l-1)}, Z_{t-T_h+2}^{(l-1)}, \dots, Z_t^{(l-1)}\right) \in \mathbb{R}^{N \times d_{model} \times T_h} \quad (5)$$

Where, N is the number of nodes, d_{model} represents the model's feature dimension, and T_h denotes the historical time window length. This design ensures the model dynamically perceives evolutionary information in both global and local contexts while considering long- and short-term historical trends, significantly enhancing the expressive power of entity representations.

3.4 Local Variation-Aware Encoder

The contribution degree of each KG snapshot at the recent m timestamps $(i.e. \left\{G_{t_q-m+1}, \dots, G_{t_q-1}\right\})$ to predicting the query $q = \left(e_{t_q}, r_{t_q}, ?, t_q\right)$is inconsistent. Therefore, it is necessary to explicitly model the relevance of KG snapshots to the current query to more precisely characterize the evolution patterns of entities and relations within adjacent time segments. To address this, we propose an Entity-Aware Attention Encoder that effectively extracts and reinforces query-relevant historical information in KG snapshots. Additionally, we introduce a Local Variation-Sensitive Encoding Module, which not only captures overall historical information but also sensitively perceives fine-grained change features evolving over time within the entity's neighborhood. Through the synergistic effect of these two mechanisms, we achieve effective modeling of KG snapshot aggregation and KG snapshot sequence evolution, significantly improving the model's performance in handling temporal KG tasks.

For each KG snapshot at a timestamp, we update entity representations by capturing spatial structural semantic information between concurrent facts. Considering that some facts in KG snapshots occur periodically (e.g., periodic meetings), we first encode temporal numerical information following [27] to obtain dynamic entity embeddings. Formally, the dynamic entity embedding is as follows:

$$\varphi(d) = \cos(d\omega_t + b_t) \quad (6)$$
$$\vec{h}_t = W_0[h_t || \varphi(d)] \quad (7)$$

Where $d = t_q - t_i$ is the time interval modeled by rescaling the learnable unit ω_t with time bias $b_t \cdot \cos(\cdot)$, $||$ is the vector concatenation operation, W_0 is a linear transformation matrix, and $\vec{h}_t \in \vec{H}_t$ is the dynamic embedding of the entity at timestamp t hen, we use R-GCN to capture structural dependencies between concurrent facts. The R-GCN is defined as:

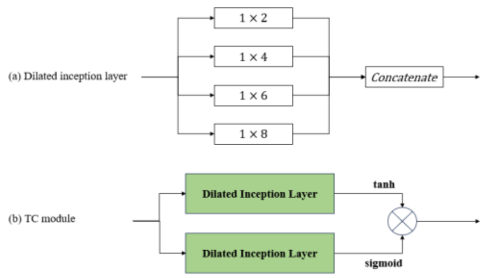

Fig.4 Change perception module

$$\vec{h}_{t,o}^{(l+1)} = \mathrm{RGCN}_{Local}\left(\vec{h}_{t,s}^{(l)}, r^{(l)}, \vec{h}_{t,o}^{(l)}\right)$$
$$= \sigma_1\left(\frac{1}{c_o}\sum_{(e_s,r),\exists(e_s,r,e_o)\in\varepsilon_t} \mathrm{W}_1^{(l)}\left(\vec{h}_{t,s}^{(l)} + r^{(l)}\right) + \mathrm{W}_2^{(l)}\vec{h}_{t,o}^{(l)}\right) \quad (8)$$

Where $\vec{\mathrm{h}}_{\mathrm{t}}^{\mathrm{l}} \in \mathbb{R}^{|\varepsilon|\times \mathrm{d}}$, $\mathrm{r}_{\mathrm{t}}^{\mathrm{l}} \in \mathbb{R}^{|\mathrm{R}|\times \mathrm{d}}$ represent the entity and relation embeddings in the l-th layer of the R-GCN at the t-th timestamp snapshot, respectively. The output of the final layer of the R-GCN is denoted as H_t^{Agg}, c_o is a normalization constant equal to the in-degree of the entity. are parameters for aggregating features and self-loops in the l-th layer, and $\sigma_1(\cdot)$ is the RReLU activation function.Variation-Sensitive Temporal Convolution

