TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 30/31 pp433–455
DOI: 10.26599/TST.2024.9010067
Volume 30, Number 1, February 2025

Exploring the Chameleon Effect of Contextual Dynamics in Temporal Knowledge Graph for Event Prediction

Xin Liu, Yi He, Wenxin Tai*, Xovee Xu, Fan Zhou, and Guangchun Luo

Abstract: The ability to forecast future events brings great benefits for society and cyberspace in many public safety domains, such as civil unrest, pandemics and crimes. The occurrences of new events are often correlated or dependent on historical and concurrent events. Many existing studies learn event-occurring processes with sequential and structural models, which, however, suffer from inefficient and inaccurate prediction problems. To better understand the event forecasting task and characterize the occurrence of new events, we exploit the human cognitive theory from the cognitive neuroscience discipline to find available cues for algorithm design and event prediction. Motivated by the dual process theory, we propose a two-stage learning scheme for event knowledge mining and prediction. First, we screen out event candidates based on historical inherent knowledge. Then we re-rank event candidates by probing into the newest relative events. Our proposed model mimics a sociological phenomenon called "the chameleon effect" and consists of a new target attentive graph collaborative learning mechanism to ensure a better understanding of sophisticated evolution patterns associated with events. In addition, self-supervised contrastive learning is employed to alleviate the over-smoothing problem that existed in graph learning while improving the model's interpretability. Experiments show the effectiveness of our approach.

Key words: temporal knowledge graph; event forecasting; graph neural networks; self-supervised learning; explainability

1 Introduction

Population-level societal events such as civil unrest and crime have significant impacts on our daily lives. Forecasting such events is of great importance for society and individuals since the accurate prediction of future events is beneficial to resource allocation[1-4], casualty prevention[5], information propagation[6, 7], risk management[8, 9], rumor/fake news detection[10, 11], crime detection[12] and epidemic prediction[13-15]. For example, the terrible crowd crush which killed more than 150 people occurred in the Halloween festivity in Itaewon of Seoul, Republic of Korea on October 29, 2022, is partly because of the unpreparedness of the police for anticipating the large gatherings in advance[16]. However, predicting future events is extremely challenging due to the lack of knowledge regarding the true causes and underlying mechanisms of event occurrence[17, 18], as the quote from Niels Bohr says: "*Prediction is very difficult, especially if it's about the future.*"

Over the past decade, considerable efforts have been dedicated to gathering important events, building event

• Xin Liu, Yi He, Wenxin Tai, Xovee Xu, Fan Zhou, and Guangchun Luo are with the University of Electronic Science and Technology of China, Chengdu 610054, China. Fan Zhou is also affiliated with Intelligent Terminal Key Laboratory of Sichuan Province, yibin 644000, Sichuan. E-mail: blurrymemory@126.com; hyjrj@foxmail.com; wxtai@std.uestc.edu.cn; xovee@std.uestc.edu.cn; gcluo@uestc.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2023-12-21; revised: 2024-02-03; accepted: 2024-04-01

databases, developing event expertise, and creating knowledge systems in order to support crucial event predictions. For instance, the United States can make policies by forecasting international crises mainly thanks to the Integrated Crisis Early Warning System (ICEWS)[19]. The Early Model Based Event Recognition using Surrogates (EMBERS)[20], as another example, is important for handling events such as influenza-like illness, civil unrest, domestic political elections, and crises. As a free open platform, Global Database of Events, Language, and Tone (GDELT)[21] monitors societal events in almost all countries and has emerged as a crucial project for researchers and practitioners.

With the availability of mass online media sources, data-driven approaches have been widely studied over the last few years. Researchers have developed various predictive analytical techniques based on historical data of events and other relevant sources for future event prediction. Many methods are explored by researchers from a variety of fields including statistics, linguistics, social science, data mining and machine learning. Nowadays, the accelerated developments of machine learning and deep learning have led to substantial progress in event prediction research. Before temporal dynamics are investigated in Refs. [19, 22-25], static reasoning is the mainstream for event forecasting methods[26-28]. Knowledge graph (KG) embedding-based models (e.g., TransE[29], RotatE[30], and ConvE[31]) map the predicates and entities to low-dimensional vector space and directly predict the event in a static manner. Meanwhile, many researchers[32-35] have found that societal events exhibit geographical properties and a high degree of temporal dependency. Based on this observation, methods based on Graph Neural Networks (GNNs)[36] and Recurrent Neural Networks (RNNs)[37] are proposed to exploit spatiotemporal characteristics of events. RE-NET[38], one of the most representative works, designs a recurrent event encoder to summarize the information of the past event sequence and utilizes a neighborhood aggregator for concurrent event with the same time snapshot. Analogously, RE-GCN[39] proposes a relation-aware Graph Convolutional Network (GCN) to capture the structural dependencies within KG at each timestamp, while the gate recurrence components are employed to extract sequential patterns between temporally adjacent facts.

Notwithstanding the increased awareness of structural and sequential information, several challenges hinder the applicability and performance of existing event prediction algorithms:

(1) Insufficient event knowledge understanding. Due to the existence of the time-variability[40], events at historical timestamps contribute differently to future event forecasting. However, it is widely acknowledged that the vanilla RNN benefits from recognizing patterns in the surrounding input features, which, however, fails to capture implicit correlations within time-variable sequences[41]. Analogously, due to the structural-variability phenomenon[42, 43], events that occurred in the neighborhood places contribute differently to the central event forecasting. However, most studies build an entity-level graph learning algorithm, updating only the representation of entities, thus failing to capture the event-level relationship.

(2) Inefficient learning algorithms. RNNs and their derivatives mainly use sequential processing over time, which means that long-term information has to sequentially travel through all cells before getting to the present ones. This is extremely time-consuming when the historical KG sequence gets longer[44]. Meanwhile, to prevent the model from missing important events that have high-order associations with the query, existing methods update node information on the whole temporal knowledge graph (TKG), which, significantly increases the computational overhead. Since practical TKGs are often very large in scale, designing an efficient learning algorithm is critical to avoid issues such as out-of-memory and expensive computational costs[45]. This, in turn, motivates recent efforts to devise efficient methods in modeling complex event data and deploying practical event prediction systems[46].

As a way of better understanding event forecasting, we try to mimic the cognitive process of humans and find clues for designing artificial architectures. In this work, we follow the *dual-process theory* – a principle proposed in neuroscience[47] – for TKG-based event forecasting. The dual-process theory suggests that human reasoning involves both heuristic and analytic processes. In the heuristic process, humans filter out appropriate judgments in their memory space based on prior experience. In the analytic process, humans use recent developments in their knowledge to adjust decision-making. This principle is widely used to

explain human predictive behaviors and has been explored to design event reasoning algorithms[43, 48-50].

Motivated by the need for more efficient modeling of historical and concurrent events, we introduce a two-stage structural and temporal learning approach for event forecasting. In the first pre-ranking stage, we leverage temporal associations between events to filter event candidates based on prior cognitive patterns. To account for unexpected and emerging events, the second re-ranking stage incorporates an event-level graph learning network. This network represents events as nodes and models their implicit relationships to capture signals from recent novel occurrences. By re-ranking based on this event graph, our model integrates both historical knowledge and new evolving dynamics for accurate forecasting. Our staged learning process allows efficiently prioritizing likely events while adapting predictions based on fresh unfamiliar events. Overall, this dual design extracts cognitive insights from the past while remaining agile to ever-changing events.

To encapsulate temporal event knowledge for the pre-ranking stage, we explicitly encode historical information as normalized frequency counts via preprocessing the event data from the TKG. These frequency counts produce timestamp representations that differentiate events with shared components but distinct occurrence times. Compared to sequential modeling techniques like using RNN[38, 39, 51], our proposed approach demonstrates substantially higher efficiency and efficacy in capturing historical data – RNN architectures model temporality through computationally-expensive sequential translations, whereas our frequency counts directly quantify event history in a lightweight tensor representation.

To model structural event knowledge for the re-ranking stage, we propose a novel target attentive graph collaborative learning algorithm. This explores event-level neighborhood interactions to better comprehend the intricate patterns within the TKG associated with events. Unlike conventional graph learning approaches such as graph attention networks, our method draws inspiration from the sociological "Chameleon Effect" phenomenon. This refers to the unconscious mimicry of behaviors to match one's social environment. Specifically, we substitute the vacant position in an event query with different entities to generate various event candidates. An event-level graph attention mechanism then adapts these event representations by modeling interactions with neighboring nodes. This allows precisely capturing concurrent structural knowledge. Additionally, we employ contrastive learning to mitigate over-smoothing in graph neural networks and enhance interpretability by maximizing the mutual information between original and event views. Our approach mimics human adaptation to social contexts, enabling more nuanced modeling of interrelated event dynamics.

The key contributions of this work are four-fold:

• We introduce a novel structural and temporal event forecasting model called TAG-Net, inspired by neurological theories of dual-process and the chameleon effect. TAG-Net employs a two-stage ranking strategy to adaptively model both historical and concurrent event interactions.

• We propose a target-attentive graph learning algorithm to collaboratively incorporate contextual information when evaluating different event candidates. To our knowledge, this represents the first technique to explicitly capture concurrent event-level relationships.

• We present a new contrastive constraint that mitigates over-smoothing in graph learning while improving model robustness during training. We provide theoretical and empirical analyses verifying its effectiveness.

• Extensive experiments on five real-world event datasets demonstrate the performance of TAG-Net, which significantly improves forecasting accuracy and efficiency compared to the state-of-the-art baseline models for event prediction.

The remainder of this paper is organized as follows. We review related work in Section 2 and introduce necessary background while formulating the event forecasting problem in Section 3. We describe the details of TAG-Net and provide theoretical model analysis in Section 4. In Section 5, we present empirical evaluations and comparisons between our model and baselines. Ablation study and parameter sensitivity are also provided. Finally, we conclude this work and point out future directions in Section 6.

2 Related Work

An event is a real-world occurrence that takes place in a specific location and time related to a particular topic. Unlike retrospective analyses such as event

summarization and event detection, event forecasting focuses on anticipating the occurrence of events in the future[46]. We review the literature on event forecasting from three categories: Recurrent Neural Network (RNN)-based methods, attention-based methods, and knowledge graph-based methods, especially with a focus on deep-learning-powered approaches and knowledge graph-based approaches. We also briefly introduce recent advances in incorporating Large Language Models (LLMs) into the event prediction task.

2.1 Sequential model-based event forecasting

Predicting events can be considered as a time series prediction problem[46], and, Recurrent Neural Networks (RNNs) are an ideal and natural choice for time series knowledge miming and have been proven to be powerful and expressive to capture long-term dependencies than traditional machine learning approaches such as hidden Markov models[52, 53] and autoregressive models[54, 55]. For example, LSTM-ARMA[56] is a pioneering work that applied LSTM[57] to solve the problem of predicting the occurrence of world news events. They take the feature representation of events as input and then feed the historical sequence to the standard LSTM architecture for predicting the next event. Many variants of LSTM such as bi-directional LSTM[58] and GRU[59] have also been used for event sequence modeling and forecasting[24]. Despite the significant improvement in capturing temporal information, several challenges have emerged in RNN-based approaches in event prediction, including the limited ability to accurately predict unrest events and the difficulty for humans to understand the model behavior.

2.2 Attention-based event prediction

Attention mechanisms enable dynamic highlights of relevant knowledge of the input data, imitating the cognitive attention of humans. This kind of method enhances the important parts of related events and fades out the trivial ones[46]. For example, a context-aware attention-based LSTM framework called CA-LSTM[51] is proposed to study the different contributions of data points in history and to improve the performance of predicting civil unrest events. CA-LSTM integrates the attention mechanism with the recurrent neural network to better understand the occurrence of events. ActAttn[60], a hierarchical attention-based spatiotemporal learning approach, tries to predict the occurrence of future protests and explains the importance of features. Specifically, the model contains a two-level attention module built on LSTM to calculate the intra-regional and inter-regional contributions. In general, compared to RNN-based methods, models that are equipped with the attention mechanism can help interpret the feature importance while underscoring the correlated events significant to the event prediction.

2.3 Knowledge graph-based event prediction

Recently, Knowledge Graphs (KGs)[61-63] are widely used in many real-world applications. Since knowledge graphs can model and reflect real-world facts, the event prediction problem can be transformed into the missing fact reasoning problem in the KGs[64]. With that, many researchers have leveraged KGs as a promising solution due to their natural ability to provide domain-specific knowledge for event reasoning and forecasting. In this work, we review KG literature from two perspectives: static KG and temporal KG-based models.

