Sequence-Based Modeling for Temporal Knowledge Graph Link Prediction

Lijie Li[1], Wenqiang Liu[1], Zuobin Xiong[2], and Ye Wang[1](✉)

[1] Harbin Engineering University, Harbin 150006, China
{lilijie,lwq,wangye2020}@hrbeu.edu.cn
[2] Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
zxiong2@gsu.edu

Abstract. Currently, the majority of research in temporal knowledge graph link prediction focuses on completing missing facts. Nevertheless, the utilization of knowledge graphs to forecast future facts has garnered significant scholarly attention. The attainment of efficient future fact prediction for time-series data hinges primarily on an in-depth exploration of both past historical facts and concurrent facts in the present. Presently, the majority of research in this domain lacks an all-encompassing integration of temporal points and durations in factual features, thereby hindering the effective management of two distinct types of facts with varying chronologies and ultimately disregarding their latent influence on future facts. This paper introduces an advanced representation model - the Progressive Representation Graph Attention Network (PRGAN) - which harnesses the potential of Graph Convolutional Neural Network and Recurrent Neural Network. PRGAN aims to ameliorate the existing shortcomings and augment the efficacy of future event prediction through attention-based learning of progressive representations of entities and relations in time series. We evaluated our proposed method with five event datasets. Extensive experimentation revealed that, in comparison with other baseline models, the PRGAN model displayed remarkable performance and efficiency in temporal reasoning tasks, thereby demonstrating its outstanding superiority.

Keywords: Graph Representation Learning · Knowledge Representation · Temporal Knowledge Graph · Event Prediction

1 Introduction

A Knowledge Graph (KG) [1] is a structured information management and knowledge representation system designed to integrate data in a human-readable and computer-friendly way. However, current research has primarily focused on using static KGs to reason with this data. In reality, however, multi-relational data is generated over time, making static KGs unsuitable for describing the dynamic nature of the world, such as the continuous changes in social networks.

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14257, pp. 550–562, 2023.
https://doi.org/10.1007/978-3-031-44216-2_45

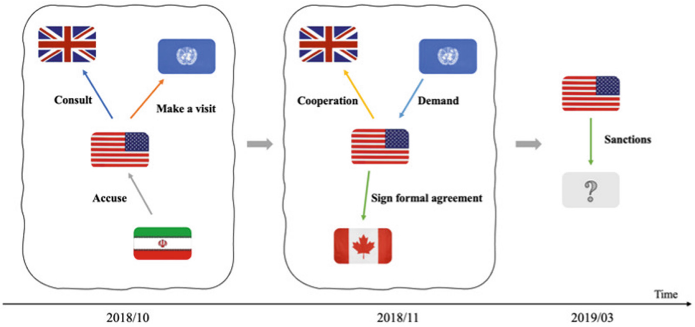

Fig. 1. Diagram of inference on TKG.

To address this issue, researchers have introduced Temporal Knowledge Graphs (TKGs). TKGs can be seen as a sequence of multiple KGs, each representing facts occurring at the same timestamp, as illustrated in Fig. 1. Yet, similar to static KGs, TKGs also face the problem of incompleteness or lack of information, which makes the task of completing TKGs a critical challenge.

Typically, Temporal Knowledge Graph (TKG) reasoning involves time stamps that vary from t_0 to t_T and can be classified into two types: internal reasoning and external reasoning. The internal reasoning approach aims to predict new facts that occur within the time period $t_0 < t < t_T$, while the external reasoning approach focuses on predicting new facts that occur after time t_T ($t > t_T$). In this area, researchers have made few attempts to infer future facts, which makes it still under-researched. Know-Evolve [2] predicts future facts by evolving embeddings of entities, while RE-Net [3] encodes past event sequences using embeddings of the original entities. However, RE-Net mainly focuses on entities in the aggregation process and ignores the relationships. Therefore, the best practice is to locally model the facts that occur in each time window as concurrent facts and then represent the historical facts across the entire time series to infer future facts.

