Artificial Intelligence Review (2024) 57:344
https://doi.org/10.1007/s10462-024-10983-0

Chronobridge: a novel framework for enhanced temporal and relational reasoning in temporal knowledge graphs

Qian Liu[1] · Siling Feng[1] · Mengxing Huang[1] · Uzair Aslam Bhatti[1]

Accepted: 26 September 2024 / Published online: 22 October 2024

Abstract

The task of predicting entities and relations in Temporal Knowledge Graph (TKG) extrapolation is crucial and has been studied extensively. Mainstream algorithms, such as Gated Recurrent Unit (GRU) models, primarily focus on encoding historical factual features within TKGs, often neglecting the importance of incorporating entities and relational features during decoding. This bias ultimately leads to loss of detail and inadequate prediction accuracy during the inference process. To address this issue, a novel ChronoBridge framework is proposed that features a dual mechanism of a chronological node encoder and a bridged feature fusion decoder. Specifically, the chronological node encoder employs an advanced recursive neural network with an enhanced GRU in an autoregressive manner to model historical KG sequences, thereby accurately capturing entity changes over time and significantly enhancing the model's ability to identify and encode temporal patterns of facts across the timeline. Meanwhile, the bridged feature fusion decoder utilizes a new variant of GRU and a multilayer perception mechanism during the prediction phase to extract entity and relation features and fuse them for inference, thereby strengthening the reasoning capabilities of the model for future events. Testing on three standard datasets showed significant improvements, with a 25.21% increase in MRR accuracy and a 39.38% enhancement in relation inference. This advancement not only improves the understanding of temporal evolution in knowledge graphs but also sets a foundation for future research and applications of TKG reasoning.

Keywords Temporal knowledge graph reasoning · Gated Recurrent Unit · Chronological node encoder · Bridged feature fusion decoder

Abbreviations

KG	Knowledge Graph
TKG	Temporal Knowledge Graph
GRU	Gated Recurrent Unit
GNN	Graph Neural Network
GCN	Convolutional Network
IGRU	Improved Gated Recurrent Unit

Extended author information available on the last page of the article

Self-GRU	Self-Gated Recurrent Unit
MLP	Multi-Layer Perceptron
MRR	Mean Reciprocal Rank
Hits@K	Hits at Rank K

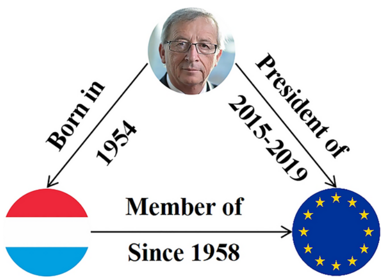
1 Introduction

In response to the dynamic nature of information in the digital era, Temporal Knowledge Graphs (TKGs) have evolved from traditional Knowledge Graphs (KGs) by integrating a time dimension to capture the transient aspects of real-world data (Ji et al. 2024; Bai 2023). TKGs enhance static triples with timestamps, creating quadruples that offer a dynamic and temporal narrative of relations, as exemplified by Jean-Claude Juncker's presidency of the European Union between 2015 and 2019, as shown in Fig. 1. This innovation enables advanced temporal reasoning, trend analysis, and predictive modeling, thereby providing a more comprehensive and precise representation of knowledge over time (Zhang et al 2023; Ge et al. 2022; Negro et al. 2023).

However, reasoning over TKGs introduces novel challenges, specifically in the encoding and decoding processes. Encoding involves transforming temporal information along with entities and relations into vector representations, whereas decoding utilizes these representations to infer and predict temporal relations. Traditional methods adept at handling static KGs fall short of capturing the temporal dynamics essential for TKGs. For example, models such as RotatE (Sun et al. 2019) excel in static environments, but lack the temporal sensitivity required for TKGs. Conversely, methods that incorporate timestamps often do so without fully leveraging the temporal dimension, leading to incomplete understanding of the evolution of knowledge graphs (Shang et al. 2019; Goel et al 2020). Advanced algorithms, such as RE-GCN (Li et al. 2021), attempt to address this by capturing the evolution of representations at individual timestamps. However, they often neglect the integration of entity and relation characteristics during decoding, which is crucial for adapting to the complexity. Thus, while effective methods are available, a comprehensive understanding of TKG evolution and efficient encoding-decoding processes remains elusive. The potential to enhance the precision of reasoning within TKGs is substantial and largely untapped. This study focused on reasoning tasks within the TKGs, as depicted in Fig. 2.

These tasks involve predicting the future state of a graph based on historical data, encompassing two specific subtasks: entity prediction and relation prediction. Entity prediction involves forecasting future relations based on a given timestamp and entity relation, such as predicting future diplomatic visits of a country. Relation prediction involves predicting

Fig. 1 An example of a TKG



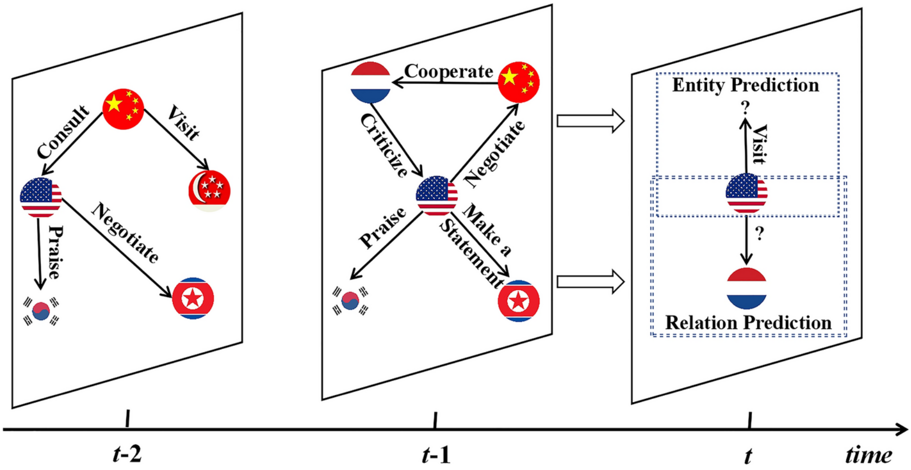


Fig. 2 Schematic diagram of temporal reasoning on TKGs, highlighting the prediction tasks at different timestamps

the future relation between a pair of entities, such as the diplomatic stance between two countries on a future timestamp.

Challenges and our approach

The task of predicting future entities and relations in TKGs is fraught with several challenges. While recurrent models, such as GRUs, are proficient at encoding historical TKG data, they struggle to integrate entity and relation features effectively, leading to a loss of detail and decreased prediction accuracy. Capturing the complex patterns of entity evolution over time and integrating bridging features to infer future events are areas that the current algorithms falter. To overcome these challenges, we propose the ChronoBridge framework, a novel approach that includes a chronological node encoder and bridged feature fusion decoder. The encoder employs an enhanced GRU-based recursive neural network to capture the temporal dynamics of the entities with greater precision. The decoder, a novel variant of the GRU combined with a multilayer perceptron, fuses the entity and relation features to improve the prediction accuracy.

Key contributions

The key contributions of our study are as follows:

(1) Pioneering Feature Fusion for TKG Decoding: We propose the first algorithm to fuse entity and relation features for TKG decoding embodied in the ChronoBridge framework with its novel chronological node encoder and bridged feature fusion decoder.
(2) Advancements in GRU Technology: We propose two innovative GRU variants, IGRU and Self-GRU, which enhance long-term dependency handling and self-memory, respectively, improving the model's ability to capture temporal dependencies.
(3) Empirical Validation: Our extensive testing on benchmark datasets reveals that the ChronoBridge significantly outperforms existing state-of-the-art methods, solidifying its position as a cutting-edge TKG reasoning framework.The subsequent sections of this paper are organized as follows. Section 2 reviews the literature on knowledge graphs and temporal reasoning, comparing traditional models with those designed for

temporal data. Section 3 elaborates on foundational concepts including knowledge graphs, TKGs, and GRUs, which are essential for comprehending the ChronoBridge framework. Section 4 explains the ChronoBridge methodology and emphasizes its core components. Section 5 presents the experimental setup and results, benchmarking the ChronoBridge against established models in various scenarios. Section 6 discusses the strengths, weaknesses, and practical implications of the proposed framework. Finally, Sect. 7 concludes with a summary of our findings, future research avenues, and the potential for chronobridge applications in broader contexts.

2 Related work

In this section, common techniques for reasoning within knowledge graphs and reasoning using TKGs are outlined.

2.1 Knowledge graph reasoning

Models for reasoning over knowledge graphs can be broadly classified into two types: those that rely on embeddings and those that utilize graph structures. The former involves mapping entities and relations from knowledge graphs into compact vector spaces for inferential processing. Common methods include TransE (Bordes et al. 2013), TransH (Wang et al. 2014) and TransR (Lin et al. 2015), etc. These models achieve prediction of new reasoning problems by learning the relation vector representation and relation conversion methods between entities, such as entity relation prediction (Yuan et al. 2021), entity classification (Sun et al. 2022), etc. The benefit of this method is its ability to grasp semantic similarities and connections between entities and relations. However, its drawback is its inability to leverage the inherent structural information of a graph. The reasoning model based on graph structure uses the graph structure of the knowledge graph for reasoning. Common methods include path-reasoning models (Bai et al. 2023) and graph neural networks (Zhou 2020). The path-reasoning model performs relation prediction or entity classification by determining the associated paths among the entities and relations within the graph. A graph neural network employs GNN techniques to characterize the vertices and links within a network, and performs reasoning through message passing, node updating, and edge modeling. The benefit is that it can fully exploit the graph's structural data, yet its downside is that it might demand extensive computational resources for reasoning over large-scale networks. Although the above two types of knowledge graph reasoning models are effective in static knowledge graphs, they only consider the static associations between entities and relations, and cannot dynamically capture changes in relations and entities.

2.2 TKG reasoning

Regarding the reasoning of TKG, it can be partitioned into two modalities: interpolation (Ding et al. 2023) and extrapolation (Zhongwu 2023).

Interpolation is the process of estimating unknown values within a range of known data points. In the context of TKG reasoning, interpolation involves predicting the state of relations at intermediate time points given the data available at specific times. This approach assumes that temporal changes between known data points follow a continuous pattern, allowing for accurate estimates of intermediate states. On the other hand, extrapolation involves predicting future values or states beyond the range of known data points. For TKGs, extrapolation refers to forecasting how relations and interactions evolve beyond the existing temporal data. This approach relies on understanding long-term patterns and trends to predict future states, often using models that generalize past observations to anticipate future developments. Both modalities were crucial for comprehensive TKG reasoning. Interpolation helps to understand and fill gaps within the temporal spectrum, enhancing the accuracy of predictions for periods where data are sparse. Extrapolation extends this understanding to the future, providing insights into potential future scenarios and enabling proactive decision making based on projected trends.

In the interpolation setting, some of the entities or events in the time-series knowledge graph and their temporal attributes are known, and entities or events at unknown time points are filled in through reasoning. For example, models (Zhu et al. 2023; Julien and Wudage 2018; Dasgupta et al. 2018) strive to discern absent details of past temporal markers. While TA-DistMult (Zhu et al. 2023) and TTransE (Zhu et al. 2023) incorporate temporal information into relation embeddings, HyTE (Dasgupta et al. 2018) uses a hyperplane to map timestamps. However, none of these models can accurately predict future timestamps, and cannot be directly applied to extrapolative reasoning scenarios to infer what may happen in the future.