2.3.1 Static KG-based models

In recent years, Graph Convolutional Networks (GCNs)[65] becomes a representative model to combine content and structural features for graph learning. Many studies have generalized it to the relation-aware GCNs so as to deal with the learning on KGs. Among them, R-GCN[66] is the first work applying GCN to model the relational KG data. It extends GCN with relation-specific filters by defining an information propagation model to calculate the forward-pass update of an entity in a relational graph. WGCN[67] is a weighted GCN that takes benefits from both GCN and ConvE[31]. Its encoder leverages knowledge graph node structure, node attributes, and edge relation types to encode event information, while the decoder models the relationship as the translation operation and captures the translational characteristic between entities and relations. In comparison with existing GCNs which cannot fully utilize multi-relation information, VR-GCN[68] utilizes a vectorized relational GCN to learn embeddings of both graph entities and relations simultaneously for modeling the multi-relational networks. Similarly, CompGCN[69] leverages a variety of entity-relation composition operations from knowledge graph embedding techniques and scales with the number of relations to jointly embed both

nodes and relations in a relational graph during GCN aggregation. However, these static KG methods neglect the dynamic evolution of the graph, which differs from real-world situations and leads to deviations in the prediction[43].

2.3.2 Temporal KG-based models

Due to its ability to incorporate and leverage time information in relational data, Temporal Knowledge Graph (TKG) has received considerable attention in the KG community. A TKG is actually a sequence of KGs corresponding to different timestamps, where all concurrent facts in each KG exhibit structural dependencies, and temporally adjacent facts carry informative sequential patterns[39]. Many new TKG models are proposed in the past few years.

The mainstream of existing approaches leverages GNNs in combination with a sequential model to integrate structural and temporal information in the entity and relation learning process. Representative methods include RE-Net[38], RE-GCN[39], TANGO[70], xERTE[71], CEN[72], and HGLS[73]. Specifically, RE-Net[38] applies a GCN and a GRU to model the sequence of 1-hop subgraphs related to the given subject entity. Different from RE-Net which neglects the structural dependencies within KGs at different timestamps and the static properties of entities, RE-GCN[39] utilizes a relation-aware GCN to capture the structural dependencies within the KG at each timestamp, while the historical KG sequence is modeled autoregressively by the gate recurrent components to capture the sequential patterns across all temporally adjacent facts. TANGO[70] utilizes neural ordinary differential equations (ODEs) to model the temporal sequences. xERTE[71] is based on temporal relational attention mechanisms. To answer a query, it extracts query-relevant subgraphs, and further computes and propagates attention scores to identify the relevant evidence in the subgraphs. CEN[72] integrates CNNs that can handle evolution patterns of variable lengths via an easy-to-difficult curriculum learning strategy. It learns the evolution patterns from short to long in an online setting and thus can adapt to changes in evolution patterns over time. HGLS[73] captures abundant semantic information of events from two different perspectives: sub-graph level and global-graph level.

Another line of the TKG-based event research predicts the event occurrence based on the appearance and repetition of historical facts. For example, CyGNet[74] takes account of temporal facts with repetitive patterns, which explicitly models the historical dependency to represent the query-related historical events. DA-Net[43] extracts the historical events from historical KGs and designs a self-attention layer to learn the attention of these events uniformly. Another attention layer in DA-Net adjusts the coefficients based on the event query and passes historical event frequency. In contrast with RNN-based methods that rely on sequential facts modeling, both CyGNet and DA-Net are much more time efficient since they encode the historical information with only one pass. However, they fail to consider the influence of unexpected emergencies while ignoring the structural information that is important for depicting event evolution and occurrence correlation.

The proposed method also discards the sequential learning scheme and enables efficient training and prediction: we explicitly encode the historical information with normalized historical frequency counts. We note that our work differs from the above two works in the ways of the encoding of timestamps and the handling of concurrent events – we encode timestamps via event historical dependency, and propose a novel event-level graph collaborative learning strategy to explore neighborhood interaction in a more natural way.

2.4 Event prediction using LLM

In recent years, LLMs have demonstrated remarkable performance across various challenging tasks, including arithmetic reasoning and multi-turn dialogue[75]. Some recent studies have explored the potential of LLMs in aiding event prediction. For example, LAMP[76] utilizes a pre-trained event sequence model to generate predictions regarding future events, which are subsequently assessed with the support of an LLM. Initially, the LLM generates potential causes to explain the likelihood of each prediction. These generated causes then serve as queries to identify similar or relevant events from past occurrences. Another model is tasked with embedding these retrievals and assessing whether they could lead to the corresponding prediction. TimeLlaMA[77] is fine-tuned on LlaMA2, using their own collection of multi-source instruction tuning datasets. Experimental results conducted on several TKG datasets demonstrate that TimeLlaMA achieves the state-of-the-art performance

in temporal prediction and explanation generation compared to a variety of existing LLMs. Note that these methods using LLMs for event prediction are not directly relevant to the scope of this work. However, we trust that this supplementary information will prove valuable to researchers engaged in endeavors within this domain.

3 Preliminary

In this section, we formally define the TKG-based event forecasting problem and then provide the necessary backgrounds w.r.t. event repetition phenomenon. Table 1 summarizes frequently used notations in this work.

3.1 Definitions

Definition 1 (Event) Let $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote a finite set of entities, relation types, and timestamps, respectively. An event can be represented as a quadruple formalized as (s,r,o,t), where $s \in \mathcal{E}$ is a subject (head) entity, $o \in \mathcal{E}$ is an object (tail) entity, $r \in R$ is the relation (predicate) occurring at timestamp t between s and o.

Definition 2 (Temporal Knowledge Graph) A temporal knowledge graph $\mathcal{G}$ is a set of quadruples. Among them, $\mathcal{G}_t$ represents a TKG snapshot which is the set of quadruples captured at time t.

Definition 3 (Event Representation) Let $\mathbf{E}_e \in \mathbb{R}^{|\mathcal{E}|\times d}$ be the embeddings of all entities, the row of which represents the embedding vector of an entity (subject or object). Similarly, let $\mathbf{E}_r \in \mathbb{R}^{|\mathcal{R}|\times d}$ be the embeddings of all relation types. We use boldfaced **s**, **r**, **o** for the embedding vectors of s, r and o, respectively.

Definition 4 (Event Forecasting) According to previous works[38, 74], event forecasting aims to predict one of the missing components of a certain event. Given a query $(s,r,?,t)$, our target is to predict the missing object o. Analogously, if the query is set to $(?,r,o,t)$, then our target is to predict s.

Without loss of generality, we describe our model as predicting the missing object entity in a temporal fact (our model can be easily extended for predicting the subject entity). Considering the task is to forecast future events, directly applying embedding techniques on timestamp t causes training issues – we discuss how to encoder timestamps in Section 4.1.1.

Table 1 Mathematical Symbols.

Symbol	Description
$\mathcal{E}$	entity set.
$\mathcal{R}$	relation set.
$\mathcal{T}$	timestamps of events.
$\mathcal{G}$	temporal knowledge graph.
$\mathcal{G}_t$	a temporal knowledge graph snapshot.
$\mathbf{E}_e$	embeddings of all entities.
$\mathbf{E}_r$	embeddings of all relation types.
$\mathbf{F}_t^{s,r}$	the normalized counted event frequencies.
$\mathbf{F}_t^{s,r}(o)$	the frequencies of a candidate entity o.
$\mathbf{I}_t^{s,r}$	indicate whether the event is historical or new.
$\mathbf{S}_{\text{his}}^{s,r}$	the score with historical dependency.
$\mathbf{S}_{\text{non}}^{s,r}$	the score with non-historical dependency.
$\mathbf{M}_{\text{his}}$	the mask to filter object candidates
P_{bs}	the score of the binary mode selector.
$\mathbf{P}(o\|s,r,\mathbf{F}_t^{s,r})$	the probability of each event candidate.

3.2 Event repetition phenomenon

The event repetition phenomenon widely exists in real-world data and applications because consistent or similar circumstances usually repeat over time[78]. Some of them regularly happen while others do not (which makes this problem challenging). Always, similar events occur over time, reflecting how society responds to the historical cycles. According to our investigations, the repetition phenomenon not only exists but also accounts for a large proportion of the interactions in real-world event datasets, showing that temporal knowledge contains conspicuous recurrence patterns along the trending timeline (Table 2). Motivated by this phenomenon, taking repetitive behaviors into account can enhance our understanding and the prediction performance of event occurrence.

4 Method

According to the dual-process theory of human cognition, people recall similar historical facts from their memory and assign the original attention to them according to prior experience. These new facts and knowledge are then utilized to adjust and select a proper decision. Inspired by this theory, we propose a two-stage spatiotemporal event learning process to

Table 2 Repetition ratio (%) on five benchmarks.

Dataset	Type	Train(%)	Validation(%)	Test(%)
ICEWS18	Event	22.64	16.98	16.32
ICEWS14	Event	65.18	26.82	26.82
GDELT	Event	35.44	23.91	24.93
WIKI	KG	65.86	69.20	72.87
YAGO	KG	40.30	78.25	83.81

model the historical information and the concurrent structural correlation for event forecasting. In the first stage, we extract the repetitive historical facts of each event query and filter out candidates on the basis of historical dependency. In the second stage, we consider the influence of the concurrent event on the prediction and propose a target-aware graph learning mechanism to help make the final decision. Figure 1 outlines the workflow of our proposed TAG-Net methodology.

4.1 Pre-rank via historical facts

According to the dual process theory, given an unknown query $(s, p, ?, t)$, humans determine the query answer by first searching their memory space for similar situations. In our case, these situations can be expressed as (s, p, o_i, k), $k < t$, i.e., historical repetitive facts. Then our brain will investigate and decide which facts are important for predicting future events[43].

4.1.1 Query encoder

Given event query $(s, r, ?, t)$, existing methods often encode the event queries into vectors[43, 74, 79]. For subject s and relation r, embedding techniques[80] can be used to represent these discrete variables as continuous low-dimensional vectors. However, it is difficult to encode timestamps directly since the event forecasting task is to predict future events. In other words, conventional time embedding methods[43] are not suitable here, given that the timestamps in the training data do not occur in the testing data will lead to unreliable predictions[81].

Previous methods primarily tried two timestamp modeling ways: capturing intricate temporal dependencies implicitly, or learning the inter-event timestamp explicitly. For the first line of work, RNNs have been used to summarize and maintain evolving entity states[38, 64]. The other line generally models inter-event time Δt with embedding techniques or conditional probability density estimation[19, 81]. In practice, RNN-based methods tend to be time-consuming when dealing with multiple snapshots (long history/sequences), while time interval-based methods make a strong assumption of the prior distribution, which, however, might be inaccurate in real-world situations. Note that the time interval-based method

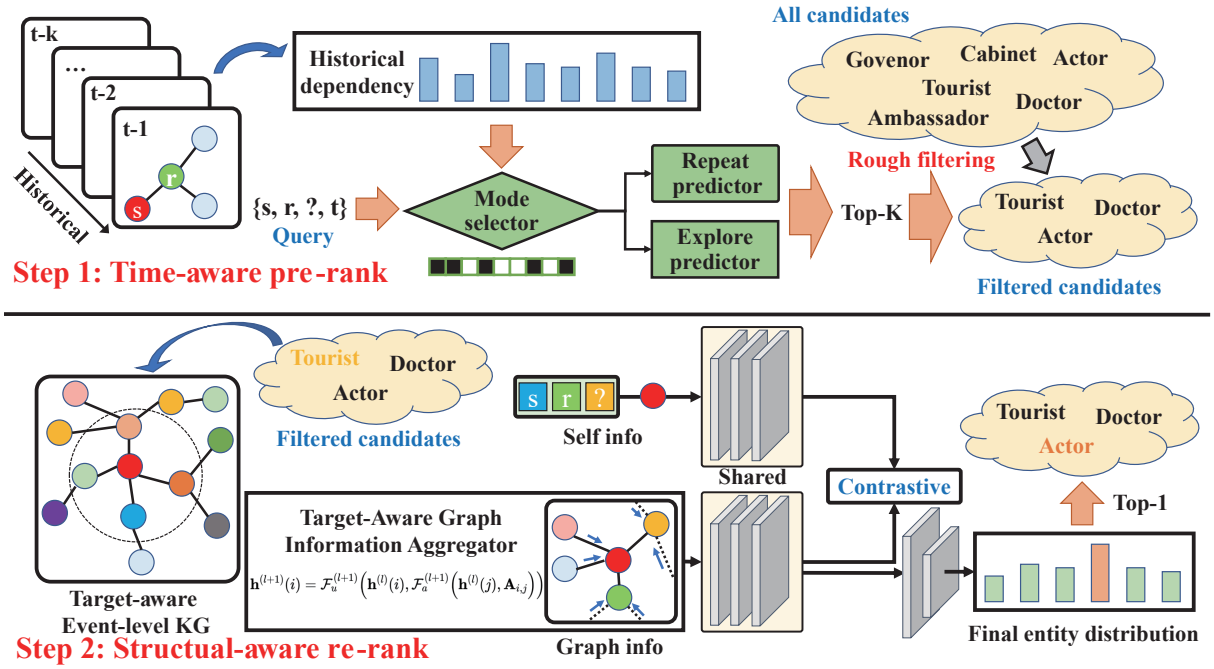


Fig. 1 The workflow of TAG-Net (Target-Attentive Graph Neural Network). (i) In the first stage, we extract event historical dependency $F_t^{s,r}$ from the temporal knowledge graph, and then we use normalized $F_t^{s,r}$ and embedding technique to generate the query vector. Next, we feed the query vector into the binary mode selector to determine whether the event forecasting can benefit from historical interactions or is a completely new event. Finally, we refine the event candidate list and maintain the top-K candidates. (ii) After extracting information from historical interactions, in the second stage, we exploit information from concurrent events. Specifically, we apply a target attentive graph collaborative learning, generating corresponding event-level graphs for each candidate. Finally, we re-rank event candidates by probing into the newest relative events.

