Graphical Abstract

Spatial-temporal Knowledge Graph Network for Event Prediction

Zepeng Huai, Dawei zhang, Guohua Yang, Jianhua Tao

Highlights

Spatial-temporal Knowledge Graph Network for Event Prediction

Zepeng Huai, Dawei zhang, Guohua Yang, Jianhua Tao

- To model the trans-regional reason for the occurrence of events, which means the cause of the incident is not local but somewhere else, we construct a novel spatial-temporal knowledge graph for event prediction, which both holds the trans-regional influence and the time sequence pattern.
- To model the continuously evolving process of the real-life world, we propose a continuous-time dynamic graph neural network to simulate and forecast the development of entities, which improves time sensitivity and robustness.

Spatial-temporal Knowledge Graph Network for Event Prediction

Zepeng Huai[a,b], Dawei zhang[a,b], Guohua Yang[a,b], Jianhua Tao[c]

[a] *School of Artificial Intelligence, University of Chinese Academy of Sciences , Beijing, China*
[b] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Beijing, China*
[c] *Department of Automation, Tsinghua University, Beijing, China*

Abstract

Predicting multiple concurrent events has a remarkable effect on understanding social dynamics and acting in advance to reduce damage. (1) From the perspective of spatial connection, trans-regional implication, which means the cause of the incident is not local but somewhere else, is an important reason for the occurrence of events. However, existing works neglect to model this spatial influence and only leverage the local information for event prediction. (2) From the perspective of temporal connection, future events are triggered by the continuous evolution of the region. Nonetheless, most studies assign events to different timestamps and recognize their sequential patterns, ignoring the continuity of the evolution process. To tackle the above two problems, we propose a **s**patial and **t**emporal **k**nowledge **g**raph neural **n**etwork (*STKGN*). Specifically, to construct the cross-regional connection, we propose a novel spatial-temporal event graph, where each region is denoted as a node and trans-regional influences are reflected by bidirectional edges. To simulate the continuously evolving process, we propose an event-driven memory network to represent the state of each entity and continually update the state embeddings by emerging events. Then we use a broadcast network to spread the local changes in the graph to obtain high-order reasons like the trans-regional implication. Extensive experiments on two real-world datasets demonstrate that STKGN achieves significant improvements over state-of-the-art methods. Further analysis shows the interpretability of the trans-regional implication.

Keywords: Multi-Event Prediction, Knowledge Graph, Dynamic Graph

1. Introduction

Social events such as organized crime, arrest, and protest parades show high occurrences in specific locations, times, and semantics. Predicting multiple co-occurring events of different types, also known as multi-event prediction, can reduce the potential social upheaval and damage, which meets the urgent need of the government. Early methods focus on mining specific patterns of the given event type, where each pattern is defined as a burst of context features. To build the mapping function from the feature to the occurrence of events, different algorithms (e.g. linear regression [3, 46] and multi-task learning [33, 51, 10, 11]) are adopted to model the underlying relations as indicators of ongoing or future events. However, they neglect to discover the hidden relational knowledge among entities.

Recently, Graph Neural Networks (GNN) have been widely researched to address non-euclidean data in many domains such as recommender systems [19], natural language processing [30], and protein interface prediction [9]. To explore the semantic correlation between entities, some works [7, 8, 39] utilize the temporal knowledge graphs to represent events extracted from formal reports or news articles by regarding subjects and objects as nodes and event types as edges, which encodes temporal text features into graphs and helps for forecasting. They follow the discrete-time dynamic graph neural network (DGNN) pattern, which means a list of graph snapshots taken at intervals in time is constructed to model the sequences of previous events. Technologically, a knowledge graph neural network, like CompGCN [44], is introduced to encode each snapshot, followed by an RNN-based module to learn the temporal patterns. Due to the advantages of GNN in better semantic embedding, these discrete-time dynamic graph based methods achieve the SOTA prediction performances. However, they neglect a key issue: **The real world is continuous and evolving (Q1)**, which means the state of an entity should continuously change, and its development should be driven by concerned events. Specifically, the state of an entity in the current timestamp is supposed to be inferred based on its previous state and emerging events. However, past studies leverage discrete-time dynamic graphs to learn entity embeddings in each timestamp, leading to a set of discrete embeddings of the same node in different timestamps. This kind of graph representation

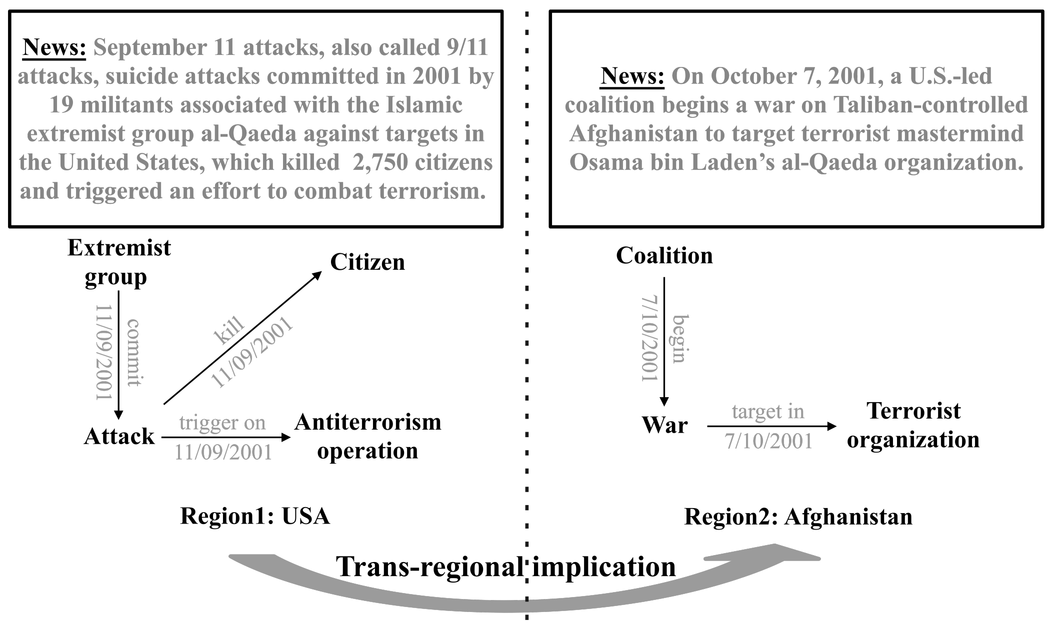


Figure 1: An example of trans-regional implications behind the occurrences of events.

learning algorithm fails to model the evolving process of the natural world and is the sub-optimal way to simulate real-life systems, which has a similar consensus in Jodie [24], TGN [36] and TGAT [50].

Another close solution is temporal knowledge graph reasoning[53, 43], where extrapolation-based methods [21, 26] aim to guess new facts at future timestamps. The problem formulation of them is predicting the missing entity(relation) given the relation(entity), a.k.a. predicting the rest given partial information of the quadruple, such as EvoKG [35], CluSTeR [25]. However, when forecasting future events in practical scenarios, it is usually challenging to be aware of the subject and object in advance, such that they are also sub-optimal methods.

What's more, we argue that a vital problem has not been considered: **there are trans-regional implications behind the occurrences of events (Q2)**. The reason why a set of events happen in one district is not here but in other regions, especially for some first-time events. Take Figure 1 as an example. The 9/11 events in the USA triggered a war and caused a chain of conflict events in Afghanistan, revealing the geopolitical influence. However, existing methods are self-contained in each region, and

Table 1: The comparison of STKGN with recent models. SP (spatial pattern) indicates whether the model takes into account the trans-regional implications, and the result is represented by ✓and ×. TP (temporal pattern) means the mothod for sequential feature extraction. IM denotes this model doesn't explicitly learn the sqeuential signals and just share one network to map current features to a future event subtype. DT and CT means discrete-time DGNN and continuous-time DGNN, respectively.