The Variation-Sensitive Temporal Convolution Module aims to effectively extract high-level dynamic change features over short periods using a set of dilated 1D convolutional filters. As shown in Figure 4, this module consists of two parallel dilated Inception layers: one serving as a filter and the other as a gating unit. The filter branch passes through a dilated Inception convolutional layer followed by a `tanh` activation function to extract rich feature representations; the gating branch employs the same structure but uses a `sigmoid` activation function to output gating signals. Subsequently, the outputs of these two branches undergo element-wise multiplication to form a gating mechanism, where the sigmoid gate finely controls the information flow of the tanh filter's output, thereby effectively filtering the most relevant temporal features. This module accepts a four-dimensional input tensor of shape (B, C, N, T) $where$ B represents the batch size, C is the number of channels (feature dimension), N denotes the number of nodes, and T is the length of the time series. After the above processing, the output tensor shape remains (B, C, N, T) meaning historical temporal information is effectively fused into each node's representation.

Selecting an appropriate convolution kernel size has always been a challenging issue for temporal convolutional networks. The traditional `Inception` strategy typically combines outputs from 2D convolutional filters of different kernel sizes (e.g., $1 \times 1, 3 \times 3$ and 5×5) to capture multi-scale features. However, considering that time series data often exhibit specific periodicities(e.g., 7, 12, 24, 48 and 60 etc.)these classical 2D filter sizes cannot adequately cover these periodic patterns. To address this, we specifically propose a time-aware inception layer design containing four 1D filters of different sizes:$1 \times 2, 1 \times 4, 1 \times 6$ and 1×8, For example, to represent a periodic pattern of length 12, the model can first acquire coarse-grained information through a 1×8 convolutional filter and then refine it further with a 1×6 filter.

Our temporal convolution module captures multi-scale temporal dependencies using four parallel 1D convolution branches with different kernel lengths and dilation coefficients. Kernels of size (1,k)slide independently along the time dimension, extracting node-specific temporal features. The receptive field size for a network with kernel size *c* and *m* layers is:

$$R = m(c-1) \times d + 1 \quad (9)$$

Where *d* is the dilation factor. We use dilated convolution to expand the receptive field without increasing depth: small kernels (e.g., sizes 2,4) capture short-term variations, while large kernels (e.g., sizes 6,8) model long-term dependencies. To enhance long-sequence capture, we design exponentially growing dilation factors q (q>1) across layers. The receptive field becomes:

$$R = 1 + (c-1)(q^m - 1)/(q-1) \tag{10}$$

This approach enables the network's receptive field size to grow exponentially at rate q as the number of hidden layers increases, thereby capturing longer sequences.

Building upon this foundation, we combine Inception with dilated convolution mechanisms to propose a Dilated Inception Layer, as shown in Figure 4(b). Formally, given a 1D sequence input $z \in R^T$ and $f_{1\times2} \in R^2$, $f_{1\times4} \in R^4, f_{1\times6} \in R^6$, $f_{1\times8} \in R^8$. our dilated inception layer adopts the following form:

$$z = concat(z \star f_{1\times2}, z \star f_{1\times4}, z \star f_{1\times6}, z \star f_{1\times8}) \tag{11}$$

Here, the outputs of the four filter branches are truncated to equal length based on the largest filter size before being concatenated along the channel dimension. The dilated convolution operation $\star$ is defined as:

$$z \star f_{1\times k}(t) = \sum_{s=0}^{k-1} f_{1\times k}(s) z(t - d \times s) \tag{12}$$

Due to differences in kernel sizes and dilation factors, the temporal lengths of branch outputs vary. In implementation, we first truncate each branch's output to the shortest length (determined by the largest kernel size and highest dilation factor) to ensure temporal alignment. Let T_m.denote this minimum output length. The aligned branch features have approximate dimensions$\left(B, \frac{C}{4}, N, T_m\right)$. Which are then concatenated to form a comprehensive multi-scale representation of shape(B, C, N, T_m).