In the extrapolation setting focused on in this article, some entities or events in the time-series knowledge graph and their temporal attributes are known, and entities or events at future time points are predicted through reasoning. For example, the models RGCRN (Zhang et al. 2022) and RE-NET (Yang and Yin 2020) attempt to perform TKG reasoning by using a message-passing mechanism, although they model the entities and relations in KG as a graph structure and use message passing in the graph. Carry out information dissemination, but they mainly focus on the relation characteristics at a particular point in time and lack the ability to capture the relation characteristics at historical time points. To solve this problem, the model's CyGNet (Zhu et al. 2021), xERTE (Han et al. 2020), TLogic (Liu et al. 2022), TITer (Sun et al. 2021), CEN (Li et al. 2022a), RE-GCN (Li et al. 2021), TiRGN (Li et al. 2022b), and RETIA (Liu et al. 2023), the relation characteristics on the timestamp can better capture the dynamics and evolution trends of the relation, and then more comprehensively understand the relations in the knowledge graph. Among them, RE-GCN and TiRGN are the algorithms most closely associated with us. While RE-GCN takes into account the structural interdependencies among facts occurring concurrently in the KG, the sequential patterns of facts that are adjacent in time, and the invariant attributes of entities, it does not consider the characteristics of entities and relations in decoding. Although TiRGN considers the order, repetition, and cycle patterns of historical facts and applies temporal features to decoding, it does not consider the characteristics of the entities and relations in decoding.

However, these two algorithms do not specifically address the complexity of TKG reasoning during encoding. In contrast, the proposed ChronoBridge in this study not only inherits the strengths of RE-GCN, but also proposes two variants of GRU. It delves deeper into the structural connections among concurrently occurring events in knowledge graphs and the sequential patterns between temporally adjacent events. Inspired by TiRGN, it integrates the features of facts and relations into a decoding process. These innovations empower ChronoBridge with significant advantages in inference tasks within the TKGs. Having outlined the advancements introduced by ChronoBridge and their relation to existing techniques, it is essential to delve into the foundational concepts that underpin these developments. Therefore, the following section covers foundational concepts that are essential for understanding TKGs, GRUs, and their associated tasks.

3 Preliminaries

This section elucidates the foundational concepts crucial for understanding the evolution and applications of TKGs and Gated Recurrent Units (GRUs), as well as the tasks associated with TKGs.

TKGs unfold as a sequence of graphs $G = \{G_1, G_2, \ldots, G_t, \ldots\}$, where each G_t captures a temporal snapshot of the Knowledge Graph. They encompass entities V and relations R, each fact presented as a tuple (s, r, o, t) , denoting subject s, relation r, object o, and the timestamp t when the fact is considered true.

Task 1: entity prediction

The entity prediction task in TKGs aims to predict future entities by scrutinizing historical KG sequences up to the last k timestamps, encapsulated in an entity embedding matrix $\mathbf{H}_t \in \mathbb{R}^{|V|\times d}$. For a query $(s, r, ?, t+1)$, a conditional probability vector $\vec{p}(o|s, r, G_{t-k+1:t})$ is derived, indicating the probability of each entity being the correct object at time $t+1$.

Task 2: relation prediction

Relation prediction seeks to identify future relations by harnessing historical KG data up to k timestamps back, represented by a relation embedding matrix $\mathbf{R}_t \in \mathbb{R}^{|R|\times d}$. For a query $(s, ?, o, t+1)$, a conditional probability vector $\vec{p}(r|s, o, G_{t-k+1:t})$ is calculated, estimating the probability of each relation being the appropriate link at $t+1$.

Fundamentals of GRU

GRUs, as shown in Fig. 3, proposed by Chung et al. (2014) in 2014, are an RNN variant designed to overcome the vanishing gradient problem of traditional RNNs. They feature a gating mechanism that controls the update and reset of the state, facilitating the capture of long-term dependencies in sequence data. GRUs consist of two gates: the update gate, which determines the retention of past information, and the reset gate, which decides how much past information to forget when assimilating new inputs. This gating mechanism enables GRUs to dynamically manage information across varying time scales, essential for sequential data processing.

4 The proposed approach

This section outlines the proposed approach, beginning with an overview of the model and detailing its core components: a Chronological Node Encoder for temporal data, a Bridged Feature Fusion Decoder for integrating features, and score functions for predicting entities and relations. It also discusses the loss function for optimization, evaluates computational complexity, and provides a detailed algorithm for implementation.

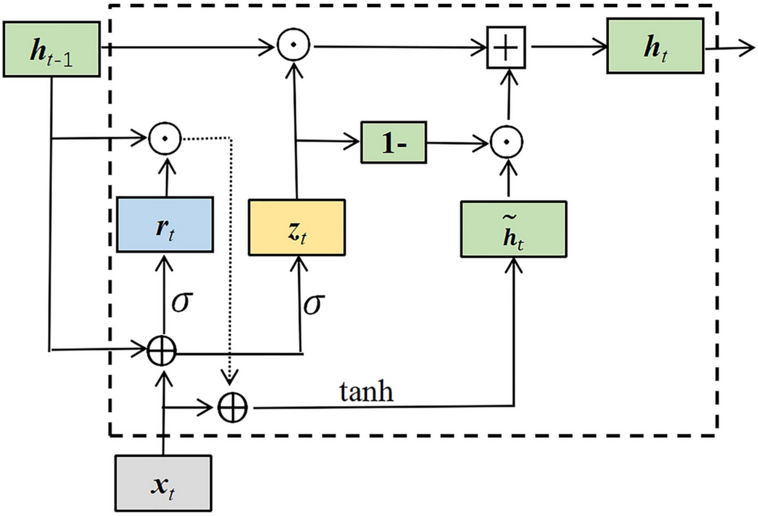


Fig. 3 The architecture of a GRU model

4.1 Model overview

The overall framework of ChronoBridge is shown in Fig. 4. ChronoBridge is a framework developed to advance TKG reasoning for future event prediction by analyzing temporal fact interactions. It remedies the shortcomings of traditional approaches that neglect the decoding phase and the significance of entity and relation features, through its innovative design consisting of a chronological node encoder and a bridged feature fusion decoder.

Chronological node encoder: This component of ChronoBridge utilizes an advanced recurrent neural network with an enhanced gated recurrent unit (GRU) to model historical KG sequences in an autoregressive manner. This innovation significantly improves the model's ability to recognize and encode temporal patterns of facts throughout the timeline.

Bridged feature fusion decoder: In contrast, this component concentrates on the extraction and integration of entity relation features, thereby providing a more nuanced interpretation of evolving relations and enriching the reasoning process.

The architecture of ChronoBridge not only tackles the temporal dynamics inherent in TKGs but also strategically incorporates entity and relation features, offering a more comprehensive approach to TKG reasoning. This framework has undergone rigorous evaluation against three real-world datasets, demonstrating substantial improvements over baseline methods. The results highlight ChronoBridge's proficiency in capturing the complex temporal and relational patterns within TKGs, signifying a significant leap forward in predictive accuracy and the understanding of knowledge graph temporal evolution.Next, a detailed introduction to ChronoBridge will be provided.

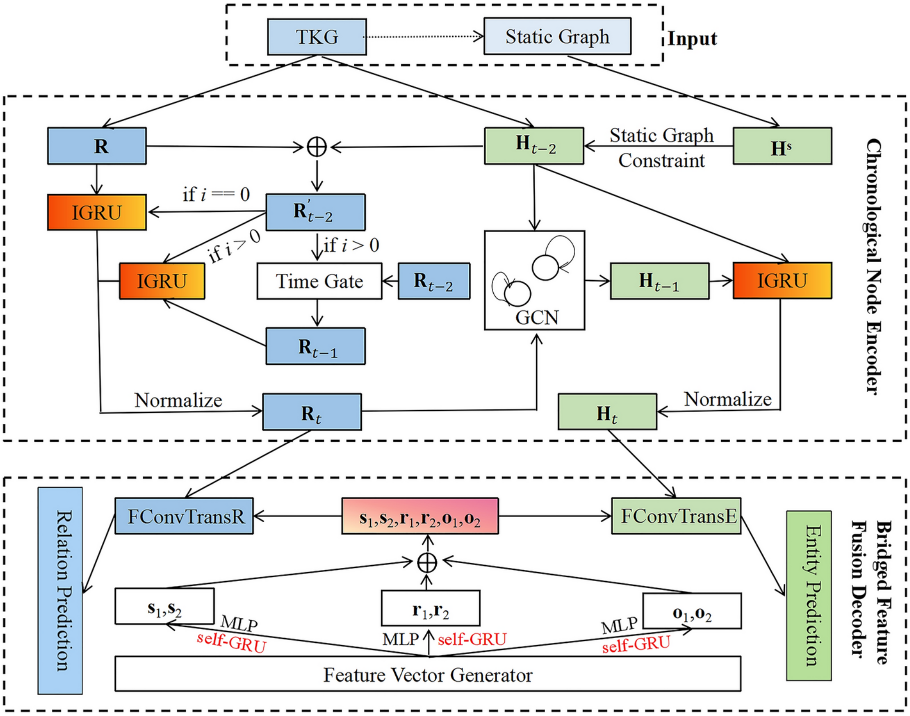


Fig. 4 Illustration of the proposed ChronoBridge model for time inference $t+1$ at timestamp

4.2 Chronological node encoder

4.2.1 Improved Gated Recurrent Unit (IGRU)

In TKG reasoning, the standard GRU is a robust model for handling dynamic changes. However, it has shortcomings such as limited capacity for long-term dependencies and high computational complexity (ArunKumar et al. 2022). To address these issues, the Improved Gated Recurrent Unit (IGRU) is proposed, incorporating the ELU activation function for better performance in TKG inference and simplifying the update and reset mechanisms to enhance long-range dependency capture. The IGRU architecture, depicted in Fig. 5, reduces computational complexity by replacing numerous multiplications with additions in each time step, resulting in more efficient inference for complex sequences.