	SIMDA	RENET	Glean	DGCN-rs	STKGN
SP	✓	×	×	×	✓
TP	IM	DT	DT	DT	CT

the judgment of whether an event will occur is only based on what happened in the same region, which fails to consider the reasoning of spatial relations and creates a critical bottleneck to improving prediction performance.

In order to concurrently address all these technical challenges, this work presents a spatial-temporal knowledge graph network(STKGN) for event prediction. To solve **Q2**, we add the region information to temporal knowledge graphs as a set of nodes. Consequently, the trans-regional influence is naturally held in this novel knowledge graph. To address **Q1**, we design an event-driven continuous-time dynamic graph neural network, including memory and broadcast networks. The memory network, whose update is driven by events, aims to simulate the gradual development of entities followed by the broadcast network, by which the local changes propagate over the graph to explore the high-order influences on evolution. In order to better understand the differences between STKGN and recent models for event prediction, we compared them in spatial and temporal dimensions, which are reported in Table 1. To summarize, the main contributions of this work are as follows:

- We construct a novel spatial-temporal knowledge graph for event prediction, which both holds the trans-regional influence and the time sequence pattern.
- We propose a continuous-time dynamic graph neural network to simulate and forecast the evolving process of entities.
- We conduct comprehensive experiments on two public datasets, which proves the effectiveness and interpretability of our proposed model.

2. Preliminaries

In this section, we introduce the newly proposed spatial-temporal event graph used in this paper and then formulate the problem with some notations.

2.1. Spatial-temporal Event Graph

We propose a novel event graph $\mathcal{G}$ to hold the spatial and temporal information. Let $\mathcal{E}$, $\mathcal{L}$ and $\mathcal{R}$ denote the set of entities, locations, and event types, respectively. When an event $ev_{so}(t)$, which belongs to event type r and is associated with the subject s and object o, happens in location l at time t, we use three quadruples[1] to record this history: (s, r, t, o), (s, r, t, l) and (o, r, t, l). To this end, the event type and temporal feature are represented by edges, while the spatial information is contained in nodes. To represent spatial influence, we add a new edge relation called *trans-regional influence* between each pair of locations, which is symmetric and exists in all timestamps. Figure 2 is an example to show the construction of the above spatial-temporal event graph using the news in Figure 1. Different from the discrete-time dynamic graph, which establishes a sequence of event graphs in ascending time order like $\{\mathcal{G}(t_1), \mathcal{G}(t_2), ...\}$, our proposed continuous-time graph continually changes the topology of the graph via the addition or deletion of node and edge. Specifically, the $i-$th node representation $\mathbf{e}_i(t)$ at timestamp t is inferred based on its previous embedding $\mathbf{e}_i(t-1)$ and the associated events at the current timestamp $ev_{ij}(t)$, that is $\mathbf{e}_i(t) = f(\mathbf{e}_i(t-1), ev_{ij}(t))$. In this way, the embedding of one node consists of a set of continuous vectors $\{\mathbf{e}(t_1), \mathbf{e}(t_2), ...\}$ and depicts a trajectory, which can describe the evolving process of its state.

2.2. Problem Setup

The goal of this study is to predict multiple co-occurring event types in the future. Formally, given historical events in past t timestamps, an embedding $\mathbf{h}_l(t+1)$ is learned to represent the state of a specific location l at time $t+1$. Based on $\mathbf{h}_l(t+1)$, we model the probabilities of event types occurring here at time $t+1$ as

$$\mathbb{P}\left(\mathbf{y}(t+1) \mid \mathbf{h}_l(t+1)\right) = \sigma\left(\mathbf{W}_\gamma \mathbf{h}_l(t+1)\right) \tag{1}$$

[1]Commonly, the subject, event type, time, and object, in that order, form a quadruple in event graph.

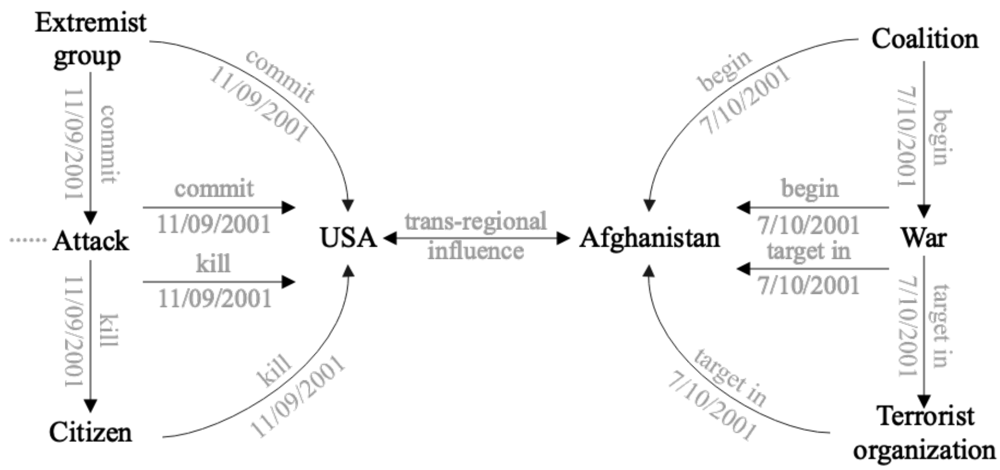


Figure 2: The spatial-temporal event graph.

where $\mathbf{y}_{t+1} \in \mathbb{R}^{|\mathcal{R}|}$ is a vector of event types. We compute the probabilities of different event types by feeding the latent embedding into a linear layer parameterized by $\mathbf{W}_\gamma \in \mathbb{R}^{|\mathcal{R}| \times d}$ followed by an element-wise sigmoid function for event prediction. In this paper, the number of the used historical timestamps is defined as a hyperparameter T, which will be analyzed in depth in Section 4.4.

3. STKGN

In this section, we first introduce the framework of the proposed model with the underlying motivation and then give the technical details of each module.

3.1. Overview

Inspired by TGN [36], we utilize a continuous-time dynamic graph to model the real world. **The core idea is that an event interacting with a node will influence its evolving direction**. We design a memory network to represent the state of an entity and update it only when events happen. Specifically, a low-dimensional vector $\mathbf{m}_i(t)$ is assigned to denote what node i has seen so far at time t. When an emerging event $ev_q(t)$ associated with node i occurs, we first extract its semantics by a text convolution layer f_1 and then pass the helpful information into node i using a message gate network

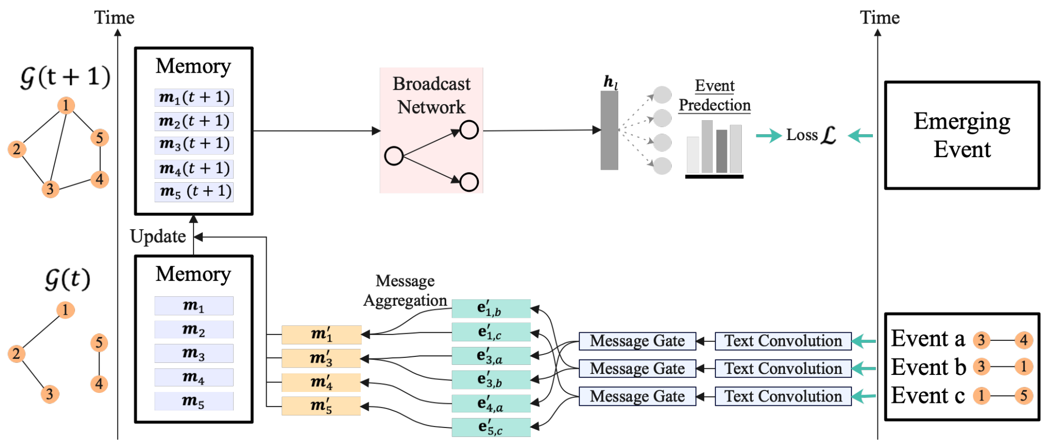


Figure 3: The framework of STKGN. Best viewed in color.

f_2. As one node usually is concerned in many events $ev_i^{all}(t)$, an aggregation function f_3 is proposed to fuse multiple messages from different events into an informative embedding, which is further fed into f_4 for memory update. When a new node first appears, its initial memory is set to be the zero vector. Thanks to this memory-based schema where memory embeddings change continuously, STKGN has the capability to represent the evolving process and gradual change of each node. **Note that the development of a node will affect both itself and the related entities especially for trans-regional influence.** Therefore, we hope to broadcast the local update throughout the graph to model the high-order reasoning chain. For example, the *9/11* events directly change the state of node *USA*, which indirectly affects the military environment of node *Afghanistan*. Technically, a GNN module f_5 is leveraged to pass the updated memory embeddings to its neighbor nodes $\mathcal{N}_i$ and learn the final representations $\mathbf{h}(t+1)$ for downstream tasks. Figure 3 presents the framework of STKGN.