The fused representation is projected to the output via two ReLU-activated linear layers. Specifically, we reshape the input tensor (B, C, N, Tm) into ($B\times N\times Tm, C$), merging temporal dimensions into the batch dimension. The first linear layer projects features from dimension C to C (matching the original channel number), followed by ReLU activation. The second linear layer projects back to CC. Finally, we reshape the output to (B, C, N, Tm), maintaining the input format with temporal length Tm. With appropriate padding, Tm can equal the original length TT. In summary, our variation-aware temporal convolution module integrates gating mechanisms for dynamic information regulation, dilated convolutions for expanded receptive fields, and multi-scale feature concatenation for precise temporal extraction. Maintaining consistent input/output shapes (B,C,N,T) enables seamless integration with spatial graph convolution modules, providing nodes with dynamic, historically enriched feature representations that significantly enhance complex temporal dependency and long-term pattern capture.

3.5 Global-Local Contrastive Learning

Evidently, the globally and locally defined encoding modules can effectively capture both global and local dependencies of queries. However, during actual model training, input data is inevitably subject to external noise interference, which significantly compromises the model's reasoning performance - an issue that existing methods have not adequately addressed. Inspired by the concept of unsupervised contrastive learning [28][29], we further propose a global-local query contrast module. By simultaneously identifying highly correlated local and global features of entities and relationships within queries, this module effectively filters out external noise.

Specifically, for each query at timestamp t_q the objective of the global-local query contrast module is to learn query representations that encompass both local and global semantic information by minimizing a supervised contrastive loss function. This ensures that the local and global representations of the same query are closer in the semantic space, while representations of different queries are more distinctly separated. Formally, we define the local and global query embedding representations as:

$$Z_t = MLP\left[h_t^{Agg}||r_t\right], \tag{13}$$

$$Z_g = MLP\left[h_g^{Agg}||r_t\right], \tag{14}$$

Where z_t represents the local query embedding at timestamp t, z_g denotes the corresponding global query embedding. The MLP network serves to normalize the query embeddings and project them into a feature space suitable for subsequent contrastive training. It should be noted that since the historical query subgraph does not explicitly encode temporal information, we employ initial relationship embeddings to represent global queries. Within our framework, global and local query representations can be viewed as two complementary perspectives of historical facts in temporal knowledge graphs. Therefore, we can consider the query representation generated by the local encoder as the baseline view, while the global encoder serves to enhance this baseline view. The local and global representations of the same query at timestamp t form a positive sample pair$\left(Z_{t,i}, Z_{g,i}\right)$, whereas representations of different queries form negative sample pairs $\left(Z_{t,i}, Z_{g,k}\right)$. Consequently, the supervised contrastive loss $\mathcal{L}_{lg}$ at timestamp t_q can be defined as:

$$\mathcal{L}_{lg} = \frac{1}{|Q_{t_q}|} log \frac{exp\left(Z_{t,i} \cdot Z_{g,i}/\tau\right)}{\sum_{k \in N_{t_q}, k \neq i}\left(Z_{t,i} \cdot Z_{g,k}/\tau\right)}, \tag{15}$$

Where Q_{t_q} denotes the set of queries at timestamp t_q, N_{t_q} represents the number of queries in the corresponding mini-batch at that timestamp, and τ is the temperature parameter controlling the contrastive distribution. The objective of loss function $\mathcal{L}_{lg}$ is to reduce the semantic distance between different views of the same query while enhancing shared features between local and global encodings, thereby effectively mitigating noise interference and improving model robustness.

To further strengthen the model's ability to distinguish between different query representations in semantic space, we apply similar constraints to local-local and global-global representations, obtaining two additional supervised contrastive losses $\mathcal{L}_{ll}$ and $\mathcal{L}_{gg}$. This yields four supervised contrastive losses in total: $\mathcal{L}_{lg}, \mathcal{L}_{gl}, \mathcal{L}_{ll}and\mathcal{L}_{gg}$ The final supervised contrastive loss is computed as:

$$L_{cl} = \left(\mathcal{L}_{lg} + \mathcal{L}_{gl} + \mathcal{L}_{ll} + \mathcal{L}_{gg}\right)/4 \tag{16}$$

This multi-view supervised contrastive mechanism enables the model to more comprehensively capture semantic differences and similarities at varying granularities, significantly enhancing reasoning performance in complex environments.