In this article, we propose a novel network model called the Progressive Representation Graph Attention Network (PRGAN) for predicting future facts in a Knowledge Graph (KG). PRGAN models the entire sequence of KG to learn progressive representations of entities and relationships. Its main contribution is summarized in the following aspects:

- Progressive representation: PRGAN uses graph convolutional neural networks and recurrent neural networks to learn progressive representations of entities and relationships throughout the entire KG sequence.
- Historical event attention mechanism: The attention mechanism is used to dynamically select and linearly combine different historical facts, allowing the model to emphasize the importance of different historical facts for the current query.

- Flexible structure: The modeling based on graph structure and the decoder is a flexible plug-in part and can be replaced/integrated with other models if there is better ones.
- Effective model: We conducted extensive experiments and evaluations of PRGAN on five temporal knowledge graph datasets, using various metrics to assess its performance. We compared our model with several baseline models commonly used in this area of research, where the results demonstrate that PRGAN outperforms most of the baseline models in terms of accuracy, efficiency, and robustness.

The remaining of this work is organized as follows. We introduce the related works of knowledge graph in Sect. 2. In Sect. 3, we formulate the studied problem of this work and then detail our proposed model in Sect. 4. Our experiments in Sect. 5 illustrate the performance of our model over selected baselines. Finally, this work is concluded in Sect. 6.

2 Related Work

2.1 Knowledge Graph Embedding

Knowledge graph embedding (KGE) [5] maps each entity $e \in E$ and each relation $r \in R$ to a low-dimensional continuous vector. Currently, there are two types of knowledge graph embeddings: static knowledge graph embedding and temporal knowledge graph embedding.

Static Knowledge Graph Embedding. Static knowledge graph embedding do not consider temporal information and has been widely studied. Typically, their embedding methods fall into three categories: (i) translation-based models, such as TransE [6]; (ii) bilinear models, such as RESCAL [7], DistMult [8], and SimplE [9]; and (iii) neural network models, such as graph convolutional network (GCN) [10]. However, these knowledge graph embedding methods are not suitable for temporal knowledge graphs (TKGs) as they cannot capture the rich dynamic temporal information of TKGs.

Temporal Knowledge Graph Embedding. Temporal Knowledge Embedding (TKE) methods aim to capture both temporal and relational information. To achieve this goal, each method either embeds discrete timestamps into a vector space or learns time-aware representations for each entity. Some popular TKE models include: TTransE [11], which extends the traditional TransE [6] model to the temporal domain by embedding time information into the scoring function; HyTE [12], which extends TransH [13] by associating each timestamp with a corresponding hyperplane and projecting entity and relation embeddings onto hyperplanes specific to the timestamp; RE-Net [3], which aggregates entity neighborhoods as historical information and models the temporal dependencies of facts using a recursive neural network; DE-SimpIE [14], which extends SimplE [9] by exploring temporal functions to combine entities and timestamps to generate time-specific representations; TPmod [15], which is a trend-guided prediction model based on attention mechanisms developed to predict missing facts in TKGs; and TempCaps [16], which is the first model that applies capsule networks [17] to temporal knowledge graph completion. Although these models perform well in their tasks, they do not capture long-term temporal dependencies of real-world facts.

2.2 Graph Representation Learning

Graph representation learning aims to transform graph data into low dimensional dense vectors either at the node level or the whole graph level. The primary goal of graph representation learning is to map nodes to vector representations while preserving as much topological information of the graph as possible. And, if the representation of graph data can contain rich semantic information, it can obtain good input features and directly use linear classifiers for classification tasks. Graph Convolutional Networks (GCN) are a type of neural network architecture that is primarily used in graph representation learning and is similar to Convolutional Neural Networks (CNN) but designed for graph data structures. GCN features can be used to perform various tasks such as node classification, link prediction, community detection, and network similarity detection, demonstrating the broad range of applications of GCN.

