Context-aware Event Forecasting via Graph Disentanglement

Yunshan Ma*
National University of Singapore
yunshan.ma@u.nus.edu

Chenchen Ye*
National University of Singapore
chenchenye.ccye@gmail.com

Zijian Wu
National University of Singapore
zijian.wu@u.nus.edu

Xiang Wang†
University of Science and Technology
of China
xiangwang1223@gmail.com

Yixin Cao
Singapore Management University
caoyixin2011@gmail.com

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Event forecasting has been a demanding and challenging task throughout the entire human history. It plays a pivotal role in crisis alarming and disaster prevention in various aspects of the whole society. The task of event forecasting aims to model the relational and temporal patterns based on historical events and makes forecasting to what will happen in the future. Most existing studies on event forecasting formulate it as a problem of link prediction on temporal event graphs. However, such pure structured formulation suffers from two main limitations: 1) most events fall into general and high-level types in the event ontology, and therefore they tend to be coarse-grained and offers little utility which inevitably harms the forecasting accuracy; and 2) the events defined by a fixed ontology are unable to retain the out-of-ontology contextual information.

To address these limitations, we propose a novel task of context-aware event forecasting which incorporates auxiliary contextual information. First, the categorical context provides supplementary fine-grained information to the coarse-grained events. Second and more importantly, the context provides additional information towards specific situation and condition, which is crucial or even determinant to what will happen next. However, it is challenging to properly integrate context into the event forecasting framework, considering the complex patterns in the multi-context scenario. Towards this end, we design a novel framework named **Se**paration and **Co**llaboration **G**raph **D**isentanglement (short as **SeCoGD**) for context-aware event forecasting. In the separation stage, we leverage the context as a prior guidance to disentangle the event graph into multiple sub-graphs, followed by a context-specific module to model the relational-temporal patterns within each context. In the collaboration stage, we design a cross-context module to retain the collaborative associations among multiple contexts. Since there is no available dataset for this novel task, we construct three large-scale datasets based on GDELT. Experimental results demonstrate that our model outperforms a list of SOTA methods. The dataset and code are released via https://github.com/yecchen/SeCoGD.

*Both authors contributed equally to the paper.

†Corresponding author. Xiang Wang is also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center.

CCS CONCEPTS

• **Computing methodologies → Temporal reasoning**.

KEYWORDS

Temporal Event Forecasting, Temporal Knowledge Graph, Graph Neural Network, Graph Disentanglement

1 INTRODUCTION

Event forecasting [51] is one of the long-standing and challenging tasks, including forecasting of pandemic outbreak [11], civil unrest [34], international conflicts [28], *etc.* Accurately predicting such vital events enables people to prepare in advance to prevent catastrophic results or minimize potential influence. Automatic event forecasting targets at modeling the rich relational and temporal patterns endowed by events observed in history, thus making accurate forecasting to events in the future. The development of data science and artificial intelligence endows human with stronger capability for automatic event forecasting, which has garnered more and more attention in recent years.

One of the prominent formulations for event forecasting is to define an event as a quadruple, *i.e.*, (s, r, o, t), where s, r, o, and t refer to subject, relation (event type) [1], object, and timestamp, respectively. At each timestamp, all the quadruples form an event graph. Given a query of $(s, r, t + 1)$ in the future and the list of historical event graphs until t, we aim to predict the missing object. Based on such a structured formulation, a plethora of works have been emerging in recent years. They have applied structured relational and temporal information (*e.g.*, RE-NET [22], RE-GCN [30]), time intervals (*e.g.*, EvoKG [36]), and texts from ontology and news articles (*e.g.*, Glean [9] and CMF [10]), *etc.* for event forecasting.

Albeit the remarkable achievements of current works [5, 9, 10, 22, 30], they still suffer from the following limitations. First, existing structured events tend to be classified as high-level general events, while more specific and informative events are few. As shown in Figure 1(a), for the well-known GDELT [28] dataset, while the hierarchical event type ontology [4] defines a large number of fine-grained event types, most actual events were being classified into

---

[1]Relation and event type are the same in this work.

hierarchical event ontology	% of event types in each level	#events per event type of each level	exemplar event types
1st level	7%	20,063	02: appeal
2nd level	43%	4,574	021: appeal for cooperation
3rd level	50%	407	0211: appeal for economic cooperation

(a) Most events fall into coarse-grained and higher-level types.

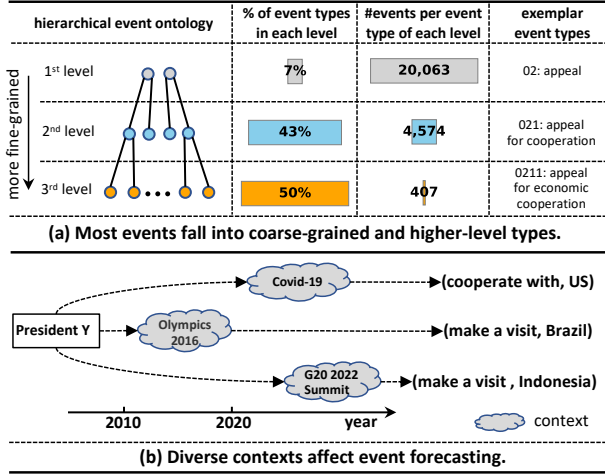**(b) Diverse contexts affect event forecasting.**

Figure 1: The motivation of context-aware event forecasting. (a) Most current events fall in the coarse-grained and higher level types of the ontology, while more informative fine-grained events are fewer. (b) Out-of-ontology and diverse contexts affect events. Context can provide more fine-grained information to enhance the event forecasting performance.

event types in the higher levels of event ontology and fine-grained events are fewer. Consequently, the expressiveness of events is severely restricted, resulting in less utility in practical scenarios. Second, events follow the types in the predefined ontology, which is usually fixed due to the difficulty of construction. For example in political event forecasting, the well-known CAMEO [4] ontology costs ten years to be finalized. It is difficult to update the ontology timely, thus newly emerging out-of-ontology information is unable to be covered by the outdated ontology [2]. Worse still, events are greatly influenced by out-of-ontology *contextual* information, such as the situation and circumstance. As the example shown in Figure 1(b), given diverse contexts, the entity *President Y* performs distinct roles and actions *w.r.t.* the various countries of the world. Such diverse contexts that provide clues for certain situations, are crucial or even determinant to event forecasting, and they cannot be adequately modeled solely based on the event ontology.

To address these limitations, we introduce *context* into existing event representation as supplementary information and define a novel task named *context-aware event forecasting*. We associate each event with a categorical context, elaborating the event's occurrence situation or condition. Then each event is extended from a quadruple to a quintuple, *i.e.*, (s, r, o, t, c), where c denotes the context. The incorporation of context brings multiple benefits to event forecasting. First, it endows more fine-grained information to each event, thus making the coarse-grained events more specific and expressive. Second, the flexibly defined context is able to offer crucial information about the circumstances or backgrounds of the events, narrowing down the potential forecasting space. As the example shown in Figure 1(b), given different contexts of *Olympics 2016* or *G20 2022 Summit*, the target countries that *President Y* would make a visit to will be different. Despite various merits, integrating context into the problem of event forecasting poses new challenges

to existing methods. First, one entity under different contexts may trigger distinctive events, and how to capture the relational and temporal patterns given a certain context is not trivial. Second, events from different contexts are also correlated with each other, and how to delicately model the collaborative associations among contexts is vital to accurate event forecasting.