3.6 Prediction and Evaluation

ConvTransE [30], a powerful scoring function widely used in recent TKG reasoning tasks, serves as our base decoder for entity prediction at timestamp t_q The entity prediction scoring function is defined as:

$$\psi\left(e_q, r_q, e, q\right) = \sigma_2\left(h_{t_q}^{e_q} ConvTransE\left(\hat{h}_{t_q}^{e_q}, r_{t_q}\right)\right), \tag{17}$$

Where $\hat{h}_{t_q}^{e_q}$ represents the query entity embedding combining global and local representations:

$$\hat{h}_{t_q}^{e_q} = \lambda \vec{h}_g^{e_q} + (1-\lambda)\vec{h}_{t_q}^{e_q}, \tag{18}$$

Where, $\lambda \in [0,1]$ is an adjustable weighting coefficient balancing global trend and local context representations. We formulate entity prediction as a multi-label learning problem, with the prediction loss $\mathcal{L}_{tkg}$ defined as:

$$\mathcal{L}_{tkg} = \sum_{t=0}^{\tau} \sum_{(e_s,r,e,t)\in\mathcal{F}_t} \sum_{e\in\varepsilon} \mathcal{Y}_t^e \log \emptyset(e_s,r,e,t) \ , \tag{19}$$

Where, $\emptyset(e_s,r,e,t)$ denotes the entity prediction score for a given triple, ε is the entity set, and$\mathcal{Y}_t^e \in \mathbb{R}^{|\varepsilon|}$ is the label vector (with value 1 at position e if the fact occurs at time t, 0 otherwise). The final loss function combines:

$$\mathcal{L} = \mathcal{L}_{tkg} + \mathcal{L}_{cl} \ , \tag{20}$$

During training, we jointly optimize both entity prediction loss $\mathcal{L}_{tkg}$ and contrastive supervised loss $\mathcal{L}_{cl}$ through synchronous learning.

Notably, to enhance training symmetry and prediction performance, we introduce inverse quadruples during both training and testing phases. The inverse query set is generated from the original query set at timestamp t_q. While constructing historical KG snapshots each epoch using combined original and inverse quadruples, direct training on this combined set risks data leakage as the entity-aware attention module might simultaneously access both subject and object entities. To prevent this, we propose a two-phase forward propagation strategy: Phase 1 processes only original queries, while Phase 2 handles inverse queries in each training cycle. This design maintains expressive power while effectively preventing information leakage. The detailed training procedure of model is summarized in table 1.

Table 1 Training procedure of TVCL-Net

Algorithm 1: Training procedure of TVCL-Net

Input: the historical KG snapshot sequence$\{G_1, G_2, \dots G_{t_q}\}$,query set Q_{query} with unknow object entities at time t_q.
Output: The reasoning results for each query are in descending order of scores.
Initialize the embeddings of entities, relation.
while $t < \lvert\tau\rvert$ do
$t' = t_q - m$
while $t' < t_q$ and $t'_q > 0$ do
$t' = t' + 1$
Aggregate local KG snapshot by Eq.2-Eq.4
Relation presentation and entity representation with mutation information are obtained by learning KG sequence through Eq5-Eq.8
end while
Entity representation with historical trend information is obtained by learning the KG sequence through formula Eq.9-Eq.11.
$t'' = t - m$
while $t'' < t$ and $t'' > 0$ do
$t'' = t'' + 1$
Compute the local-global query contrast loss by Eq.11-Eq.15.
end while
Replace the missing object entities for each query $(s, r, ?, t) \in \mathcal{Q}_{query}$ and calculate the scoring function by Eq.16-Eq.20.
$t = t + 1$
end while

4. Experiments

4.1 Datasets and Baselines

4.1.1 Datasets

To comprehensively evaluate the performance of TVCL-Net in entity prediction tasks, we employed four benchmark datasets widely used for TKG extrapolation: ICEWS14, ICEWS18, ICEWS05-15 [31], and GDELT [32]. Among these, ICEWS14, ICEWS05-15, and ICEWS18 are subsets derived from the Integrated Crisis Early Warning System (ICEWS), containing numerous political events with specific timestamps. GDELT is a global event database encompassing 20 types of events. Following the preprocessing strategies in [33][34], we split all datasets into training, validation, and test sets at a ratio of 80%/10%/10%. The statistical details of all datasets are provided in Table 2.