The IGRU operates as follows:

a. Reset gate r_t:

$$r_t = \vartheta\left(f\left([x_t; h_{t-1}]\right)\right) \tag{1}$$

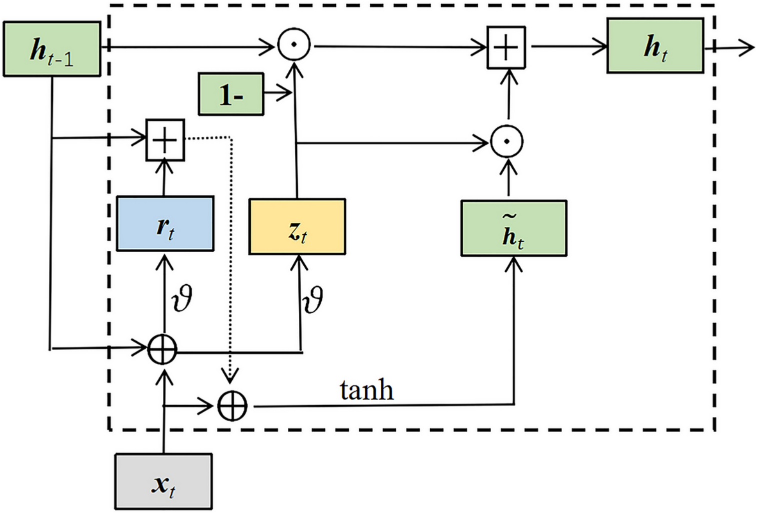


Fig. 5 IGRU model architecture diagram

where $\vartheta(\cdot)$ is a activation function, $f(\cdot)$ is a linear transformation, x_t is the input at the current time step *t*, h_{t-1} is the hidden state from the previous time step, and [;] denotes vector concatenation.

b. Update gate z_t :

$$z_t = \vartheta\left(f\left([x_t; h_{t-1}]\right)\right) \tag{2}$$

c. Candidate hidden state $\tilde{h}$:

$$\tilde{h} = \tanh\left(f\left([x_t; (r_t \odot h_{t-1})]\right)\right) \tag{3}$$

where $\odot$ is element-wise multiplication, $\tanh(\cdot)$ is hyperbolic tangent activation function, which outputs values between -1 and 1.

d. Final hidden state h_t :

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h} \tag{4}$$

4.2.2 Static attribute definition

In the context of TKG reasoning, static attributes provide a foundational framework for constructing static graphs, which are pivotal for encoding persistent relations among entities. These graphs are instrumental in supporting various reasoning tasks by offering a consistent structure that underpins the dynamic nature of TKGs (Li et al. 2021). To leverage the informative power of static attributes, their incorporation into the recurrent encoding scheme is proposed. For example, within the ICEWS dataset (Ferreira et al. 2021), static representations are derived from entity names, which typically adhere to a structured format like "Entity Type (Country)." A novel update mechanism for these static graphs is proposed, encapsulated by the following formula:

$$\vec{h}_i^{st} = \sigma\left(\sum_{(r^{st},j)\in\mathcal{E}^s} \frac{\mathbf{W}_{r^{st}}\vec{h}_j^{st}}{|\mathcal{N}(i)|}\right) \tag{5}$$

where, $\vec{h}_i^{st}$ denotes the static attribute embedding for entity *i*, $\mathbf{W}_{r^{st}}$ is the relation-specific transformation matrix for the static graph, and $\mathcal{E}^s$ represents the set of static edges connected to entity *i*. The normalization factor $|\mathcal{N}(i)|$ is the cardinality of the neighborhood set of entity *i*, ensuring that the influence of the entity's degree on its embedding is mitigated. The function $\sigma(\cdot)$ is an activation function, such as the sigmoid or hyperbolic tangent, which introduces non-linearity into the static attribute embeddings.

4.2.3 Encoding interdependencies among co-occurring events

Events occurring concurrently within a TKG often have intrinsic interdependencies that reflect complex relational structures. Capturing these interdependencies is crucial for a comprehensive understanding of the graph's dynamics. To this end, an innovative encoding strategy is proposed that leverages a multi-channel convolutional approach, extending beyond the capabilities of traditional GCN models like RE-GCN (Li et al. 2021). This approach allows for the simultaneous processing of multiple relations and the nuanced integration of their respective embeddings. The updated formula, which encapsulates the novel encoding strategy, is as follows:

$$\vec{h}_{o,t}^{l+1} = \sigma\left(\sum_{(s,r,o)\in\mathcal{F}_t} \mathbf{M}_r^l * \left(\vec{h}_{s,t}^l \circ \vec{h}_{r,t}\right)\right) \tag{6}$$

where, $\vec{h}_{o,t}^{l+1}$ signifies the next-level embedding for the target entity *o* at time *t*. The convolution operation $*$ is applied across the embeddings of source entities $\vec{h}_{s,t}^l$ and their corresponding relation embeddings $\vec{h}_{r,t}$, with a relation-specific convolutional filter $\mathbf{M}_r^l$ that adapts to the relational context. The set $\mathcal{F}_t$ represents the set of facts occurring at time *t*. The element-wise product $\circ$ ensures that the entity and relation embeddings are combined in a manner that preserves the unique characteristics of each relation. The activation function $\sigma(\cdot)$ is introduced to incorporate non-linearities into the learning process, enabling the model to capture complex patterns within the TKG.

4.2.4 Sequential event patterns between adjacent facts

To capture the trends and preferences of entities, historical facts at adjacent time points are considered by the model. This holistic view allows for a better understanding of behavioral evolution. In the case of non-initial cycles, to avoid over-smoothing and vanishing gradient problems, a time-gated recurrent component is applied to each relation. This component dynamically manages relation information across timestamps, maintaining distinct relation representations. The formulae for these operations are as follows:

First, the entity matrix $\mathbf{H}_{t-2}$ is combined with the static graph constraints and a randomization matrix $\mathbf{R}$:

$$\mathbf{R}'_{t-2} = [\mathbf{R}; \mathbf{H}_{t-2}] \tag{7}$$

Then, the relation matrix $\mathbf{R}_t$ at timestamp *t* is updated using IGRU and normalized:

$$\mathbf{R}_t = \mathrm{Normalize}\left(\mathrm{IGRU}\left(\mathbf{R}'_{t-2}, \mathbf{R}\right)\right) \tag{8}$$

For the time-gated component, the following is established:

$$\mathbf{R}_{t-1} = \mathbf{U}_t \otimes \mathbf{R}_{t-2} + (1 - \mathbf{U}_t) \otimes \mathbf{R}'_{t-2} \tag{9}$$

$$\mathbf{U}_t = \theta\left(\mathbf{W}_1 \mathbf{R}'_{t-2} + \mathbf{b}_1\right) \tag{10}$$

Finally, the entity matrix $\mathbf{H}_{t-2}$ and the relation matrix $\mathbf{R}_t$ are embedded into GCN, and the entity matrix $\mathbf{H}_t$ at time *t* is updated using IGRU and normalization:

$$\mathbf{H}_t = \mathrm{Normalize}\left(\mathrm{IGRU}\left(\mathbf{H}_{t-1}, \mathbf{H}_{t-2}\right)\right) \tag{11}$$

4.3 Bridged feature fusion decoder

4.3.1 Self-Gated Recurrent Unit (Self-GRU)

Within the ChronoBridge framework, it is crucial that not only the features of entities and their relations, including those of adjacent entities and relations, are captured, but also the intrinsic features of the focal entity are extracted. Entities and relations are treated as sequences, and to more accurately extract their features, the Self-Gated Recurrent Unit (Self-GRU) is proposed? as shown in Fig. 6. The Self-GRU utilizes the sequence itself both as the current input and as the output from the previous timestep, building upon the IGRU model. Compared to IGRU, the Self-GRU is streamlined, shedding unnecessary contextual information to fully exploit the sequence's inherent features. These enhancements enable the Self-GRU to extract more effective sequence features, proving highly effective in TKG reasoning tasks.

The data formula for the Self-GRU is as follows:

a. Reset gate r_t:

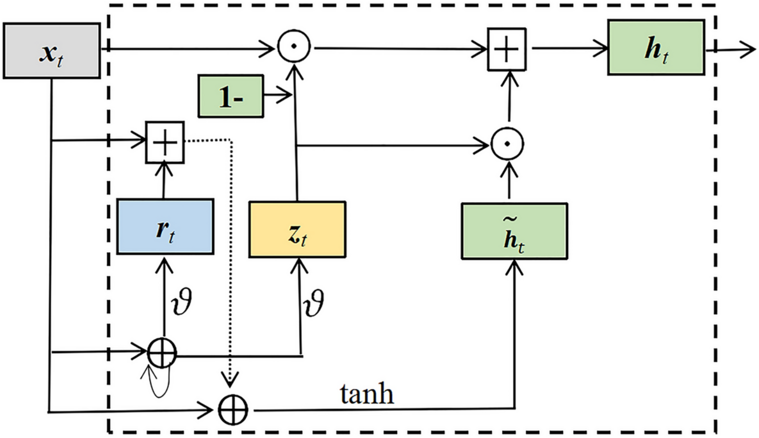


Fig. 6 Self-GRU model architecture diagram

$$r_t = \vartheta(f([x_t; x_t])) \tag{12}$$

where $\vartheta(\cdot)$ is the Exponential Linear Unit (ELU) activation function, $f(\cdot)$ is a linear transformation, and [;] denotes vector concatenation.

b. Update gate z_t:

$$z_t = \vartheta(f([x_t; x_t])) \tag{13}$$

The process for calculating z_t mirrors that of r_t.

c. Candidate activation $\tilde{h}$:

$$\tilde{h} = \tanh(f([x_t; (r_t \odot x_t)])) \tag{14}$$

where $\odot$ denotes element-wise multiplication.

d. Final output h_t:

$$h_t = (1 - z_t) \cdot x_t + z_t \cdot \tilde{h} \tag{15}$$

4.3.2 Extraction and fusion of entity and relation features

In TKG reasoning, Multi-Layer Perceptrons (MLPs) can be leveraged for feature extraction of entities and relations. The MLP is adept at learning non-linear feature transformations and managing complex combinations of features. With backpropagation, the MLP adapts its parameters, enabling the model to discern data patterns and relations autonomously. Consequently, MLPs are employed for feature extraction in TKGs, enhancing the modeling capabilities for entities and relations. Concurrently, the novel Self-GRU algorithm is applied to further refine feature extraction, aiming to boost inference accuracy. Specifically, the dataset is traversed, compiling all unique timestamped subject entities into matrix $\mathbf{s}$, relations into matrix $\mathbf{r}$, and object entities into matrix $\mathbf{o}$. These matrices are then processed by MLPs and Self-GRU to yield the final feature embeddings.

$$\mathbf{s}_1 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_1 \odot \mathbf{s} + \mathbf{c}_1))) \tag{16}$$

$$\mathbf{s}_2 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_2 \odot \mathbf{s} + \mathbf{c}_2))) \tag{17}$$

$$\mathbf{r}_1 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_3 \odot \mathbf{r} + \mathbf{c}_3))) \tag{18}$$

$$\mathbf{r}_2 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_4 \odot \mathbf{r} + \mathbf{c}_4))) \tag{19}$$

$$\mathbf{o}_1 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_5 \odot \mathbf{o} + \mathbf{c}_5))) \tag{20}$$

$$\mathbf{o}_2 = \text{Self-GRU}(\text{MLP}(f(\mathbf{U}_6 \odot \mathbf{o} + \mathbf{c}_6))) \tag{21}$$

where, $\odot$ signifies the Hadamard product, and $f(\cdot)$ extracts the upper triangular part of the matrix to eliminate redundant information, easing the model's computational load. Matrices $\mathbf{U}_1 \ldots \mathbf{U}_6$ and vectors $\mathbf{c}_1 \cdots \mathbf{c}_6$ are learnable parameters.Next, the features of entities and relations are fused. Concatenation not only combines different features into a comprehensive tensor but also, when dealing with sequential data, aligns feature tensors over time to construct the input sequence. Hence, entity and relation features are concatenated for fusion.

$$\mathbf{F} = [\mathbf{s}_1 \| \mathbf{s}_2 \| \mathbf{r}_1 \| \mathbf{r}_2 \| \mathbf{o}_1 \| \mathbf{o}_2] \tag{22}$$

where $\|$ denotes vector concatenation, and $\mathbf{F}$ is the fused feature matrix.