To tackle the above challenges, we borrow the idea from graph disentanglement representation learning and propose a general framework **SeCoGD** (**Se**paration and **Co**llaboration **G**raph **D**isentanglement), for context-aware event forecasting. It consists of two stages: *separation* and *collaboration*. First, in the separation stage, we utilize the context as a prior guidance to separate the event graphs into multiple sub-graphs. Then, we resort to established relational-temporal models, such as RE-GCN [30], to capture the context-specific patterns within each sub-graph. Second, in the collaboration stage, we construct hypergraphs among the disentangled embeddings and leverage GNN to learn the collaborative associations among contexts. Different from current graph disentanglement methods that just focus on how to separate the graph, our framework considers both separation and collaboration due to the prior guidance of context for the separation. Moreover, our method is a general framework, of which the key modules can be replaced by alternative designs. At the same time, considering that there is no available dataset that has context information, we build three new datasets based on GDELT. Extensive experiments demonstrate that our proposed framework outperforms various SOTA methods. The main contributions of our work are summarized as:

- We propose to introduce the categorical contextual information into the structured event forecasting problem.
- To tackle the new task, we build a novel framework SeCoGD, and the two-stage design of separation and collaboration is effective in capturing the complex patterns in the multi-context scenario.
- We build three datasets based on GDELT to facilitate current and future studies for context-aware event forecasting. Our method significantly outperforms SOTA methods on the three datasets.

2 PRELIMINARY

We first give a formal definition for the task of context-aware event forecasting. Then, we introduce the newly constructed datasets.

2.1 Problem Formulation

We first present the problem formulation of conventional event forecasting, which does not consider the contextual information. Then we present how to introduce the context and formulate the new task of context-aware event forecasting.

Conventional Event Forecasting. We define an event as a quadruple (s, r, o, t), where $s \in \mathcal{E}$, $r \in \mathcal{R}$, and $o \in \mathcal{E}$ corresponds to subject entity, relation, and object entity, respectively; t is the timestamp when this event happens; $\mathcal{E}$ and $\mathcal{R}$ are the entity and relation set, respectively. All the quadruples in the same timestamp t form an event graph, denoted as $G_t = \{(s_n, r_n, o_n, t)\}_{n=1}^{N}$, where (s_n, r_n, o_n, t) is the n-th event, and N is the number of events at timestamp t. Given the historical event graphs at and before time t, denoted as $\mathbf{G}_{\leq t} = \{G_1, G_2, \cdots, G_t\}$, and a query, denoted as $(s, r, t+1)$, we aim to predict the object o.

---

[2]Both out-of-ontology and outdated ontology refer to the same problem in this work.

Context-aware Event Forecasting. We define context $c \in C$ as a categorical value denoting certain situations or conditions shared by a group of events, where $C = \{c_1, c_2, \cdots, c_K\}$ is the set of contexts and K is the number of a few contexts. In practice, for historical events, the context can be obtained from human annotation, crowd-sourcing tags, or automatic information extraction systems. We assign a context c to each event, thus extending its quadruple representation into a quintuple representation, denoted as (s, r, o, t, c). Correspondingly, the event graph at each timestamp t will be extended as $G_t = \{(s_n, r_n, o_n, t, c_n)\}_{n=1}^N$, where c_n is the context of the n-th event. Given the historical event graphs $\mathbf{G}_{\leq t}$, a query $(s, r, t + 1)$ and a specified context c in which the query event is supposed to be, we target at predicting the object o. Please note that specifying the **categorical** context during inference will not leak information about the predicted object. For example, given the context of Covid-19, the query "which country that President Y will cooperate with" will not be leaked to the model. And we assume that it is not difficult for human to provide such contextual information for a certain event he/she wants to predict.

2.2 Dataset Construction

Existing datasets for event forecasting are different cropped versions of GDELT [28] and ICEWS [34]. For example, among the datasets used by current works [15, 16, 22, 29, 30], ICEWS14, ICEWS18, ICEWS05-15 include events in the ICEWS dataset of year 2014, 2018, and 2005-2015, respectively; and GDELT covers January 2018 of the original GDELT dataset. However, all of these versions only use the existing quadruple data while overlooking the context information.

To facilitate the study of context-aware event forecasting, we build three benchmark datasets based on the GDELT dataset [28], which provides the original news article URLs of the extracted events. Following previous works [9, 10], we crop three subsets of GDELT according to the regions of the events, *i.e.*, Egypt (EG), Iran (IR), and Israel (IS), spanning from February 2015 to March 2022. According to a previous systematical study [47], the structured events extracted by GDELT have high recall while low precision, which means there are many false positive events. Such noise could be caused by the event extraction system used in GDELT or the low quality of original articles. Since the GDELT event extraction system is unavailable to the public, we aim to remove low-quality articles to eliminate these noises by the following data preprocessing steps.

First, we keep the event with a valid URL. Second, we sort the domain names of the URLs, which correspond to different news agencies. In total, there are around 20K domain names, and the top 69 cover 40% of the events. After checking these top domain names, we confirm that their news articles are of higher quality and reliability. Therefore, we remove the remaining 60% of events that are published in long-tailed domain names, which are usually from less influential agencies or personal blogs and are likely to be of low quality or even fake. Third, even though the interval of two consecutive timestamps in the original GDELT data is 15 minutes, it is unnecessary to have such precise timestamps for political events. Following ICEWS, we take the one-day time interval and collapse the 15 minutes-level timestamps of events on the same day to the day-level timestamp. Finally, we obtain the datasets and split them into training/validation/testing with a ratio of 8/1/1 over the

Table 1: Dataset Statistics.

	$|\mathcal{V}|$	$|\mathcal{E}|$	#urls	#days	#train	#valid	#test
EG	2,594	225	96,081	2,584	377,430	36,588	28,644
IR	2,988	236	223,616	2,584	973,752	69,827	76,239
IS	3,456	238	345,611	2,584	1,430,389	171,518	156,695

timeline. Due to the severe popularity bias of the dataset, several most frequent entities in the validation and testing set are masked. The statistics are shown in Table 1.

Since there is no context label in the original GDELT dataset, we leverage the textual content and topic model (*i.e.,* LDA [1]) as a *proxy* to generate contexts. In particular, we use an LDA model to extract the topic distribution of each article, where the topic with the highest weight is treated as the context of an article. An event is assigned with its corresponding article's context. More studies about the topic models as well as alternative context generation approaches are presented in Section 4.4. To be noted, during inference, people can provide certain context for the query to be predicted.

3 METHODOLOGY

To solve the new problem of context-aware event forecasting, we propose a novel framework **Se**paration and **Col**laboration **G**raph **D**isentanglement (**SeCoGD**), as shown in Figure 2. It consists of two stages: the separation stage and the collaboration stage.

3.1 Separation

In the separation stage, we first use the context as a prior guidance to disentangle the event graph into multiple sub-graphs. Then we devise a context-specific modeling module to capture the relational and temporal patterns within each context.