3 Problem Formulation

A event-based Temporal Knowledge Graph (TKG) can be viewed as a series of static Knowledge Graphs (KGs) sorted by event timestamp. Therefore, the TKG G can be formally represented as a sequence of KGs with timestamps, i.e., $G = \{G_1, G_2, ..., G_t, ..., G_T\}$ (that is $G = G_1 \cup G_2 \cup \; ... \cup G_t ... \cup G_T$), where each $G_t = (V, R, F_t)$ is a directed multi-relational graph at timestamp t. Any event in G_t can be represented as a timestamped quadruple, i.e., (subject, relation, object, time), and is dented by the quadruple $(s, r, o, t) \in G_t$. The important mathematical symbols used in this paper are shown in Table 1.

Table 1. Important symbols and their descriptions.

Symbol	Descriptions
G, G_t	TKG, the KG of the timestamp t
V, R, F_t	Entity set, relationship set, and fact set in TKG (at timestamp t)
E_t, R_t	Embedding matrix of entities and relationships at time t
H, R	Embedding matrix of random initialized entities and relationships

In extrapolation prediction tasks, facts are sorted based on their timestamps. Given an incomplete event (missing either the object entity $(s, r, ?, t)$, or the subject entity $(?, r, o, t)$, or the relation $(s, ?, o, t)$), we should predict the missing object, entity or relation based on the historical facts before time t. For each event $(s, r, o, t) \in F_t$, we can obtain three sub-prediction tasks: $(s, r, ?, t)$, $(?, r, o, t)$ and $(s, ?, o, t)$. Therefore, the overall joint probability is calculated as follows:

$$P(F_t) = \frac{(P_o(F_t) + P_s(F_t) + P_r(F_t))}{3}, \tag{1}$$

where $P_o(F_t)$, $P_s(F_t)$, and $P_r(F_t)$ represent the joint probability of $(s, r, ?, t)$, $(?, r, o, t)$ and $(s, ?, o, t)$. Applying the division rule in Eq. (1) ensures that the resulting probability $P_o(F_t)$ is normalized to the range of $[0, 1]$.

Suppose that the prediction of future facts at time stamp t depends on the facts of its previous k time steps, i.e., $G_{t-k}, G_{t-k+1}, ..., G_{t-1}$. We use $E_t \in R^{|V|\times d}$ to represent the entity embedding matrix modeled in the historical KG sequence and $R_t \in R^{|R|\times d}$ to represent the relation embedding matrix modeled in the historical KG sequence (where d is the embedding dimension). The problem of predicting future facts can be formalized into two categories: entity prediction and relation prediction, both of which can be ultimately formulated as a ranking problem.

The prediction of subject entities is defined as follows:

$$P_s(F_t) = P(s|G_{t-k:t-1}, r, o) = P(s|E_t, R_t, r, o). \tag{2}$$

The prediction of object entities is defined as follows:

$$P_o(F_t) = P(o|G_{t-k:t-1}, s, r) = P(o|E_t, R_t, s, r). \tag{3}$$

The prediction of relations is defined as follows:

$$P_r(F_t) = P(r|G_{t-k:t-1}, s, o) = P(r|E_t, R_t, s, o). \tag{4}$$

4 The PRGAN Model

PRGAN is a novel network model proposed by this work for predicting future facts in Temporal Knowledge Graphs (TKGs). As shown in Fig. 2, the framework of PRGAN consists of an encoder and a decoder. PRGAN utilizes a relation-aware graph neural network [4] to encode concurrent facts at each timestamp and capturing structural dependencies. It further models the historical KG sequence via an autoregressive approach using a time-gated recurrent module and a recurrent neural network based on an attention mechanism to obtain a progressive representation of entities and relations over time.