CyGNet[74] sets an infinite magnitude of the time embedding ($t_k = t_{k-1} + \Delta t$) – i.e., the magnitude of the time embedding will increase when the timestamp increases, resulting in an unstable training problem.

The core desirability of temporal modeling lies in that, given two event queries $(s, r, ?, t_1)$ and $(s, r, ?, t_2)$, the query encoder should be able to generate inconsistent representations. In this work, we propose a simple but effective way to encode the event timestamp. Specifically, we use the normalized ($L1$-Norm) counted frequencies $\mathbf{F}_t^{s,r} \in \mathbb{R}^{|\mathcal{E}|}$ as the event-aware time representation. For each query quadruple $(s, r, ?, t)$, $\mathbf{F}_t^{s,r}(o)$ represents the frequencies of an event candidate entity o associated with subject s and relation r from the previous snapshots:

$$\mathbf{F}_t^{s,r}(o) = \sum_{k<t} |\{o \mid (s, r, o, k) \in \mathcal{G}_k\}| \tag{1}$$

Directly using $\mathbf{F}_t^{s,r}$ as the timestamp representation has several benefits: (i) the model can distinguish two event queries that have the same subject and relation, but with different timestamps t; (ii) using the normalized counted frequencies as time representation retains the information of historically related events in a more elegant way; (iii) the magnitude of the normalized timestamp vector $\mathbf{F}_t^{s,r}$ in our method is constant to 1, preventing the model from unstable training.

Considering the sparsity problem of $\mathbf{F}_t^{s,r}$, we use a non-linear transformation to reduce the dimension of the time representation:

$$\mathbf{t} = \tanh(\mathbf{W}_q \operatorname{norm}(\mathbf{F}_t^{s,r})) \tag{2}$$

where tanh is the activation function $\tanh(x) = \dfrac{e^{2x} - 1}{e^{2x} + 1}$ and $\mathbf{W}_q \in \mathbb{R}^{d \times |\mathcal{E}|}$ are trainable parameters. Then the representation of the event query given $(s, r, ?, t)$ is defined as:

$$\mathbf{q} = (\mathbf{s} \oplus \mathbf{r} \oplus \mathbf{t}) \tag{3}$$

4.1.2 Historical dependency learning

As discussed above, explicitly considering the repetitive behavior in algorithm design could enhance future event prediction accuracy. Following previous works[74, 82, 83], we propose to model historical dependency explicitly and apply it as an additional domain expert bias to forcibly change the distribution of model outputs. Normally, we can reuse the counted frequencies $\mathbf{F}_t^{s,r}$ as the representation of historical dependency. However, since the statistical frequency is closely related to the data distribution, directly using it as an additional artificial bias may cause the model susceptible to the long-tailed distribution. In other words, the model may predict entities that frequently interact with other entities in the training dataset, which, however, might not be the ground-truth candidates. Thereupon, the frequency-biased representation of historical dependency typically restricts the reliability of the model which tends to focus on dominant types of events and exhibits a poor performance on tail types – infrequent events tend to have greater impact on society[84].

To this end, we transform $\mathbf{F}_t^{s,r}$ into $\mathbf{I}_t^{s,r} \in \mathbb{R}^{|\mathcal{E}|}$ where each slot only indicate whether the corresponding event is historical or new:

$$\mathbf{I}_t^{s,r}(o) = \begin{cases} +1, & \mathbf{F}_t^{s,r}(o) \geqslant 1; \\ -1, & \mathbf{F}_t^{s,r}(o) = 0 \end{cases} \tag{4}$$

Then we define a *repeat predictor* to forecast future events based on the assumption that it has already occurred in the past. In concrete, we design a non-linear layer to transform the query embedding into candidate scores which can be defined as:

$$\mathbf{S}_{\text{his}}^{s,r} = \tanh(\mathbf{W}_{\text{his}}\mathbf{q} + \mathbf{b}_{\text{his}})\mathbf{E}_e^T \tag{5}$$

where $\mathbf{W}_{\text{his}} \in \mathbb{R}^{3d \times d}$ and $b \in \mathbb{R}^d$ are trainable parameters. We add $\mathbf{I}_t^{s,r}$ to change the index scores of historical entities in $\mathbf{S}_{\text{his}}^{s,r}$ to higher values without contributing to the gradient update:

$$\mathbf{S}_{\text{his}}^{s,r} = \tanh(\mathbf{W}_{\text{his}}\mathbf{q} + \mathbf{b}_{\text{his}})\mathbf{E}_e^T + \eta\mathbf{I}_t^{s,r} \tag{6}$$

As a result, we pay more attention to historical entities by adding $\mathbf{I}_t^{s,r}$ into $\mathbf{S}_{\text{his}}^{s,r}$.

4.1.3 Non-historical event exploring

In most TKGs, although many events often show repeated occurrence patterns, new events may have no historical events as referred by Ref. [74]. Thus, we should take not only historical but also non-historical entities into consideration. Analogously, for non-historical dependency, the score for all candidates is defined as:

$$\mathbf{S}_{\text{non}}^{s,r} = \tanh(\mathbf{W}_{\text{non}}\mathbf{q} + \mathbf{b}_{\text{non}})\mathbf{E}_e^T \tag{7}$$

We cut down the impact of events that have occurred before by changing the candidates' scores to:

$$\mathbf{S}_{\text{non}}^{s,r} = \tanh(\mathbf{W}_{\text{non}}\mathbf{q} + \mathbf{b}_{\text{non}})\mathbf{E}_e^T - \eta\mathbf{I}_t^{s,r} \tag{8}$$

4.1.4 Binary mode selector

We further design a *binary mode selector* which is

motivated by that, once we have trained an accurate mode selector, our model can learn to predict from a delimited candidate space rather than the entire entity vocabulary, reducing the difficulty of event forecasting and boosting the prediction accuracy.

Specifically, we feed query representation $\mathbf{q}$ to a simple MLP, using cross-entropy loss $\mathcal{L}_{\text{bs}}^{\text{ce}}$ as the guidance to minimize the difference between the output of binary mode selector and the ground truth label (whether the missing entity of query q exists in the set of historical entities). Then, based on the prediction P_{bs} of the binary mode selector, our model will choose the corresponding mask $\mathbf{M}$ to filter object candidates:

$$\mathbf{M}_{\text{his}}(o)=\begin{cases}1, & \mathbf{I}_t^{s,r}(o)=+1;\\ 0, & \mathbf{I}_t^{s,r}(o)=-1\end{cases} \tag{9}$$

$$\mathbf{M}_{\text{non}}(o)=\begin{cases}1, & \mathbf{I}_t^{s,r}(o)=-1;\\ 0, & \mathbf{I}_t^{s,r}(o)=+1\end{cases} \tag{10}$$

$$\mathbf{S}_t^{s,r}=\mathbf{S}_{\text{his}}^{s,r}\cdot\mathbf{M}_{\text{his}}\cdot P_{\text{bs}}+\mathbf{S}_{\text{non}}^{s,r}\cdot\mathbf{M}_{\text{non}}\cdot(1-P_{\text{bs}}) \tag{11}$$

And the probability of each event candidate is:

$$\mathbf{P}(o|s,r,\mathbf{F}_t^{s,r})=\text{softmax}(\mathbf{S}_t^{s,r})(o) \tag{12}$$

At last, we have entities with the maximum values:

$$\tilde{o}=\text{argmax}_{o\in\mathcal{E}}(\mathbf{P}(o|s,r,\mathbf{F}_t^{s,r})) \tag{13}$$

4.2 Re-rank via recent concurrent events

In the last subsection, we get the event prediction o given event query $(s,r,?,t)$. However, as reported in Ref. [85], belief bias occurs when we draw conclusions solely based on prior, old-fashioned beliefs[86]. Therefore, we consider the influence of the newest unexpected emergencies on the prediction, i.e., the relevant concurrent events at the nearest timestamp τ, for adjusting or rearranging the event candidates obtained during the pre-rank stage.

4.2.1 Collaborative event learning

In TAG-Net, we also introduce GNNs to facilitate event collaborative learning as previous works do[38, 39, 64]. The core insight of the graph collaborative learning is to enrich the node representation learning via aggregating neighborhood information from TKG[87]. Many of the existing works[38, 39] adopt GCNs and their variants such as relational GCN[64] to learn a better representation of entities for every snapshot. However, these graph-based learning methods are resource-consuming and, more importantly, restricted to low-level entity representations that only factor in subject entities and relations when calculating attention scores, making the model unable to adjust adaptively to different event candidates.

Taking inspiration from the natural behavior of chameleons who change their color in response to their surroundings, in this work, we propose a target attentive graph learning approach that can adaptively incorporate information from concurrent events when faced with different object candidates. Specifically, we substitute the vacant position in the event query $(s,r,?,t)$ with different candidate entities to form complete candidate events. We then propose an event-level attention mechanism to update the representation of candidate events. For an event-level TKG $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$ at the latest timestamp τ (τ is omitted for simplicity), every node $v\in\mathcal{V}$ represent a certain event and will be paired with a node representation $\mathbf{h}(v)$ initialized as $\mathbf{h}^{(0)}:=(\mathbf{s}\oplus\mathbf{r}\oplus\mathbf{o})$, which will serve as input to the first GNN layer. Events that have at least one common entity will serve as neighbors. In each graph convolution layer, node representations are updated in two steps: neighbors propagating and node updating.

For neighbor propagation, we compute a pair-wise unnormalized attention score between two neighbors. Here, it first concatenates two event representations and takes a dot product with a learnable weight vector $\mathbf{a}^{(l)}$:

$$e_{ij}^{(l)}=\text{LeakyReLU}(\mathbf{a}^{(l)}(\mathbf{h}^{(l)}(i)\oplus\mathbf{h}^{(l)}(j))) \tag{14}$$

We normalize coefficients across all choices of neighbors j using the softmax function to make them easily comparable across different nodes:

$$\alpha_{ij}^{(l)}=\frac{\exp(e_{ij}^{(l)})}{\sum_{k\in\mathcal{N}(i)}\exp(e_{ik}^{(l)})} \tag{15}$$

Then the embeddings from neighbors are aggregated together, scaled by the attention scores:

$$\mathbf{h}^{(l+1)}(i)=\sigma\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}^{(l)}\mathbf{h}^{(l)}(j)\right) \tag{16}$$

The above aggregation process can be expressed as $\mathcal{F}_a^{(l+1)}\left(\mathbf{h}^{(l)}(j),\mathbf{A}_{i,j}\right)$.

For node updating during the l-th iteration, each $\mathbf{h}^{(l)}(i)$ is updated using i's neighborhood information and self-information $\mathbf{h}^{(l)}(i)$:

$$\mathbf{h}^{(l+1)}(i)=\mathcal{F}_u^{(l+1)}\left(\mathbf{h}^{(l)}(i),\mathcal{F}_a^{(l+1)}\left(\mathbf{h}^{(l)}(j),\mathbf{A}_{i,j}\right)\right), \tag{17}$$

where $\mathcal{F}_u : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ is a non-linear layer that maps both the current representation and aggregated vector to update the node's representation.

4.2.2 Re-rank on candidate subset

Note that this vanilla version of the target attentive graph learning has to compute graph attention $|\mathcal{E}|$ times ($i \in |\mathcal{E}|$) for each sample, limiting its practical values in real situations. To solve this problem, the first *pre-rank* stage should provide preliminary filtering results to narrow down the item list delivered to the second *re-rank* stage. Given the probability of each candidate from Eq. (12), we select the top K items for the second stage:

$$\tilde{\mathcal{E}} = \underset{\text{top}K}{\text{argmax}}(\mathbf{P}(o|s,r,\mathbf{F}_t^{s,r})) \tag{18}$$

where K is a hyper-parameter and $\tilde{\mathcal{E}}$ is the subset of entity candidates. In the second stage, with a small portion of object candidates, the target attentive graph collaborative learning can be considered to enrich the event representation. Since we reduce the number of candidates from $|\mathcal{E}|$ to $|\tilde{\mathcal{E}}|$ (dozens), the computational complexity is significantly reduced.

4.2.3 Over-smoothing and interpretability

Graph neural networks integrate the comprehensive relation of graph data and the representation learning capability of neural networks. However, the phenomenon occurs that the learned node representations become indistinguishable – due to the nature of the message passing scheme[88] and overfitting problem[89] – the updated event representation might be totally different from the original one – hurting the performance of the model on downstream tasks (*a.k.a.* over-smoothing problem)[88-90]. In addition, most GNNs are deployed as black boxes, lacking explicit declarative knowledge representations (inexplicable problem). As a result, they have difficulty in generating the required underlying explanatory structure[91], and, without understanding and verifying the inner working mechanism, GNNs cannot be fully trusted, which prevents their use in critical applications pertaining to fairness, privacy and safety[92].