4.1.2 Baselines

To comprehensively evaluate the reasoning capabilities of TVCL-Net, we selected multiple representative and widely-covered baseline methods for comparison, encompassing two major categories: static knowledge graph methods and temporal extrapolation approaches:

(1) Static Knowledge Graph (SKG) :Methods:

DisMult [35], ComplEx [36], Conv-TransE [30], RotatE [37]Temporal extensions include: TTransE [38], TA-DisMult [40], De-SimIE [39], TNT-ComplEx [41], TANGO-Tucker [42]

(2) Temporal Knowledge Graph Extrapolation Models:

xERTE [43], TITer [44], CyGNet [25], RE-NET [3], RE-GCN [22], CEN [45]Latest methods include: TiRGN [6], HisMatch [5], RETIA [46], CENET [24]

These methods exhibit distinct emphases in terms of encoding mechanisms, approaches to historical information utilization, and temporal modeling capabilities, collectively forming a systematic evaluation framework for TVCL-Net.

4.2 Experimental Settings

4.2.1 Evaluation Metrics

We employ two widely-used evaluation metrics for assessing TKG reasoning methods: Mean Reciprocal Rank (MRR) and$Hit@k(k = 1,3,10)$, MRR calculates the average reciprocal rank of the ground truth across all queries, while $Hit@k$ measures the proportion of cases where the correct entity appears in the top-k ranked candidates. Recent studies [47][48] have shown that traditional static filtering settings are unsuitable for TKG extrapolation as they ignore temporal dimensions of facts. In practice, only concurrently occurring facts should be filtered. Therefore, we report results using the temporally-aware filtering setting that has been widely adopted in recent work, which only filters quadruples occurring at the query time.

Table 2 Details of the dataset

Dataset	ICEWS14	ICEWS05-15	ICEWS18	GDELT
Entities	6,869	10,094	23,033	7,691
Relations	230	256	251	240
Training	74,845	373,018	368,868	1,734,399
Validation	8,514	45,995	46,302	238,765
Test	7,371	49,545	46,159	305,241
Time granularity	24 hours	24 hours	24 hours	15 mins
Snapshot numbers	365	365	4,017	2,975

4.2.2 Implementation Details

For all datasets, we set the embedding dimension d to 200 and the learning rate to 0.001. The batch size was configured according to the number of quadruples per timestamp. TVCL-Net parameters were optimized using Adam [49] during training. The R-GCN layers in both the local entity-aware attention recursive encoder and global entity attention-aware encoder were set to 2, with a dropout rate of 0.2 per layer. The optimal local historical KG snapshot sequence lengths for ICEWS14, ICEWS18, ICEWS05-15, and GDELT were set to 7, 7, 9, and 7 respectively. The prediction weight λ was set to 0.9 for all datasets, while the temperature coefficients were configured as 0.03, 0.03, 0.07, and 0.07 for ICEWS14, ICEWS18, ICEWS05-15, and GDELT correspondingly. For the decoder across all datasets, we used 50 filters with kernel size 2×3 and a dropout rate of 0.2.

Table 3 Ablation experiment

Model	ICEWS14				ICEWS18				ICEWS05-15			
	MRR	Hits @1	Hits @3	Hits@ 10	MRR	Hits@ 1	Hits @3	Hits@ 10	MRR	Hits@ 1	Hits @3	Hits @10
TVCL-Net	50.20	37.88	55.99	72.85	37.76	24.98	41.20	58.32	57.02	46.51	64.11	78.02
TVCL-Net-T	49.33	37.13	54.14	72.69	37.02	23.96	39.22	56.78	56.29	45.03	63.48	76.60
TVCL-Net-A	48.95	36.76	49.04	70.56	36.08	23.57	38.08	55.26	56.51	44.67	64.06	77.24

4.2.3 Ablation experiment

To analyze individual component contributions, we conducted ablation studies (Table 3) with two variants:

(1) TVCL-Net-A: Removes the trend-aware module while retaining the variation-aware component, isolating the impact of short-term variation features.

(2) TVCL-Net-T: Removes the variation-aware module while keeping the trend-aware component, evaluating long-term trend effects.

Results demonstrate that while both mechanisms contribute to performance gains, the trend-aware module shows more significant improvements across all datasets, highlighting the critical importance of long-term evolutionary information for entity prediction. The variation-aware module specializes in capturing local abrupt events, effectively addressing limitations of traditional methods in handling non-stationary patterns.