4.1 Concurrent Dependency Modeling Module

To predict future facts, the concurrent facts that occur at the same time can have a significant impact. Therefore, PRGAN utilizes a relation-aware graph neural network (RGCN) to model concurrent facts at each timestamp and uses a w-layer relation-aware GCN to model structural dependency relationships. Specifically,

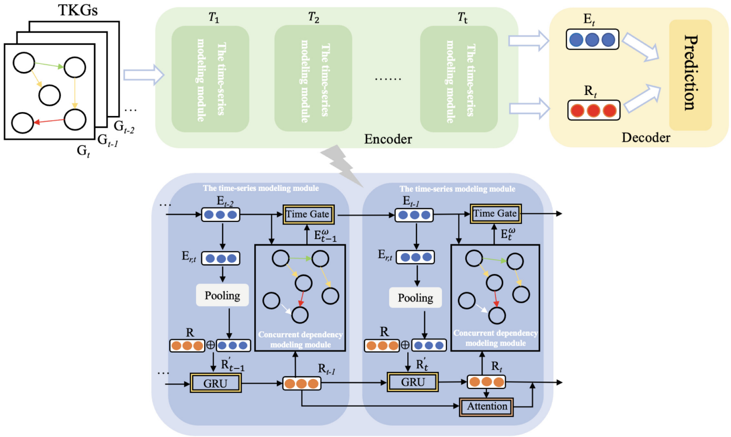

Fig. 2. The framework of proposed PRGAN.

for each graph with a timestamp, the embedding of the object entity o at layer $l \in [0, w-1]$ is considered to obtain representation learning information under a messaging framework with l layers. Its embedding in layer $l+1$ is obtained using the following formula:

$$Z_o^l = \frac{1}{k} \sum_{(s,r),\exists(s,r,o)\in G_t} W_1^l \left(e_{s,t}^l + r_t\right) + W_2^l e_{0,t}^l \tag{5}$$

$$e_{o,t}^{l+1} = \sigma\left(Z_o^l\right) \tag{6}$$

where, $e_{s,t}^l$, r_t, and $e_{0,t}^l$ represent the l layer embeddings of subject entity s, relationship r and object entity o at timestamp t. W_1^l and W_2^l are the parameters for aggregating features and self-looping in l layer. k is a normalization constant equal to the in-degree of object entity o. σ is the ReLU activation function.

4.2 The Time-Series Modeling Module

For predicting future facts, the potential impact behind all the facts that occurred at every time is also significant. Therefore, PRGAN utilizes a time-gated recurrent unit (GRU) [18] with an attention mechanism to model the historical facts of the entire knowledge graph (KG) time sequence. The latent time sequence problem of historical facts is solved by stacking the ω-layer RGCN. However, it should be considered that, firstly, if the same entity pairs appear with repeated relationships at adjacent timestamps, the obtained entity embeddings may converge to the same value, resulting in the problem of excessive smoothing in the graph. Secondly, considering the length of the time sequence, a large number of stacked RGCNs may eventually encounter the problem of gradient disappearance, causing the model's weights to remain unchanged during the training process. Considering the above two issues, a time-gated unit is added to the model to alleviate these influences. Thus, the final entity embedding matrix E_t is composed of the entity embedding matrix E_{t-1} at timestamp $t-1$ and the current output E_t^ω at timestamp t, as shown in the following formula:

$$E_t = G_t \bigotimes E_t^\omega + (1 - G_t) \bigotimes E_{t-1}, \tag{7}$$

where $\bigotimes$ denotes dot product operation.

Besides, the non-linear transformation of time gate $G_t \in R^{d \times d}$ is performed as follows:

$$G_t = \alpha \left(W_3 E_{t-1} + b\right) \tag{8}$$

where α is the sigmoid function, $W_3 \in R^{d \times d}$ is the weight matrix of the time gate.