4.3.3 Building the feature fusion decoders

Having obtained the feature matrices for entities and relations, two feature fusion decoders, FConvTransE and FConvTransR, were devised to concurrently predict entities and relations. The decoders perform convolution operations on the composite embeddings-comprised of the entity embedding matrix $\mathbf{H}_t$, relation embedding matrix $\mathbf{R}_t$, and the fused feature matrix $\mathbf{F}$-and then split the resulting composite representation.

$$\mathbf{EM} = [\mathbf{F} \| \mathbf{H}_t \| \mathbf{R}_t] \tag{23}$$

The convolution operation is calculated as follows:

$$\text{conv}_c^n = \text{conv}_c(\mathbf{EM}, n) = \sum_{i=0}^{K-1} \mathbf{w}_c \cdot (\mathbf{EM}_i(n+i)) \tag{24}$$

where c is the number of convolution kernels, K is the size of the kernels, n ranges from 0 to $d-1$, d is the dimensionality of the output vector, and $\mathbf{w}_c$ are the learnable kernel weights.Post convolution, the final feature output for FConvTransE is:

$$\boldsymbol{\zeta}(\mathbf{EM}) = \mathrm{ReLU}\left(\mathrm{vector}(\mathbf{CONVF}_{\mathrm{conv}}) \cdot \mathbf{W}\right) \cdot \mathbf{H}_{o,t} \tag{25}$$

where, vector denotes a feature mapping operation, and $\mathbf{W} \in \mathbb{R}^{cd \times d}$ is a linear transformation matrix. FConvTransR computes scores similarly, substituting $\mathbf{R}_t$ with $\mathbf{H}_{o,t}$.

4.4 Score functions for entity and relation prediction tasks

In the ChronoBridge framework, the intricacies of temporal and relational data are modeled using Graph Convolutional Networks (GCNs), which are integrated into our decoder design to improve entity and relation predictions in TKGs.

For the entity prediction task, the FConvTransE decoder integrates a one-dimensional convolutional layer with a subsequent dense layer to process the features. Similarly, the FConvTransR decoder is structured for the relation prediction task. The score functions FConvTransE(.) and FConvTransR(.) correspond to these two-layered structures, respectively. The probability vectors representing all possible entities and relations are computed as follows:

For entities:

$$\vec{p_e} = \sigma\left(\mathbf{H}_t \cdot \mathrm{FConvTransE}\left(\vec{s_t}, \vec{r_t}, \mathbf{F}\right)\right) \tag{26}$$

For relations:

$$\vec{p_r} = \sigma\left(\mathbf{H}_t \cdot \mathrm{FConvTransR}\left(\vec{s_t}, \vec{o_t}, \mathbf{F}\right)\right) \tag{27}$$

where, $\sigma(\cdot)$ denotes the sigmoid activation function, which maps the output to a probability distribution over entities and relations. The vectors $\vec{s}_t$, $\vec{r}_t$, and $\vec{o_t}$ represent the embeddings for subjects, relations, and objects at time t, extracted from the temporal entity embeddings $\mathbf{H}_t$ and temporal relation embeddings $\mathbf{R}_t$. Both FConvTransE(.) and FConvTransR(.) produce embeddings of dimensionality $\mathbb{R}^{1 \times d}$, where d is the embedding size.By leveraging these score functions, the model can predict the likelihood of a given entity or relation being the correct completion of a TKG triplet at a specific timestamp. This approach allows for a nuanced understanding of temporal dynamics in TKGs, providing a robust mechanism for reasoning over time-sensitive information.

4.5 Loss function

The ChronoBridge framework proposes innovative methodologies for predicting entities and relations by treating them as multi-label classification tasks with a temporal dimension. The vectors $\overrightarrow{x^e_{\mathrm{t+1}}} \in \mathrm{R}^{|V|}$ and $\overrightarrow{x^r_{\mathrm{t+1}}} \in \mathrm{R}^{|R|}$ represent the multi-label aspect of the tasks at timestamp $t+1$, where elements are assigned a value of 1 for verified facts and 0 otherwise. The innovation in the loss function is its capacity to encompass both the temporal dynamics and the complexity of multi-label scenarios.

The loss for entity prediction, $\mathcal{L}_{entity}$, is reformulated to mirror the log probabilities of the accurate entities, given a source entity s, a relation r, and the historical embeddings up to time t, across all entities in the vocabulary:

$$\mathcal{L}_{entity} = \sum_{t=0}^{T-1} \sum_{\forall(s,r,o,t+1)\in\mathcal{E}_{t+1}} \sum_{i=0}^{|V|-1} \overrightarrow{x_{t+1,i}^{e}} \cdot \log \mathcal{P}_i\left(o|s, r, \mathbf{R}_t, \mathbf{H}_t, \mathbf{F}\right) \tag{28}$$

In a similar vein, the loss for relation prediction, $\mathcal{L}_{relation}$, accounts for the log probabilities of the correct relations, given the source and target entities as well as the historical backdrop:

$$\mathcal{L}_{relation} = \sum_{t=0}^{T-1} \sum_{\forall(s,r,o,t+1)\in\mathcal{E}_{t+1}} \sum_{i=0}^{|R|-1} \overrightarrow{x_{t+1,i}^{r}} \cdot \log \mathcal{P}_i\left(r|s, o, \mathbf{R}_t, \mathbf{H}_t, \mathbf{F}\right) \tag{29}$$

where,$\mathcal{P}_i\left(o|s, r, \mathbf{R}_t, \mathbf{H}_t, \mathbf{F}\right)$: The predicted probability of the target entity o, given the source entity s, relation r, the embeddings of relations up to time t ($\mathbf{R}_t$), historical embeddings ($\mathbf{H}_t$), and additional features ($\mathbf{F}$). $\mathcal{P}_i\left(r|s, o, \mathbf{R}_t, \mathbf{H}_t, \mathbf{F}\right)$: The predicted probability of the relation r, given the source entity s, target entity o, the embeddings of relations up to time t ($\mathbf{R}_t$), historical embeddings ($\mathbf{H}_t$), and additional features ($\mathbf{F}$).$\mathcal{E}_{t+1}$: The set of confirmed facts at time $t+1$, each fact represented as a tuple $(s, r, o, t+1)$, where s is the source entity, r is the relation, and o is the target entity.The objective of the loss functions is to minimize the prediction errors for entities and relations while considering the dynamics over time and the intricacies of a multi-label context, thus enabling the model to account for the characteristics of temporal sequence data in its predictions.

This paper proposes a new mathematical formulation for the static graph constraint loss, aimed at penalizing large deviations in the embedding space while ensuring temporal consistency of the embeddings. The approach involves introducing a time-weighted loss function that penalizes the Euclidean distance between embeddings, allowing for a natural evolution of the embeddings over time. The formula is as follows:

$$\mathcal{L}_{static} = \sum_{y=0}^{m} \sum_{i=0}^{|V|-1} \omega(y) \cdot \left\| \overrightarrow{h_i^s} - \overrightarrow{h_{t-m+y,i}} \right\|_2^2 \tag{30}$$

where, $\|\cdot\|_2$ denotes the Euclidean (L2) norm, and $\omega(y)$ is a time-decay function that can be defined to control how much change is allowed over time. For instance, $\omega(y)$ could be a decreasing function such as an exponential decay:

$$\omega(y) = e^{-\alpha y} \tag{31}$$

where, α is a hyperparameter that determines the rate of decay, allowing for more change in the embeddings as time progresses. This alternative loss function still captures the essence of penalizing large deviations in the embedding space, but does so in a way that is more aligned with traditional regularization techniques in machine learning.The total loss function, $\mathcal{L}_{total}$, as defined by the equation, represents an innovative approach to combining different aspects of a learning task. Here's an in-depth explanation of each component:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{entity} \cdot \exp(-\alpha \mathcal{L}_{relation}) + \lambda_2 \frac{\mathcal{L}_{relation}}{1 + \beta \mathcal{L}_{static}} + \lambda_3 \sin(\gamma \mathcal{L}_{static}) \tag{32}$$

The first term, $\lambda_1 \mathcal{L}_{entity} \cdot \exp(-\alpha \mathcal{L}_{relation})$, captures the entity loss, $\mathcal{L}_{entity}$, modulated by an exponential function of the relation loss, $\mathcal{L}_{relation}$. The weight λ_1 is a hyperparameter that determines the overall impact of the entity loss on the total loss. The exponential term, $\exp(-\alpha \mathcal{L}_{relation})$, serves to dynamically scale down the entity loss as the relation loss increases, with α controlling the sensitivity of this attenuation. This means that when the model is more confident about the relations within the data (i.e., $\mathcal{L}_{relation}$ is low), the entity loss has a stronger influence on the total loss.The second term, $\lambda_2 \frac{\mathcal{L}_{relation}}{1+\beta \mathcal{L}_{static}}$, represents the relation loss, $\mathcal{L}_{relation}$, adjusted by the static loss, $\mathcal{L}_{static}$. The hyperparameter λ_2 sets the weight of this term within the total loss. The relation loss is divided by a term that increases with the static loss, moderated by the hyperparameter β. This creates a conditional dependency where the influence of the relation loss on the total loss is inversely related to the magnitude of the static loss. In other words, when the static loss is high, the contribution of the relation loss is scaled down, which could be useful if the static features' correctness implies less concern about the relations.

The third term, $\lambda_3 \sin(\gamma \mathcal{L}_{static})$, incorporates a sinusoidal modulation of the static loss, $\mathcal{L}_{static}$. The amplitude of this term in the total loss is determined by the hyperparameter λ_3. The sinusoidal function proposes a periodic characteristic to the impact of the static loss, with γ adjusting the frequency of this periodicity. This term allows the influence of the static loss on the total loss to vary in a non-linear and potentially cyclic manner, which could be beneficial for capturing patterns that have a periodic nature in the static component of the data.

In this loss function, the hyperparameters λ_1, λ_2, and λ_3 are crucial for balancing the different components according to their importance to the specific learning task. Adjusting these weights allows the model to prioritize different aspects of the data during training. The use of non-linear and conditional terms in the loss function is designed to capture complex dependencies and interactions between different types of losses, which could lead to more robust learning in complex domains. However, this complexity also requires careful tuning of the hyperparameters to ensure that the model does not overfit and can generalize well to unseen data.

The innovation in the logical flow of the loss function is the integration of temporal reasoning within a multi-task learning framework, which is particularly challenging due to the dynamic nature of TKGs. The proposed loss functions aim to cohesively capture the temporal evolution of entities and relations while respecting the inherent structure of the graph. This approach is expected to enhance the predictive performance of the Chrono-Bridge framework by effectively leveraging temporal information and graph topology to inform the learning process.

4.6 Computational complexity analysis

The ChronoBridge framework's computational complexity is determined by the interplay of its chronological node encoder and bridged feature fusion decoder components. The encoder utilizes an Improved Gated Recurrent Unit (IGRU) with a per-timestep complexity of $O(d_x d_h + 2d_h^2 + d_h)$, aggregating to $O(T(d_x d_h + 2d_h^2 + d_h))$ over a sequence of length T. The decoder involves a Self-Gated Recurrent Unit (Self-GRU) and Multi-Layer Perceptrons (MLPs), with complexity depending on the layer sizes and feature dimensions. The loss function adds complexity based on the number of entities $|V|$, relations $|R|$, edges $|E|$, and timestamps T. The overall framework complexity can be approximated as $O\left(T(d_x d_h + 2d_h^2 + d_h) + ld_{in}d_{out} + d_{features} + T|V||E| + T|V||R| + m|V|d_h\right)$, with the terms involving $|V|$, $|R|$, and T being potentially dominant due to their multiplicative nature. Practical optimizations may reduce actual computational load, but the theoretical complexity highlights scalability and potential performance constraints of the framework.