For the sake of simplicity, we eliminate the time information of a vector, which is t by default, a.k.a. $\mathbf{m}_i = \mathbf{m}_i(t)$, unless we specifically notate it.

3.2. Memory network

Text convolution With an event represented by a quadruple, we combine its subject, event type, and object to export a sentence, such as (*extremist group*, *commit*, 11/09/2001, *attack*) $\rightarrow$ *extremist group commit attack*. Our task is to extract its semantics by learning a low-dimensional vector. To this end, a text convolution network is utilized. Specifically, given a document (a.k.a. sentence) consisting of n words as $\{w_1, w_2, ...w_n\}$, we initialize their features using the pretrained embeddings $\mathbf{w}$ from Google News[2] [32] and project the event to an embedding matrix: $\mathbf{E}_{ev} = [\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_n] \in \mathbb{R}^{n \times d_{pre}}$, where d_{pre} is the pretrained embedding dimension. Inspired by textCNN [4], we slide a convolution filter $\mathbf{cf} \in \mathbb{R}^{s \times d_{pre}}$ along rows to extract contextual features from a window of s words, followed by a max-pooling and activation function as

$$c = \sigma_1(\max(\mathbf{cf} * \mathbf{E}_{ev})) \tag{2}$$

where c is a scalar and σ_1 is the *Relu* activation function. Then d' filters are used to learn multiple features and infer an informative representation. What's more, we adopt two window sizes, which are 2 and 3, to capture different aspects of word relations, leading to the final representation of event ev_q as $\mathbf{e}_q = [c_1, c_2, ..., c_{2d'}] \in \mathbb{R}^{d \times 1}, d = 2d'$.

Message gate An emerging event will affect the subject's and object's evolving direction. For node i (a subject or obeject), we use a gate control network to decide what information of the event will flow into its memory unit, which is formulated as

$$\mathbf{e}'_{i,q} = (\mathbf{W}_1 \mathbf{e}_q + \mathbf{b}_1) \odot \sigma_2(\mathbf{W}_2[\mathbf{e}_q \parallel \mathbf{m}_i] + \mathbf{b}_2) \tag{3}$$

where the second term on the right side works as a soft on-off switch controlling the degree how much the memory unit accepts. $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{d \times 1}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times 2d}$ and $\mathbf{b}_2 \in \mathbb{R}^1$ are all trainable parameters. σ_2 is the *sigmoid* activation function. $\parallel$ is the concatenation operator. $\odot$ is the element-wise product operation.

Message aggregation Resorting to batch processing for efficiency reasons may lead to multiple events involving the same node i in one batch. Furthermore, critical events have a higher priority and greater impact on the node state. Therefore, we adopt an attention mechanism to fuse multiple messages from various events as

$$b_{i,q} = \mathbf{MLP}(\mathbf{e}'_{i,q} \parallel \mathbf{m}_i) \tag{4}$$

$$w_{i,q} = \frac{\exp\, b_{i,q}}{\sum_{ev_{q'} \in ev_i^{all}} \exp\, b_{i,q'}} \tag{5}$$

[2]https://code.google.com/archive/p/word2vec/

$$\mathbf{m}'_i = \sum_{ev_q \in ev_i^{all}} w_{i,q} \mathbf{e}'_{i,q} \tag{6}$$

where ev_i^{all} denotes all events associated with node i at the current timestamp, **MLP** is a two-layers feedforward neural network with the *LeakyReLU* as activation function. In this way, we attentively collect all messages which appear in current timestamp t and are associated with this node.

Memory Update With a low-dimensional embedding to represent the upcoming change on this node, we update its memory in a continuous way to simulate the evolving process. Inspired by neural turing machines [15], we use two controllers of its memory write module, which are **add** and **erase** operations. Specifically, a new value that will be added to the memory states is dependent on both the input $\mathbf{m}'_i$ and the current states $\mathbf{m}_i$ as

$$\mathbf{add}_i = \sigma_1 \left(\mathbf{W}_3 \left[\mathbf{m}_i \parallel \mathbf{m}'_i \right] + \mathbf{b}_3 \right) \tag{7}$$

Meanwhile, a scalar $erase_i$ is leveraged to control how much the current states are involved in generating the new add-on value as

$$erase_i = \sigma_2 \left(\mathbf{W}_4 \left[\mathbf{m}_i \parallel \mathbf{m}'_i \right] + \mathbf{b}_4 \right) \tag{8}$$

Then the memory states are updated by:

$$\mathbf{m}_i(t+1) = (1 - erase_i) \cdot \mathbf{m}_i + erase_i \cdot \mathbf{add}_i \tag{9}$$

$\mathbf{W}_3 \in \mathbb{R}^{d \times 2d}$, $\mathbf{W}_4 \in \mathbb{R}^{1 \times 2d}$, $\mathbf{b}_3 \in \mathbb{R}^d$, and $\mathbf{b}_4 \in \mathbb{R}^1$ are all learnable parameters.

Essentially, Equation 9 adopts a linear combination between previous states $\mathbf{m}_i$ and the new add-on vector $\mathbf{add}_i$, which prevents the consecutive dynamic embeddings of a node from varying too much and drives the evolutionary change of the memory vector along a continuous trajectory in the latent space.

3.3. Broadcast network

As aforementioned in 3.1, related nodes have potential bidirectional influences, especially in two location nodes. The message about one node may indirectly affect the development of neighbor nodes. Therefore, we learn the relational embeddings to capture influence flow in the graph. Inspired by CompGCN [44], we perform a multi-layer graph convolution network as

$$\mathbf{h}_i^{(l+1)} = f \left(\sum_{(j,i) \exists (j,r,t,i) \in \mathcal{G}_t} \mathbf{W}_5^{(l)} \phi \left(\mathbf{h}_j^{(l)}, \mathbf{o}_r^{(l)} \right) \right) \tag{10}$$

where $\mathbf{h}_i^{(l)}$ and $\mathbf{o}_r^{(l)}$ denote representations in l-th layer for node i and event type r, respectively. For the initial layer, $\mathbf{h}_i^{(1)} = \mathbf{m}_i(t+1)$. $\mathbf{W}_5^{(l)} \in \mathbb{R}^{d\times d}$ is the weight parameter for aggregating operation in the l-layer. ϕ is the multiplication operation to combing node and relation embeddings. f is the $Tanh$ activation function. Then the relation embeddings are updated as follows

$$\mathbf{o}_r^{(l+1)} = \mathbf{W}_{rel}\mathbf{o}_r^{(l)} \tag{11}$$

where $\mathbf{W}_{rel} \in \mathbb{R}^{d\times d}$ is the transformation matrix, which projects all the relations to the same embedding space and allows them to be utilized in the next layer.

