# Complex Evolutional Pattern Learning for Temporal Knowledge Graph Reasoning

**Zixuan Li[1,2,3]*, Saiping Guan[1,2], Xiaolong Jin[1,2], Weihua Peng[3], Yajuan Lyu[3], Yong Zhu[3], Long Bai[1,2], Wei Li[3], Jiafeng Guo[1,2], Xueqi Cheng[1,2]**

[1]School of Computer Science and Technology, University of Chinese Academy of Sciences;
[2]CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences; [3]Baidu Inc.
{lizixuan,guansaiping,jinxiaolong}@ict.ac.cn
{pengweihua,lvyajuan,zhuyong}@baidu.com

## Abstract

A Temporal Knowledge Graph (TKG) is a sequence of KGs corresponding to different timestamps. TKG reasoning aims to predict potential facts in the future given the historical KG sequences. One key of this task is to mine and understand evolutional patterns of facts from these sequences. The evolutional patterns are complex in two aspects, length-diversity and time-variability. Existing models for TKG reasoning focus on modeling fact sequences of a fixed length, which cannot discover complex evolutional patterns that vary in length. Furthermore, these models are all trained offline, which cannot well adapt to the changes of evolutional patterns from then on. Thus, we propose a new model, called Complex Evolutional Network (CEN), which uses a length-aware Convolutional Neural Network (CNN) to handle evolutional patterns of different lengths via an easy-to-difficult curriculum learning strategy. Besides, we propose to learn the model under the online setting so that it can adapt to the changes of evolutional patterns over time. Extensive experiments demonstrate that CEN obtains substantial performance improvement under both the traditional offline and the proposed online settings.

## 1 Introduction

Temporal Knowledge Graph (TKG) (Boschee et al., 2015; Gottschalk and Demidova, 2018, 2019; Zhao, 2020) has emerged as a very active research area over the last few years. Each fact in TKGs is a quadruple *(subject, relation, object, timestamp)*. A TKG can be denoted as a sequence of KGs with timestamps, each of which contains all facts at the corresponding timestamp. TKG reasoning aims to answer queries about future facts, such as *(COVID-19, New medical case occur, ?, 2022-1-9)*.

To predict future facts, one challenge is to dive deep into the related historical facts, which reflect the preferences of the related entities and affect their future behaviors to a certain degree. Such facts, usually temporally adjacent, may carry informative sequential patterns, called evolutional patterns in this paper. For example, [*(COVID-19, Infect, A, 2021-12-21), (A, Discuss with, B, 2021-12-25), (B, Go to, Shop, 2021-12-28)*] is an informative evolutional pattern for the above query implied in historical KGs. There are two kinds of models to model evolutional patterns, namely, query-specific and entire graph based models. The first kind of models (Jin et al., 2020; Li et al., 2021a; Sun et al., 2021; Han et al., 2020a, 2021; Zhu et al., 2021) extract useful structures (i.e., paths or subgraphs) for each individual query from the historical KG sequence and further predict the future facts by mining evolutional patterns from these structures. This kind of models may inevitably neglect some useful evolutional patterns. Therefore, the entire graph based models (Deng et al., 2020; Li et al., 2021a) take a sequence of entire KGs as the input and encode evolutional patterns among them, which exhibit superiority to the query-specific models.

However, they all ignore the length-diversity and time-variability of evolutional patterns. **Length-diversity**: The lengths of evolutional patterns are diverse. For example, [*(COVID-19, Infect, A, 2021-12-21), (A, Discuss with, B, 2021-12-25), (B, Go to, Shop, 2021-12-28)*] is a useful evolutional pattern of length 3 to predict the query *(COVID-19, New medical case occur, ?, 2022-1-9)* and [*(COVID-19, Infect, A, 2021-12-21), (A, Go to, Shop, 2021-12-30)*] is also a useful evolutional pattern of length 2 for this query. Previous models extract evolutional patterns of a fixed length, which cannot handle evolutional patterns of diverse lengths. **Time-variability**: Evolutional patterns change over time. For example, *(COVID-19, Infect, A, 2019-12-9)* and *(COVID-19, Infect, A, 2022-1-9)* may lead to different results due to the wide usage of the COVID-19 vaccines. Previous models learn from

*This work was done while the first author was doing internship at Baidu Inc.



*Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*
*Volume 2: Short Papers*, pages 290 - 296

the historical training data, which fail in modeling the time-variability of evolutional patterns after that.

Upon the above observations, we propose Complex Evolutional Network (CEN) to deal with the above two challenges. For length-diversity, CEN learns evolutional patterns from historical KG sequences of different lengths via an Relational Graph Neural Network (RGCN) based KG sequence encoder and a length-aware Convolutional Neural Network (CNN) based evolutional representation decoder. Besides, the model is trained via an easy-to-difficult curriculum learning strategy incrementally according to the length of KG sequences. For time-variability, we learn CEN under an online setting and combine CEN with a temporal regularization unit to alleviate the catastrophic forgetting problem (Mccloskey and Cohen, 1989).

In general, this paper makes the following contributions:

- We address, for the first time, the problems of length-diversity and time-variability of evolutional patterns for TKG reasoning.
- For length-diversity, we propose a length-aware CNN to learn evolutional patterns with different lengths in a curriculum learning manner. For time-variability, we propose to learn the model under an online setting to adapt to the changes of evolutional patterns.
- Experiments demonstrate that the proposed CEN model achieves better performance on TKG reasoning under both the traditional offline and the proposed online settings.

## 2 Related Work

The TKG reasoning task primarily has two settings, interpolation and extrapolation. This paper focus on the extrapolation setting. In what follows, we will introduce related work on both settings:

**TKG Reasoning under the interpolation setting.** This setting aims to complete the missing facts at past timestamps (Jiang et al., 2016; Leblay and Chekol, 2018; Dasgupta et al., 2018; Garcia-Duran et al., 2018; Goel et al., 2020; Wu et al., 2020). For example, TTransE (Leblay and Chekol, 2018) extends TransE (Bordes et al., 2013) by adding the temporal constraints; HyTE (Dasgupta et al., 2018) projects the entities and relations to time-aware hyperplanes to generate representations for different timestamps. Above all, they cannot obtain the representations of the unseen timestamps and are not suitable for the extrapolation setting.

**TKG Reasoning under the extrapolation setting** This setting aims to predict facts at future timestamps, which can be categorized into two groups: query-specific and entire graph based models. Query-specific models focus on modeling the query-specific history. For example, RE-NET (Jin et al., 2020) captures the evolutional patterns implied in the subgraph sequences of a fixed length specific to the query. CyGNet (Zhu et al., 2021) captures repetitive patterns by modeling repetitive facts. xERTE (Han et al., 2020a) learns to find the query-related subgraphs of a fixed hop number. CluSTeR (Li et al., 2021a) and TITer (Sun et al., 2021) both adopt reinforcement learning to discover evolutional patterns in query-related paths of a fixed length. Unlike the query-specific models, entire graph based models encode the latest historical KG sequence of a fixed-length. RE-GCN (Li et al., 2021b) captures the evolutional patterns into the representations of all the entities by modeling KG sequence of a fixed-length at lastest a few timestamps. Glean (Deng et al., 2020) introduces event descriptions to enrich the information of the entities.

## 3 Problem Formulation

A TKG $G = \{G_1, G_2, ..., G_t, ...\}$, where $G_t = (\mathcal{V}, \mathcal{R}, \mathcal{E}_t)$, is a directed multi-relational graph. $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{E}_t$ is the set of facts at timestamp $t$. The TKG reasoning task aims to answer queries like $(s, r, ?, t_q)$ or $(?, r, o, t_q)$ with the historical KG sequence $\{G_1, G_2, ..., G_{t_q-1}\}$ given, where $s, o \in \mathcal{V}$, $r \in \mathcal{R}$ and $t_q$ are the subject/object entity, the relation and the query timestamp, respectively. Following Jin et al. (2020), KGs from timestamps 1 to $T_1$, $T_1$ to $T_2$, $T_2$ to $T_3$ ($T_1 < T_2 < T_3$) are used as the training, validation and test sets, respectively. Under the traditional offline setting, models are trained only using the training set ($t_q \leq T_1$), while under the online setting, the model will be updated by KGs before $t_q$ ($T_1 < t_q \leq T_3$) continually. Without loss of generality, we describe our model as predicting the missing object entity.