4.7 Algorithm steps

The complete detailed algorithm with all steps is shown below:

Method: The Procedure of ChronoBridge method

Input: Train set G_{train} , static graph G^s and KG at timestamp t in the TKG_t, Epoch q

Output: The minimum loss $\mathcal{L}_{min}$,$eScore$,$rScore$

```
W1, U1, U2, U3, U4, U5, U6 ← normal(), b1, c1, c2, c3, c4, c5, c6 ← init.zeros()
for epoch ← 1 to q do
    L_end ← 0.0, L_entity ← 0.0, L_relation ← 0.0, L_static ← 0.0
    for (s, r, o, t) G_train do
        split (s, r, o, t) into two tasks for query s, r, ?, t + 1) and (s, ?, o, t + 1)
        Randomly initialize the matrix H^s and R
        for each task in (s, r, o, t)
            Build chronological node encoder by Equation(1) to (11)
            Build Feature fusion decoders by Equation(26) to (28)
        Calculate the decoding score eScore and rScore of the entities and relation
        by Equation(26)(27), respectively
        Calculate L_entity, L_relation, L_static and L_end by Equation(28) to (32)
    L_min = min(L_min, L_end)
    Update parameters with Adam optimizer[34] and the cosine annealing algorithm[35]
return L_min, eScore, rScore
```

5 Experiments

In this section, we comprehensively evaluate the ChronoBridge framework, detailing datasets, metrics, baseline comparisons, and implementation nuances, culminating in a multifaceted analysis that includes entity and relation prediction performance, computational efficiency, and robustness across various scenarios and datasets, including a practical case study on taxi fare prediction and processing the latest ICEWS 2024 dataset.

Table 1 Summary of TKG datasets

Datasets	$\|Entities\|$	$\|Predicates\|$	$\|Train\|$	$\|Validation\|$	$\|Test\|$	Hourliness
ICEWS14s	6869	230	74,845	8514	7371	24
ICEWS18	23,033	256	373,018	45,955	49,545	24
ICEWS05-15	10,094	251	368,868	46,302	46,159	24

5.1 Datasets

To accurately assess the performance of the ChronoBridge model, we evaluated our method on three public datasets, all sourced from the popular TKG resource ICEWS[32]. Table 1 summarizes the basic statistics for these three datasets, and the details for each dataset are provided below:

- **ICEWS 14:** This subset of the ICEWS resource, released by TA-DistMult (Zhu et al. 2023), covers a short time span from January 1, 2014, to December 31, 2014, with daily granularity. It includes 7,128 distinct entities and 230 types of relations.
- **ICEWS 05-15:** This is a long-range subset of the ICEWS resource, also released by TA-DistMult, and is nearly five times larger than ICEWS 14. It encompasses data from January 1, 2005, to December 31, 2015, with daily granularity, featuring 10,488 distinct entities and 251 types of relations.
- **ICEWS 18:** Another short-range subset of the ICEWS resource, this dataset, released by xERTE (Han et al. 2020), covers the year 2018 with daily granularity. It contains 23,033 distinct entities and 256 types of relations.

5.2 Evaluation metrics

In TKG reasoning, the Mean Reciprocal Rank (MRR) and Hits at ranks 1, 3, and 10 are critical metrics for assessing the accuracy of entity and relation predictions, as detailed in Li et al. (2021). The MRR is the average of the inverse ranks at which correct answers are found across all queries, with a higher score indicating a model that more frequently predicts correct answers at higher ranks. Hits at K, with K being either 1, 3, or 10, represents the proportion of queries for which the model's prediction is ranked within the top K positions, serving as a direct measure of the model's precision in ranking correct answers at the very top.

In the experiment, we adopt MRR and Hits@1, 3, 10 as the evaluation metrics. The MRR and Hits@K are defined as follows:

$$\mathrm{MRR} = \frac{1}{|G|} \sum_{(s,r,o,t)\in G} \frac{1}{rank(s,r,o,t)} \tag{33}$$

$$\mathrm{Hits@}K = \frac{1}{|G|} \sum_{(s,r,o,t)\in G} int(rank(s,r,o,t) \leq K) \tag{34}$$

Where *rank*(s, r, o, t) refers to the rank of the correct quadruple (s, r, o, t) in the prediction table, $K = 1, 3, 10$, *int*(.) is the indicator function.To validate ChronoBridge's effectiveness, it is contrasted with both static and TKG representation learning techniques. For static KG reasoning, the well-known RotatE (Sun et al. 2019) is opted for. In the realm of TKG inference, for interpolation scenarios, the established TA-DistMult (Zhu et al. 2023) and HyTE (Dasgupta et al. 2018) are selected, while for extrapolation contexts, the notable RE-GCN (Li et al. 2021) along with the cutting-edge TiRGN (Li et al. 2022b) and RETIA (Liu et al. 2023) are picked.

5.3 Implementation details

In these experiments, the ChronoBridge framework was meticulously configured to optimize reasoning over TKG. A hidden state dimension of 256 was set for the Improved Gated Recurrent Unit (IGRU) within the chronological node encoder for mid-sized datasets, leveraging the ELU activation function and initialized using the Xavier method. The learning rate was fixed at 0.001, with a decay factor of 0.96 being applied every 10,000 steps. Static attribute embeddings were uniformly randomized between [− 0.01, 0.01], and the relation-specific transformation matrix was initialized via the He method, with non-linearity proposed by the sigmoid function. Convolutional channels were matched to the count of unique relations, utilizing filters that were initialized using the He method and ReLU for activation. Hidden states within the bridged feature fusion decoder's Self-Gated Recurrent Unit (Self-GRU) were initialized between [− 0.05, 0.05], employing the ELU activation function, with a learning rate matching that of the encoder. Feature extraction was carried out by MLPs with two layers of 512 neurons each, and Hadamard product matrices were initialized between [− 0.1, 0.1], with bias vectors set to zero. FConvTransE and FConvTransR produced 128-dimensional embeddings, with scores mapped to probabilities using the sigmoid function. The loss functions were dictated by hyperparameters with $\alpha = 0.1,\ \beta = 0.01$, and $\gamma = 0.001$, featuring component weights of $\lambda_1 = 1$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.1$. The time-weighted loss function was set to decay exponentially at a rate of $\alpha = 0.01$. Implementation of the framework was done in TensorFlow 2.x, utilizing the Adam optimizer with a batch size of 128 and incorporating early stopping after 10 epochs to counter overfitting. Dropout was applied at a rate of 0.2 to the MLP layers, with the training being limited to a maximum of 100 epochs. Optimal settings, including the historical lengths for various ICEWS datasets and the learning rate adjustments via the cosine annealing algorithm, were determined through grid search and manual tuning. For FConvTransE and FConvTransR, 50 convolution kernels of size 2×3 with a dropout rate of 0.2 were utilized. The hardware setup included a Precision 7920 Tower with 128 GB RAM and an Nvidia TITAN RTX graphics card, ensuring robust computational support for the experiments.

Table 2 Performance (in percentage) for the entity prediction task on ICESW14s and ICEWS05-15 with raw metrics

Models	ICESW14s				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RotatE [2019]	25.71	16.41	29.01	45.16	19.01	10.42	21.35	36.92
HyTE [2018]	16.78	2.13	24.84	43.94	16.05	6.53	20.20	34.72
TA-DistMult [2018]	26.22	16.83	29.72	45.83	27.51	17.57	31.46	47.32
RE-GCN [2021]	41.31	30.31	46.81	62.26	46.41	35.17	52.76	67.64
TiRGN [2022]	43.88	33.12	49.48	64.98	48.72	37.17	55.48	70.53
RETIA [2023]	<u>45.29</u>	<u>34.60</u>	<u>50.88</u>	<u>66.06</u>	<u>52.17</u>	<u>40.21</u>	<u>59.42</u>	<u>73.98</u>
ChronoBridge	**52.01**	**42.63**	**57.00**	**69.83**	**71.62**	**65.13**	**75.71**	**83.33**

The best scores are marked in bold, and the second-best scores are underscored to highlight their significance

Table 3 Performance (in percentage) for the entity prediction task on ICEWS18 with raw metrics

Models	MRR	Hits@1	Hits@3	Hits@10
RotatE [2019]	14.53	6.47	15.78	31.86
HyTE [2018]	7.41	3.10	7.33	16.01
TA-DistMult [2018]	16.42	8.60	18.13	32.51
RE-GCN [2021]	30.55	20.00	34.73	51.46
TiRGN [2022]	32.06	21.08	6.75	53.62
RETIA [2023]	<u>34.16</u>	<u>22.97</u>	<u>39.27</u>	<u>55.96</u>
ChronoBridge	**36.74**	**24.89**	**41.56**	**57.29**

The best scores are marked in bold, and the second-best scores are underscored to highlight their significance

Table 4 Performance (in percentage) for the relation prediction task on ICESW14s, ICEWS05-15 and ICEWS18 with raw metrics

Models	ICEWS14s	ICEWS05-15	ICEWS18
RE-GCN [2021]	41.06	40.63	40.53
TiRGN [2022]	41.80	42.12	42.18
RETIA [2023]	<u>47.63</u>	<u>46.70</u>	<u>46.59</u>
ChronoBridge	**63.78**	**80.01**	**48.27**

The best scores are marked in bold, and the second-best scores are underscored to highlight their significance

5.4 Performance comparison for entity prediction

Tables 2 and 3 highlights the comparative performance of various methods in the entity prediction task, with the best scores marked in bold and the second-best underscored. Several key insights emerge from these results:

ChronoBridge significantly outperforms baseline models, taking the lead in the MRR metric across all datasets. It shows notable improvements over the conventional RE-GCN model, with MRR enhancements of 8.79%, 25.21%, and 6.19% on each dataset respectively, particularly excelling with the ICEWS05-15 dataset. This dataset's time-stamped events and comprehensive facts facilitate the capture of continuous event relations and the structural intricacies of knowledge graphs, aiding in the effective extraction of entity and relation features. While improvements on ICEWS18 are evident, they are less pronounced compared to ICEWS05-15, likely due to the latter's broader temporal range, encompassing 15 years of event data versus one year in ICEWS18, providing a richer source for analysis.

ChronoBridge significantly outperforms both classic models like RE-GCN and cutting-edge models such as TiRGN and RETIA. It not only accounts for the influence of historical fact features on future predictions but also integrates an innovative component, the IGRU, for historical predictions. Its autoregressive modeling of KG sequences proves beneficial in identifying patterns across all facts. The newly proposed self-GRU plays a crucial role in extracting and fusing entity and relation features, which are then leveraged in the decoder for inference, fostering a comprehensive understanding of the facts and significantly enhancing prediction accuracy through deep mining and reasoning.