The final representation of the location node in time t is infered via sum pooling as

$$\mathbf{h}_l = \sum_{l=1}^{L} \mathbf{h}_l^{(l)} \tag{12}$$

3.4. Model Optimization

We regard the target task, a.k.a. event prediction, as a multi-label classification. Specifically, given a location $\mathbf{h}_l(t)$ and its ground truth label $\hat{\mathbf{y}}(t+1) = \{\hat{y}_i\}^{|\mathcal{R}|}$, we unequally treat different event types and transfer the true label set to a distribution via the frequency. Therefore, $\hat{y}_i$ is a decimal in $[0,1]$ and $\sum_{i=1}^{|\mathcal{R}|}\hat{y}_i = 1$. Then we normalize them via softmax operation. The categorical cross-entropy loss [29, 31] is designed as

$$\mathcal{L}_l = -\frac{1}{|\mathcal{R}|}\sum_{i\in|\mathcal{R}|}\hat{y}_i \log\left(\frac{\exp(y_i)}{\sum_{j\in|\mathcal{R}|}\exp(y_j)}\right) \tag{13}$$

Then we traverse all locations and summarize losses to optimize the model as

$$\mathcal{L} = \sum_{l\in\mathcal{L}} \mathcal{L}_l \tag{14}$$

We adopt the schema of next event prediction to regularize the training process. In each batch, all observed data within the given time windows $\{t, t+1, ..., t+T-1\}$ of a fixed length T are leveraged to predict what events will happen in time $t+T$. For inference, we use a sigmoid function over the predicted score y_i and set a threshold, which is 0.5 by default, to decide the occurrence of an event.

4. Experiments

We evaluate our proposed method to answer the following research questions:

- **RQ1**: Does our proposed STKGN outperform the state-of-the-art methods?
- **RQ2**: From the spatial pattern perspective, can STKGN provide potential explanations about trans-regional implications?
- **RQ3**: From the temporal pattern perspective, does the scheme of continuous-time dynamic graph proposed in STKGN perform better in time sensitivity and robustness compared with past temporal models in event prediction?
- **RQ4**: How do different components (i.e., memory network and broadcast network) and the critical hyperparameter (i.e. GNN layer numbers) affect STKGN?

4.1. Experimental Settings

Dataset Setup. We use two common datasets in event prediction, which are the *Integrated Conflict Early Warning System* (ICEWS) and *Global Database of Events, Language, and Tone* (GDELT) [48]. Both of them are publicly available[3] with a similar data format, and we introduce one in detail for simplicity. ICEWS contains abundant political and crisis events in 260 countries or districts. These events are coded using 20 main categories (e.g. *make public statement* and *appeal*) and 296 subcategories (e.g. *decline comment* and *appeal for diplomatic cooperation*). Each event is represented by its time (day, month, year), location (city, district, province), entity (subject, object) and event type, etc. To better reflect the trans-regional influence, we select three countries (a.k.a. $|\mathcal{L}| = 3$) with spatial proximity in each dataset: (1) Iraq, Afghanistan, and Iran in ICEWS. (2) Iraq, Turkey, and Iran in GDELT. All data range from Jan. 1, 2000 to Jan. 1, 2010 and the time granularity is one day. For long-range event which occurs across multiple time steps, we update the node state using this event at multiple time steps, which is

[3] ICEWS: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/TYHZAK; GDELT: http://data.gdeltproject.org/events/index.html

Table 2: Statistics of two datasets.

		$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#events
ICEWS	Iraq	6,581	224	32,107
	Afghanistan	8,002	239	36,285
	Iran	7,362	219	17,924
	All	17,529	245	86,316
GDELT	Iraq	4,903	189	21, 901
	Turkey	6,183	206	28, 850
	Iran	5,875	197	29,755
	All	15,943	215	80,506

reasonable because this even affects the node for a long time. To ensure the data quality, we filter out events whose time or location is missing. Data statistics are shown in Table 2.

Baselines. To demonstrate the effectiveness, we compare STKGN with four kinds of methods: designed specifically for event prediction without the usage of GNN (DNN [8, 39], SIMDA [11]), designed specifically for dynamic graph representation learning rather than event prediction (T-GCN [52], EvolveGCN [34], Jodie [24], TGN [36]), designed specifically for temporal knowledge graph reasoning rather than event prediction (RE-GCN [26], EvoKG [35]), and designed specifically for event prediction based on GNN (RENET [21], Glean [8], DGCN-rs [39]).

- **DNN** Followed the setting in [8, 39], it is a deep neural network consisting of three dense layers. The initial features are generated using TF-IDF from all event contents.
- **SIMDA** [11] Based on the first law of geography, it shares similar event subtype patterns across spatially closed tasks.
- **T-GCN** [52] It combines the GCN and GRU for traffic prediction, where the GCN is used to learn complex topological structures and the GRU is employed to learn dynamic traffic changes.
- **EvolveGCN** [34] It use an RNN to evolve the GCN parameters without resorting to node embeddings, which is more applicable to the frequent change of the node set.

- **Jodie** [24] It employs two recurrent neural networks to update the representations of users and items at every interaction and learns their continuous embedding trajectories.

- **TGN** [36] It proposes a generic inductive framework of temporal graph networks operating on continuous-time dynamic graphs.

- **RE-GCN** [26] It learns the evolutional representations of entities and relations at each timestamp by modeling the KG sequence recurrently.

- **EvoKG** [35] It simultaneously models the timing of events and the evolving network structure via the conditional probability.

- **RENET** [21] It employs a neighborhood aggregator to model the connection of facts at each timestamp and a recurrent event encoder to infer future events.

- **Glean** [8] It introduces CompGCN [44] to aggregate the entity and event type embeddings in each event snapshot, which are fed into a recurrent encoder to model temporal information for final prediction.

- **DGCN-rs** [39] It is a variant of Glean and replaces the RNN module in Glean with a dilated casual convolutional network.

Note that some baselines (e.g. Glean, DGCN-rs) are self-contained in one region and only leverage local events for prediction. To make the most of their potential, we follow the input consistency principle and set the following two versions of such baselines:

- **baseline$_{\mathbf{local}}$** just collects local events to build graph [8, 39]. To this end, when forecasting future events in one country, this version only uses local histories.

- **baseline$_{\mathbf{global}}$** constructs a spatial-temporal event graph proposed in this paper, which leverages both the local and nonlocal histories.

For those models that already take into account geographic information, we use complete data and no additional annotations to the model.
Parameter Settings. We implement our STKGN model in Pytorch and Deep Graph Library (DGL)[4]. For the same side information, we initialize

[4]https://github.com/dmlc/dgl

node features via the same pretrained embeddings from Google News[5] followed by a transformation matrix to obtain the expected embedding size. We train all models by splitting the data by time to simulate the real situation. Therefore, we train all models on the first 80% of events, validate on the next 10%, and test on the last 10% of data. For GNN-based methods, the length of historical time windows and the propagation layers are set to 14 and 2 separately. For T-GCN, the number of hidden units is set to 100 for best performance. For Jodie, the subject and object are regarded as the user and item in an interaction. For TGN, the MLP and multi-head attention are adopted as the message function and embedding module separately. For RE-GCN, the ascending pace of the angle γ is set to 9° for best performance. For DGCN-rs, dilated factor d is set to be $1, 2, 4$. The embedding size of all models is set to be 64 for a fair comparison. We adopt Adam as the optimizer and the batch size is fixed at 1024 for all methods. We apply a grid search for hyper-parameters: the learning rate and the coefficient of L2 normalization are searched in $\{10^{-5}, 10^{-4}, \cdots, 10^{-1}\}$, and the dropout ratio is tuned in $\{0.0, 0.1, \cdots, 0.9\}$.
Evaluation Metrics. We use Recall and Hit@K to evaluate models.

- **Recall** evaluates whether the model can predict the event types that will occur in the future as fully as possible.
- **Hit@K** is the percentage of correct entities in the top K predictions. We set K as 3 because too small K is not enough to reflect multiple event types and too large K is meaningless.