3.1.1 Context-aware Graph Disentanglement. Generally, events in the same context exhibit similar or correlated patterns, while events in different contexts demonstrate distinctive characteristics. Current works [10, 22, 30, 36] connect all the quadruples at the same timestamp as a unified event graph and learn a single embedding for each entity and relation via GNN models. However, such unified entity and relation embeddings are highly entangled *w.r.t.* diverse context [32], failing to capture the context-specific patterns.

Inspired by recent progress in disentangled representation learning [32, 46, 48], we seek graph disentanglement for context-aware event forecasting. Most existing works solely rely on the inherent structural information for graph disentanglement. For example, MaridVAE [33] and DGCF [45] utilize the user-item interactions to learn disentangled representations for different intents; DisenKGAT [48] tackles the heterogeneous knowledge graph and disentangles the entity embedding with respect to different topics and clusters. Nonetheless, these methods are incapable of disentangling event graphs since the events are too coarse-grained, and pure structural information is unable to well disentangle the graph.

We employ the context as a prior guidance to disentangle the event graph. Formally, given K contexts, we separate the original entangled event graph G_t into K sub-graphs $\{G_t^{c_1}, \cdots, G_t^{c_k}, \cdots, G_t^{c_K}\}$, where each sub-graph $G_t^{c_k}$ contains all the events within context c_k, denoted as $G_t^{c_k} = \{(s_n, r_n, o_n, c_k, t)\}_{n=1}^{N_t^{c_k}}$, where $N_t^{c_k}$ is the number of events in timestamp t within context c_k. Note that we make use

of the external prior knowledge, *i.e.,* the context, to disentangle the original graph. Meanwhile, previous works exploit end2end solutions, which either adopt attention mechanism [32] or incorporate auxiliary distance regularizer [45] to directly learn disentangled representations solely relying on the graph data. This prior-guided disentanglement is better than the end2end solutions in separating the graph due to the incorporation of external knowledge. Based on the disentangled event graph, the core of the model is to model the patterns within each context and across multiple contexts.

3.1.2 Context-specific Modeling. Each separated graph preserves distinct characteristics for both concurrent relations and evolving patterns under its corresponding context. Towards this end, we build a context-specific modeling module for each context. Given a list of historical event graphs $G^c_{\leq t}$ under a certain context c, the context-specific modeling module aims to learn entity and relation representations. We inherit the design of RE-GCN [30] to build this context-specific modeling module, which encompasses two parts: concurrent events modeling and temporal event modeling.

Concurrent Event Modeling is curated to model the relationship among events occurring in the same timestamp. We make use of RGCN [37], which is capable of modeling multi-relation graphs, as the graph kernel to learn the entity representation. At timestamp t under context c, for each layer l of the graph propagation, the message obtained by each object o is $\mathbf{e}^l_{o,t,c} \in \mathbb{R}^d$, defined as:

$$\mathbf{e}^l_o = f\left(\frac{1}{|\mathcal{E}_o|} \sum_{(s,r)\in\mathcal{E}_o} \mathbf{W}^l_1(\mathbf{e}^{l-1}_s + \mathbf{r}) + \mathbf{W}^l_2 \mathbf{e}^{l-1}_o\right), \tag{1}$$

where d is the dimensionality of the message, $\mathcal{E}_o$ stands for all the events of which o is the object, $\mathbf{W}^l_1, \mathbf{W}^l_2 \in \mathbb{R}^{d\times d}$ are the parameters of the convolutional kernel in layer l, and $f(\cdot)$ is the activation function which we use RReLU. To be noted, $\mathbf{e}^l_o, \mathcal{E}_o, \mathbf{e}^{l-1}_s, \mathbf{r}, \mathbf{e}^{l-1}_o$ all stand for their corresponding representations in time t of context c, where we omit the subscript (t, c) for simplicity. After performing multi-layer message passing, we aggregate the messages obtained from multi-layer propagation and yield the entity representation at time t under context c, defined as:

$$\mathbf{e}_{o,t,c} = \sum_{l=0}^{L} \mathbf{e}^l_{o,t,c}, \tag{2}$$

where $\mathbf{e}^0_{o,t,c} = \mathbf{e}^0_{o,c} \in \mathbb{R}^d$ is randomly initialized for each entity o under each context c. And the representations for all entities at timestamp t within context c are denoted as $\mathbf{E}'_{t,c} \in \mathbb{R}^{|\mathcal{E}|\times d}$.

Temporal Pattern Modeling is designed to capture the temporal evolution of entities and relations. Following the previous study [30], we devise a learnable gate mechanism to reserve the entities' evolving patterns. It is formally defined as:

$$\mathbf{E}_{t,c} = \mathbf{U}_{t,c}\mathbf{E}'_{t,c} + (1 - \mathbf{U}_{t,c})\mathbf{E}_{t-1,c}, \tag{3}$$

where $\mathbf{U}_{t,c} \in \mathbb{R}^{d\times d}$ is the learnable gate, which is calculated by a nonlinear transformation:

$$\mathbf{U}_{t,c} = \sigma(\mathbf{W}_4 \mathbf{E}_{t-1,c} + \mathbf{b}), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid activation function, and $\mathbf{W}_4$ and $\mathbf{b}$ are trainable parameters for the gate. For efficiency during implementation, we take the recent D steps of historical graphs to capture the

temporal evolving patterns, following the typical TKG (Temporal Knowledge Graph) solutions [5]. Then, the entity embeddings in the last step preserve all the context-aware relational and temporal patterns, and we denote them as $\mathbf{E}_c$.

For the relation representation, we concatenate its embedding and associated entities, thus each relation embedding is updated as:

$$\mathbf{r}'_{t,c} = \left[\mathbf{r}_c; \frac{1}{|\mathcal{V}_{r,t,c}|} \sum_{v\in\mathcal{V}_{r,t,c}} \mathbf{e}_{v,t,c}\right], \tag{5}$$

where $\mathcal{V}_{r,t,c}$ is the set of entities that connect to the relation r, $\mathbf{e}_{v,t,c} \in \mathbb{R}^d$ is the representation of entity v in $\mathbf{E}_{t,c}$, and $[;]$ is the concatenation operation. Then a GRU is applied to deduce the temporal relation representation $\mathbf{r}_{t,c}$, calculated by:

$$\mathbf{r}_{t,c} = \text{GRU}(\mathbf{r}_{t-1,c}, \mathbf{r}'_{t,c}). \tag{6}$$

And all the relations' representations at time t for context c are defined as $\mathbf{R}_{t,c} \in \mathbb{R}^{|\mathcal{R}|\times d}$. Going through D steps of recurrent units, we obtain all the relations' representations that retain relational and temporal information conditioned on context c, denoted as $\mathbf{R}_c$.

3.2 Collaboration

In the collaboration stage, we leverage hypergraphs to model the cross-context collaborative associations. Then we perform context-aware prediction and optimization.

3.2.1 Cross-context Modeling. Even though the same entity demonstrates different characteristics in various contexts, these contexts are not independent but correlated with each other. For example, given the contexts of *Covid-19 Pandemic* and *Russia-Ukraine War*, many countries must consider them simultaneously to make economic policies, in order to minimize the influence on their economy as well as social stability. To this end, capturing such correlation is crucial for some events that are affected by multiple contexts. Furthermore, after disentangling the event graph into multiple contexts, each sub-graph will be sparser than the original unified graph. Some entities that do not have sufficient occurrence in a certain context will not be well-trained for accurate forecasting. For such few-shot entities and relations, transferring knowledge from other contexts that have sufficient training data is a promising solution.