## 4 Methodology

We propose CEN to deal with the length-diversity and time-variability challenges of evolutional pat-

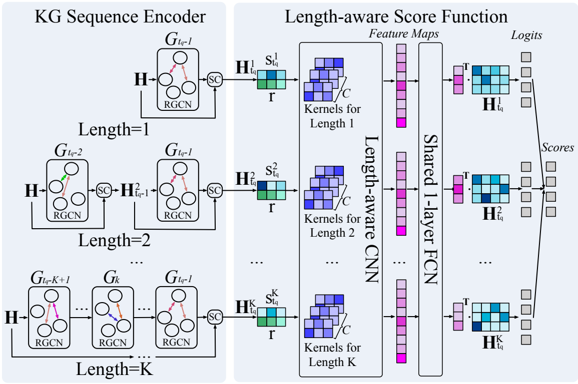


Figure 1: An diagram of the basic CEN model.

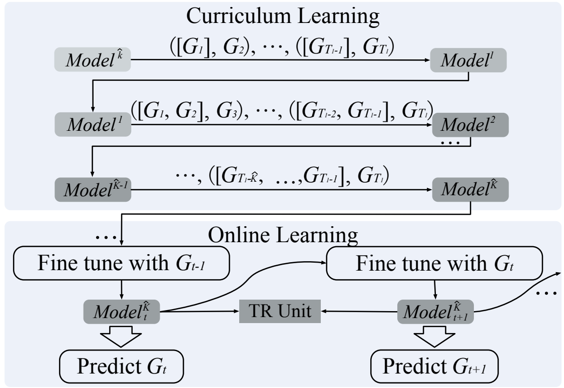


Figure 2: The learning procedure of the proposed model.

tern learning for TKG reasoning. Specifically, CEN consists of a basic model as well as a curriculum learning strategy for the former challenge and an online learning strategy for the latter challenge.

## 4.1 Basic CEN Model

As shown in Figure 1, the basic model of CEN contains a KG sequence encoder and an evolutional representation decoder. The KG sequence encoder encodes the latest historical KG sequences of different lengths to corresponding evolutional representations of entities. Then, the evolutional representation decoder calculates the scores of all entities for the query based on these representations.

**KG Sequence Encoder.** Its inputs include the lastest historical KG sequences of lengths from 1 to $K$, initial representations of entities $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and relation representations $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}| \times d}$, where $d$ is the dimension of the representations. Take the KG sequence of length $k = 2$ for example, for each KG in the input sequence $\{G_{t_q-2}, G_{t_q-1}\}$, it iteratively calculates the evolutional representations of entities $\mathbf{H}_t^2$ at the corresponding timestamps $t \in \{t_q - 1, t_q\}$ as follows:

$$\hat{\mathbf{H}}_t^2 = RGCN(\mathbf{H}_{t-1}^2, \mathbf{R}, G_{t-1}), \quad (1)$$

$$\mathbf{H}_t^2 = SC(\hat{\mathbf{H}}_t^2, \mathbf{H}_{t-1}^2), \quad (2)$$

where $RGCN(\cdot)$ and $SC$ denote the shared RGCN layer and the skip connection unit proposed in RE-GCN (Li et al., 2021b). For the initial timestamp $t_q - 1$, $\mathbf{H}_{t_q-2}^2$ is set to $\mathbf{H}$. $\mathbf{R}$ is shared across timestamps, which is different from RE-GCN. By reusing the encoder for KG sequences of different lengths, we obtain $K$ entity evolution representations at the query timestamp: $\{\mathbf{H}_{t_q}^1, ..., \mathbf{H}_{t_q}^k, ..., \mathbf{H}_{t_q}^K\}$.