4.2. Performance Comparision (RQ1)

Due to space limitations, we split the empirical results on the two datasets into 2 tables. The number is in the percentage formula (%). The best result in each country is highlighted in bold face and the second best are in underlined. ▲% denotes the improvement of STKGN over the scond best algorithm. Table 3 and Table 4 are results on ICEWS and GDELT, respectively. The observations are as follows:

- STKGN consistently achieves the best performance on two datasets in terms of all measures. Specifically, it achieves significant improvements over the strongest baselines w.r.t. Recall by 4.35% and 1.72%

[5]https://code.google.com/archive/p/word2vec/

Table 3: Performance comparision on ICEWS.

	ICEWS					
	Iraq		Afghanistan		Iran	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
DNN	66.16	20.87	68.78	24.32	65.53	18.75
SIMDA	71.28	25.43	75.06	29.47	72.21	24.62
T-GCN$_{\text{local}}$	72.40	26.79	76.23	31.97	74.50	26.87
T-GCN$_{\text{global}}$	76.92	27.71	80.65	33.45	75.95	27.30
EvolveGCN$_{\text{local}}$	68.57	23.27	66.38	23.51	69.29	22.65
EvolveGCN$_{\text{global}}$	69.56	24.00	67.78	24.16	69.51	23.03
Jodie$_{\text{local}}$	77.63	30.67	74.72	33.68	75.69	28.22
Jodie$_{\text{global}}$	79.56	33.96	80.33	35.55	76.48	30.92
TGN$_{\text{local}}$	83.59	36.74	81.17	38.60	82.71	35.28
TGN$_{\text{global}}$	84.27	37.14	83.58	40.07	<u>84.34</u>	37.69
RE-GCN$_{\text{local}}$	80.82	35.69	83.01	34.84	80.22	34.31
RE-GCN$_{\text{global}}$	81.10	36.33	83.66	36.10	81.80	35.54
EvoKG$_{\text{local}}$	83.45	37.01	84.17	40.88	83.72	36.12
EvoKG$_{\text{global}}$	<u>85.13</u>	<u>37.99</u>	84.58	<u>41.30</u>	84.32	37.77
RENET$_{\text{local}}$	74.40	30.32	75.97	31.43	73.92	29.19
RENET$_{\text{global}}$	75.95	31.41	77.64	33.89	75.71	32.20
Glean$_{\text{local}}$	77.64	34.37	79.22	38.05	78.27	36.32
Glean$_{\text{global}}$	79.48	34.90	83.13	39.94	79.64	36.53
DGCN-rs$_{\text{local}}$	81.12	34.85	83.68	39.29	80.08	35.78
DGCN-rs$_{\text{global}}$	83.58	36.41	<u>84.73</u>	41.02	83.26	<u>37.94</u>
STKGN	**83.06**	**36.13**	**84.74**	**38.12**	**82.04**	**37.32**
▲%	3.22	2.03	4.35	2.88	2.28	1.94

in Afghanistan and Turkey, respectively. There are two main reasons: (1) STKGN incorporates the location nodes, which helps to model the trans-regional implications and reveal the spatial relations of event occurrences, and (2) the scheme of the continuous-time dynamic graph has SOTA ability to model the teporal pattern of events.

- Despite some particular cases, the global version of baselines outperforms its local version. The performance gain benefits from the introduction of additional nonlocal information. Moreover, the degrees

Table 4: Performance comparision on GDELT.

	GDELT					
	Iraq		Turkey		Iran	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
DNN	60.74	16.44	65.11	19.88	59.77	16.66
SIMDA	67.91	19.59	71.65	21.00	68.14	25.05
T-GCN$_{\text{local}}$	70.18	22.05	76.67	24.15	70.76	28.58
T-GCN$_{\text{global}}$	69.68	23.89	75.11	26.74	72.97	29.79
EvolveGCN$_{\text{local}}$	64.20	18.72	70.36	24.16	65.14	20.86
EvolveGCN$_{\text{global}}$	64.55	19.62	69.52	25.42	64.41	21.88
Jodie$_{\text{local}}$	71.25	24.37	78.61	32.69	72.89	29.40
Jodie$_{\text{global}}$	73.87	26.10	78.83	32.15	72.87	29.07
TGN$_{\text{local}}$	77.83	32.79	82.23	35.86	79.48	34.92
TGN$_{\text{global}}$	79.96	<u>34.09</u>	<u>83.02</u>	36.69	80.16	<u>35.99</u>
RE-GCN$_{\text{local}}$	76.51	30.55	79.11	34.01	73.67	33.57
RE-GCN$_{\text{global}}$	77.56	31.46	80.23	34.62	75.11	34.69
EvoKG$_{\text{local}}$	78.71	33.05	81.29	35.86	78.62	35.92
EvoKG$_{\text{global}}$	79.88	33.91	82.99	<u>36.81</u>	79.41	36.11
RENET$_{\text{local}}$	72.43	26.43	75.98	28.23	71.05	27.70
RENET$_{\text{global}}$	71.15	28.04	74.82	30.87	71.15	29.80
Glean$_{\text{local}}$	74.82	31.37	80.04	35.49	74.81	33.58
Glean$_{\text{global}}$	77.12	32.78	81.87	36.05	77.75	34.74
DGCN-rs$_{\text{local}}$	79.28	30.69	81.92	34.46	79.29	33.10
DGCN-rs$_{\text{global}}$	<u>80.12</u>	33.15	82.38	36.17	<u>80.32</u>	35.89
STKGN	**83.06**	**36.13**	**84.74**	**38.12**	**82.04**	**37.32**
▲%	2.94	2.04	1.72	1.31	1.72	1.33

of improvement from the local to global version vary over baselines, which is due to the different applicabilities of the model to this novel spatial-temporal knowledge graph.

- Among dynamic graph representation learning models (T-GCN, EvolveGCN, Jodie and TGN) and GNN-based event prediction models (RENET, Glean, DGCN-rs and STKGN), continuous-time DGNN based methods outperform discrete-time DGNN based methods, like T-GCN → TGN and Glean → STKGN. The reason is the superiority of continuous-time

DGNN over discrete-time DGNN.

- Among temporal knowledge graph reasoning methods, RE-GCN, learning representations in a continuous way, underperforms EvoKG, learning representations in a discrete way. The possible reason is EvoKG uses max pooling over all entities, which can capture global information leading to better prediction. Another potential reason is EvoKG designs a novel framework using conditional density estimation of event triple, which is more effective than event triple completion in RE-GCN.
- Jointly analyzing all models in the same country (e.g. Iran and Iraq) across two datasets, we find the performance in GDELT is poor than that in ICEWS. Moreover, the improvement from the local to the global version is also slightly more than that in ICEWS. A possible reason is that GDELT includes many aspects of society, such as military, political, business, and social media, while ICEWS mainly consists of conflict-related events. Therefore, it is more difficult to recognize the event occurrences and evolution patterns in GDELT.

4.3. Case Study of Trans-regional Implication (RQ2)

To prove the interpretability of STKGN, we use the events of the Iraq War that broke out on Mar. 20, 2003, as an example to predict what type of events will happen in Iran. We visualize the predicted probability of the relationship *provide aid*. The ground truth is that Iran provided humanitarian aid on Mar. 21. Thus, we expect its predicted probability to be greater than 0.5, as aforementioned in Section 3.4. We conduct two tests based on whether or not to remove the influence between Iraq and Iran. The results in Figure 4 show that removing this edge leads to wrong predictions and a sharp drop in the predicted probability, from 0.61 to 0.07. On the contrary, the introduction of the relationship *trans-regional influence* successfully predicts the event type *provide aid* in Iran, which especially is a first-time event in the past month.

4.4. Time Sensitivity and Robustness Analysis (RQ3)

In this section, we explore the following two aspects: (1) the sensitivity to the number of historical timestamps T, and (2) the robustness when partial past data is missing, which is common in practical scenarios.