Based on the above motivations, we devise a collaboration module to model the collaborative effects among multiple contexts, aiming to achieve potential knowledge transfer for sparse entities. It is worth mentioning that we do not have supervised information to quantify the correlations among contexts, thus, we are unable to explicitly model the collaborative effects. Considering this, we resort to hypergraph to model the latent collaborations. Concretely, for each entity v, we construct a hypergraph among its sub-embeddings in different contexts, where the nodes are the separated embeddings of all entities (relations) in different contexts and every hyper-edge connects the separated embeddings of the same entity (relation). Then we leverage a multi-layer LightGCN [17] to propagate over every hypergraph, and $\hat{\mathbf{e}}^p_{v,c} \in \mathbb{R}^d$ is the p-th layer propagated information to node v under context c, obtained by:

$$\hat{\mathbf{e}}^p_{v,c} = \frac{1}{|C_v| - 1} \sum_{i\in C_v\backslash\{c\}} \hat{\mathbf{e}}^{p-1}_{v,i}, \tag{7}$$
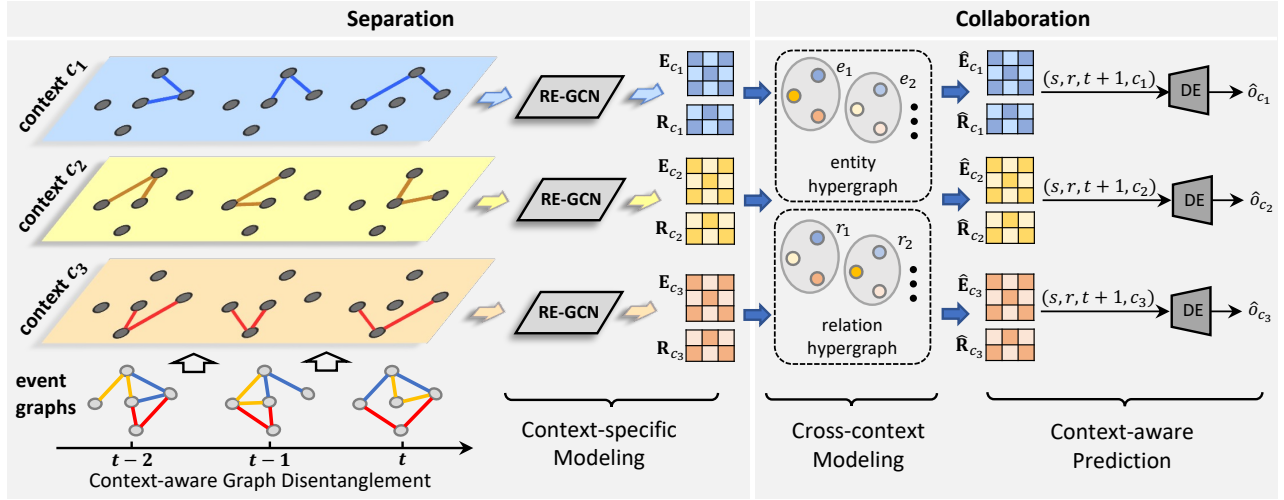
Figure 2: The overall framework of SeCoGD consists of two stages: separation and collaboration. The separation stage includes the context-aware graph disentanglement and context-specific modeling modules, and the collaboration stage comprises the cross-context modeling and context-aware prediction modules.

where C_v are all the contexts that the entity v has been in. After P layers of propagation, we aggregate each layer's embedding and yield the final entity representation:

$$\hat{\mathbf{e}}_{v,c} = \sum_{p=0}^{P} \hat{\mathbf{e}}_{v,c}^{p}, \tag{8}$$

where $\hat{\mathbf{e}}_{v,c}^{0}$ is the representation of entity v in $\mathbf{E}_c$. After the hypergraph propagation, all entities are represented as $\hat{\mathbf{E}}_c$.

Analogous to entities, relations' representations in different contexts are also totally isolated during the context-specific modeling. Thereby, for each relation, we also build a hypergraph and take advantage of a multi-layer LightGCN kernel to capture the collaborative associations among different contexts, defined as:

$$\hat{\mathbf{r}}_{x,c}^{p} = \frac{1}{|C_x| - 1} \sum_{i \in C_x \backslash \{c\}} \hat{\mathbf{r}}_{x,i}^{p-1}, \tag{9}$$

where $\hat{\mathbf{r}}_{x,c}^{p} \in \mathbb{R}^d$ is information propagated to relation x in layer p, and C_x is the set of contexts that relation x has been in over the historical observations. With P layers of graph propagation, we aggregate multiple layers' representations and obtain the final relation embedding $\hat{\mathbf{r}}_{x,c}$, formally written as: $\hat{\mathbf{r}}_{x,c} = \sum_{p=0}^{P} \hat{\mathbf{r}}_{x,c}^{p}$, where $\hat{\mathbf{r}}_{x,c}^{0}$ is relation x's embedding in $\mathbf{R}_c$.

3.2.2 Context-aware Prediction and Optimization. With the context-specific and cross-context modeling modules, we learn the entity and relation representations that not only capture context-aware characteristics but also preserve transferred knowledge from other contexts. Following the established approach to event forecasting [29, 30], we devise a decoder based on ConvTransE [38]. In particular, given a query quadruple (s, r, t, c), we first use a ConvTransE to produce the query's representation, then score the candidate entities $\mathcal{E}$ via inner-product between the query and candidate representations. Formally, we calculate the prediction scores for all candidate entities given the query (s, r) at time $t + 1$ under context

c as follows:

$$\hat{\mathbf{p}}(\mathcal{E}|s, r, c, G_{\leq t}) = \text{softmax}(\hat{\mathbf{E}}_c \text{ConvTransE}(\hat{\mathbf{e}}_{s,c}, \hat{\mathbf{r}}_c)), \tag{10}$$

where $\text{softmax}(\cdot)$ is the softmax function, $\text{ConvTransE}(\cdot)$ is the ConvTransE decoder, and $\hat{\mathbf{e}}_{s,c}$ and $\hat{\mathbf{r}}_c$ are the representations for s and r, respectively. The predicted object is presented as:

$$\hat{o}_{(s,r,t+1,c)} = \arg\max_{\mathcal{E}} \hat{p}(\mathcal{E}|s, r, c, G_{\leq t}). \tag{11}$$

We employ cross-entropy loss to optimize the whole framework in an end-to-end fashion, and the loss is defined as:

$$\mathcal{L} = \sum_{t=0}^{T-1} \sum_{c \in C} \sum_{(s,r) \in G_{t+1}^{c}} \mathbf{y}_{(s,r,t+1,c)} \log \hat{\mathbf{p}}(\mathcal{E}|s, r, c, G_{\leq t}), \tag{12}$$

where T is the total number of timestamps in the training set, and $\mathbf{y}_{(s,r,t+1,c)}$ is the one-hot representation of ground-truth object o.

3.3 Discussion

To further highlight the key contributions of this work, we discuss the generalization capability of SeCoGD, as well as the rationale behind separation and collaboration.