Meanwhile, the embedding of the relationship contains information about the entities involved in the corresponding fact. That is, the embedding of relationship r_t at timestamp t is influenced by two factors: (i) the progressive representation of the relevant entities $C_{r,t} = e|\,(e, r, o, t)\, or\, (s, r, e, t) \in F_t$ of relationship r at timestamp t in the time series; (ii) the embedding representation of relationship r itself at timestamp $t-1$.

In addition, in order to better capture the progressive representation of historical facts, we use an attention mechanism for historical facts, allowing the model to dynamically select and linearly combine different historical facts of the relationship, namely:

$$e_\tau = \nu_e^T tanh(W_e R_t + U_e R_\tau) \tag{9}$$

$$\alpha_\tau = \frac{exp(e_\tau)}{\sum_{\tau=1}^{t} exp(e_\tau)} \tag{10}$$

$$R_t = \sum_{\tau=1}^{t} \alpha_\tau R_\tau \tag{11}$$

where ν_e^T, W_e and U_e are parameters that determine the importance of different historical facts when making predictions for a given query.

4.3 Loss Calculation

Predicting future facts based on the historical KG sequence can be viewed as a multi-classification task for predicting the object entity given (s, r, t), where each class corresponds to a possible object entity. Similarly, predicting the relationship given (s, o, t) and predicting the subject entity given (r, o, t) can also be viewed as multi-classification tasks. For the sake of simplicity, we omit the symbols for previous facts here. Therefore, the loss function can be formulated as follows:

$$\mathcal{L}^e = \sum_{t=1}^{T} \sum_{(s,r,o,t) \in G_t} \sum_{i=0}^{|V|-1} y_{t,i}^e log[p_i\left(o|E_{t-1}, R_{t-1}, s, r\right)]. \tag{12}$$

$$\mathcal{L}^r = \sum_{t=1}^{T} \sum_{(s,r,o,t)\in G_t} \sum_{i=0}^{|R|-1} y_{t,i}^r log[p_i\left(r|E_{t-1}, R_{t-1}, s, o\right)]. \tag{13}$$

where $p_i\left(o|E_{t-1}, R_{t-1}, s, r\right)$ and $p_i\left(r|E_{t-1}, R_{t-1}, s, o\right)$ represents the probability score of entity i and relationship i, respectively. $y_{t,i}^e$ represents the i-th vector in y_t^e. $y_{t,i}^r$ represents the i-th vector in y_t^r. If $y_{t,i}^e$ and $y_{t,i}^r$ correspond to a true fact, their values are 1, otherwise 0. T represents the total number of KG timestamps in the training dataset. It is worth noticing that $y_t^e \in \mathbb{R}^{|V|}$ represents the label vector representation of entity prediction task at timestamp t; $y_t^r \in \mathbb{R}^{|R|}$ represents the label vector representation of relationship prediction task.

The final loss can be represented sum of two tasks as follows.

$$\mathcal{L}_{final} = \lambda_1 \mathcal{L}^e + \lambda_2 \mathcal{L}^r, \tag{14}$$

where λ_1 and λ_2 are weighted parameters that control the weight of each loss term. λ_1 and λ_2 can be selected based empirically based on the tasks. If the task is to predict the object entity o given (s, r), we can set relatively small values for λ_1. And visa versa for λ_2.

5 Experiments

5.1 Experimental Setup

Datasets. To evaluate the performance of our PRGAN model, we conducted experiments on five commonly used datasets, namely WIKI [19], YAGO [20], GDELT [21], ICEWS05-15 [22], and ICEWS18 [22]. To ensure fairness and accuracy, we split each dataset into training, validation, and test sets in a ratio of 80%, 10%, and 10%, respectively. Table 2 presents the statistical information of the experimental datasets.

Table 2. More about datasets.