4.2.4 Self-supervised event learning

As for the over-smoothing problem, our key insight is to add a constraint to let the model be able to identify the original node after the graph collaborative learning. For the interpretability problem, we aim to identify a subgraph that is most influential for the prediction[92]. Accordingly, edges and nodes that have limited influences will not be selected for prediction explanation since they cannot influence the performance of GNNs. This effect also implies that the graph with strong interpretability tends to be a small connected subgraph. In this work, we introduce self-supervised contrastive learning[93] to solve the above two problems simultaneously.

As shown in Fig. 2, we feed the original view and the graph learning view of the candidate event samples into the same network to extract features and learn to make the cross-correlation matrix between these two groups of output features close to the identity. The goal is to keep the representation vectors of one sample's different views similar while minimizing the redundancy between these vectors. Let C be a cross-correlation matrix computed between outputs from two identical networks along the batch dimension. C is a square matrix with the size same as the feature network's output dimensionality. Each entry in the matrix C_{ij} is the cosine similarity between network output vector dimension at index i, j and batch index b, $\mathbf{z}_{bj}^A, \mathbf{z}_{bj}^B$, with a value between -1 (i.e., the perfect anti-correlation) and 1 (i.e., the perfect correlation):

$$C_{ij} = \frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b \left(z_{b,i}^A\right)^2} \sqrt{\sum_b \left(z_{b,j}^B\right)^2}} \tag{19}$$

The contrastive loss function is then defined as:

$$\mathcal{L}^{cl} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{20}$$

where λ is a positive constant to balance the importance of the invariance term and the redundancy reduction term of the loss. Intuitively, the invariance term of the objective tries to equate the diagonal

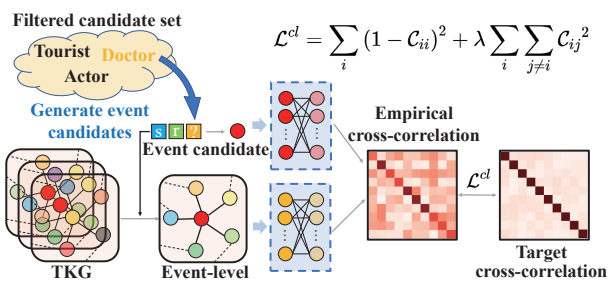


Fig. 2 Illustration of self-supervised contrastive learning between the self-information view and graph learning view for a certain event candidate.

elements of the cross-correlation matrix to 1, making the event embeddings from collaborative learning still retain the original event information. Meanwhile, the redundancy reduction term tries to equate the off-diagonal elements of the cross-correlation matrix to 0, decorrelating the representations of different events.

4.2.5 Theoretical analysis

In this subsection, we will derive the reason why our model suffers from the over-smoothing problem (Theorem 1). Then, we prove that the proposed additional contrastive constraints can prevent the model from learning homogeneous representations, as formally induced in Theorem 2. Finally, we give a deep-insight analysis of how self-supervised contrastive learning enhances interpretability (Discussion 1).

Specifically, we first detail why GNNs are prone to the over-smoothing problem:

Lemma 1 Suppose a connected graph $\mathcal{G}$ with N nodes $\mathcal{V}=\{v_1, v_2, \ldots, v_N\}$ has no bipartite components, defining $\mathbf{v}_i^{(\kappa)}$ as the embedding of node i after κ times GCN message passing. Then for large enough κ, $\forall i, j \in \{1,2,\ldots,\mathcal{N}\}$, $\|\mathbf{V}_i^{(\kappa)}-\mathbf{V}_j^{(\kappa)}\|_2 \leqslant \|\mathbf{V}_i^{(0)}-\mathbf{V}_i^{(0)}\|_2$[94].

Proof Given the message passing of vanilla GCN,

$$\mathbf{V}^{(t+1)}=\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{V}^{(t)} \tag{21}$$

where $\hat{A}=A+I$ and $\hat{\mathbf{D}}_i=\sum_j \hat{\mathbf{A}}_{ij}$, we can then rewrite the message passing as

$$\mathbf{V}^{(t+1)}=\left(\mathbf{I}-\boldsymbol{L}_{\text{sym}}\right)\mathbf{V}^{(t)} \tag{22}$$

where $\mathbf{L}_{\text{sym}}=\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{L}}\hat{\mathbf{D}}^{-\frac{1}{2}}, \hat{\mathbf{L}}=\hat{\mathbf{D}}-\hat{\mathbf{A}}$. Let $(\lambda_1,\lambda_2,\ldots,\lambda_N)$ and $(\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_N)$ respectively denote the eigenvalue and eigenvector of matrix $\mathbf{I}-\mathbf{L}_{\text{sym}}$. With the property of the symmetric Laplacian matrix for non-bipartite and connected graph, we have:

$$-1<\lambda_1<\cdots<\lambda_N=1, \quad \mathbf{e}_N=\hat{\mathbf{D}}^{-\frac{1}{2}}[1,1,\ldots,1]^T \tag{23}$$

and

$$\begin{aligned}&\mathbf{V}_i^{(\kappa)}-\mathbf{V}_j^{(\kappa)}\\&=\left[\left(\mathbf{I}-\mathbf{L}_{\text{sym}}\right)^{\kappa}\mathbf{V}\right]_i-\left[\left(\mathbf{I}-\mathbf{L}_{\text{sym}}\right)^{\kappa}\mathbf{V}\right]_j\\&=\left[\lambda_1^{\kappa}\left(\mathbf{e}_1^{(i)}-\mathbf{e}_1^{(j)}\right),\ldots,\lambda_{n-1}^{\kappa}\left(\mathbf{e}_{N-1}^{(i)}-\mathbf{e}_{N-1}^{(j)}\right),0\right]\hat{\mathbf{V}}\end{aligned} \tag{24}$$

where $\mathbf{e}_k^{(i)}$ denotes the i-th element of eigenvector $\mathbf{e}_k$. $\hat{\mathbf{V}}$ is the coordinate matrix of $\mathbf{V}$ in the space spanned by eigenvectors. We can write the $\|\mathbf{V}_i^{(\kappa)}-\mathbf{V}_j^{(\kappa)}\|_2$ as

$$\left\|\mathbf{V}_i^{(\kappa)}-\mathbf{V}_j^{(\kappa)}\right\|_2=\sqrt{\sum_{m=1}^{N}\left[\sum_{k=1}^{N-1}\lambda_k^{\kappa}\left(\mathbf{e}_k^{(i)}-\mathbf{e}_k^{(j)}\right)\hat{\mathbf{V}}_{\text{km}}\right]^2} \tag{25}$$

Since we have $-1<\lambda_1<\cdots<\lambda_{N-1}<1$, for a large κ, $\|\mathbf{V}_i^{(\kappa)}-\mathbf{V}_j^{(\kappa)}\|_2 \leqslant \|\mathbf{V}_i^{(0)}-\mathbf{V}_i^{(0)}\|_2$. ■

Note that smoothness (Euclidean distance) is a metric that reflects the similarity of node representations[89], so, according to the above lemma, the Frobenius norm of the graph after κ-times message passing is less than that of the original graph. That is to say, when κ gets larger, the over-smoothing problem occurs. As we introduce an attention mechanism into message passing, the above derivation cannot directly apply to our scenario. To facilitate the analysis of settings of our model, we focus on the attention aggregation and simplify Eq. (16) in terms of matrix operation as $\mathbf{V}^{(t+1)}=\mathbf{A}\mathbf{V}^{(t)}$. Among them, $\mathbf{A}_{N\times N}$ is redefined as the attention weight matrix (different from the above discretized $\mathbf{A}$ in vanilla GCN): $\mathbf{A}_{ij}^{(t)}=\alpha_{ij}^{(t)}$, if $j\in\mathcal{N}(i)$ else 0, and $\sum_{i=1}^{N}\alpha_{i,j}^{(t)}=1$. Let $D=\text{diag}(d_1,d_2,\ldots d_N)$ and $d_i=\sum_{j=1}^{N}\mathbf{A}_{i,j}$, then the graph Laplacian of $\mathcal{G}$ is defined as $\mathbf{L}=\mathbf{D}-\mathbf{A}$. According to Ref. [95], $\mathbf{L}_{rw}=\mathbf{D}^{-1}\mathbf{L}$ is the random walking normalized Laplacian of graph $\mathcal{G}$. Then we have:

Lemma 2 Single-layer attention message passing is equivalent to a specific Laplacian smoothing operation[90].

Proof According to random walking normalized Laplacian smoothing[95], each row of the input feature in matrix $\mathbf{V}$ is updated as:

$$\mathbf{v}_i=(1-\lambda)\mathbf{v}_i+\lambda\sum_j^{N}\frac{\alpha_{i,j}}{d_i}\mathbf{v}_j \tag{26}$$

where $0<\lambda\leqslant 1$ is a parameter to control the smoothness, i.e., the importance weight of the node's features with respect to the features of its neighbors. We can rewrite the Laplacian smoothing in Eq. (26) as:

$$\mathbf{V}=\left(\mathbf{I}-\lambda\mathbf{D}^{-1}\mathbf{L}\right)\mathbf{V}=(\mathbf{I}-\lambda\mathbf{L}_{\text{rw}})\mathbf{V} \tag{27}$$

As $d_i=\sum_{j=1}^{N}\mathbf{A}_{i,j}=1$, then we have $\mathbf{D}=\mathbf{I}$ and $\mathbf{L}_{\text{rw}}=\mathbf{D}^{-1}\mathbf{L}=\mathbf{I}^{-1}(\mathbf{I}-\mathbf{A})=\mathbf{I}-\mathbf{A}$. Consequently, we can term that graph attention as a special form of the Laplacian smoothing with $\lambda=1$. ■

Theorem 1 Event-level graph collaborative learning suffers from the over-smoothing problem.

Proof For a non-bipartite, connected graph $\mathcal{G}$, there exists a random walk on $\mathcal{G}$ with transition probability matrix $\mathbf{P}$ that converges to a unique stationary

distribution π[96]. That is, if $\mathbf{A}$ is fixed at each layer, for any pair of node (v_i, v_j), $\lim_{t\to\infty}\mathbf{P}^{(t)}\left(v_i, v_j\right) = \pi\left(v_j\right) = d_{v_j}/\sum_{k=1}^{N} d_k$[90]. However, in practice, attention weight matrices vary in different layers. Stacking multiple attention layers is equivalent to a matrix chain multiplication with different attention-weight matrices. Let $\mathbf{A}_t$ be the attention matrix derived in t-th layer. Let f_i^{κ} be the i-th row of $\sum_{t=1}^{\kappa}\mathbf{A}_{\kappa}$, according to the converge analysis of random walk in Ref. [96], we have: $\left\|f_i^{\kappa}-\pi\right\| \leqslant \lambda_{\kappa}\left\|f_i^{\kappa-1}-\pi\right\| \leqslant \cdots \leqslant \Pi_{t=1}^{\kappa}\lambda_t\left\|f_i^{1}-\pi\right\|$, where λ_{κ} is the mixing rate of random walk with $\mathbf{A}_{\kappa}$. As we generate the event graph based on a certain event query, it is a query-dependent dynamic graph and each node has direct or indirect connections with the central event. Therefore, our event-level query-dependent graph is a strongly connected graph because if we start from a certain vertex (central event query) and use BFS (Breadth First Search) or DFS (Depth First Search) to conduct searches, we can traverse all vertices – every vertex is accessible from another vertex, fulfilling the definition of a strongly connected graph. Drawn conclusion from Ref. [96] that for strongly connected graph, the mixing rate $0 < \lambda_{\kappa} < 1$, we can derive that $\lim_{\kappa\to\infty} f_i^{\kappa} = \pi$. ■

Theorem 2 Additional contrastive constraints $\mathcal{L}^{cl}$ between the original view and the enriched graph view can mitigate the negative influence of the over-smoothing problem.

Proof Recent literature identifies two key properties related to contrastive learning: *alignment* and *uniformity*[97]. Given a distribution of positive pairs, alignment calculates the expected distance between the embeddings of all paired instances:

$$\mathcal{L}^{\text{align}} \triangleq \mathop{\mathbb{E}}_{(\boldsymbol{h}^{+},\boldsymbol{h})\sim p_{\text{pos}}}\left[\|\boldsymbol{h}^{+}-\boldsymbol{h}\|_2^2\right] \tag{28}$$

while uniformity measures how well the uniform distribution of the embeddings is:

$$\mathcal{L}^{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{\boldsymbol{h}_1,\boldsymbol{h}_2 \overset{i.i.d.}{\sim} p_{\text{data}}}\left[e^{-2\|\boldsymbol{h}_1-\boldsymbol{h}_2\|_2^2}\right] \tag{29}$$

In general contrastive learning, positive instances should stay close and the embeddings for random instances should scatter on the hypersphere[97]. Due to the uniformity property, the over-smoothing problem can be alleviated. ■

Theorem 2 states that when κ gets larger, the transition matrix from the attention mechanism becomes (nearly) static, thus the graph attention neural network on a strongly connected graph also suffers from the over-smoothing problem[90].