4.2.4 Comparative Results

Table 4 presents comprehensive experimental results across all four benchmark datasets, demonstrating TVCL-Net's superior overall performance. Key findings include:

(1) TVCL-Net outperforms state-of-the-art methods on all four benchmarks, achieving improvements of 5.9%, 5.4%, 2.3%, and 3.6% on ICEWS14, ICEWS18, ICEWS05-15, and GDELT respectively.

(2) While HisMatch considers local historical information queries, it lacks capability to evaluate the importance of different historical information elements.

(3) Although CENET employs contrastive learning, its performance is limited by not accounting for trend patterns and abrupt changes in historical information.

(4) Comparative analysis confirms that our contrastive learning approach effectively enhances both model performance and prediction accuracy.

5. Conclusion

This paper presents TVCL-Net, a novel contrastive learning model integrating trend and variation awareness mechanisms for extrapolation reasoning tasks in Temporal Knowledge Graphs (TKGs). The proposed model adopts a dual-path global-local encoding architecture to comprehensively model entity evolution behaviors across different timescales. In the global encoding path, we introduce a trend-aware convolution mechanism to effectively extract latent patterns from long-term historical evolution, while the local encoding path incorporates a variation-aware module to capture fine-grained changes occurring in node neighborhoods over short time intervals.

Furthermore, we propose a global-local contrastive learning mechanism that enhances the model's discriminative capability for key features in historical events through contrastive optimization of positive and negative sample pairs, significantly improving entity prediction accuracy. Experimental results on four authoritative datasets demonstrate that TVCL-Net outperforms existing mainstream methods in both accuracy and robustness. Future work may explore incorporating multimodal information (e.g., text and images) into temporal-aware graph modeling to further enhance the model's applicability in complex real-world scenarios.

References

[1] Trivedi, R., Dai, H., Wang, Y., & Song, L. (2017, July). Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In international conference on machine learning (pp. 3462-3471). PMLR.

[2] Trivedi, R., Farajtabar, M., Biswal, P., & Zha, H. (2019, May). Dyrep: Learning representations over dynamic graphs. In International conference on learning representations.

[3] W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2022-11-24, pp. 6669–6683

[4] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-GCN: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems, 21(9), 3848-3858.

[5] Z. Li, Z. Hou, S. Guan, X. Jin, W. Peng, L. Bai, Y. Lyu, W. Li, J. Guo, and X. Cheng, "Hismatch: Historical structure matching based temporal knowledge graph reasoning," in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, 2022, pp. 7328–7338.

[6] Y. Li, S. Sun, and J. Zhao, "Tirgn: time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning," in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022. ijcai. org, 2022, pp. 2152–2158

[7] Lin, Y., Wang, H., Sun, Y., & Zha, D. (2025). LTGQ: Learning Temporal Granularity for Temporal Knowledge Graph Forecasting with Quadruplet Networks. In Proceedings of the 2025 International Conference on Learning Representations (ICLR).

[8] Ju, W., Luo, X., Ma, Z., Yang, J., Deng, M., & Zhang, M. (2022). GHNN: Graph harmonic neural networks for semi-supervised graph-level classification. Neural Networks, 151, 70-79.

[9] Liu, W., Zhang, C., Lin, G., & Liu, F. (2020). Crnet: Cross-reference networks for few-shot segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4165-4173).

[10] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "Amie: association rule mining under incomplete evidence in ontological knowledge bases," in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 413–422.

[11] Omran, P. G., Wang, K., & Wang, Z. (2018). Scalable rule learning via learning representation. In IJCAI-18. International Joint Conferences on Artificial Intelligence. Ji, S. et al. (2021). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE TNNLS.

[12] A. Bordes, N. U sunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi relational data," in Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.

[13] Nickel, M., Tresp, V., & Kriegel, H. P. (2011, June). A three-way model for collective learning on multi-relational data. In Icml (Vol. 11, No. 10.5555, pp. 3104482-3104584). T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in ICLR, 2017.

[14] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, ser. Lecture Notes in Computer Science, vol. 10843. Springer, 2018, pp. 593–607.

[15] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020, April). K-bert: Enabling language representation with knowledge graph. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 03, pp. 2901-2908).