**Evolutional Representation Decoder.** Multiple evolutional representations contain evolutional patterns of multiple lengths. To distinguish the influences of the length-diverse evolutional patterns, we design a length-aware CNN, which uses $K$ separate channels to model the above $K$ evolutional representations. Specifically, for a query $(s, r, ?, t_q)$, the representations of $s$ ($\mathbf{s}_{t_q}^1, ..., \mathbf{s}_{t_q}^k, ..., \mathbf{s}_{t_q}^K$) and $r$ ($\mathbf{r}$) are looked up from multiple representations of entities $\{\mathbf{H}_{t_q}^1, ..., \mathbf{H}_{t_q}^k, ..., \mathbf{H}_{t_q}^K\}$ and the shared relation representations $\mathbf{R}$. For historical KG sequence of length $k$, $k^{th}$ channel with $C$ different kernels of size $2 \times M$ is used to decode the concatenation of $\mathbf{s}_{t_q}^k$ and $\mathbf{r}$. Specifically, the feature maps are calculated as below,

$$\mathbf{m}_c^k(s, r, t_q) = Conv_{2D}(\mathbf{w}_c^k, [\mathbf{s}_{t_q}^k; \mathbf{r}]), \quad (3)$$

where $Conv_{2D}$ denotes the 2D convolution operation, $\mathbf{w}_c^k$ $(0 \leq c < C)$ are the trainable parameters in $c^{th}$ kernel of $k^{th}$ channel and $\mathbf{m}_c^k(s, r, t_q) \in \mathbb{R}^{1 \times d}$. After that, it concatenates the output vectors from $C$ kernels yielding a vector: $\mathbf{m}^k(s, r, t_q) \in \mathbb{R}^{C \times d}$. For $K$ channels, it outputs a list of vectors: $[\mathbf{m}^1(s, r, t_q)$,... , $\mathbf{m}^k(s, r, t_q)$,..., $\mathbf{m}^K(s, r, t_q)]$. Then, each vector is fed into a shared 1-layer Fully Connected Network (FCN) with $\mathbf{W}_3 \in \mathbb{R}^{Cd \times d}$ as its parameters and the final score of a candidate entity $o$ is the sum of the logits from multiple evoltional representations: $\sum_{k=1}^{K} \mathbf{m}^k(s, r, t_q)\mathbf{W}_3\mathbf{o}^k$, where $\mathbf{o}^k$ is the evolutional representation of length $k$ for $o$. Then we seen it as a multi-class learning problem and use the cross-entropy as its objective function.

## 4.2 Curriculum Learning for Length-diversity

Longer historical KG sequences contain more historical facts and longer evolutional patterns, which is more challenging to learn. Similar to human learning procedures, the models can benefit from an easy-to-difficult curriculum. Besides, how to

| Datasets | ICEWS14 | ICEWS18 | WIKI |
|---|---|---|---|
| $\#\mathcal{E}$ | 6,869 | 23,033 | 12,554 |
| $\#\mathcal{R}$ | 230 | 256 | 24 |
| $\#Train$ | 74,845 | 373,018 | 539,286 |
| $\#Valid$ | 8,514 | 45,995 | 67,538 |
| $\#Test$ | 7,371 | 49,545 | 63,110 |
| $T_3$ | 1 day | 1 day | 1 year |

Table 1: Statistics of the datasets. $\#Train$, $\#Valid$, $\#Test$ are the numbers of facts in the training, validation and test sets.

choose the maximum length of evolutional patterns is vital to CEN. Thus, we design the curriculum learning strategy to learn the length-diverse evolutional patterns from short to long and adaptively select the optimal maximum length $\hat{K}$. As shown at the top of Figure 2, we start from the minimum length $\hat{k}$ ($\hat{k} = 1$ for example) and gradually move on to longer history in the training set. The model stops the curriculum and gets the optimal $\hat{K}$ when the MRR metric decreases or the length is up to maximum length $K$. Note that, curriculum learning is conducted under the traditional offline setting and $Model^{\hat{K}}$ is used as the pre-trained model for online learning.