According to the method of modeling temporal signals, we classify past works into four types with some necessary explanations: (1) **Feature mapping** (DNN and SIMDA). (2) **RNN** including its variants like GRU (T-GCN,

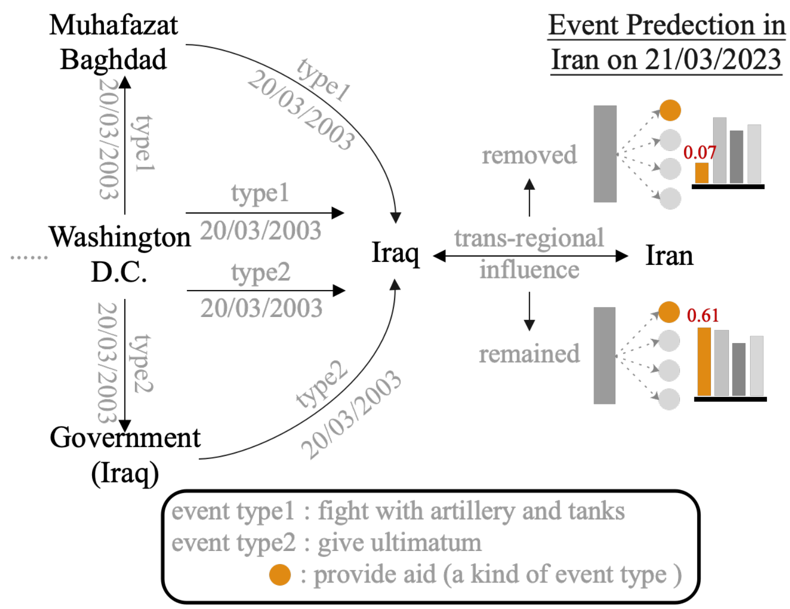
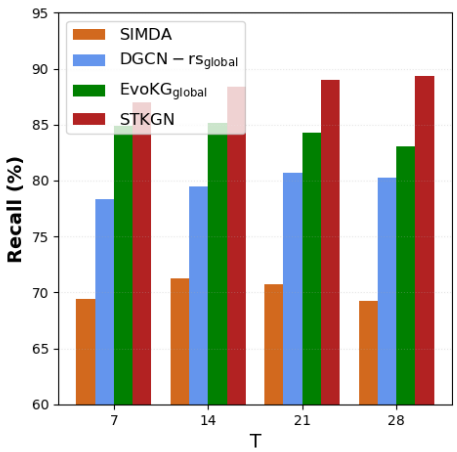


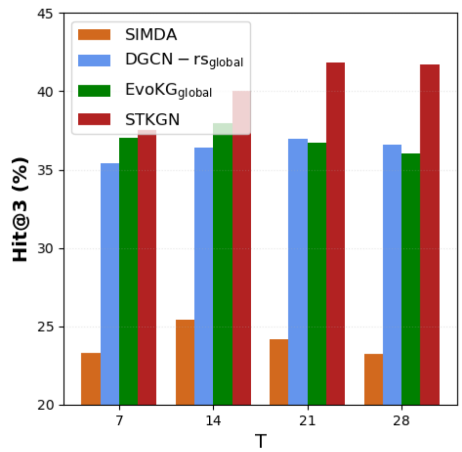
Figure 4: Case Study of Trans-regional Implication. Best viewed in color.

EvolveGCN, EvoKG, RENET, Glean). All of them adopts RNN as core module to learn sequential patterns. Therefore we call them RNN-based methods just from the temporal pattern perspective. (3) Dilated casual convolutional network, abbreviated as **DCC** (DGCN-rs). (4) Continuous-time DGNN, abbreviated as **CTDGNN** (Jodie, TGN, RE-GCN, STKGN). For each kind of method, we choose one SOTA model according to the performances in Table 3. They are SIMDA, EvoKG, DGCN-rs and STKGN. Since there is the similar findings across model versions, cities and datasets, we just show the performance of the global version in Iraq using the ICEWS dataset for simplicity.

Sensitivity analysis. We search T in the range of $\{7, 14, 21, 28\}$. The results of Recall and Hit@3 are in Figure 5(a) and (b). We observe that CT-DGNN is more suitable for capturing long-term dependence among events, while the performances of the other three methods all drop slightly as the T becomes larger. The underlying reason is the adaptability of the memory network to social events. Social events hide potential long-term temporal dependence while the memory unit has a natural advantage for modeling long-term regularities, which has been demonstrated in recommender sys-



(a)



(b)

Figure 5: Time Sensitivity Analysises. Best viewed in color.

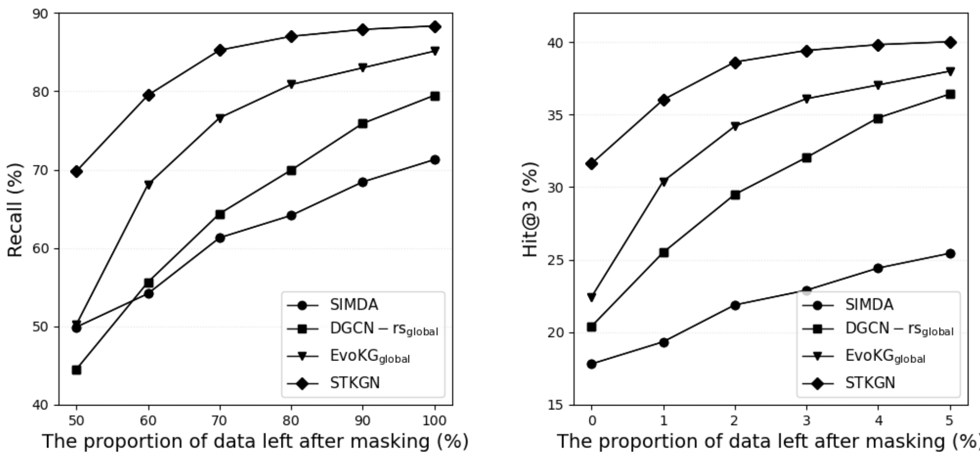
tems [20, 41]. Furthermore, in the baselines, DCC outperforms the feature mapping and RNN based mothods because the dilated convolution operation expands the receptive field, which has a similar finding in [39].

Robustness analysis. We randomly mask different ratios of training data, which are$\{10\%, 20\%, 30\%, 40\%, 50\%\}$, and keep the test data unchanged. T is fixed to 14. Figure 5(c) shows that STKGN has clearly lower performance degradation compared to other methods when keeping less and less data, which is of great significance in practical scenarios. A possible reason is that the memory embedding representing the node state in CTDGNN is continuously evolving, and missing a limited part of past events will not drastically change its development direction. In addition, an interesting phenomenon is that RNN is not as robust as the feature mapping, which may be due to that the feature mapping does not rely on time-series signals but only on the sufficiency of datas.

4.5. Study of STKGN (RQ4)

Impact of some important modules. We compare STKGN with five variants.

- **STKGN-w/o m** removes the memory network. In order to keep fairness and the same introduction of pre-training knowledge, we keep the



(a) (b)

Figure 6: Time Robustness Analysises. 100 % data left means no masking.

text convolution layer to generate the initial embedding of the broadcast network.

- **STKGN-w/o b** removes the broadcast network.
- **STKGN-mean** replaces the add-erase operation(aeop) with average pooling.
- **STKGN-LSTM** replaces the aeop with LSTM recommended in TGN to update memory.
- **STKGN-GRU** replaces the aeop with GRU recommended in TGN.

From table 5, the major findings are as follows: (1) The performance of STKGN-w/o m and STKGN-w/o b sharply degrades, which proves the necessity of memory and broadcast network. (2) Among three variants of the memory update module, STKGN-mean slightly underperforms STKGN while the other two achieve the worst performances. A possible reason is that average pooling maintains a linear combination and proves a continuous evolution of node states while the other two operations do not.

Impact of GNN layer numbers. The number of GNN layers affects the scope of the local update. We search L in the range of $\{1, 2, 3\}$ for all

Table 5: Impact of some important modules and the number is in the percentage formula (%).