[16] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Karsai, M., & Benson, A. R. (2020). EvolveGCN:

Evolving Graph Convolutional Networks for Dynamic Graphs. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04), 5363–5370.
[17] Sankar, A., Wu, Y., Gou, W., Zhang, Y., Yang, J., & Sundaram, H. (2020). Dynamic Graph Representation Learning via Self-Attention Networks. In ICLR 2020.
[18] Rossi, E., Chamberlain, B. P., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. M. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. ICML 2020 Workshop. arXiv preprint.
[19] Wang, Xuhong, et al. "Apan: Asynchronous propagation attention network for real-time temporal graph embedding." Proceedings of the 2021 international conference on management of data. 2021.Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive representation learning on temporal graphs. In International Conference on Learning Representations (ICLR).
[20] He, Y., Zhang, P., Liu, L., Liang, Q., Zhang, W., & Zhang, C. (2024). Hip network: Historical information passing network for extrapolation reasoning on temporal knowledge graph. arxiv preprint arxiv:2402.12074.
[21] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive representation learning on temporal graphs. arXiv preprint arXiv:2002.07962.
[22] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "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. Association for Computing Machinery, 2021, pp. 408–417.
[23] Wang, Y., He, X., Cao, Y., Zhang, M., & Chua, T.-S. (2021). CAWN: Causal Anonymous Walks for Temporal Graphs. In NeurIPS 2021.
[24] Y. Xu, J. Ou, H. Xu, and L. Fu, "Temporal knowledge graph reasoning with historical contrastive learning," in Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. AAAI Press, 2023, pp. 4765–4773
[25] C. Zhu, M. Chen, C. Fan, G. Cheng, and Y. Zhang, "Learning from history: Modeling temporal knowledge graphs with sequential copygeneration networks," vol. 35, no. 5, pp. 4732–4740, 2021, number: 5.
[26] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan,"Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in Advances in Neural Information Processing Systems, 2019, pp. 5244–5254
[27] Z. Li, Z. Hou, S. Guan, X. Jin, W. Peng, L. Bai, Y. Lyu, W. Li, J. Guo, and X. Cheng, "Hismatch: Historical structure matching based temporal knowledge graph reasoning," in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, 2022, pp. 7328–7338.
[28] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.
[29] Y. Xu, J. Ou, H. Xu, and L. Fu, "Temporal knowledge graph reasoning with historical contrastive learning," in Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. AAAI Press, 2023, pp. 4765–4773
[30] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press, 2019, pp. 3060–3067.
[31] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, and M. Ward, "Icews coded event data. harvard dataverse," V4 (http://dx. doi. org/10.7910/DVN/28075), 2015.
[32] K. Leetaru and P. A. Schrodt., "Gdelt: Global data on events, location, and tone, 1979-2012," ISA Annual Convention, vol. 4, pp. 1–49, 2013.
[33] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "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. Association for Computing Machinery, 2021, pp. 408–417.
[34] W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2022-11-24, pp. 6669–6683.
[35] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015
[36] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, ´ "Complex embeddings for simple link prediction," in Proceedings of the 33nd International Conference on Machine Learning, ICML 2016,
[37] Z. Sun, Z. Deng, J. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex

space," in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[38] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds. ACM, 2018, pp. 1771–1776

[39] R. Goel, S. M. Kazemi, M. A. Brubaker, and P. Poupart, "Diachronic embedding for temporal knowledge graph completion," in The ThirtyFourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020, pp. 3988–3995

[40] A. Garc´ıa-Duran, S. Dumancic, and M. Niepert, "Learning sequence ´ encoders for temporal knowledge graph completion," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. Association for Computational Linguistics, 2018, pp. 4816–4821.

[41] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[42] Z. Han, Z. Ding, Y. Ma, Y. Gu, and V. Tresp, "Learning neural ordinary equations for forecasting future links on temporal knowledge graphs," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021, pp. 8352–8364.

[43] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021.

[44] H. Sun, J. Zhong, Y. Ma, Z. Han, and K. He, "TimeTraveler: Reinforcement learning for temporal knowledge graph forecasting," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021, pp. 8306–8319.

[45] Z. Li, S. Guan, X. Jin, W. Peng, Y. Lyu, Y. Zhu, L. Bai, W. Li, J. Guo, and X. Cheng, "Complex evolutional pattern learning for temporal knowledge graph reasoning," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, 2022, pp. 290– 296.