Datasets	$\|\mathcal{V}\|$	$\|\mathcal{R}\|$	$\|\mathcal{T}rain\|$	$\|\mathcal{V}alid\|$	$\|\mathcal{T}est\|$	$\mathcal{T}ime\ Gap$
WIKI	12,554	24	539,286	67,538	63,110	1 year
YAGO	10,623	10	161,540	19,523	20,026	1 year
GDELT	7,691	240	1,734,399	238,765	305,241	15 min
ICEWS05-15	10,094	251	368,868	46,302	46,159	24 h
ICEWS18	23,033	256	373,018	45,995	495,45	24 h

Decoder. For knowledge graph link prediction tasks, we usually use specific scoring functions to measure the plausibility of quadruples. Previous research has shown that some GCNs with convolutional scoring functions perform well in knowledge graph link prediction tasks. In order to better reflect the translation property of the progressive embedding of entities and relationships implied in Eq. (5), we chose ConvTransE [23] as our decoder. As we mentioned in previous section, ConvTransE decoder can also be replaced by other scoring functions.

Optimizer. In our model, we used the latest AI optimizer proposed by Google Brain - VeLO [24]. It is constructed entirely based on AI and does not require manual adjustment of any hyperparameters, which can be applied on different tasks and has an acceleration effect on training 83 tasks, surpassing a series of currently available optimizers.

Metrics. We use four evaluation metrics to measure our model, namely MRR and Hits@1/3/10.

Baselines. The PRGAN model is mainly compared with two types of models: the SKG reasoning model and the TKG reasoning model. The SKG reasoning model includes DistMult [8], RGCN [4], ConvTransE [23], and RotatE [25]. The TKG reasoning model includes TA-DistMult [26], HyTE [12], R-GCRN [27], and RE-Net [3].

5.2 Experimental Results

We conducted a detailed comparison of our proposed method with the above baseline models. The best score of each column is represented in bold black, and the second-best score is represented with an underline notation.

Results of Entity Prediction Task. The results of the entity prediction task are presented in Table 3 and Table 4.

As we can see, SKG reasoning methods perform much worse than PRGAN, mainly because they cannot capture the temporal information in facts. Moreover, the experimental results on the WIKI and YAGO datasets show a more significant improvement. This is mainly because the time intervals in these two datasets are much larger than the other two datasets. Therefore, we can conclude that at each timestamp, PRGAN can capture more structural dependencies between concurrent facts, which also justify that modeling complex structural dependencies between concurrent facts is necessary.

However, PRGAN did not achieve the best results on the GDELT dataset. Through our analysis, we found that in the GDELT dataset, most entities are abstract concepts. Thus, when PRGAN models the entire KG sequence, these abstract concept entities might affect the progression representation of other concrete entities.

Results of Relation Prediction Task. The results of the relation prediction tasks are presented in Table 5.

Since currently most models are mainly focused on entity prediction, we selected two typical baseline models, ConvTransE [23] and R-GCRN [27] for comparison. In Table 5, we only selected the MRR metric to analyze the results. After analysis, we found that the performance gap of PRGAN in relation prediction tasks is much smaller than in entity prediction tasks. This is mainly

Table 3. Results of entity prediction task on WIKI, YAGO, GDELT.

Model	WIKI			YAGO			GDELT			
	MRR	H@3	H@10	MRR	H@3	H@10	MRR	H@1	H@3	H@10
DistMult	27.96	32.45	39.51	44.05	49.70	59.94	8.61	3.91	8.27	17.04
R-GCN	13.96	15.75	22.05	20.25	24.01	37.30	12.17	7.40	12.37	20.63
ConvTransE	<u>30.89</u>	<u>34.30</u>	<u>41.45</u>	46.67	52.22	<u>62.52</u>	19.07	11.85	20.32	<u>33.14</u>
RotatE	26.08	31.63	38.51	42.08	46.77	59.39	3.62	0.52	2.26	8.37
TA-DistMult	26.44	31.36	38.97	44.98	50.64	61.11	10.34	4.44	10.44	21.63
HyTE	25.40	29.16	37.54	14.42	39.73	46.98	6.69	0.01	7.57	19.06
R-GCRN	28.68	31.44	38.58	43.71	48.53	56.98	18.63	11.53	19.80	32.42
RE-NET	30.87	33.55	41.27	<u>46.81</u>	<u>52.71</u>	61.93	**19.60**	**12.03**	**20.56**	**33.89**
PRGAN	**38.44**	**44.09**	**53.21**	**58.46**	**64.11**	**74.78**	<u>19.11</u>	<u>11.80</u>	<u>20.41</u>	32.08