	ICEWS					
	Iraq		Afghanistan		Iran	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
STKGN-w/o m	73.06	29.11	80.78	34.02	73.99	25.65
STKGN-w/o b	83.13	36.62	83.19	35.41	76.53	29.75
STKGN-mean	85.83	38.26	86.38	42.30	85.13	38.82
STKGN-LSTM	82.58	36.15	80.50	34.96	77.29	35.57
STKGN-GRU	82.66	36.47	80.70	34.31	77.53	35.99
	GDELT					
	Iraq		Turkey		Iran	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
STKGN-w/o m	68.53	24.01	75.31	25.88	71.35	30.66
STKGN-w/o b	73.56	27.83	79.69	33.08	74.53	32.84
STKGN-mean	81.47	35.66	83.69	37.82	81.47	36.26
STKGN-LSTM	79.63	34.91	81.72	36.12	79.62	32.66
STKGN-GRU	79.01	34.24	82.32	36.44	79.77	32.21

GNN-based baselines. Table 6 shows the results in Iraq of two datasets, and it has similar conclusions in other countries, which are omitted. From the observations above, we can find two uniform phenomenons: (1) $L = 2$ has the best performance, and (2) continuing to stack more GNN layers leads to overfitting. The reason is the inherent topology of this spatial-temporal event graph, where all important information of a location node is stored within its 2-hop range. For example, $Extremist\,group \xrightarrow{commit} USA \xleftrightarrow{trans-regional\,influence} Afghanistan$ has fully represented the impact on Afghanistan caused by related actors in another region. Therefore, $L = 2$ is the best, and too many layers can bring noisy signals.

5. Related work

Our study is closely related to a large body of literature on event prediction, dynamic graph representation, and temporal knowledge graph reasoning.

Table 6: Impact of GNN layer numbers and the number is in the percentage formula (%).

	Iraq in ICEWS					
	L=1		L=2		L=3	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
T-GCN$_{\text{global}}$	76.07	26.25	76.92	27.71	74.78	21.92
EvolveGCN$_{\text{global}}$	68.76	22.15	69.56	24.00	66.77	20.82
Jodie$_{\text{global}}$	77.91	22.44	79.56	33.96	76.67	20.10
TGN$_{\text{global}}$	83.18	35.05	84.27	37.14	82.41	33.59
RE-GCN$_{\text{global}}$	79.92	35.00	81.10	36.33	74.58	32.69
EvoKG$_{\text{global}}$	84.66	36.80	85.13	37.99	81.03	34.02
RENET$_{\text{global}}$	73.69	30.17	75.95	31.41	73.96	30.33
Glean$_{\text{global}}$	78.71	34.53	79.48	34.90	78.05	33.89
DGCN-rs$_{\text{global}}$	82.37	35.54	83.58	36.41	82.01	35.12
STKGN	87.07	38.15	88.35	40.02	86.69	37.92
	Iraq in GDELT					
	L=1		L=2		L=3	
	Recall	Hit@3	Recall	Hit@3	Recall	Hit@3
T-GCN$_{\text{global}}$	68.04	21.37	69.68	23.89	65.37	18.23
EvolveGCN$_{\text{global}}$	63.31	18.19	64.55	19.62	60.11	16.46
Jodie$_{\text{global}}$	72.46	24.25	73.87	26.10	71.99	23.72
TGN$_{\text{global}}$	79.15	33.82	79.96	34.09	75.72	31.31
RE-GCN$_{\text{global}}$	76.25	30.82	77.56	31.46	73.72	28.31
EvoKG$_{\text{global}}$	79.15	31.02	79.88	33.91	76.36	28.31
RENET$_{\text{global}}$	70.58	27.66	71.15	28.04	70.37	27.19
Glean$_{\text{global}}$	76.34	32.12	77.12	32.78	75.20	31.34
DGCN-rs$_{\text{global}}$	79.17	32.79	80.12	33.15	77.62	31.22
STKGN	81.31	34.99	83.06	36.13	80.86	32.63

Event prediction. There are two kinds of algorithms for event prediction. (1) Early studies take text features as input and use a multi-classification paradigm to predict event types. According to the feature mapping method, there are two solutions: linear regression [3, 46] and multi-task learning [33, 51, 10, 11]). Linear regression based methods first extract domain-specific features from the text and then uses a linear model to calculate the probability. [3] analyzes the text content of daily Twitter feeds to predict the stock market by two mood tracking tools, namely OpinionFinder and Google-Profile of Mood States (GPOMS). [46] discovers word-based topics via latent Dirichlet allocation (LDA) and uses the generalized linear regression model to predict the likelihood of an incident. In order to further consider the geographical heterogeneity, the formulation of a multi-task learning framework for event forecasting is proposed. [51] builds a forecasting model for all locations simultaneously by extracting and utilizing appropriate shared information that effectively increases the sample size for each location, thus improving the forecasting performance. [11] shares similar event subtype patterns across spatially closed tasks and regard different timestamps as independent variables to extract temporal features. However, none of the above methods fully consider the deep connection between the actors of the events. (2) Recently, GNN-based methods show full potential in event prediction [21, 8, 39]. They follow the pattern of discrete-time dynamic graph neural network (DGNN) using a set of graph snapshots taken at intervals in time. Glean [8] employs CompGCN [44] to encode each snapshot and leverages RNN to model temporal signals. DGCN-rs [39] just replaces the RNN module in Glean with a dilated casual convolutional network to better capture long-term dependence. Although they perform better than feature mapping methods, they are still limited by discrete-time dynamic graph representations.

Dynamic graph representation. According to the approaches to model the system's time domain, there are two categories for dynamic graph representation. (1) discrete-time approaches [38, 52, 34, 14]. DySAT [45] generates a dynamic node representation by joint self-attention along two dimensions: structural neighborhoods and temporal dynamics. The former extracts feature from local node neighborhoods in each snapshot, while the latter captures graph evolution over multiple time steps. DyGGNN [40] leverages the gated graph neural networks (GGNNs) to learn graph's topology at each time step and LSTMs to propagate the temporal information among the time steps. EvolveGCN [34] proposes a novel dynamic evolution paradigm that evolves GCN parameters instead of nodes. They leverages GNN to learn graph's topology and RNN to model the temporal information. However, most real-life systems of interactions such as traffic networks are dynamic. Applying static graph deep learning models to dynamic graphs by ignoring the temporal evolution [27], which is sub-optimal [50]. However, the evolution of a dynamic graph can not be sufficiently described by a sequence of static graphs. (2) continuous-time approaches [24, 36, 6, 47, 28].

[6] proposes a continuous-time version of node2vec [16] for a more efficient dynamic link prediction. Jodie [24] aims to learn the continuous embedding trajectories of user and item in recommendation via updating them by two recurrent neural networks. DyRep [43] proposes a novel modeling framework for dynamic graphs that posits representation learning as a latent mediation process bridging two observed processes, which are the topological evolution and activities between nodes. TGAT [47] uses the self-attention mechanism as a building block and develops a novel functional time encoding technique based on the classical Bochner's theorem from harmonic analysis. The above methods design for deep learning on continuous-time dynamic graphs. However, the downstream task of them is link prediction or node classification [1, 22], not specifically for event prediction. To our best knowledge, STKGN is the first attempt to introduce the continuous-time dynamic graph for event prediction.