Discussion 1 *Contrastive learning can enhance the interpretability of graph learning.*

To illustrate how self-supervised contrastive learning boosts the interpretability of graph learning, in the following, we detail existing works on graph explainer. Specifically, previous works[91, 92] formalize the notion of importance using Mutual Information (MI) and formulate the graph explainer as an optimization learning task:

$$\max_{\mathcal{G}_S} \text{MI}\left(Y, \mathcal{G}_S\right) = H(Y) - H\left(Y \mid \mathcal{G}=\mathcal{G}_S\right) \tag{30}$$

where $\mathcal{G}_S$ is the explanation subgraph of the full TKG graph $\mathcal{G}$. Examining this equation, we can see that the entropy term $H(Y)$ is constant because the parameters Φ are fixed for a trained model. As a result, maximizing mutual information between predicted label distribution Y and explanation graph $\mathcal{G}_S$ is equivalent to minimizing conditional entropy $H\left(Y \mid \mathcal{G}=\mathcal{G}_S\right)$, which can be expressed as follows:

$$H\left(Y \mid \mathcal{G}=\mathcal{G}_S\right) = -\mathbb{E}_{Y\mid\mathcal{G}_S}\left[\log P_{\Phi}\left(Y \mid G=G_S\right)\right] \tag{31}$$

The explanation for prediction Y is thus a subgraph $\mathcal{G}_S$ that minimizes the uncertainty of the model when the graph scale is limited to $\mathcal{G}_S$. In practice, we can modify the conditional entropy objective in the above equation with a cross-entropy objective between the label class and the model prediction. Thus, the objective we optimize using gradient descent is rewritten as follows:

$$\min -\sum_{c=1}^{C}\mathbb{I}[y=c]\left[\log P_{\Phi}\left(Y=y \mid G=G_S\right)\right] \tag{32}$$

To obtain a compact explanation, existing methods impose an explicit constraint on $\mathcal{G}_S'$s size as $|\mathcal{G}_S| \leqslant K_N$, so that $\mathcal{G}_S$ has at most K_N nodes. Taking the additional constraint proposed in Ref. [92] as an example: they penalize the large size of the graph explainer by adding the sum of all elements in the graph as the regularization term. However, we argue that such a pre-defined constraint on the neighbor numbers might be inappropriate for various samples[98]. Our method, which leverages self-supervised contrastive learning to overcome the unexplainable problem, is much more effective than the explicit-constraint one.

In summary, the proposed self-supervised contrastive learning retains non-redundant information about the event, so as to prevent the model from learning

homogeneous representation, which, thereby, alleviates the over-smoothing problem. In addition, since the representation of the central event is the product of the message fusion between the original self-information and its concurrent neighbor events, maximizing mutual information (enhancing connections) between such two views is equivalent to aggregating information from neighbors as less as possible. In other words, this objective is a surrogate constraint that suggests the graph explainer output a small subgraph with minimally influential neighbors[99]. Apart from that, contrastive learning also augments the consistency between two different views, preventing one-sided judgments or over-reliance on structural information[98].

4.3 Inference and training

The followings are the details of event forecasting and network training. We also analyze the time complexity in data structures and algorithms.

4.3.1 Event forecasting

Given an event query $(s, r, ?, t)$, the inference process consists of several steps. First, we retrieve historical events from the event database to obtain the representation of historical dependency $\mathbf{F}_t^{s,r}$. Then, we use $\mathbf{F}_t^{s,r}$ to represent the timestamp t and generate the representation $\mathbf{q}$ of the event query via Eq. (3). Next, we feed the query vector into the *binary mode selector* to choose the corresponding module for event forecasting. We then filter out apparent inappropriate candidates and maintain the top K candidates. After that, for each candidate $o_i \in \tilde{\mathcal{E}}$, we substitute it into the event query to complete the event form as (s, r, o_i, t) and utilize a target-attentive collaborative learning mechanism to extract associations via Eqs. (14)–(17). At last, after getting the enriched event representation, the score of each candidate is defined as:

$$\mathbf{S}^q = \tanh(\mathbf{W}\mathbf{q} + \mathbf{b})\tilde{\mathbf{E}}_e^T \tag{33}$$

where $\tilde{\mathbf{E}}_e$ represents the embedding dict of the filtered set. The entity with the maximum value is the most likely entity the component predicts.

4.3.2 Network training

Predicting the entity (object) when given an event query can be viewed as a multi-class classification task, where each class corresponds to one object entity. Thus, the learning objective can be defined as minimizing the following loss:

$$\mathcal{L} = \mathcal{L}^{\text{ce}} + \alpha \mathcal{L}^{\text{cl}} \tag{34}$$

where α is a hyper-parameter and $\mathcal{L}^{ce}$ is the cross-entropy loss across two stages[†]. Considering the instability of the model in the early stage of training, we do not directly optimize the whole network simultaneously. Instead, we adopt a *curriculum* learning approach – first training the components belonging to the pre-rank stage until convergence, then training the components belonging to the re-rank stage.

4.3.3 Complexity analysis

The main computational overheads of TAG-Net lie on three parts: (1) historical dependency; (2) pre-rank stage; (3) re-rank stage.

Preprocessing historical dependency needs to traverse all events in the TKG to count the event frequency, which is $O(N \log N)$ complexity, here N is the number of training samples. During the **pre-rank stage**, we train three components: *mode selector*, *repeat predictor* and *explore predictor*. Since candidate score calculation is the dominant computational consumption in each component, the overall complexity of the pre-rank stage can be approximated as $O(|\mathcal{E}|d)$. The **re-rank stage** primarily contains two components: *self-supervised contrastive learning* and *target-attentive graph collaborative learning*. Compared to the latter one, the complexity of the former one can be omitted. In practice, by conducting a pre-filtering operation, target attentive graph learning only requires $O(|\tilde{\mathcal{E}}|d^2)$ complexity.

Generally, the complexity of our model is mainly from the data preprocessing stage and the re-rank stage. It is worth mentioning that our model is still much more efficient than previous RNN-based methods. Their computations of historical events are sequentially dependent, while our method can learn event interactions in parallel. As for concurrent event collaborative learning, the improved performance comes at the cost of additional computation time. We apply a pre-rank filtering strategy to reduce the number of candidates from $|\mathcal{E}|$ to $|\tilde{\mathcal{E}}|$ (dozens), significantly reducing the complexity without sacrificing forecasting performance.

5 Experiments

We now present evaluations of TAG-Net against baselines on event forecasting. After discussing the

[†]The binary mode selector ($\mathcal{L}^{\text{bs}}$) can be trained independently.

details of the experimental setup, we provide performance comparison results, ablation studies, timestamp influence, over-smoothing investigation, interpretability visualization, and parameter sensitivity.

5.1 Experimental settings

In this subsection, we introduce five event datasets, 13 baselines, two metrics and implementation details.

5.1.1 Data

We select five benchmark datasets including three event-based TKGs and two public TKGs. The former three event-based datasets consist of *Integrated Crisis Early Warning System* (ICEWS18)[100], ICEWS14[19] and *Global Database of Events, Language, and Tone* (GDELT)[21]. The latter two public KG-based datasets are WIKI[101] and YAGO[102]. In our experiments, we follow the preprocessing of CyGNet and split the datasets into training, validation and test sets in proportions of 80%, 10% and 10%[38, 74]. Table 3 summarizes the data statistics.

5.1.2 Baselines

We select the following 11 up-to-date *open-sourced* event forecasting (knowledge reasoning) models, including both static and temporal approaches. Note that we omit two recent works here: CEN[72] (uses length-aware CNNs to learn evolutional patterns with different lengths in a curriculum learning manner) and HGLS[73] (captures semantic information of events from sub-graph view and complete graph view) – due to prohibitive computational cost – out of memory (OOM) on a single RTX-3090.

• **TransE[29]:** is a classic knowledge base embedding learning model, which focuses on the minimal parameterization of the model to represent hierarchical relations.

• **DistMult[103]:** proposes a simple bilinear formuation, and the embeddings learned from the bilinear objective are particularly good at capturing the relational semantics.

• **ComplEx[104]:** describes a simple approach of matrix and tensor factorization for link prediction. It users vectors with complex values and retains the mathematical definition of the dot product.

• **R-GCN[66]:** shows that GCN framework can be applied to modeling relational data for link prediction and entity classification tasks.

• **ConvE[31]:** introduces a link prediction model that users 2D convolution over embeddings and nonlinear feature layers to model KGs.

• **TeMP[105]:** learns TKG completion by computing entity representation via joint modeling of multi-hop structural information and temporal facts from nearby timestamps.

• **RE-NET[38]:** proposes a method for modeling temporal, multi-relational and concurrent interactions between entities. It defines event joint probabilities and is able to infer graphs in a sequential manner.

• **xERTE[71]:** proposes an explainable reasoning approach for link prediction on TKGs. This model extracts a query-dependent subgraph from a given TKG and performs an attention propagation process to reason the subgraph.

• **TLogic[106]:** is based on temporal random walks in temporal knowledge graphs. It learns temporal logical rules from TKGs and applies these rules to the link forecasting task.

• **RE-GCN[39]:** reasons TKGs by learning evolution representations of entities and relations and capturing structural dependencies among concurrent facts and informative sequential patterns across temporal adjacent facts.

• **TANGO[70]:** proposes a multi-relational GCN layer to capture structural dependencies on TKGs and learn continuous dynamic representations using graph ODEs.

• **CyGNet[74]:** leverages the copy mechanism to tackle the event forecasting problem. It hypothesizes that a future fact can be predicted from the facts in history.

5.1.3 Metrics

We report the results of Mean Reciprocal Rank (MRR) and hits at 1/3/10 (H@1/3/10) in our evaluation. MRR is the average of the reciprocal of the rank of each test

Table 3 Statistics of five event datasets.

Dataset	# Entities	# Relations	# Training	# Validation	# Test	# Time gap
ICEWS18	23 033	256	373 018	45 995	49 545	1 day
ICEWS14	12 498	260	323 895	-	341 409	1 day
GDELT	7691	240	1 734 399	238 765	305 241	15 min
WIKI	12 554	24	539 286	67 538	63 110	1 year
YAGO	10 623	10	161 540	19 523	20 026	1 year

fact:

$$\mathrm{MRR} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(s,r,o,t)\in\mathcal{D}_{\text{test}}} \frac{1}{\mathrm{Rank}(s,r,o,t))} \tag{35}$$

where $|\mathcal{D}_{\text{test}}|$ is the size of the test set. H@K is calculated over top K items, i.e., the proportion of the correct recommended entities in the K previous positions in a ranking list: $H@K = n_{\text{hit}}/|\mathcal{D}_{\text{test}}|$, where n_{hit} is the number of times that the desired quadruple appears in the top K-ranked quadruples. Higher MRR and H@K indicate better performance.

To avoid counting higher ranks from other valid predictions as errors and having flaws in the metrics, we follow previous works[29, 38, 74] to remove all triples (except the triples of interest) that appear in the training, validation and test sets form the list of corrupted triples.

5.1.4 Implementation details

We set the dimension of embedding vectors to 200 (which is consistent with the experimental settings in CyGNet). We train our model using Adam optimizer with a learning rate of $1e-3$. We tune other hyperparameters using grid search for optimization within 30 epochs. We train parameters in the first stage with 15 epochs. Then we froze these parameters and train the second stage for another 15 epochs. The experiments are conducted on Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz, one NVIDIA GeForce RTX 3090, with 128GB CPU memory. We note that we are unable to run xERTE baseline on ICEWS14 and WIKI datasets due to resource constraints.

5.2 Comparison with prior models

The overall performance comparison between TAG-Net and baselines is shown in Tables 4 and 5, from which we have the following observations:

(O1): **Temporal-aware event forecasting methods outperform the static one.** In most cases, static KG-based models such as TransE and R-GCN generally performed worse than others and cannot meet the requirement for the event forecasting task. They rely on time-missing event triplets and are unable to capture the intrinsic temporal evolution patterns of events. In other words, the upper bound forecasting accuracy of the neural network will be restricted by the expected error minimization under limited static triplet information. In contrast to static methods that are unable to distinguish two similar events with different timestamps, temporal methods can capture temporally sequential patterns and thus, reason more accurately for unobserved timestamps. It suggests that additional consideration on time information encoding can help improve event forecasting performance.

(O2): **Baselines perform inconsistently across datasets.** Reviewing all existing methods, some of them are better at exploiting historical information, while others are good at enriching entity representations via leveraging structural information. However, none of them perform consistently well on

Table 4 Performance comparisons on three event-based TKGs. Best results are in bold font and the second-best results are underlined.