3.3.1 Generalization Capability. We argue that our method SeCoGD is a general framework instead of a specific model. The key contribution of SeCoGD lies in two aspects: 1) it makes use of context as a prior guidance to disentangle the event graph; and 2) it proposes a novel graph disentanglement idea under prior-guided disentanglement, that is to model the collaborative association among the disentangled representations. First, the context can be flexibly defined according to various application scenarios. For example, the tag of the news article that an event belongs to can be used as its context. Alternatively, similar to our solution of the latent topic model, various automatic text clustering algorithms, such as K-means or GMM (Gaussian Mixture Model), are plausible to identify the latent contexts of events. Second, each component of SeCoGD has various alternatives. For example, RE-NET [22]

can be used to replace the RE-GCN module for context-specific modeling, hypergraph can be replaced by some regularizers (*i.e.,* L2 distance) that pull closer the disentangled representations, and multiple modules [12, 40, 49] could be manipulated as the decoder.

3.3.2 Separation and/or Collaboration. Our work strengthens that it is crucial to incorporate the collaboration stage on top of the separation stage. However, most previous works on graph disentanglement solely focus on the separation part. For example, they either leverage regularization terms to maximize the mutual information [45] among multiple chunked representations or use attention mechanism [32] to make different disentangled representations attend on various sub-graphs. We assume that such contradictory modeling philosophy roots in two reasons. First, we have prior knowledge as the guidance for the disentanglement, therefore, we do not need any heuristically manipulated disentanglement strategies, such as mutual information maximization or attention. Second and more importantly, we believe that the crux of an effective graph disentanglement model lies in a good balance of separation and collaboration. Current works are built upon a unified graph model, which is highly intertwined. Thereby, a separation module is necessary to eliminate the entanglement. Meanwhile, we disentangle event graphs by the prior contextual information, where the sub-graphs are well or even over separated, thus a collaboration module is required to rectify the separation.

4 EXPERIMENTS

We aim to answer the following research questions:

- **RQ1:** Does our framework outperform the SOTA methods?
- **RQ2:** Are the design of two stages, *i.e.,* separation and collaboration, effective in terms of event forecasting?
- **RQ3:** How does the context affect event forecasting?

4.1 Experimental Settings

We conduct experiments on the three datasets that we constructed, *i.e.,* EG, IR, and IS. The construction and statistics of the datasets can be found in Section 2.2. Following previous settings [30], we use Mean Reciprocal Rank (MRR) and HIT@{1, 3, 10} as the evaluation metrics. We use MRR to select the best model based on the validation set and record its corresponding performance on the testing set.

4.1.1 Compared Methods. Since current works have never studied the newly proposed problem of context-aware event forecasting on temporal event graph data, we select several strands of the most relevant works to compare with our proposed method.

- **Static KG completion methods** treat event forecasting as a link prediction task on the static event graph. We select the following representative methods: **DistMult** [49], **ConvE** [12], **ConvTransE** [38], **RotatE** [40], and **RGCN** [37].
- **Temporal KG forecasting methods** are designed for temporal event forecasting. These methods consider both relational and temporal information for link prediction in the next timestamp. We consider the following SOTA methods: **TANGO** [16], **RE-NET** [22], **RE-GCN** [30], **EvoKG** [36], and **HiSMatch** [29].
- **Temporal event forecasting methods with texts** incorporate textual information into the event forecasting model, while the standard temporal KG forecasting methods only use structural

information. In particular, we implement two versions of a representative method: 1) **CMF**$_{ont}$ [10], by faithfully following the settings of the original work and incorporating the event textual description defined in the CAMEO [4] *ontology* into the structural event forecasting model. In addition, we also re-implemented 2) **CMF**$_{art}$ [10], which differs from **CMF**$_{ont}$ by using the original *article* embeddings extracted by doc2vec [26] instead of using the texts in the ontology. CMF is originally designed for binary classification of a event happening or not. We replace their task head with a typical ConvTransE decoder to enable link prediction.

- **Graph disentanglement methods** aim to separate the intertwined relational information into disentangled representations. We take into account two representative graph disentanglement methods: **DisenGCN** [32] and **DisenKGAT** [48]. Both of the methods are designed for static graphs.

4.1.2 Hyper-parameter Settings. We implement all the static methods using OpenKE [3], for TANGO, RE-NET, RE-GCN, and EvoKG, we use their released code. For CMF and HiSMatch, we re-implement them by ourselves since these methods have not released the code. To be fair and following previous settings [30], for all the baselines and our method, we set $d = 200$, use cross-entropy loss, search learning rate from {0.01, 0.001, 0.0001} and weight decay from $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. For temporal methods, we search the historical graph length D in the range of {1,3,7}. For our method, we search the number of RGCN propagation layers L from {1,2,3}, the number of hypergraph propagation layers P from {1,2}, the number of LDA topics (*aka.* contexts) K from {3, 5, 7}. We use Adam [24] optimizer and Xavier [13] initialization for all the parameters.
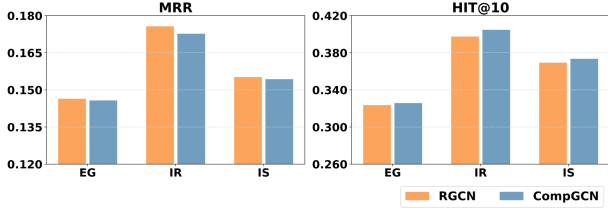
4.2 Performance Comparison (RQ1)

Table 2 shows the overall performance of our model and baselines. First of all, our method outperforms all the baselines on all three datasets. Among all the metrics, the improvement on HIT@1 is the highest, which is truly helpful in practice. Second, for all the baselines, RE-GCN achieves the best performance and even beats the models with textual inputs (*i.e.,* CMF$_{ont}$ and CMF$_{art}$), demonstrating its superiority in modeling temporal event graphs. This is why we select RE-GCN for context-specific modeling in our implementation. Third, in terms of the methods with textual inputs, CMF$_{ont}$ and CMF$_{art}$ perform well, beating most of the static and temporal methods. The results imply that the additional textual information offers valuable clues that are crucial to forecast future events. However, they are not the strongest baseline, probably because they are originally designed for binary event classification and the link prediction head is not perfectly adapted. Finally, for the two graph disentanglement-based methods, *i.e.,* DisenGCN and DisenKGAT, they do not perform very well. There are two possible reasons: 1) they rely on the static global graph, which cannot model the temporal evolving patterns; and 2) more importantly, the events in current datasets are coarse-grained and less discriminative, therefore, the methods that solely rely on structured data fail to learn disentangled representations. The results also justify our method that leverages the context as a prior guidance, instead of graph structure, to separate the event graph.

---

[3]https://github.com/thunlp/OpenKE

Table 2: The overall performance comparison between SeCoGD and baselines.