[46] K. Liu, F. Zhao, G. Xu, X. Wang, and H. Jin, "Retia: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation," in IEEE International Conference on Data Engineering. IEEE, 2023.

[47] Z. Han, Z. Ding, Y. Ma, Y. Gu, and V. Tresp, "Learning neural ordinary equations for forecasting future links on temporal knowledge graphs," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021, pp. 8352–8364

[48]] Z. Li, Z. Hou, S. Guan, X. Jin, W. Peng, L. Bai, Y. Lyu, W. Li, J. Guo, and X. Cheng, "Hismatch: Historical structure matching based temporal knowledge graph reasoning," in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, 2022, pp. 7328–7338

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

Table 4 Comparison of predicted performance

Model	ICEWS14				ICEWS18				ICEWS05-15				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
DisMult(2014)	15.44	10.91	17.24	23.92	11.51	7.03	12.87	20.86	17.95	13.12	20.71	29.32	8.68	5.58	9.96	17.13
ComplEx(2016)	32.54	23.43	36.13	50.73	22.94	15.19	27.05	42.11	32.63	24.01	37.50	52.81	16.96	11.25	19.52	32.35
ConvE(2018)	35.09	25.23	39.38	54.68	24.51	15.23	29.25	44.51	33.81	24.78	39.00	54.95	16.55	11.02	18.88	31.60
Conv-TransE(2019)	33.80	25.40	38.54	53.99	22.11	13.94	26.44	42.28	33.03	24.15	38.07	54.32	16.20	10.85	18.38	30.86
RotatE(2019)	21.31	10.26	24.35	44.75	12.78	4.01	14.89	31.91	24.71	13.22	29.04	48.16	13.45	6.95	14.09	25.99
TTransE(2016)	13.72	2.98	17.70	35.74	8.31	1.92	8.56	21.89	15.57	4.80	19.24	38.29	5.50	0.47	4.94	15.25
TA-DisMult(2018)	25.80	16.94	29.74	42.99	16.75	8.61	18.41	33.59	24.31	14.58	27.92	44.21	12.00	5.76	12.94	23.54
DE-SimIE(2020)	33.36	24.85	37.15	49.82	19.30	11.53	21.86	34.80	35.02	25.91	38.99	52.75	19.70	12.22	21.39	33.70
TNTComplEx (2020)	34.05	25.08	38.50	50.92	21.23	13.28	24.02	36.91	27.54	9.52	30.80	42.86	19.53	12.41	20.75	33.42
RE-NET(2020)	36.93	26.83	39.51	54.78	28.81	19.05	32.44	47.51	43.32	33.43	47.77	63.05	19.62	12.42	21.00	34.01
CyGNet(2020)	35.05	25.73	39.01	53.55	24.93	15.90	28.28	42.61	36.81	26.61	41.63	56.22	18.48	11.52	19.57	31.98
TANGO-Tucker (2021)	36.80	27.43	40.89	54.93	28.68	19.35	32.17	47.04	42.86	32.72	48.14	62.34	19.53	12.43	20.79	33.19
xERTE (2021)	40.02	32.06	44.63	56.17	29.98	22.05	33.46	44.83	46.62	37.84	52.31	63.92	18.09	12.30	20.06	30.34
RE-GCN (2021)	40.39	30.66	44.96	59.21	30.58	21.01	34.34	48.75	48.03	37.33	53.85	68.27	19.64	12.42	20.90	33.69
TiRGN (2022)	44.04	33.83	48.95	63.84	33.66	23.19	37.99	54.22	50.04	39.25	56.13	70.71	21.67	13.63	23.27	37.60
HisMatch (2022)	46.42	35.91	51.63	66.84	33.99	23.91	37.90	53.94	52.85	42.01	59.05	73.28	22.01	14.45	23.80	36.61
RETIA (2023)	42.76	32.28	47.77	62.75	32.43	22.23	36.48	52.94	47.26	36.64	52.90	67.76	20.12	12.76	21.45	34.49
CENET (2023)	39.02	29.62	43.23	57.49	27.85	18.15	31.63	46.98	41.95	32.17	46.93	60.43	20.23	12.69	21.70	34.92
TVCL-Net	50.20	37.88	55.99	72.85	37.76	24.98	41.20	58.32	57.02	46.51	64.11	78.02	24.03	14.88	26.30	42.99