5.5 Performance comparison for relation prediction

Table 4 in the study presents a comparative analysis of various models on the relation prediction task within TKGs, with ChronoBridge achieving the best performance metrics, notably in the Mean Reciprocal Rank (MRR). The table highlights the model's outcomes in bold for the best performance and underlines the second-best, emphasizing the stark contrast with other models such as RE-GCN, TiRGN, and RETIA. Some models, like RE-NET and CyGNet, are deemed unsuitable for this task and are therefore excluded from the comparison. The improvements of ChronoBridge are particularly noteworthy against the RE-GCN model, with MRR gains of 22.72%, 39.38%, and 7.74% across various datasets.

The detailed analysis of why ChronoBridge outperforms others can be broken down into several key innovations and compatibilities with the ICEWS05-15 dataset:

(1) Temporal Encoding: ChronoBridge's advanced temporal encoding techniques are adept at capturing the dynamic nature of TKGs, understanding historical patterns, and predicting future trends, which are essential for relation prediction.
(2) Feature Fusion: The bridged feature fusion decoder in ChronoBridge goes beyond traditional models by integrating temporal and relational features, creating a richer representation that more accurately captures the complexities of temporal relations.
(3) Self-GRU Algorithm: The innovative self-GRU algorithm is tailored for TKGs, outperforming traditional RNNs in maintaining sophisticated temporal states of entities and relations, which is vital for accurate predictions.
(4) Decoding Strategy: ChronoBridge's decoding strategy avoids the information loss typical in other models during the decoding phase, ensuring that the rich features extracted and fused earlier are effectively utilized for more informed inferences, as seen in the improved MRR scores.
(5) Dataset Compatibility: The ICEWS05-15 dataset's extensive event data and rich relational information provide an ideal testing ground for ChronoBridge's capabilities. The model's architecture is well-suited to leverage such datasets, contributing to its significant performance improvement.
(6) Comparative Performance: ChronoBridge's substantial improvements in MRR over classic models like RE-GCN and newer benchmarks like TiRGN and RETIA highlight its enhanced ability to capture temporal dynamics and provide a more robust representation of temporal knowledge.In summary, the superior performance of ChronoBridge as shown in Table 4 is attributed to its sophisticated temporal encoding, effective feature fusion, the introduction of a self-GRU algorithm, and a decoding strategy that ensures the comprehensive utilization of temporal and relational features. These innovations result in ChronoBridge's distinct advantage for predicting relations within TKGs, leading to its outstanding results across all datasets, with particularly pronounced improvements on the ICEWS05-15 dataset. Unlike most methods that focus solely on historical event features, ChronoBridge addresses the gap by thoroughly investigating the characteristics of entities and relations from various angles and integrating these features into the decoding process, which is more conducive to generating accurate inferences and significantly enhancing prediction accuracy.

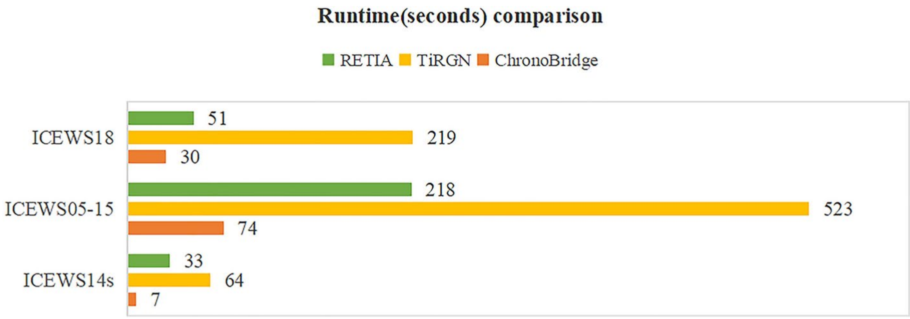


Fig. 7 Comparison chart of prediction time

5.6 Comparison of prediction time

This section focuses on the eficiency of the ChronoBridge model by comparing theprediction times, The aim is to assess how quickly the model can deliver predictionsin practical scenarios.

The exceptional time efficiency of the ChronoBridge model, as depicted in Fig. 7, can be attributed to its innovative design which streamlines the feature extraction process. Unlike the TiRGN model, which requires a comprehensive analysis of historical data to identify recurring entities and relations, ChronoBridge employs an optimized Incremental Gated Recurrent Unit (IGRU) that facilitates autoregressive modeling of historical KG sequences more effectively. This allows ChronoBridge to rapidly and accurately capture the dynamics of entities and relations without the exhaustive data processing that TiRGN entails. Furthermore, while the RETIA model's predictive accuracy benefits from its sophisticated hypergraph algorithm, this complexity also translates into greater computational demands, reducing its time efficiency. In contrast, ChronoBridge's self-GRU algorithm simplifies the feature extraction while maintaining high predictive performance, significantly reducing the overall time complexity and enhancing the model's operational efficiency. This strategic balance between complexity and efficiency is what enables ChronoBridge to maintain a competitive edge in both speed and accuracy within the realm of TKG reasoning.

5.7 Ablation study

To minimize discrepancies between training and testing and gain clearer insights into the impact of different components of the model, an ablation analysis was conducted.In these experiments, we systematically removed or replaced key components of the model to evaluate their impact on overall performance. Specifically, we compared the performance differences between IGRU and self-GRU with propagation GRU, and analyzed how each of these algorithms, both individually and collectively, contributes to the model's performance.Next, we will outline the design of our ablation study.

Our ablation study follows these steps:

(1) **Baseline model (GRU Only)**: Use the standard GRU as the encoder and decoder, referred to as GRU-Baseline.

Table 5 Ablation studies on entity prediction

models	ICEWS14s	ICEWS05-15	ICEWS18
GRU-Baseline	41.31	46.41	30.55
IGRU-Enhanced	42.01	47.32	31.56
self-GRU-Enhanced	42.35	46.93	31.95
BFFD-Added	46.57	64.12	33.87
ChronoBridge-Complete	**48.06**	**68.46**	**35.28**

The best scores are marked in bold

Table 6 Ablation studies on relation prediction

models	ICEWS14s	ICEWS05-15	ICEWS18
GRU-Baseline	41.06	40.63	40.53
IGRU-Enhanced	42.23	43.69	42.21
self-GRU-Enhanced	42.62	43.52	41.79
BFFD-Added	55.89	72.24	46.08
ChronoBridge-Complete	**58.23**	**75.29**	**48.23**

The best scores are marked in bold

(2) **IGRU replacing GRU**: Replace the standard GRU with IGRU while keeping all other model components unchanged, referred to as IGRU-Enhanced.
(3) **self-GRU replacing GRU**: Replace the standard GRU with self-GRU while keeping all other model components unchanged, referred to as self-GRU-Enhanced.
(4) **Bridged feature fusion decoder**: Add our bridged feature fusion decoder to the standard GRU model without using IGRU and self-GRU, referred to as BFFD-Added (bridged feature fusion decoder).
(5) **Complete ChronoBridge model**: The full model that includes IGRU, self-GRU, and the bridged feature fusion decoder, referred to as ChronoBridge-Complete.Each step is tested on the same dataset to ensure comparability of results.The results of the experiments are presented in Tables 4 and 5, meticulously isolates the contribution of each component of the ChronoBridge model to the overall performance.Next, we will analyze the ablation studies for entity prediction and relation prediction separately.

(1) Entity prediction ablation study analysisIn the ablation study for entity prediction, we observed a step-by-step improvement from GRU-Baseline to ChronoBridge-Complete as evidenced in Table 5. The IGRU-Enhanced model showed a slight improvement over the GRU-Baseline across all datasets, which can be attributed to IGRU's enhanced handling of temporal sequence data, capturing the changes in entity states over time. In addition, the self-GRU-Enhanced model performed slightly better than the IGRU-Enhanced model, indicating the key role of the self-attention mechanism in the model, helping it to capture complex, time-spanning dependencies between entities.

With the addition of the bridged feature fusion decoder (BFFD-Added), we saw a significant leap in performance on the ICEWS05-15 dataset, suggesting that the feature fusion decoder is particularly effective when dealing with large amounts of historical information and data spanning longer time frames. This is likely because the FFD is better at integrating features from different time points, thus providing the model with a richer context for predicting the current state of an entity. Ultimately, the ChronoBridge-Complete model achieved the highest performance across all datasets, reflecting the synergistic effect of IGRU, self-GRU, and the feature fusion decoder, complementing each other to provide a robust framework for comprehensively capturing and utilizing dynamic information in TKGs.

(2) Relation prediction ablation study analysisAs shown in Table 6, the ablation study results for relation prediction showed a similar trend, but with more significant improvements across different models. The enhancements seen with the IGRU-Enhanced and self-GRU-Enhanced models in relation prediction underscore the importance of temporal awareness and capturing long-distance dependencies, although these effects were not as pronounced as in entity prediction. This may be due to the inherently more complex nature of relation prediction, which involves more interactions between entities.

The significant improvement with the bridged feature fusion decoder (BFFD-Added) in relation prediction, especially on large-scale datasets like ICEWS05-15, suggests that it provides an effective means of integrating complex interaction features between entities. This indicates that FFD is crucial for understanding and predicting how entities interact with each other at different time points. The exceptional performance of the ChronoBridge-Complete model further proves the complementarity of the model components, combining the temporal awareness of IGRU, the long-distance dependency capture of self-GRU, and the feature fusion capability of FFD, to provide a comprehensive and powerful framework for relation prediction.

Overall, these ablation study results emphasize the importance of effectively integrating temporal sequence information, capturing long-term dependencies, and fusing complex features when designing models for TKG reasoning. The innovations in the ChronoBridge model have set a new standard for performance, offering valuable insights and potential directions for future research.

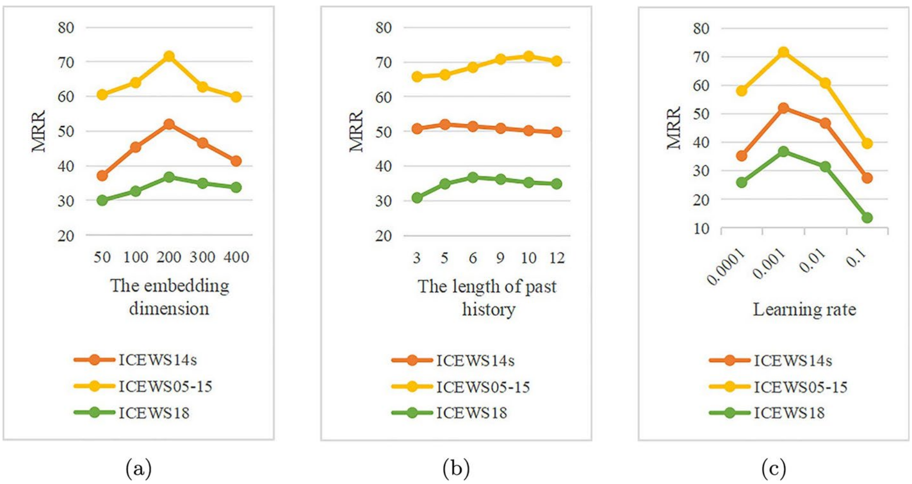


Fig. 8 Analysis results of sensitive parameters for entity prediction

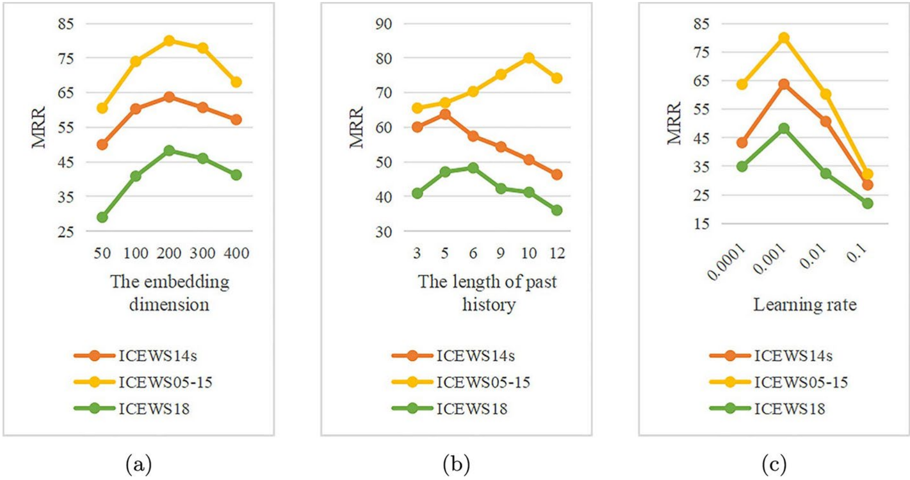


Fig. 9 Analysis results of sensitive parameters for relation prediction

5.8 Sensitivity analysis

To understand how ChronoBridge adapts to various dataset categories, the responsiveness of its critical hyperparameters is examined. By adjusting these parameters, the performance changes of ChronoBridge on three datasets on ICEWS are analyzed.