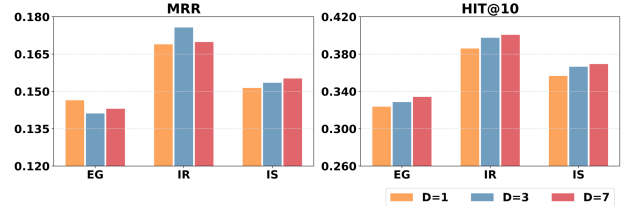| Model	EG				IR				IS			
	MRR	HIT@1	HIT@3	HIT@10	MRR	HIT@1	HIT@3	HIT@10	MRR	HIT@1	HIT@3	HIT@10
DistMult [49]	0.1164	0.0344	0.1214	0.2927	0.1349	0.0392	0.1468	0.3379	0.1031	0.0223	0.0929	0.2950
ConvE [12]	0.1151	0.0312	0.1272	0.2882	0.1365	0.0409	0.1485	0.3400	0.1060	0.0251	0.0984	0.2935
ConvTransE [38]	0.1205	<u>0.0377</u>	0.1305	0.2921	0.1405	0.0462	0.1529	0.3412	0.1079	0.0287	0.0994	0.2930
RotatE [40]	0.0892	0.0125	0.0772	0.2748	0.1055	0.0125	0.1074	0.3152	0.0879	0.0132	0.0714	0.2638
RGCN [37]	0.0974	0.0279	0.1046	0.2377	0.1185	0.0366	0.1301	0.2860	0.0861	0.0242	0.0652	0.2307
TANGO [16]	0.1043	0.0240	0.1106	0.2761	0.1249	0.0281	0.1367	0.3314	0.0972	0.0171	0.0852	0.2889
RE-NET [22]	0.1212	0.0413	0.1224	0.2932	0.1401	0.0451	0.1501	0.3452	0.1064	0.0263	0.1016	0.2894
RE-GCN [30]	<u>0.1245</u>	0.0352	<u>0.1366</u>	<u>0.3101</u>	<u>0.1647</u>	<u>0.0622</u>	<u>0.1796</u>	<u>0.3838</u>	<u>0.1301</u>	0.0408	<u>0.1281</u>	<u>0.3346</u>
EvoKG [36]	0.0797	0.0012	0.0775	0.2529	0.0892	0.0011	0.0767	0.3120	0.0779	0.0008	0.0518	0.2789
HiSMatch [29]	0.1126	0.0275	0.1279	0.2906	0.1469	0.0496	0.1599	0.3572	0.1283	<u>0.0434</u>	0.1248	0.3017
CMF$_{ont}$ [10]	0.1206	0.0348	0.1298	0.3015	0.1527	0.0529	0.1643	0.3673	0.1248	0.0368	0.1224	0.3256
CMF$_{art}$ [10]	0.1202	0.0345	0.1293	0.3027	0.1510	0.0496	0.1636	0.3716	0.1263	0.0382	0.1236	0.3261
DisenGCN [32]	0.0849	0.0196	0.0805	0.2198	0.1084	0.0275	0.1096	0.2793	0.0833	0.0162	0.0633	0.2427
DisenKGAT [48]	0.0801	0.0083	0.0822	0.2382	0.0895	0.0059	0.0977	0.2744	0.0724	0.0106	0.0429	0.2322
SeCoGD(ours)	**0.1464**	**0.0593**	**0.1605**	**0.3236**	**0.1757**	**0.0724**	**0.1902**	**0.3975**	**0.1552**	**0.0595**	**0.1588**	**0.3693**
%Improv.	17.59	57.29	17.50	4.35	6.68	16.40	5.90	3.57	19.29	37.10	23.97	10.37
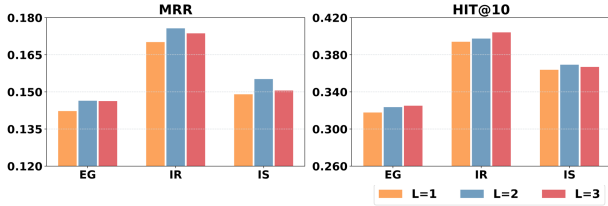
Figure 3: Results comparison of propagating different number of layers L in the context-specific modeling module.

Figure 4: Results of using different graph kernels.

Figure 5: Results of with different historical length D.

Table 3: Study of the cross-context modeling and context-aware prediction.

Model	EG		IR		IS	
	MRR	H@10	MRR	H@10	MRR	H@10
SeCoGD	0.146	0.324	0.176	0.397	0.155	0.369
w/o Ent HG	0.139	0.315	0.168	0.391	0.147	0.359
w/o Rel HG	0.143	0.331	0.170	0.400	0.147	0.362
w/o Ent or Rel HG	0.138	0.315	0.163	0.386	0.144	0.355
Avr. Context	0.130	0.309	0.163	0.373	0.129	0.331

4.3 Study of Key Modules (RQ2)

We conduct model studies to analyze the effect of the key modules in the two stages, *i.e.,* separation and collaboration.

4.3.1 Study of the Separation Stage. For the **concurrent event modeling**, we use the RGCN kernel. We tune the number of propagation layers, and the results are shown in Figure 3. Basically, two and three layers are better than one layer, depicting that higher-order information propagation over the concurrent event graph is beneficial to capture the context-specific signals. We also replace RGCN with CompGCN [42], and Figure 4 illustrates the results. Overall speaking, CompGCN and RGCN perform similarly to each other on the three datasets, and they differ slightly in terms of different evaluation metrics. It shows that our framework is not sensitive to relational modeling models.

For the **temporal pattern modeling** module, we tune the length of historical graphs D that we used to generate the entity and relation embeddings. We try different historical length D within {1, 3, 7} and visualize the results in Figure 5. For HIT@10, the longer the historical length is, the better the performance will be. But for MRR, EG and IR achieve the best performance with D=1 and D=2, respectively. This difference reminds practitioners to properly select evaluation metrics according to the application scenarios. For example on the EG dataset, if we care more about the ranking of the prediction, we need to choose MRR and set D=1. Meanwhile, if we pay more attention to the hit rate of the top-10 predicted results, HIT@10 with D=7 should be a better option. In addition, longer historical length takes extra computational costs. Therefore, it is a trade-off between efficacy and efficiency in practice.

4.3.2 Study of the Collaboration Stage. We construct a hypergraph over the sub-embeddings of each entity and relation to retain the collaborative associations across multiple contexts. To test the efficacy of the collaboration stage and the implementation of the
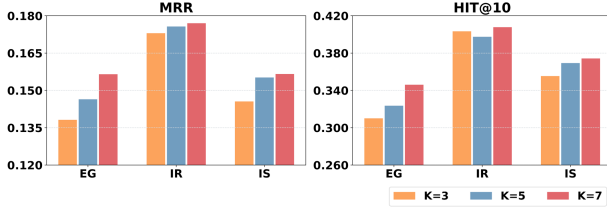
Figure 6: Results with different number of contexts.

Table 4: Alternative context generation methods.

Model	EG		IR		IS	
	MRR	H@10	MRR	H@10	MRR	H@10
RE-GCN	0.125	0.310	0.165	0.384	0.130	0.335
K-means	0.139	0.314	0.169	0.388	0.145	0.352
GMM	0.139	0.316	0.165	0.375	0.134	0.339
LDA(SeCoGD)	0.146	0.324	0.176	0.397	0.155	0.369

hypergraph, we design several ablated models by progressively removing the two hypergraphs of entity and relation. In Table 3, "w/o Ent HG", "w/o Rel HG", and "w/o Ent or Rel HG" refer to *without relation hypergraph*, *without entity hypergraph*, and *without either entity or relation hypergraph*, respectively. From the results, we can see that the results of removing either relation or entity hypergraph are worse than SeCoGD but better than that of removing both, demonstrating the efficacy of both hypergraphs. More interestingly, the performance drop of removing the entity hypergraph is generally larger than that of removing the relation hypergraph, implying that the collaboration of entities is more valuable.