Method	ICEWS18				ICEWS14				GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE[29]	18.86	4.11	25.46	44.73	20.69	2.12	32.74	48.37	17.35	0.63	26.92	42.37
DistMult[103]	22.87	12.87	28.61	42.61	20.88	10.05	21.51	35.33	19.74	13.18	20.96	32.64
ComplEx[104]	32.57	21.05	34.18	45.78	24.93	18.52	27.71	42.59	22.58	15.89	24.68	37.77
R-GCN[66]	24.23	15.36	25.56	37.79	27.82	19.61	32.96	44.82	24.02	16.02	26.72	33.82
ConvE[31]	38.29	28.83	39.68	50.55	39.95	33.06	42.32	55.22	35.51	29.03	38.87	49.23
TeMP[105]	41.73	32.07	43.51	52.42	45.07	35.04	46.64	55.84	39.66	29.24	41.31	48.79
RE-NET[38]	41.03	37.03	47.83	56.68	46.03	37.12	49.76	58.66	42.33	34.81	43.87	55.68
xERTE[71]	37.75	30.89	41.42	51.66	34.81	25.09	35.73	46.81	–	–	–	–
TLogic[106]	37.52	30.09	40.87	52.27	38.19	32.23	41.05	49.58	22.73	17.65	24.66	32.59
RE-GCN[39]	31.45	25.79	35.63	50.28	33.37	24.45	36.23	50.78	28.77	23.27	34.67	44.94
TANGO-TuckER[70]	45.31	36.78	47.33	57.25	46.87	39.44	50.16	59.14	40.88	28.22	42.56	54.17
TANGO-Distmult[70]	44.23	37.66	46.07	56.14	46.62	43.75	47.36	59.17	41.02	35.81	43.22	53.75
CyGNet[74]	47.89	39.59	49.03	57.95	47.09	40.51	54.72	59.39	50.33	44.09	56.03	59.39
TAG-Net	**48.77**	**40.43**	**51.11**	**59.34**	**48.76**	**44.32**	**56.34**	**61.82**	**51.52**	**45.61**	**57.24**	**61.51**

Table 5 Performance comparisons on two public TKGs.

Method	WIKI			YAGO		
	MRR	H@1	H@3	MRR	H@1	H@3
TransE	46.66	38.59	51.64	50.03	48.26	61.02
DistMult	48.36	36.93	50.71	61.71	54.49	62.75
ComplEx	48.41	37.59	50.55	61.22	55.42	61.96
R-GCN	37.94	28.01	38.85	43.53	34.62	45.27
ConvE	46.95	40.17	49.09	62.01	56.91	63.59
TeMP	49.53	46.18	51.01	64.86	57.25	65.28
RE-NET	50.26	49.16	54.29	65.43	63.34	66.57
xERTE	–	–	–	59.92	58.35	60.32
TLogic	57.73	57.43	57.88	1.29	0.49	0.85
RE-GCN	43.92	39.34	47.78	64.08	58.99	68.92
TANGO-TuckER	53.51	<u>52.63</u>	<u>54.98</u>	67.54	66.39	69.66
TANGO-Distmult	<u>53.98</u>	52.25	54.02	<u>68.65</u>	<u>67.34</u>	<u>70.89</u>
CyGNet	49.03	46.56	51.41	64.36	62.81	65.57
TAG-Net	**68.14**	**67.51**	**68.10**	**83.38**	**82.27**	**84.33**

all datasets. For example, CyGNet investigates the repetition phenomenon of events occurrence and thereupon introduces copy mechanism to strengthen the influence of historical dependency. It outperforms other baselines on three event-related datasets, however, is inferior under two KG-based datasets YAGO and WIKI. Analogously, RE-GCN and RE-Net extract inherent spatial knowledge to updating entity representation, they perform well on YAGO but are unstable on other datasets. These observations are consistent with our expectations: The three event-based datasets have a small timestamp gap (see Table 3), so we need to explore the long-term historical dependency. In contrast, the timestamp gap of other KG-based datasets YAGO and WIKI are up to a year, so finding clues for event forecasting will be efficient and precise if we just focus on recent concurrent events (structural knowledge) rather than long-term historical dependency. To date, none of the existing methods can precisely understand the occurrence pattern of events.

(O3): **TAG-Net consistently outperforms all baselines for event forecasting.** Our approach outperforms baselines in terms of two evaluation metrics on all five datasets. For example, TAG-Net achieves up to 1.84%, 3.55%, and 2.37% improvements of MRR on three event-based TKGs datasets, respectively. Benefiting from the two-stage learning process, TAG-Net can model the historical information as well as the concurrent structural correlation more effectively. For KG datasets WIKI and YAGO, TAG-Net has 26.23% and 21.45% improvements of MRR against the best baseline. As discussed above, TAG-Net underscores the impact of the newest concurrent events via the target attentive graph learning mechanism, thus achieving significant improvements on WIKI and YAGO. Generally, these results suggest that our proposed method is effective for event pattern understanding and forecasting.

5.3 Ablation study

To investigate the contributions of each component in TAG-Net, we conduct ablation study by examining the performance change after removing each component. Specifically, the ablation study is conducted on two representative datasets: the KG-based YAGO dataset with a high repetition rate, and the event-based ICEWS18 dataset with a low repetition rate. We design three variants of TAG-Net:

• **w/o S1**: this variant removes the pre-rank stage from TAG-Net, which ignores the historical dependency.

• **w/o S2**: this variant removes the second re-rank stage from TAG-Net, which ignores the influence of the concurrent events.

• **GCN**: contrary to *TAG-Net w/o S2*, we maintain the pre-rank stage and just use vanilla GCN model to replace the target attentive graph learning net.

The results are shown in Table 6 and we have the following findings. First, when we discard the historical information and only focus on concurrent events (w/o S1), the performance drops drastically on both datasets. And, the pre-rank stage appears to have a

Table 6 Ablation study of TAG-Net.

Method	ICEWS18				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TAG-Net w/o S1	25.41	23.21	28.41	32.35	38.19	37.71	39.06	41.56
TAG-Net w/o S2	46.92	36.84	51.99	54.06	80.58	79.96	80.72	81.73
TAG-Net-GCN	48.12	40.07	51.73	59.38	80.36	80.26	80.73	81.33
TAG-Net	**48.77**	**40.43**	**51.11**	**59.34**	**83.38**	**82.27**	**84.33**	**84.44**

greater influence on KG-based datasets with less timestamp gap. Second, paying extra attention to the newest concurrent events will help the model enhance the forecasting accuracy, regardless of whether using a standard graph learning method (e.g., GCN) or the proposed target attentive graph learning mechanism. This phenomenon highlights the importance of recent knowledge developments when making decisions. Third, the proposed target attentive graph learning achieves better performance than vanilla GCN, indicating the superiority of our method.

5.4 Influence of time encoding

One of the pros of TAG-Net is to emphasize the importance of characterizing time information and accordingly propose an efficient time encoding technique. In this subsection, we compare TAG-Net with two variants that have different ways of time-encoding. Specifically, *TAG-Net w/o t* represents event query without timestamps, and *TAG-Net* Δt represents event query with time interval-based technique[74].

The prediction results are shown in Table 7. We can see that neglecting temporal information will significantly lower the event forecasting performance. As for TAG-Net Δt, its performance is close to TAG-Net, which, again, emphasizes the importance of modeling time information for event forecasting.

5.5 Effectiveness of contrastive learning

To verify whether the self-supervised contrastive learning approach can alleviate the over-smoothing problem and enhance the interpretability of the model, we visualize the attention scores before and after adding the contrastive constraint into the training objectives, as shown in Fig. 3. We can observe that, without the contrastive constraint, the model learns a trivial solution that aggregates information from surrounding neighbors almost uniformly. In contrast, after training with the proposed contrastive learning objective, the model only focuses on limited neighbors, in line with our assumptions and derivations. On one hand, it verifies that contrastive learning learns distinguished representation that can prevent the model from the over-smoothing problem; on the other, it shows that the model learned with contrastive constraints has a stronger interpretability than the original one.

5.6 Target attentive graph learning

Target attentive graph collaborative learning can improve the performance of the event forecasting task and, to provide more insight, we visualize the most influential neighbors to better understand the working principles. As in Table 8, we select an event query $(s, r, ?, t)$ from the test set as the focal event, and then calculate the cosine similarity between representations of the focal event and other events. We show events (three for each graph type) with the highest similarity scores by using a normal graph and two target attentive graphs (TAGs). We can observe that the similar events obtained by the normal graph confuse the object types – two of them are country names and the other is people names. When we use target attentive graphs, the objects of similar events are the same as the object of the focal event. As expected, the target attentive graph matches similar events with the target event for

Table 7 Influence of time encoding. Among them, TAG-Net w/o t denotes representing event query without considering timestamp; TAG-Net Δt denotes representing timestamp with the time interval-based technique.

Method	ICEWS18				YAGO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TAG-Net w/o t	30.16	23.13	35.36	46.82	63.35	58.96	64.22	67.61
TAG-Net Δt	46.96	37.43	49.95	55.93	76.34	73.94	77.33	76.87
TAG-Net	**48.77**	**40.43**	**51.11**	**59.34**	**83.38**	**82.27**	**84.33**	**84.44**

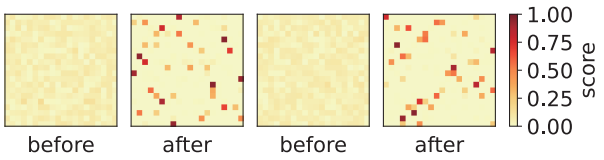


Fig. 3 We calculate the attention scores against neighbors (after 4 times message-passing) before and after adding the contrastive constraint into the training objectives. Each row represents the attention scores of a specific sample. Note that this process is a reflection of message passing, thus it can't reflect the coefficient of self-information.

obtaining better representation.

5.7 Parameter sensitivity

We conduct parameter sensitivity experiments for four important hyperparameters in TAG-Net: the number of candidates K, scalers α, λ, and η. As shown in Fig. 4, we adjust the values of hyperparameters and then observe the performance change of our model on YAGO and ICEWS18. The top K event candidates are the entities filtered out in the pre-rank stage. We select K from 50 to 300. We can see that larger K consistently improves the prediction performance. The improvements become saturated when K is sufficiently large. Due to resource limitations, we do not test K larger than 300. Scaler α controls the ratio of losses $\mathcal{L}^{ce}$ and $\mathcal{L}^{cl}$. When α increases from 0 to 1, the results on the two datasets first increase and then start to decrease. A suitable value of α is around 0.4. Scaler λ is a positive constant trading off the importance of the invariance term and the redundancy reduction term of loss $\mathcal{L}^{cl}$. The performance change trend of λ is similar to that of the α's, but the suitable value of λ is around 0.6. A larger value of η means that we pay more attention to historical entities. We alter the η's value from {2, 4, 6, 8, 10}, and the results suggest that 6 is a better choice for η.

Table 8 Case study on target attentive graph (TAG) collaborative learning net on ICEWS18.

Normal	Focal event	Donald Trump	Accuse	-
	Similar event	Donald Trump	Accuse	Russia
		Donald Trump	Accuse	Pakistan
		Donald Trump	Accuse	Vladimir Putin
TAG 1	**Focal event**	Donald Trump	Accuse	Russia
	Similar event	Donald Trump	Accuse	China
		Donald Trump	Accuse	Pakistan
		Donald Trump	Accuse	North Korea
TAG 2	**Focal event**	Donald Trump	Accuse	Vladimir Putin
	Similar event	Donald Trump	Accuse	Justin Trudeau
		Donald Trump	Accuse	Kim Jong-Un
		Donald Trump	Accuse	Bashar al-Assad

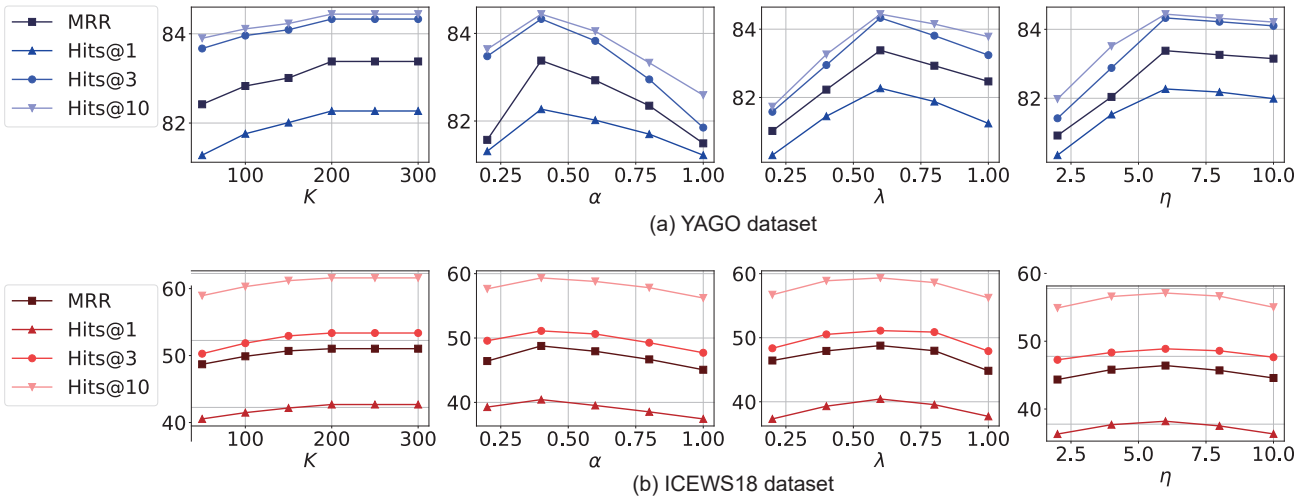


Fig. 4 Impact of four important parameters K, α, λ, and η.