Table 4. Results of entity prediction task on ICEWS05-15, ICEWS18.

Model	ICEWS05-15				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
DistMult	19.91	5.63	27.22	47.33	13.86	5.61	15.22	31.26
R-GCN	27.13	18.83	30.41	43.16	15.05	8.13	16.49	29.00
ConvTransE	30.28	20.79	33.80	49.95	23.22	14.26	26.13	41.34
RotatE	19.01	10.42	21.35	36.92	14.53	6.47	15.78	31.86
TA-DistMult	27.51	17.57	31.46	47.32	16.42	8.60	18.13	32.51
HyTE	16.05	6.53	20.20	34.72	7.41	3.10	7.33	16.01
R-GCRN	35.93	26.23	40.02	54.63	23.46	14.24	26.62	41.96
RE-NET	<u>36.86</u>	<u>26.24</u>	<u>41.85</u>	<u>57.60</u>	<u>26.17</u>	<u>16.43</u>	<u>29.89</u>	<u>44.37</u>
PRGAN	**38.16**	**27.23**	**42.87**	**60.03**	**27.29**	**18.62**	**30.22**	**47.10**

Table 5. Results of relation prediction task.

Model	WIKI	YAGO	GDELT	ICEWS05-15	ICEWS18
ConvTransE	86.64	<u>90.98</u>	**18.97**	38.26	<u>38.00</u>
R-GCRN	<u>88.88</u>	90.18	18.58	<u>38.37</u>	37.14
PRGAN	**93.57**	**93.11**	<u>18.82</u>	**38.40**	**38.94**

because in the dataset, the number of relations is much smaller than the number of entities. At the same time, the improvement on the WIKI and YAGO datasets is slightly larger than on the other three datasets because they contain fewer relations. In summary, the analysis of the results on the relation prediction task once again confirms the superiority of our model against baselines.

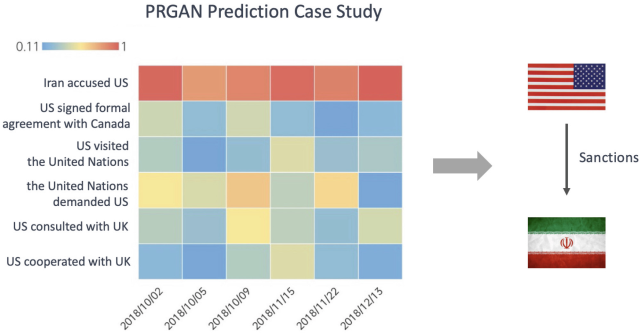

Fig. 3. Case Study - Heat Map.

5.3 Case Study

To further evaluate the performance of our PRGAN model, we conducted a case study on a selected subset of the test dataset's event-based Temporal Knowledge Graph (TKG). The subset was chosen based on its relevance to real-world events and the potential impact of predicting future facts accurately. As shown in Fig. 3, we obtained historical facts from the TKG at the timestamps of October 2018 and November 2018 and attempted to predict which country the United States would sanction at the timestamp of March 2019. Our hypothesis was that the United States and the United Kingdom had established a good relationship due to their "cooperation", while the United States and Iran had a hostile relationship due to "accusation". Since "sanction" is a hostile relationship, we predicted that "Iran" was more likely to be the entity that the United States would sanction based on the relationships between the entities in the TKG.