It is observed from Figs. 8a to 9a that when the embedding dimension of the recurrent encoder is set to 200, the MRR of the three datasets reaches the optimal value. Should the embedding dimension become too large, improvements in MRR are no longer seen; similarly, a too small embedding dimension also negatively impacts MRR. This suggests that embedding dimensions either too large or too small may affect the dependencies between events, thus affecting the training efficacy of the model.

According to Figs. 8b and 9b, the optimal history lengths for the three datasets ICEWS14s, ICEWS05-15, and ICEWS18 are identified to be 5, 10, and 6, respectively. An increase in event history length leads to better model performance. However, exceeding a certain history length threshold results in a decrease in MRR, and consequently, a decline in model performance. This reflects that the model's performance is highly dependent on the duration of event history, highlighting the importance of selecting an appropriate length for historical events to accurately capture the temporal relations among them.

Ultimately, as depicted in Figs. 8c and 9c, it is apparent that when the learning rate is 0.001, the MRR of the model reaches the optimal value. A higher learning rate accelerates parameter adjustments, yet it might cause fluctuations or a lack of stability; conversely, a lower learning rate results in more gradual updates to parameters. Hence, selecting a suitable learning rate is essential for sustaining enhancements in predictive precision. In addition, adjusting the learning rate can also avoid the common gradient explosion or gradient disappearance problems in deep learning, thereby improving the stability and accuracy of training. In summary, the selection of optimal hyperparameters is key not only for bolstering the stability and precision of the model's training process but also for hastening its convergence and augmenting its capacity to generalize.

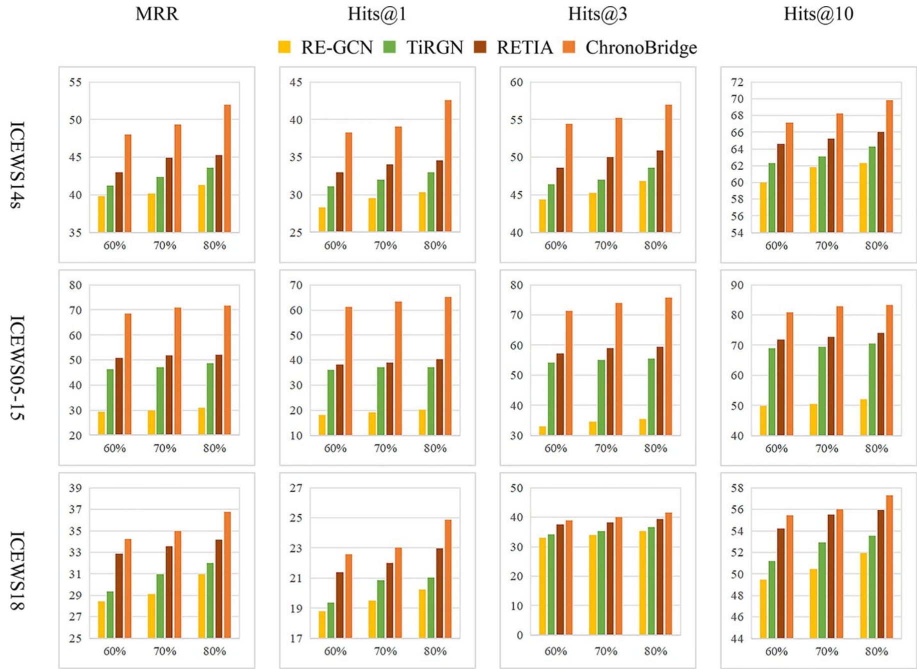


Fig. 10 Performance analysis of different algorithm with training data variation for entity prediction task on ICESW14s, ICEWS05-15 and ICEWS18 with raw metrics

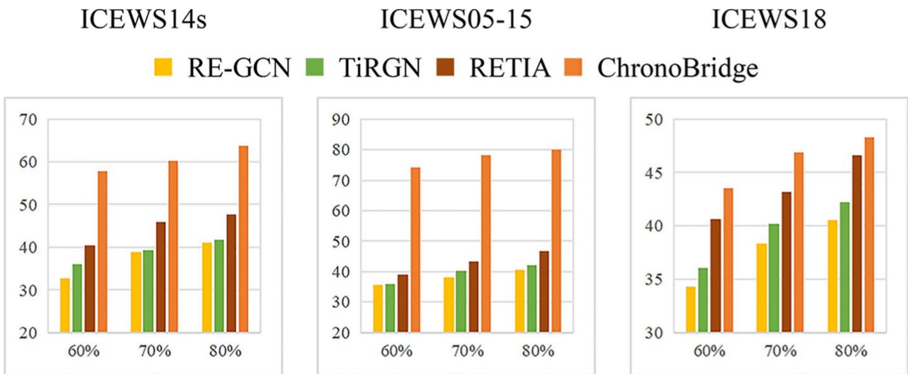


Fig. 11 Performance analysis of different algorithm with training data variation for relation prediction task on ICESW14s, ICEWS05-15 and ICEWS18 with raw metrics

5.9 Impact of training ratio

To thoroughly investigate how the proportion of training data influences the efficacy of the ChronoBridge framework, a range of experiments were performed, adjusting the quantity of training data in relation to the entire dataset. This examination aids in comprehending how the model's performance fluctuates with varying volumes of training data and offers insights for the distribution of data resources in real-world scenarios.

In the experiments, the training data ratio was set to 0.6, 0.7, 0.8, and 0.9, and the performance of the ChronoBridge framework was compared with other algorithms under different training ratios. The most typical RE-GCN algorithm and the most advanced TiRGN and RETIA models were used for comparison.

Figures 10 and 11 illustrate that an incremental rise in the share of training data corresponds with an enhancement in the performance of each algorithm. However, ChronoBridge consistently outperforms other algorithms at all training rates and reaches optimal performance at a training rate of 80%. These results carry significant weight in ascertaining the ideal quantity of training data necessary for attaining desirable outcomes and further aid in augmenting the model's capacity to learn and generalize.

In real-world applications, the quantity and quality of training data are critical factors that significantly affect a model's performance. A scarcity of training data can hinder the model's ability to fully capture the characteristics of the data, resulting in inferior performance. Conversely, an excess of training data can bring in noise and superfluous details, which might compromise the model's ability to generalize. Hence, selecting the right amount of training data is essential for various datasets and tasks.

Based on the aforementioned analysis, the subsequent conclusions can be deduced: The ratio of training data plays a crucial role in determining the performance of a model. As the proportion of training data increases, the performance of each algorithm improves. The ChronoBridge framework always outperforms other algorithms in entity prediction and relation prediction. Under different training ratios, ChronoBridge's performance is better than the most typical RE-GCN algorithm and the most advanced TiRGN and RETIA models.

The ChronoBridge framework has good stability and generalization capabilities. Under different training ratios, the performance difference of ChronoBridge is small, indicating that the performance of the framework on different data sets is relatively stable. Choosing the appropriate proportion of training data is the key to improving model performance. By adjusting the proportion of training data, the best model performance can be obtained. In practical applications, the optimal training data ratio should be determined based on specific tasks and data sets.

Therefore, by studying the influence of training data volume proportion on the performance of the ChronoBridge framework, it is found that ChronoBridge has excellent performance in entity prediction and relation prediction. Furthermore, selecting the optimal ratio of training data is a critical element in enhancing model performance. These insights contribute to the advancement of the model's learning and generalization abilities and offer direction for the distribution of data resources in real-world applications.

5.10 Case study

This section demonstrates the practical applications and effectiveness of ChronoBridge by illustrating its use in predicting taxi fares and processing the latest ICEWS 2024 data.

5.10.1 Predicting taxi fare using ChronoBridge

The ChronoBridge algorithm stands out in its application to the New York City Taxi Trip Dataset (Huang et al. 2023), which includes a comprehensive collection of over 600 million taxi trips from 2009 to 2016. This dataset is rich with details such as the taxi provider's ID, pickup and drop-off timings, locations, passenger count, trip distance, duration, and payment methods. For effective model training and evaluation, the data was divided into an 80% training set, with the remaining 20% split equally for validation and testing.

ChronoBridge utilizes this data within a TKG, where timestamps are derived from pickup and drop-off times, and taxi fares are the target relations to predict. Taxi identifiers and trip details serve as the entities in the TKG. The algorithm's Graph Convolutional Network framework is pivotal in structuring this data for the model.

Training ceases after 500 iterations, similar to ChronoBridge's validation procedure, and each session requires 7–8 h on a high-performance Nvidia TITAN RTX GPU. Post-training, ChronoBridge can infer and predict fares in about 30 s, outpacing the previously leading algorithm in terms of efficiency.

ChronoBridge's prediction accuracy is remarkable, with an MRR of 98.92%, and Hits@1, Hits@3, and Hits@10 at 98.72%, 99.87%, and 99.99%, respectively. These metrics reflect the algorithm's adeptness at learning and encoding historical data patterns through its advanced gated recurrent units. Furthermore, the decoder's ability to extract and fuse entity and relation features leads to a robust reasoning process.

In essence, ChronoBridge's effectiveness in fare prediction enhances the taxi service experience by ensuring customer satisfaction and improving operational efficiency. The algorithm's dual-component structure, which combines a sophisticated encoder with a powerful decoder, allows for a deep understanding of temporal and relational data dynamics. This approach not only achieves high accuracy in predictions but also sets a new standard for real-time data processing in urban transportation and the broader field of smart city infrastructure.

5.10.2 Latest ICEWS 2024 data processing using ChronoBridge

To validate the ChronoBridge framework's efficacy, the recurrent encoder's capability to identify sequential data trends and the decoder's adeptness in feature extraction and fusion were tested using data from March and April 2024, as depicted in Table 7 of the latest ICEWS dataset. The results, as indicated in the Answer section, align perfectly with real-world outcomes, thus confirming the predictions made by ChronoBridge.