6 Conclusion

In this work, we presented TAG-Net, a novel method for TKG-based event forecasting. To better understand event forecasting, we mimicked the cognitive process of humans and proposed a two-stage learning framework on the basis of the dual-process theory. In the first stage, we extracted temporal associations from historical dependency; in the second stage, we modeled the influence of the newest, unexpected, and emergent events. We devised a target attentive graph collaborative learning algorithm to explore the event-level neighborhood interactions. We evaluated TAG-Net over five real-world datasets, and the experimental results showed the superiority of our method.

In addition to the performance improvements our model achieved, we note that there are still space to further improve the performance of the proposed model. First, the target attentive graph collaborative learning network brings additional computational overhead, even with the proposed candidate-reducing strategy. Second, traditional event forecasting methods mainly focus on training a classifier in the closed-set world, where training and testing samples may share the same label space. Intuitively, the out-of-distribution or data shift problem occurs when we try to predict future events. To date, event forecasting is still a very challenging task and is worth more research investigations.

In our future work, we plan to enhance our method to address the above-mentioned challenges. First, we are seeking more efficient architectures and algorithms that can filter out irrelevant event candidates more precisely via some anomaly detection methods[107]. Second, we will consider incorporating uncertainty learning, adversarial training as well as self-supervised learning[108, 109] to improve the reliability of event forecasting methods on new events.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62176043 and 62072077), and the Grant SCITLAB-30002 of Intelligent Terminal Key Laboratory of Sichuan Province.

References

[1] G. Zeng, J. Zhuang, H. Huang, M. Tian, Y. Gao, Y. Liu, and X. Yu, Use of deep learning for continuous prediction of mortality for all admissions in intensive care units, *Tsinghua Science Technology.*, vol. 28, no. 4, pp. 639–648, 2023.

[2] X. Yang and J. A. Esquivel, Time-aware LSTM neural networks for dynamic personalized recommendation on business intelligence, *Tsinghua Science Technology*, vol. 29, no. 1, pp. 185–196, 2024.

[3] Z. Han, L. Shi, L. Liu, L. Jiang, J. Fang, F. Lin, J. Zhang, J. Panneerselvam, and N. Antonopoulos, A survey on event tracking in social media data streams, *Big Data Mining and Analytics*, vol. 7, no. 1, pp. 217–243, 2024.

[4] Y. Peng, S. Xu, Q. Chen, W. Huang, and Y. Huang, A novel popularity extraction method applied in session-based recommendation, *Tsinghua Science Technology*, vol. 29, no. 4, pp. 971–984, 2024.

[5] F. Pappenberger, J. Bartholmes, J. Thielen, H. L. Cloke, R. Buizza, and A. de Roo, New dimensions in early flood warning across the globe using grand-ensemble weather predictions, *Geophys. Res. Lett.*, vol. 35, no. 10, pp. L10404, 2008.

[6] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, A survey of information cascade analysis: Models, predictions, and recent advances, *ACM Comput. Surv.*, vol. 54, no. 2, pp. 27, 2021.

[7] X. Xu, F. Zhou, K. Zhang, and S. Liu, CCGL: contrastive cascade graph learning, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4539–4554, 2023.

[8] V. Kumar, S. S. Saheb, Preeti, A. Ghayas, S. Kumari, J. K. Chandel, S. K. Pandey, and S. Kumar, AI-based hybrid models for predicting loan risk in the banking sector, *Big Data Mining and Analytics*, vol. 6, no. 4, pp. 478–490, 2023.

[9] W. Tai, T. Zhong, Y. Mo, and F. Zhou, Learning sentimental and financial signals with normalizing flows for stock movement prediction, *IEEE Signal Process. Lett.*, vol. 29, pp. 414–418, 2021.

[10] W. Lam, H. M. L. Meng, K. L. Wong, and J. C. H. Yen, Using contextual analysis for news event detection, *Int. J. Intell. Syst.*, vol. 16, no. 4, pp. 525–546, 2001.

[11] X. Chen, F. Zhou, F. Zhang, and M. Bonsangue, Catch me if you can: A participant-level rumor detection framework via fine-grained user representation learning, *Inf. Process. Manag.*, vol. 58, no. 5, pp. 102678, 2021.

[12] J. Yu, Y. Lee, K. C. Yow, M. Jeon, and W. Pedrycz, Abnormal event detection and localization via adversarial event prediction, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3572–3586, 2022.

[13] S. Mehrmolaei and M. R. Keyvanpour, An enhanced hybrid model for event prediction in healthcare time series, *Int. J. Knowl. Based Intell. Eng. Syst.*, vol. 23, no. 3, pp. 131–147, 2019.

[14] M. F. Myers, D. J. Rogers, J. Cox, A. Flahault, and S. I. Hay, Forecasting disease risk for increased epidemic preparedness in public health, *Adv. Parasitol.*, vol. 47, pp. 309–330, 2000.

[15] Y. Zhang, M. Sheng, R. Zhou, Y. Wang, G. Han, H. Zhang, C. Xing, and J. Dong, HKGB: an inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians' expertise

incorporated, *Inf. Process. Manag*., vol. 57, no. 6, p. 102324, 2020.

[16] C. Sang-Hun, J. Yoon, P. Mozur, V. Kim, L. Su-Hyun, and J. Y. Young, A 'sea of bodies': How a festive night in seoul turned deadly, *The New York Times*, Oct, 2022.

[17] L. Zhao, Event prediction in the big data era: A systematic survey, *ACM Comput. Surv.*, vol. 54, no. 5, pp. 94, 2021.

[18] X. Xu, T. Qian, Z. Xiao, N. Zhang, J. Wu, and F. Zhou, PGSL: A probabilistic graph diffusion model for source localization, *Expert Syst. Appl*., vol. 238, p. 122028, 2024.

[19] R. Trivedi, H. Dai, Y. Wang, and L. Song, Know-evolve: Deep temporal reasoning for dynamic knowledge graphs, in *Proc. 34th Int. Conf. Machine Learning*, Sydney, NSW, Australia, 2017, pp. 3462–3471.

[20] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al., 'Beating the news' with EMBERS: Forecasting civil unrest using open source indicators, in *Proc. 20th ACM SIGKDD Int. Conf. Knowledge discovery and data mining*. New York, USA., 2014, pp. 1799–1808.

[21] K. Leetaru and P. A. Schrodt, Gdelt: Global data on events, location, and tone, 1979–2012, in *Proc. ISA Annual Convention*, Citeseer, 2013, pp. 1–49.

[22] A. Desantis, C. Roussel, and F. Waszak, The temporal dynamics of the perceptual consequences of action-effect prediction, *Cognition*, vol. 132, no. 3, pp. 243–250, 2014.

[23] C. Esteban, O. Staeck, S. Baier, Y. Yang, and V. Tresp, Predicting clinical events by combining static and dynamic information using recurrent neural networks, in *Proc. IEEE Int. Conf. Healthcare Informatics* (*ICHI*), Chicago, IL, USA, 2016, pp. 93–101.

[24] I. Mele, S. Ali Bahrainian, and F. Crestani, Event mining and timeliness analysis from heterogeneous news streams, *Inf. Process. Manag. Int. J.*, vol. 56, no. 3, pp. 969–993, 2019.

[25] X. Zhang and W. Gao, Predicting viral rumors and vulnerable users with graph-based neural multi-task learning for infodemic surveillance, *Inf. Process. Manag.*, vol. 61, no. 1, pp. 103520, 2024.

[26] J. O'Madadhain, J. Hutchins, and P. Smyth, Prediction and ranking algorithms for event-based network data, *SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 23–30, 2005.

[27] S. Van Landeghem, S. Pyysalo, T. Ohta, and Y. Van de Peer, Integration of static relations to enhance event extraction from text, in *Proc. 2010 Workshop on Biomedical Natural Language Processing*, Uppsala, Sweden, 2010, pp. 144–152.

[28] L. Zhuang, H. Fei, and P. Hu, Syntax-based dynamic latent graph for event relation extraction, *Inf. Process. Manag.*, vol. 60, no. 5, pp. 103469, 2023.

[29] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Proc. 27th Conf. Neural Information Processing Systems*, Lake Thoe, Nevada, USA, 2013, pp. 2787–2795.

[30] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in *Int. Conf. Learn. Represent*., 2018.

[31] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, Convolutional 2D knowledge graph embeddings, in *Proc. Thirty-Second AAAI Conf.* on *Artificial Intelligence*, New Orleans, Louisiana, USA, 2018, pp. 1811–1818.

[32] L. Zhao, F. Chen, C.-T. Lu, and N. Ramakrishnan, Spatiotemporal event forecasting in social media, *in Proc. 2015 SIAM Int. Conf. Data Mining*., Philadelphia, PA, USA, 2015, pp. 963–971.

[33] G. Atluri, A. Karpatne, and V. Kumar, Spatio-temporal data mining: A survey of problems and methods, *ACM Comput. Surv.*, vol. 51, no. 4, pp. 83, 2018.

[34] J. Tuke, A. Nguyen, M. Nasim, D. Mellor, A. Wickramasinghe, N. Bean, and L. Mitchell, Pachinko Prediction: A Bayesian method for event prediction from social media data, *Inf. Process. Manag.*, vol. 57, no. 2, pp. 102147, 2020.

[35] G. Liao, X. Deng, C. Wan, and X. Liu, Group event recommendation based on graph multi-head attention network combining explicit and implicit information, *Inf. Process. Manag.*, vol. 59, no. 2, pp. 102797, 2022.

[36] P. VELIČKOVIĆ, G. Cucurull, A. Casanova, A. Romero, P. LIÒ, and Y. Bengio, Graph attention networks, in *Proc. 6th Int. Conf. on Learning Representations*, Vancouver, Canada, 2018.

[37] L. R. Medsker and L. Jain, Recurrent neural networks, *Des. Appl.*, vol. 5, no. 2, pp. 64–67, 2001.

[38] W. Jin, M. Qu, X. Jin, and X. Ren, Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Stroudsburg, PA, USA, 2020, pp. 6669–6683.

[39] 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 *Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, Canada, 2021, pp. 408–417.

[40] 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 *Proc. Findings of the Association for Computational Linguistics*: *EMNLP 2022*, Abu Dhabi, United Arab Emirates, 2022, pp. 7328–7338.

[41] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, Neural attentive session-based recommendation, in *Proc. 2017 ACM on Conf. Information and Knowledge Management*, Singapore, 2017, pp. 1419–1428.

[42] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, and G. Chen, Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems, in *Proc. WWW '19*: *The World Wide Web Conf*., San Francisco, CA, USA, 2019, pp. 2091–2102.

[43] K. Liu, F. Zhao, H. Chen, Y. Li, G. Xu, and H. Jin, DA-Net: Distributed attention network for temporal knowledge graph reasoning, in *Proc. 31st ACM Int. Conf.*

Information & Knowledge Management, Atlanta, GA, USA, 2022, pp. 1289–1298.

[44] Z. He, B. Hui, S. Zhang, C. Xiao, T. Zhong, and F. Zhou, Exploring indirect entity relations for knowledge graph enhanced recommender system, *Expert Syst. Appl.*, vol. 213, pp. 118984, 2023.

[45] J. Gastinger, T. Sztyler, L. Sharma, A. Schuelke, and H. Stuckenschmidt, Comparing apples and Oranges? On the Evaluation of Methods for Temporal knowledge graph forecasting. *Machine Learning and Knowledge Discovery in Databases*: *Research Track*. Cham: Springer Nature Switzerland, 2023, pp. 533–549.

[46] S. Deng and Y. Ning, A survey on societal event forecasting with deep learning, arXiv preprint arXiv:2112.06345, 2021.

[47] J. St B. T. Evans, Heuristic and analytic processes in reasoning, *Br. J. Psychol.*, vol. 75, no. 4, pp. 451–468, 1984.

[48] A. P. Banks and C. Hope, Heuristic and analytic processes in reasoning: An event-related potential study of belief bias, *Psychophysiology*, vol. 51, no. 3, pp. 290–297, 2014.

[49] M. E. Roser, J. St. B. T. Evans, N. A. McNair, G. Fuggetta, S. J. Handley, L. S. Carroll, and D. Trippas, Investigating reasoning with multiple integrated neuroscientific methods, *Front. Hum. Neurosci.*, vol. 9, pp. 41, 2015.

[50] Z. Li, X. Jin, S. Guan, W. Li, J. Guo, Y. Wang, and X. Cheng, Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs, in *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. Natural Language Processing*, Virtual Event, 2021, pp. 4732–4743.

[51] X. Wang, H. Chen, Z. Li, and Z. Zhao, Unrest news amount prediction with context-aware attention LSTM, Geng X and Kang BH, *Pacific Rim International Conference on Artificial Intelligence*, Cham: Springer, 2018, pp. 369–377.