During prediction, our context-aware event forecasting will pair each query $(s, r, t + 1)$ with an auxiliary context c. By specifying the context, its corresponding branch of the decoder will be selected and performs the forecasting. We argue that such a context-aware prediction narrows down the candidate space and performs better. To justify our hypothesis, we curate a variant, in which we do not specify the context during inference while just averaging the prediction scores from all the context decoders, corresponding to the row "Avr. Context" in Table 3. We can observe that "Avr. Context" performs much worse than SeCoGD. This phenomenon indicates that the specification of the proper context during inference is crucial to SeCoGD, justifying our hypothesis that the context plays a pivotal role in accurate event forecasting.

4.4 Study of the Context (RQ3)

4.4.1 Effect of the Number of Contexts. We vary the number of LDA topics K when we generate the context based on the news article, resulting in multiple versions of datasets with different number of contexts. We implement SeCoGD on all the versions of dataset and obtain the results, which are presented in Figure 6. In general, more contexts yield better performance. This is natural and reasonable because when the number of contexts increases, each context will be more specific, resulting in more fine-grained information being injected into the event. However, more contexts inevitably introduce more computational expenses. We leave the study of efficiency and scalability improvement in future work.

4.4.2 Effect of the Context Curation Methods. We define the context as a categorical label for each event, while LDA is just one of the

automatic methods in order to avoid extensive labor and costs for context annotation. We argue that alternative automatic approaches are also workable for our framework. To illustrate this property, we leverage two prominent text clustering methods, *i.e.,* K-means and GMM (Gaussian Mixture Model) using the article embeddings pre-trained by doc2vec [26], to generate contexts. Results based on the newly-generated contexts are shown in Table 4. From the results, we can conclude that SeCoGD is generally able to outperform RE-GCN by leveraging the contexts generated with alternative clustering methods. This further illustrates that our method is robust to diverse context sources. Nonetheless, our proposal of using LDA to generate contexts performs best, thus we take it as the default setting.

4.4.3 Case Study. We seek to elicit the content of each context, thus to elaborate how different events are predicted under distinct contexts. As shown in Figure 7, for each dataset, we illustrate the top words of each context in the form of word cloud [19] (the size of each word is proportional to its weight in the LDA topic distribution). From word clouds, we observe rich information within each context and clear content differences among contexts. For example, each context in the EG dataset covers background information such as popular actors, important cities, and critical actions; meanwhile, they are about economic, military, and political events respectively.

We also pick an example query $(s, r, t + 1)$ from the testing set of each dataset to concretely explicate the benefits of context-aware event forecasting. We list the object with the highest score predicted by RE-GCN and SeCoGD, and we observe that SeCoGD generates more accurate prediction results compared to RE-GCN. With a given context c, general event types such as 'Consult', 'Negotiate', and 'Host a visit' are now narrated with more supplementary information modeled in the context, leading to better results. We also observe that given the same query, SeCoGD sometimes predicts distinct objects under different contexts. For example, for the example query in IS dataset, SeCoGD predicts that *Israel* will host a visit for *Ukraine students* under *Context 2*, and predicts *UK* instead under *Context 3*. We notice that both events exist in the dataset and thus are both correct, and the two predictions are in line with the contents of their context. As shown in the word cloud for IS Context 2 and 3, the former prediction might take more military factors into consideration, and the latter is more related to government affairs. This demonstrates the flexibility in depicting the event by context.

5 RELATED WORK

5.1 Temporal Event Forecasting

Temporal event forecasting aims to forecast future events based on a list of observed historical events. It has been studied in various application scenarios, including criminal activities [44], disease outbreaks [11], stock markets [2], as well as international political events [28, 34]. Various problem formulations are utilized with regard to different event types, such as time series forecasting, natural language generation, and link prediction. In this work, we follow the typical formulations of link prediction, which is also called temporal knowledge graph completion. It inherits from static knowledge graph completion, where the key is to learn relational embeddings via various scoring functions, such as TransE [3], DistMult [49], ComplEx [41], RotatE [40], ConvE [12], ConvTransE [38]
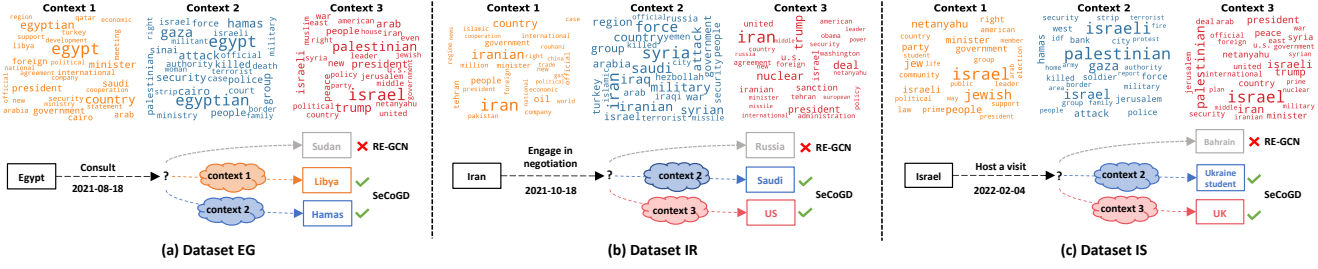
Figure 7: Case study on three datasets. In each sub-figure, the context number K is set as three, the top shows the word cloud of each context, and the bottom illustrates several exemplar forecasting results by SeCoGD and RE-GCN.

etc. To tackle the temporal evolving patterns and forecast future events, recurrent neural network (RNN) [20] has been included. RE-NET [22] proposes to use RGCN [37] to capture the relational patterns in each timestamp and GRU [7] to model the dynamics of embeddings over time. RE-GCN [30] additionally incorporates a static graph to learn the static properties of the entities and adopts ConvTransE [38] as the decoder. TANGO [16] models the structure of candidate entities via neural ordinary differential equations; EvoKG [36] considers the time information for event forecasting; HiSMatch [29] reformulates event forecasting as a query-candidate matching problem and proposes a two-branch framework to match a query to candidate entities. More related works [15, 18, 27, 39, 53] can be seen in the survey [5]. Most of these TKG methods only operate on pure structured data, overlooking the rich semantic or contextual information. To address these limitations, Glean [9] and CMF [10] propose to use the textual information. They simplify the event forecasting problem from fine-grained link prediction to an easier binary classification problem, *i.e.,* predicting whether an event will happen or not. In addition, the textual information is only available for historical events but unavailable for future events, thus this additional information cannot directly narrow down the candidate space. Some works also use context for event prediction, while they are either for event status classification [8] or time series forecasting [35], which are different tasks.