We conducted a heatmap analysis on this prediction, and as shown in Fig. 3, PRGAN successfully obtained the correct answer - Iran. This indicates that our model has good performance as expected.

6 Conclusion

In this paper, we proposed PRGAN. The model reduces the event prediction problem to an extrapolation reasoning problem in the temporal knowledge graph by learning the temporal and structural information of facts from the perspectives of historical facts and concurrent facts, and obtaining the progressive representation of entities and relations in the time series to make future predictions. The experimental results on five datasets demonstrate that PRGAN outperforms the baseline models in predicting future facts in Temporal Knowledge Graphs (TKGs), as measured by various metrics. Moreover, the use of the VeLO optimizer developed by Google Brain greatly improved the speed and effectiveness of training, enabling PRGAN to achieve state-of-the-art performance in predicting future facts in TKGs.

Acknowledgment. This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, China Higher Education Innovation Fund No. 2021ITA05010.

References

1. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: Knowledge graphs: a survey of techniques and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2016)
2. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th Proceedings of the International Conference on Machine Learning, vol. 70, pp. 3462–3471. ACM (2017)
3. Jin, W., Zhang, C., Szekely, P., Ren, X.: Recurrent event network for reasoning over temporal knowledge graphs. In: Proceedings of the 7th International Conference on Learning Representations (2019)
4. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
5. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2018)
6. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS (2013)
7. Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
8. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
9. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: NeurIPS (2018)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., et al.: Towards time-aware knowledge graph completion. In: Proceedings of the COLING, Osaka, Japan, pp. 1715–1724 (2016)
12. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the EMNLP, Brussels, Belgium, pp. 2001–2011 (2018)
13. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI, Quebec City, Quebec, Canada, pp. 1112–1119 (2014)
14. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI, New York, New York, USA, pp. 3988–3995 (2020)
15. Bai, L., Ma, X., Zhang, M., Wenting, Yu.: TPmod: a tendency-guided prediction model for temporal knowledge graph completion. ACM Trans. Knowl. Discov. Data **15**(3), 17 (2021). Article 41
16. Fu, G., et al.: TempCaps: a capsule network-based embedding model for temporal knowledge graph completion. In: Proceedings of the Sixth Workshop on Structured Prediction for NLP, pp. 22–31 (2022)

17. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems, pp. 3859–3869 (2017)
18. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
19. Yasseri, T., Bright, J., Margetts, H.: Wikipedia as a data source for political scientists: accuracy and completeness of coverage. PS: Polit. Sci. Polit. **45**(04), 711–716 (2012)
20. Mahdisoltani, F., Biega, J., Suchanek, F.: YAGO3: a knowledge base from multilingual Wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference (2014)
21. Leetaru, K., Schrodt, P.A.: GDELT: global data on facts, location, and tone, 1979–2012. In: ISA annual convention, vol. 2, 1–49. Citeseer (2013)
22. King, G., Lam, P., Roberts, M.: The integrated crisis early warning system (ICEWS)-a framework for the automated analysis of societal-level crisis early warning. J. Peace Res. **50**(2), 275–285 (2013)
23. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B.: End-to-end structure-aware convolutional networks for knowledge base completion (2019)
24. Lample, G., Sablayrolles, A., Ranzato, M., Larochelle, H.: Vector-Less objectives for neural machine translation. arXiv preprint arXiv:1902.01370 (2019)
25. Sun, Z., Deng, H., Nie, L., Tang, L., Yang, Y., Liu, Z.: RotatE: knowledge graph embedding by relational rotation in complex space. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4602–4609 (2019)
26. Du, X., Dai, H., He, M., Zhang, Y.: Temporal attentive knowledge graph embedding for predicting social facts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7488–7498 (2020)
27. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) ICONIP 2018. LNCS, vol. 11301, pp. 362–373. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04167-0_33