[52] F. Qiao, P. Li, X. Zhang, Z. Ding, J. Cheng, and H. Wang, Predicting social unrest events with hidden Markov models using GDELT, *Discrete Dyn. Nat. Soc.*, vol. 2017, pp. 8180272, 2017.

[53] F. Qiao, X. Zhang, and J. Deng, Learning evolutionary stages with hidden semi-markov model for predicting social unrest events, *Discrete Dyn. Nat. Soc.*, vol. 2020, pp. 3915036, 2020.

[54] G. Schneider, M. Bussmann, and C. Ruhe, The dynamics of mass killings: Testing time-series models of one-sided violence in the bosnian civil war, *Int. Interact.*, vol. 38, no. 4, pp. 443–461, 2012.

[55] N. B. Weidmann and M. D. Ward, Predicting conflict in space and time, *J. Confl. Resolut.*, vol. 54, no. 6, pp. 883–901, 2010.

[56] E. M. Smith, J. Smith, P. Legg, and S. Francis, Predicting the occurrence of world news events using recurrent neural networks and auto-regressive moving average models, Chao F, Schockaert S, and Zhang Q, *UK Workshop on Computational Intelligence*, Cham: Springer, 2018, pp. 191–202.

[57] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[58] H. Kiyomaru, K. Omura, Y. Murawaki, D. Kawahara, and S. Kurohashi, Diversity-aware event prediction based on a conditional variational autoencoder with reconstruction, in *Proc. First Workshop on Commonsense Inference in Natural Language Processing*, Hong Kong, China, 2019, pp. 113–122.

[59] H. Duan, Z. Sun, W. Dong, K. He, and Z. Huang, On clinical event prediction in patient treatment trajectory using longitudinal electronic health records, *IEEE J. Biomed. Health Inform.*, vol. 24, no. 7, pp. 2053–2063, 2020.

[60] A. M. Ertugrul, Y.-R. Lin, W.-T. Chung, M. Yan, and A. Li, Activism via attention: Interpretable spatiotemporal learning to forecast protest activities, *EPJ Data Sci.*, vol. 8, pp. 5, 2019.

[61] X. Chen, S. Jia, and Y. Xiang, A review: Knowledge reasoning over knowledge graph, *Expert Syst. Appl.*, vol. 141, pp. 112948, 2020.

[62] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, 2022.

[63] Z. Xie, R. Zhu, J. Liu, G. Zhou, and J. X. Huang, An efficiency relation-specific graph transformation network for knowledge graph representation learning, *Inf. Process. Manag.*, vol. 59, no. 6, pp. 103076, 2022.

[64] Z. Li, S. Feng, J. Shi, Y. Zhou, Y. Liao, Y. Yang, Y. Li, N. Yu, and X. Shao, Future event prediction based on temporal knowledge graph embedding, *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2411–2423, 2023.

[65] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, Simplifying graph convolutional networks, in *Proc. 36th Int. Conf. on Machine Learning*, California, USA, 2019, pp. 6861–6871.

[66] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, Modeling relational data with graph convolutional networks, *European Semantic Web Conference*, Cham: Springer, 2018, pp. 593–607.

[67] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 3060–3067, 2019.

[68] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang, A vectorized relational graph convolutional network for multi-relational network alignment, in *Proc. Twenty-Eighth Int. Joint Conf. Artificial Intelligence*, Macao, China, 2019, pp. 4135–4141.

[69] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, Composition-based multi-relational graph convolutional networks, in *Proc. 8th Int. Conf. on Learning Representations*, Virtual Event, 2020.

[70] Z. Han, Z. Ding, Y. Ma, Y. Gu, and V. Tresp, Learning neural ordinary equations for forecasting future links on

temporal knowledge graphs, in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing*, Virtual Event, 2021, pp. 8352–8364.

[71] Z. Han, P. Chen, Y. Ma, and V. Tresp, Explainable subgraph reasoning for forecasting on temporal knowledge graphs, in *Proc. 9th Int. Conf. on Learning Representations*, Virtual Event, 2021.

[72] 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 *Proc. 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, 2022, pp. 290–296.

[73] M. Zhang, Y. Xia, Q. Liu, S. Wu, and L. Wang, Learning long- and short-term representations for temporal knowledge graph reasoning, in *Proc. ACM Web Conf. 2023*, Austin, TX, USA, 2023, pp. 2412–2422.

[74] C. Zhu, M. Chen, C. Fan, G. Cheng, and Y. Zhang, Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, pp. 4732–4740, 2021.

[75] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, A brief overview of ChatGPT: The history, status quo and potential future development, *IEEE/CAA J. Autom. Sin.*, vol. 10, no. 5, pp. 1122–1136, 2023.

[76] X. Shi, S. Xue, K. Wang, F. Zhou, J. Y. Zhang, J. Zhou, C. Tan, and H. Mei, Language models can improve event prediction by few-shot abductive reasoning, in *Proc. 37th Conf. Neural Information Processing Systems*, New Orleans, LA, USA, 2023, pp. 29532–29557.

[77] C. Yuan, Q. Xie, J. Huang, and S. Ananiadou, Back to the future: Towards explainable temporal reasoning with large language models, arXiv preprint arXiv:2310.01074, 2023.

[78] J. C. Gomez and M. M. Morcos, Voltage Sag and recovery time in repetitive events, *IEEE Trans. Power Deliv.*, vol. 17, no. 4, pp. 1037–1043, 2002.

[79] C. Guo, T. Wang, Y. Lin, H. Chen, H. Yu, C. Zhu, and J. Qiu, Modeling unseen entities from a semantic evidence view in temporal knowledge graphs, in *Proc. 7th IEEE Int. Conf. Data Science in Cyberspace*, Guilin, China, 2022, pp. 333–339.

[80] J. Weston, F. Ratle, and R. Collobert, Deep learning via semi-supervised embedding, in *Proc. 25th Int. Conf. Machine learning*, Helsinki, Finland, 2008, pp. 1168–1175.

[81] N. Park, F. Liu, P. Mehta, D. Cristofor, C. Faloutsos, and Y. Dong, EvoKG: jointly modeling event time and network structure for reasoning over temporal knowledge graphs, in *Proc. 15th ACM Int. Conf. Web Search and Data Mining*, Virtual Event, 2022, pp. 794–803.

[82] J. Gu, Z. Lu, H. Li, and V. O. K. Li, Incorporating copying mechanism in sequence-to-sequence learning, in *Proc. 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1631–1640.

[83] Y. Xu, J. Ou, H. Xu, and L. Fu, Temporal knowledge graph reasoning with historical contrastive learning, *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 4, pp. 4765–4773, 2023.

[84] J. Li, L. Meng, Z. Zhang, and K. Yang, Low-frequency, high-impact: Discovering important rare events from UGC, *J. Retail. Consum. Serv.*, vol. 70, pp. 103153, 2023.

[85] E. J. N. Stupple, L. J. Ball, J. St. B. T. Evans, and E. Kamal-Smith, When logic and belief collide: Individual differences in reasoning times support a selective processing model, *J. Cogn. Psychol.*, vol. 23, no. 8, pp. 931–941, 2011.

[86] K. C. Klauer, J. Musch, and B. Naumer, On belief bias in syllogistic reasoning, *Psychol. Rev.*, vol. 107, no. 4, pp. 852–884, 2000.

[87] Y. Yang, Z. Wei, Q. Chen, and L. Wu, Using external knowledge for financial event prediction based on graph neural networks, in *Proc. 28th ACM Int. Conf. Information and Knowledge Management*, Beijing, China, 2019, pp. 2161–2164.

[88] K. Zhang, Y. Zhu, J.Wang, and J. Zhang, Adaptive structural fingerprints for graph attention net-works, in Proc. 7th Int. Conf. on Learning Representations, Virtual Event, 2019.

[89] M. Liu, H. Gao, and S. Ji, Towards deeper graph neural networks, in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 338–348.

[90] G. Wang, R. Ying, J. Huang, and J. Leskovec, Improving graph attention networks with large margin-based constraints, arXiv preprint arXiv:1910.11945, 2019.

[91] H. Yuan, J. Tang, X. Hu, and S. Ji, XGNN: towards model-level explanations of graph neural networks, in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 430–438.

[92] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, GNNExplainer: generating explanations for graph neural networks, in *Proc. Adv. Neural Inf. Process. Syst*., Vancouver, Canada, 2019, pp. 9240–9251.

[93] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, Barlow twins: Self-supervised learning via redundancy reduction, in *Proc. 38th Int. Conf. on Machine Learning*, Virtual Event, 2021, pp. 12310–12320.

[94] J. Xia, L. Wu, G. Wang, J. Chen, and S. Z. Li, Progcl: Rethinking hard negative mining in graph contrastive learning, in *Proc. 39th Int. Conf. on Machine Learning*, Virtual Event, 2022, pp. 24332–24346.

[95] Q. Li, Z. Han, and X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 3538–3545, 2018.

[96] D. Randall, Rapidly mixing Markov chains with applications in computer science and physics, *Comput. Sci. Eng.*, vol. 8, no. 2, pp. 30–41, 2006.

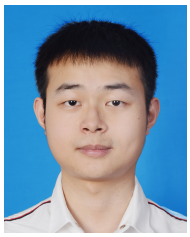
[97] T. Wang and P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in *Proc. 37th Int. Conf. Machine*

Learning, Virtual Event, 2020, pp. 9929–9939.
[98] Q. Wen, Z. Ouyang, C. Zhang, Y. Qian, Y. Ye, and C. Zhang, Graph contrastive learning with cross-view reconstruction, in *Proc. 36th Conf. on Neural Information Processing Systems*, New Orleans, LA, USA, 2022.
[99] T. M. Cover and J. A. Thomas, Entropy, relative entropy, and mutual information, *Elem. Inf. Theory*, vol. 2no.1, pp. 13–55, 2005.
[100] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, and M. Ward, Icews coded event data, *Harvard Dataverse*, vol. 12, 2015.
[101] J. Leblay and M. W. Chekol, Deriving validity time in knowledge graph, in *Proc. ACM Web Conf.*, Lyon, France, 2018, pp. 1771–1776.
[102] F. Mahdisoltani, J. Biega, and F. Suchanek, Yago3: A knowledge base from multilingual wikipedias, in *Proc. 7th Biennial Conf. on Innovative Data Systems Research*, Asilomar, CA, USA, 2014.
[103] B. Yang, S. W-t. Yih, X. He, J. Gao, and L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in *Proc. Int. Conf. on Learning Representations*, 2015.
[104] T. Trouillon, J.Welbl, S. Riedel, ′ E. Gaussier, and G. Bouchard, Complex embeddings for simple link prediction, in *Proc. 33rd Int. Conf. on Machine Learning*, pp. 2071–2080, PMLR, 2016, pp. 2071–2080.
[105] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, TeMP: Temporal message passing for temporal knowledge graph completion, in *Proc. 2020 Conf. Empirical Methods Natural Language Processing*, Stroudsburg, PA, USA, 2020, pp. 5730–5746.
[106] Y. Liu, Y. Ma, M. Hildebrandt, M. Joblin, and V. Tresp, TLogic: temporal logical rules for explainable link forecasting on temporal knowledge graphs, *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 4, pp. 4120–4127, 2022.
[107] F. Zhou, G. Wang, K. Zhang, S. Liu, and T. Zhong, Semi-supervised anomaly detection via neural process, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10423–10435, 2023.
[108] F. Zhou, P. Wang, X. Xu, W. Tai, and G. Trajcevski, Contrastive trajectory learning for tour recommendation, *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 1, pp. 4, 2021.
[109] W. Tai, T. Lan, Z. Wu, P. Wang, Y. Wang, and F. Zhou, Improving session-based recommendation with contrastive learning, *User Model. User Adapt. Interact.*, vol. 33, no. 1, pp. 1–42, 2023.

Xin Liu is currently a PhD candidate in computer science from University of Electronic Science and Technology of China (UESTC). His research interests include knowledge graphs, machine learning, graph neural networks, natural language processing, data mining and knowledge discovery.

Wenxin Tai is currently a second-year PhD student in software engineering, UESTC. He received the BS degree in communication engineering and MS degree in software engineering at UESTC in 2019 and 2022, respectively. His research interests include controllable generative probabilistic models and trustworthy artificial intelligence.

Fan Zhou received the BS degree in computer science from Sichuan University, China, in 2003, and the MS and PhD degrees from the UESTC, in 2006 and 2011, respectively. He is currently a full professor with the School of Information and Software Engineering, UESTC. His research interests include machine learning and social network knowledge discovery.



Yi He received the BS degree in software engineering from UESTC, 2022. He is currently pursuing a MS degree in software engineering at UESTC. His research interests include graph neural networks, natural language processing, data mining and knowledge discovery.

Xovee Xu received the BS degree and MS degree in software engineering from UESTC in 2018 and 2021, respectively, where he is currently pursuing the PhD degree in computer science. His research focuses on understanding information diffusion, user-generated content, and human behaviors in social network.

Guangchun Luo received the PhD degree in computer science from UESTC, China, in 2004. He is currently a professor of computer science at UESTC. His research interests include computer networks and big data.