5.2 Graph Disentanglement

Graph neural networks [14, 25, 43] have been the defacto solutions for graph representation learning. Graph disentanglement is the extension of disentangled representation learning from the general domain to the graph data. Disentangled representation learning focuses on separating the unified representation into multiple disentangled components, thus achieving many excellent modeling properties such as enhanced representation capability or explainability. Various studies have been conducted on CV [6], NLP [23], as well as recommender system [21, 33, 45]. For graph representation learning, disentanglement has also garnered particular attention. DisenGCN [32] is one of the pioneering works to use multiple disentangled graph convolutional kernels to learn disentangled node representations. FactorGCN [50] factorizes the node embedding into multiple blocks, which captures interpretable global topological semantics. IPGDN [31] leverages the Hilbert-Schmidt Independence Criterion (HSIC) to achieve disentanglement. ADGCN [52] introduces adversarial learning to graph disentanglement representation learning. DisenHAN [46] is designed for heterogeneous

graph, where multiple node and relation types are involved. The most relevant work for event forecasting is DisenKGAT [48], which aims to learn disentangled representations for knowledge graph. Despite various studies on graph disentanglement, our work differs from current works and promotes these works in several aspects. First, we are the first to introduce graph disentanglement learning to temporal event forecasting. Second, most of the existing works aim to directly disentangle the graph purely using the graph's own features, ignoring the contextual information.

6 CONCLUSION AND FUTURE WORK

In this work, we explored the incorporation of context into the problem of event forecasting and proposed a novel task of context-aware event forecasting. To tackle this novel problem, we borrowed the idea from graph disentanglement and designed an overall framework SeCoGD. Specifically, we utilized the context as prior guidance to separate the event graph and incorporated a context-specific modeling module to capture the relational and temporal patterns in each context. In addition, we designed a cross-context modeling module to model the collaborative associations among multiple contexts. Since there are no available datasets for this new task, we built three large-scale datasets based on GDELT. Extensive experiments on these three datasets demonstrated that our framework outperforms all the SOTA methods. Various model studies further elaborated more details about the effectiveness of the key modules and various contexts of the framework.

Despite the progress achieved by this work, there are several limitations, thus motivating multiple potential research directions in the future. First, the implementation of context generation is based on unsupervised methods, while human-generated contexts, such as tags and categories, could be more useful in practice. Second, the original articles of these events are just used as a proxy to generate contexts, of which just a little information has been utilized. More effective approaches to mining more beneficial patterns from raw texts are promising. Third, more advanced graph disentanglement methods are expected to be explored and enhance the performance. Finally, in addition to next step prediction, the more important yet challenging multi-horizon forecasting should be studied in future.

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[2] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. 2011. Twitter mood predicts the stock market. *J. Comput. Sci.* 2, 1 (2011), 1–8.

[3] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[4] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. CAMEO.CDB.09b5.pdf. In *ICEWS Coded Event Data*. Harvard Dataverse. https://doi.org/10.7910/DVN/28075/SCJPXX

[5] Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2022. Temporal Knowledge Graph Completion: A Survey. *CoRR* abs/2201.08236 (2022).

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NIPS*. 2172–2180.

[7] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*. ACL, 1724–1734.

[8] Zeyu Dai, Wenlin Yao, and Ruihong Huang. 2017. Using Context Events in Neural Network Models for Event Temporal Status Identification. In *IJCNLP(2)*. Asian Federation of Natural Language Processing, 234–239.

[9] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic Knowledge Graph based Multi-Event Forecasting. In *KDD*. ACM, 1585–1595.

[10] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2021. Understanding Event Predictions via Contextualized Multilevel Feature Learning. In *CIKM*. ACM, 342–351.

[11] Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. 2020. Cola-GNN: Cross-location Attention based Graph Neural Networks for Long-term ILI Prediction. In *CIKM*. ACM, 245–254.

[12] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*. AAAI Press, 1811–1818.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS (JMLR Proceedings, Vol. 9)*. JMLR.org, 249–256.

[14] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[15] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. Explainable Subgraph Reasoning for Forecasting on Temporal Knowledge Graphs. In *ICLR*. OpenReview.net.

[16] Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning Neural Ordinary Equations for Forecasting Future Links on Temporal Knowledge Graphs. In *EMNLP (1)*. Association for Computational Linguistics, 8352–8364.

[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM, 639–648.

[18] Yongquan He, Peng Zhang, Luchen Liu, Qi Liang, Wenyuan Zhang, and Chuang Zhang. 2021. HIP Network: Historical Information Passing Network for Extrapolation Reasoning on Temporal Knowledge Graph. In *IJCAI*. ijcai.org, 1915–1921.

[19] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. 2014. Word Cloud Explorer: Text Analytics Based on Word Clouds. In *IEEE*. 1833–1842.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *ACL*. Association for Computational Linguistics, 4255–4264.

[22] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530* (2019).

[23] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *ACL (1)*. Association for Computational Linguistics, 424–434.

[24] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[25] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016).

[26] Quoc V. Le and Tomás Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, 1188–1196.

[27] Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving Validity Time in Knowledge Graph. In *WWW (Companion Volume)*. ACM, 1771–1776.

[28] Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, Vol. 2. Citeseer, 1–49.

[29] Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022. HiSMatch: Historical Structure Matching based Temporal Knowledge Graph Reasoning. *CoRR* abs/2210.09708 (2022).

[30] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning. In *SIGIR*. ACM, 408–417.

[31] Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. 2020. Independence Promoted Graph Disentangled Networks. In *AAAI*. AAAI Press, 4916–4923.

[32] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled Graph Convolutional Networks. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 4212–4221.

[33] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. 5712–5723.

[34] Sean P O'brien. 2010. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International studies review* 12, 1 (2010), 87–104.

[35] Maya Okawa, Tomoharu Iwata, Yusuke Tanaka, Takeshi Kurashima, Hiroyuki Toda, and Hisashi Kashima. 2022. Context-aware spatio-temporal event prediction via convolutional Hawkes processes. *Mach. Learn.* 111, 8 (2022), 2929–2950.

[36] Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. EvoKG: Jointly Modeling Event Time and Network Structure for Reasoning over Temporal Knowledge Graphs. In *WSDM*. ACM, 794–803.

[37] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC (Lecture Notes in Computer Science, Vol. 10843)*. Springer, 593–607.

[38] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*. AAAI Press, 3060–3067.

[39] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting. In *EMNLP (1)*. Association for Computational Linguistics, 8306–8319.

[40] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR (Poster)*. OpenReview.net.

[41] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML (JMLR Workshop and Conference Proceedings, Vol. 48)*. JMLR.org, 2071–2080.

[42] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *ICLR*. OpenReview.net.

[43] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

[44] Xiaofeng Wang, Matthew S. Gerber, and Donald E. Brown. 2012. Automatic Crime Prediction Using Events Extracted from Twitter Posts. In *SBP (Lecture Notes in Computer Science, Vol. 7227)*. Springer, 231–238.

[45] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.

[46] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM*. ACM, 1605–1614.

[47] Michael D Ward, Andreas Beger, Josh Cutler, Matthew Dickenson, Cassy Dorff, and Ben Radford. 2013. Comparing GDELT and ICEWS event data. *Analysis* 21, 1 (2013), 267–297.

[48] Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network. In *CIKM*. ACM, 2140–2149.

[49] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR (Poster)*.

[50] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. 2020. Factorizable Graph Convolutional Networks. In *NeurIPS*.

[51] Liang Zhao. 2022. Event Prediction in the Big Data Era: A Systematic Survey. *ACM Comput. Surv.* 54, 5 (2022), 94:1–94:37.

[52] Shuai Zheng, Zhenfeng Zhu, Zhizhe Liu, Shuiwang Ji, and Yao Zhao. 2021. Adversarial Graph Disentanglement. *CoRR* abs/2103.07295 (2021).

[53] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. In *AAAI*. AAAI Press, 4732–4740.