Temporal knowledge graph reasoning. According to inferring future or present information, temporal knowledge graph reasoning can be divided into two categories. (1) interpolation-based methods [13, 18, 37, 49]. They learn representations for time-augmented KG facts that can be used in conjunction with scoring functions for link prediction. HyTE [5] explicitly incorporates time in the entity-relation space by assigning a hyperplane to each timestamp. Dyernie [18] leverages a non-Euclidean embedding approachthat learns evolving entity representations in a product of Riemannian manifolds to capture hierarchical and cyclic structures. [13] designs equipping static models with a diachronic entity embedding function to provide the characteristics of entities at any point in time. However, all of them are not able to predict events at future timestamps. (2) extrapolation-based methods [42, 43, 53, 17, 35]. DyRep [43] adopts a temporal-attentive representation network that encodes temporally evolving structural information into node representations to drive the nonlinear evolution of the observed graph dynamics. CyGNet [53] is capable of identifying facts with repetition and accordingly predicting such future facts with reference to the known facts in the past via the whole entity vocabulary. CluSTeR [25] proposes a novel framework using a two-stage manner and reinforcement learning (RL) to induce multiple clues for prediction. [17] extends the idea of neural ordinary differential equations to multi-relational graph convolutional networks to infer continuous embeddings. They captures the ever-changing structural and temporal dynamics in TKGs via recurrent event modeling. However, these methods need to assume that some members of the knowledge graph quadruple have been given, which is not applicable to cold-start prediction task.

6. Conclusion

In this paper, we aim to mine the spatial and temporal patterns of event occurrences. We propose a novel spatial-temporal knowledge graph to model the trans-regional influence and a continuous-time dynamic graph neural network to simulate the evolving process of nodes. Comprehensive experiments are conducted to demonstrate the effectiveness and explainability of the above methods. For future work, we plan to investigate the effect of personalization between a pair of locations because the degree of correlation varies between countries. Another direction is to use multiple memory embeddings to represent entities' long-term and short-term states.

References

[1] Barros, C.D., Mendonça, M.R., Vieira, A.B., Ziviani, A.: A survey on embedding dynamic graphs. ACM Computing Surveys (CSUR) **55**(1), 1–37 (2021)

[2] Bi, Y., Song, L., Yao, M., Wu, Z., Wang, J., Xiao, J.: A heterogeneous information network based cross domain insurance recommendation system for cold start users. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2211–2220 (2020)

[3] Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of computational science **2**(1), 1–8 (2011)

[4] Chen, Y.: Convolutional neural network for sentence classification. Master's thesis, University of Waterloo (2015)

[5] Dasgupta, S.S., Ray, S.N., Talukdar, P.: Hyte: Hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 conference on empirical methods in natural language processing. pp. 2001–2011 (2018)

[6] De Winter, S., Decuypere, T., Mitrović, S., Baesens, B., De Weerdt, J.: Combining temporal aspects of dynamic networks with node2vec for a

more efficient dynamic link prediction. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). pp. 1234–1241. IEEE (2018)

[7] Deng, S., Rangwala, H., Ning, Y.: Learning dynamic context graphs for predicting social events. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1007–1016 (2019)

[8] Deng, S., Rangwala, H., Ning, Y.: Dynamic knowledge graph based multi-event forecasting. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1585–1595 (2020)

[9] Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. Advances in neural information processing systems **30** (2017)

[10] Gao, Y., Zhao, L.: Incomplete label multi-task ordinal regression for spatial event scale forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)

[11] Gao, Y., Zhao, L., Wu, L., Ye, Y., Xiong, H., Yang, C.: Incomplete label multi-task deep learning for spatio-temporal event subtype forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3638–3646 (2019)

[12] García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. arXiv preprint arXiv:1809.03202 (2018)

[13] Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3988–3995 (2020)

[14] Goyal, P., Chhetri, S.R., Canedo, A.: dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowledge-Based Systems **187**, 104816 (2020)

[15] Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)

[16] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)

[17] Han, Z., Ding, Z., Ma, Y., Gu, Y., Tresp, V.: Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 8352–8364 (2021)

[18] Han, Z., Ma, Y., Chen, P., Tresp, V.: Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. arXiv preprint arXiv:2011.03984 (2020)

[19] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 639–648 (2020)

[20] Ji, W., Wang, K., Wang, X., Chen, T., Cristea, A.: Sequential recommender via time-aware attentive memory network. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 565–574 (2020)

[21] Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. arXiv preprint arXiv:1904.05530 (2019)

[22] Kazemi, S.M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., Poupart, P.: Representation learning for dynamic graphs: A survey. J. Mach. Learn. Res. **21**(70), 1–73 (2020)

[23] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[24] Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th

ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1269–1278 (2019)

[25] Li, Z., Jin, X., Guan, S., Li, W., Guo, J., Wang, Y., Cheng, X.: Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. arXiv preprint arXiv:2106.00327 (2021)

[26] Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., Cheng, X.: Temporal knowledge graph reasoning based on evolutional representation learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 408–417 (2021)

[27] Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology **58**(7), 1019–1031 (2007)

[28] Ma, Y., Guo, Z., Ren, Z., Tang, J., Yin, D.: Streaming graph neural networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 719–728 (2020)

[29] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., Van Der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European conference on computer vision (ECCV). pp. 181–196 (2018)

[30] Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. arXiv preprint arXiv:1703.04826 (2017)

[31] Menon, A.K., Rawat, A.S., Reddi, S., Kumar, S.: Multilabel reductions: what is my loss optimising? Advances in Neural Information Processing Systems **32** (2019)

[32] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013)

[33] Ning, Y., Tao, R., Reddy, C.K., Rangwala, H., Starz, J.C., Ramakrishnan, N.: Staple: Spatio-temporal precursor learning for event forecasting. In: Proceedings of the 2018 SIAM International Conference on Data Mining. pp. 99–107. SIAM (2018)

[34] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C.: Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5363–5370 (2020)

[35] Park, N., Liu, F., Mehta, P., Cristofor, D., Faloutsos, C., Dong, Y.: Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 794–803 (2022)

[36] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:2006.10637 (2020)

[37] Sadeghian, A., Rodriguez, M., Wang, D.Z., Colas, A.: Temporal reasoning over event knowledge graphs. In: Workshop on Knowledge Base Construction, Reasoning and Mining (2016)

[38] Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th international conference on web search and data mining. pp. 519–527 (2020)

[39] Song, X., Wang, H., Zhou, B.: Dgcn-rs: A dilated graph convolutional networks jointly modelling relation and semantic for multi-event forecasting. In: International Conference on Neural Information Processing. pp. 666–676. Springer (2021)

[40] Taheri, A., Gimpel, K., Berger-Wolf, T.: Learning to represent the evolution of dynamic graphs with recurrent models. In: Companion proceedings of the 2019 world wide web conference. pp. 301–307 (2019)

[41] Tan, Q., Zhang, J., Liu, N., Huang, X., Yang, H., Zhou, J., Hu, X.: Dynamic memory based attention network for sequential recommenda-

tion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4384–4392 (2021)

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

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

[44] Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.: Composition-based multi-relational graph convolutional networks. arXiv preprint arXiv:1911.03082 (2019)

[45] Wang, X., He, X., Nie, L., Chua, T.S.: Item silk road: Recommending items from information domains to social users. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 185–194 (2017)

[46] Wang, X., Gerber, M.S., Brown, D.E.: Automatic crime prediction using events extracted from twitter posts. In: International conference on social computing, behavioral-cultural modeling, and prediction. pp. 231–238. Springer (2012)

[47] Wang, Y., Chang, Y.Y., Liu, Y., Leskovec, J., Li, P.: Inductive representation learning in temporal networks via causal anonymous walks. arXiv preprint arXiv:2101.05974 (2021)

[48] Ward, M.D., Beger, A., Cutler, J., Dickenson, M., Dorff, C., Radford, B.: Comparing gdelt and icews event data. Analysis **21**(1), 267–297 (2013)

[49] Wu, J., Cao, M., Cheung, J.C.K., Hamilton, W.L.: Temp: Temporal message passing for temporal knowledge graph completion. arXiv preprint arXiv:2010.03526 (2020)

[50] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: Inductive representation learning on temporal graphs. arXiv preprint arXiv:2002.07962 (2020)

[51] Zhao, L., Sun, Q., Ye, J., Chen, F., Lu, C.T., Ramakrishnan, N.: Multi-task learning for spatio-temporal event forecasting. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1503–1512 (2015)

[52] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems **21**(9), 3848–3858 (2019)

[53] Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y.: Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4732–4740 (2021)