Table 7 2024 latest ICEWS case studies. The first row is the case of entity prediction, and the last row is the case of entity prediction and relation prediction

History at $t-2$	History at $t-1$	Query at t	Answer
World Health Organization-Global, Make a visit, China	World Health Organization Global, Make statement, China	China, Share intelligence or information?	World Health Organization Global
Ukraine, Host a visit, Moldova	Moldova? Military, Russia Slovenia, Praise or endorse,Ukraine	Russia? Retreat militarily,? or Russia?, Ukraine	Ukraine or Retreat militarily

In the first scenario, the encoder successfully deduced that the sequence of events (A, Make a visit, B, $t-2$) followed by (A, Make statement, B, $t-1$) would logically lead to (B, Share intelligence or information, A, t), exemplifying its proficiency in capturing and processing temporal patterns of facts. The second scenario further demonstrates the decoder's sophistication, where it utilizes the context provided by (A, Host a visit, B, $t-2$), (B, Military, C, $t-1$), and (D, Praise or endorse, A, $t-1$) to not only predict the event (B, Retreat militarily, ?, t) but also to infer the nature of the relation between entities B and A at time t. This underscores the decoder's logical reasoning in handling simultaneous fact dependencies.

The recurrent encoder in ChronoBridge, enhanced with an Improved Gated Recurrent Unit, is particularly adept at autoregressive modeling of historical knowledge graph sequences. This enables a more effective capture of factual sequence patterns. The decoder complements this by meticulously extracting and fusing features related to entity relations, thereby refining the accuracy of predictions.

In essence, the precision of ChronoBridge's predictions demonstrates its substantial practical and application significance. It not only processes datasets with high accuracy but also provides insights into the temporal evolution of knowledge graphs, marking a significant leap in the field of TKG reasoning. This underscores the potential of ChronoBridge to be a valuable tool for future applications requiring dynamic and accurate predictions of events based on historical data.

6 Discussion

In the discussion of the ChronoBridge framework, it is important to delve into the implications and potential areas for further exploration that stem from its innovative approach to TKG Reasoning.

Firstly, the use of an advanced recurrent neural network with a revamped Gated Recurrent Unit (GRU) in the chronological node encoder represents a significant technical enhancement. This allows for a more effective autoregressive modeling of historical knowledge graph sequences, which is crucial for recognizing and encoding temporal patterns. The implications of this could be vast, potentially leading to improvements in other areas that require sequence prediction, such as natural language processing, financial forecasting, and even predictive healthcare.

Secondly, the bridged feature fusion decoder's focus on entity and relation feature extraction and fusion is a strategic move away from traditional approaches that might neglect these aspects during the decoding phase. This could have important consequences for the development of reasoning systems that are more aligned with the way humans understand and interact with temporal information, suggesting that future systems could become more intuitive and context-aware.

Moreover, the holistic approach that ChronoBridge takes by integrating both temporal dynamics and entity-relation features could inspire new lines of research. For instance, it raises questions about the potential for cross-domain applications and whether the principles of ChronoBridge could be adapted to other types of graphs or networks that are not strictly temporal in nature.

The empirical validation of ChronoBridge against three real-world datasets and its demonstrated improvements over baseline methods also opens up a discussion about the benchmarks used in TKG reasoning. There may be a need to develop more comprehensive and challenging datasets that can better test the limits of such advanced frameworks and drive the field towards more innovative solutions.

Lastly, while ChronoBridge marks a substantial advancement, there are always limitations and considerations for future work. For example, the computational complexity of the framework, its scalability to even larger datasets, and the interpretability of its reasoning processes are all areas that could benefit from further research. Additionally, exploring how ChronoBridge handles noisy, incomplete, or evolving data could also be a valuable direction, as real-world TKGs are often imperfect and subject to change.

In summary, the discussion around ChronoBridge touches upon its technical contributions, its potential impact beyond TKG reasoning, the need for better benchmarks in the field, and considerations for future research directions. The framework sets a new precedent in the field and offers a rich ground for future exploration and innovation.

7 Conclusion

In this paper, a novel ChronoBridge framework is proposed that incorporates a chronological node encoder and a bridged feature fusion decoder, enhancing the model's understanding of temporal evolution and its ability to infer future events. This approach addresses issues of detail loss and inadequate prediction accuracy in reasoning processes.Additionally, two variants of Gated Recurrent Units (GRUs) are proposed: IGRU and self-GRU. These are utilized within the ChronoBridge framework to respectively tackle problems of complex sequence inference and long-term dependency capture. Extensive experiments validate the superiority of our method in extrapolating entity reasoning tasks. Looking ahead, several directions can be explored to further advance ChronoBridge in TKG reasoning. Firstly, exploring how to extend the framework to handle more complex temporal patterns and a wider variety of relations. Secondly, considering the integration of multimodal information such as text and image data to enrich the model's input. Additionally, researching methods to optimize the framework's efficiency and scalability to meet the reasoning demands of large-scale knowledge graphs.

Author contributions [Qian Liu] contributed to the conception and design of the study, acquisition of data, analysis and interpretation of the results, and drafting of the manuscript. [Siling Feng] provided methods, ideas, and critical revisions to the manuscript, contributing important intellectual content and funding support. [Mengxing Huang] participated in data analysis, interpretation of the results, and critical revisions of the manuscript. [Uzair Aslam Bhatti] participated in data analysis, interpretation of the results, and critical revisions of the manuscript. All authors have read and approved the final manuscript.

Funding This research was supported by the National Natural Science Foundation of China project under Grant 62466016 and 62241202,and the National Key Research and Development Program of China under Grant 2021ZD0111000, and Key research and development plan of the Ministry of Science and Technology: 2021ZD0111002.

Data availability The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethical approval In this study, all data used were obtained in accordance with the relevant ethical guidelines and regulations. Informed consent was obtained from all participants involved in the data collection process. Any identifying information has been removed to ensure confidentiality and privacy. This research has been approved by the College of Information and Communication Engineering Review Board or Ethics Committee at Hainan Universiy.

References

ArunKumar KE et al (2022) Comparative analysis of Gated Recurrent Units (GRU), long short-term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends. Alexandria Eng J 61(10):7585–7603

Bai L et al (2023) Multi-hop temporal knowledge graph reasoning with temporal path rules guidance. Expert Syst Appl 223:119804

Bai L et al (2023) Temporal knowledge graphs reasoning with iterative guidance by temporal logical rules. Inf Sci 621:22–35

Bordes A et al (2013) Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, p. 26

Chung J et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555

Dasgupta SS, Ray SN, Talukdar P (2018) Hyte: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 2001–2011

Ding Z et al (2023) Learning meta-representations of one-shot relations for temporal knowledge graph link prediction. 2023 international joint conference on neural networks (IJCNN). IEEE, pp 1–10

Ferreira Leonardo N et al (2021) The small-world network of global protests. Sci Rep 11(1):19215

Ge X et al (2022) Spatio-temporal knowledge graph based forest fire prediction with multi source heterogeneous data. Remote Sensing 14(14):3496

Goel R et al (2020) Diachronic embedding for temporal knowledge graph completion. Proc AAAI Conf Artif Intell 34(4):3988–3995

Han Z et al (2020) xerte: explainable reasoning on temporal knowledge graphs for forecasting future links. arXiv preprint arXiv:2012.15537

Huang Xiaohui et al (2023) Multi-view dynamic graph convolution neural network for traffic flow prediction. Expert Syst Appl 222:119779

Ji G et al (2024) Decision optimization in cooperation innovation: the impact of big data analytics capability and cooperative modes. Ann Oper Res 333(2):871–894

Julien L, Wudage CM (2018) Deriving validity time in knowledge graph. Companion Proc Web Conf 2018:1771–1776

Li Z et al (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, pp. 408–417

Li Z et al (2022a) Complex evolutional pattern learning for temporal knowledge graph reasoning. arXiv preprint arXiv:2203.07782

Lin Y et al (2015) Learning entity and relation embeddings for knowledge graph completion. Proc AAAI Conf Artif Intell 29:1
Li Y, Sun S, Zhao J (2022b) TiRGN: time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. IJCAI, pp. 2152–2158
Liu Y et al (2022) Tlogic: temporal logical rules for explainable link forecasting on temporal knowledge graphs. Proc AAAI Conf Artif Intell 36(4):4120–4127
Liu K et al (2023) RETIA: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation. 2023 IEEE 39th international conference on data engineering (ICDE). IEEE, pp 1761–1774
Liu Y et al (2023) Entity relationship extraction based on a multi-neural network cooperation model. Appl Sci 13(11):6812
Mat ANA, Ritahani IA (2023) Study of Adam and Adamax optimizers on Alexnet architecture for voice biometric authentication system. 2023 17th international conference on ubiquitous information management and communication (IMCOM). IEEE, pp 1–4
Negro A et al (2023) Analysis of the evolution of COVID-19 disease understanding through temporal knowledge graphs. Front Res Metrics Anal 8:1204801
Shang C et al (2019) End-to-end structure-aware convolutional networks for knowledge base completion. Proc AAAI Conf Artif Intell 33(1):3060–3067
Sun H et al (2021) Timetraveler: reinforcement learning for temporal knowledge graph forecasting. arXiv preprint arXiv:2109.04101
Sun Z et al (2019) Rotate: knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197
Sun K et al (2022) Abnormal entity-aware knowledge graph completion. 2022 IEEE international conference on data mining workshops (ICDMW). IEEE, NY, pp 891–900
Wang Z et al (2014) Knowledge graph embedding by translating on hyperplanes. Proc AAAI Conf Artif Intell 28:1
Yang H, Yin L (2020) Re-net: a relation embedded deep model for au occurrence and intensity estimation. In: Proceedings of the Asian conference on computer vision
Yi Z et al (2022) Temporal knowledge graph embedding for link prediction. International conference on web information systems and applications. Springer, pp 3–14
Yuan L et al (2021) Predicting entity relations across different security databases by using graph attention network. 2021 IEEE 45th annual computers, software, and applications conference (COMPSAC). IEEE, pp 834–843
Zhang M et al (2023) Learning latent relations for temporal knowledge graph reasoning". In: Proceedings of the 61st annual meeting of the association for computational linguistics, Vol 1: long papers, pp. 12617–12631
Zhongwu C et al (2023) Meta-learning based knowledge extrapolation for temporal knowledge graph. Proc ACM Web Conf 2023:2433–2443
Zhou J et al (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81
Zhu C et al (2021) Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. Proc AAAI Conf Artif Intell 35(5):4732–4740
Zhu L et al (2023) Few-shot link prediction with meta-learning for temporal knowledge graphs. J Comput Design Eng 10(2):711–721

Authors and Affiliations

Qian Liu[1] · Siling Feng[1] · Mengxing Huang[1] · Uzair Aslam Bhatti[1]

✉ Siling Feng
fengsiling@hainanu.edu.cn

Qian Liu
liuqian2693@hainanu.edu.cn

Mengxing Huang
huangmx09@163.com

Uzair Aslam Bhatti
uzair@hainanu.edu.cn

1 School of Information and Communication Engineering, Hainan University, 58 Renmin Avenue, Haikou 570228, Hainan, China