### 4.3 Online Learning for Time-variability

To handle the time-variability of evolutional patterns, one simple and direct method is to update the model according to the newly occurred facts. Thus, as shown in the bottom of Figure 2, for timestamp $t+1$ ($T_1 < t+1 < T_3$), $Model_t^{\hat{K}}$ is fine-tuned to get $Model_{t+1}^{\hat{K}}$ by predicting the facts in the KG at the last timestamp $G_t$ with historical KG sequences as inputs. Furthermore, to balance the knowledge of new evolutional patterns and the existing ones, we use a Temporal Regularization unit (TR unit) (Daruna et al., 2021; Wu et al., 2021). We apply an $L2$ regularization constraint between two temporally adjacent models to smooth the drastic change of the parameters.

### 4.4 Analysis on Computational Complexity

We analyze the computational complexity of CEN. We view the computational complexities of the RGCN unit and ConvTransE as constants. Then, the time complexity of the RGCN at a timestamp $t$ is $O(|\mathcal{E}|)$, where $|\mathcal{E}|$ is the maximum number of facts at timestamps in history. As we unroll $m$ ($m = \hat{K} - \hat{k}$) sequences, the time complexity of the KG sequence encoder is finally $O(m^2|\mathcal{E}|)$. Thus, the time complexity of CEN is $O(m^2|\mathcal{E}| + m)$.

## 5 Experiments

**Experimental Setup.** We adopt three widely-used datasets, ICEWS14 (Li et al., 2021b), ICEWS18 (Jin et al., 2020), and WIKI (Leblay and Chekol, 2018) to evaluate CEN. Dataset statistics are demonstrated in Table 1. Due to the space limitation, the CEN model is only compared with the latest models of TKG reasoning: CyGNet (Zhu et al., 2021), RE-NET (Jin et al., 2020), xERTE (Han et al., 2020a), TG-Tucker (Han et al., 2021), TG-DistMult (Han et al., 2021), TiTer (Sun et al., 2021) and RE-GCN (Li et al., 2021b). In the experiments, we adopt MRR (Mean Reciprocal Rank) and Hits@{1,3,10} as the metrics for TKG reasoning. We averaged the metrics over five runs. Note that, following Han et al. (2020b), we adopt an improved filtered setting where the timestamps of facts are considered, called time-aware filtered setting. Take a typical query $(s, r, ?, t_1)$ with answer $o_1$ in the test set for example, and assume there is another two facts $(s, r, o_2, t_2)$ and $(s, r, o_3, t_1)$. Under this time-aware filtered setting, only $o_3$ will be considered as a correct answer and thus removed from the ranking list of candidate answers.

**Implementation Details.** In the experiments, the optimal minimum lengths of evolutional patterns $\hat{k}$ for ICEWS14, ICEWS18, WIKI are 3, 3, 2, respectively. The maximum length $K$ for all datasets is set to 10. For all datasets, the kernel width $M$ is set to 3, and $C$ is set to 50. For each fact $(s, r, o, t)$ in the test set, we evaluate CEN on two queries $(s, r, ?, t)$ and $(?, r, o, t)$. The dimension $d$ of relation representations and entity representations is set to 200 on all datasets. Adam (Kingma and Ba, 2014) is adopted for parameter learning with the learning rate of 0.001 on all datasets. The number of RGCN layers is set to 2 and the dropout rate for each layer to 0.2. For the online setting, we set the max epochs of the fine-tuning at each timestamp to 30. For predicting $G_t$, $G_{t-2}$ is used as the validation set. We fine tune the pre-trained CEN from $T1 + 1$ to $T_3$ and report the results at the test timestamps ($T_2$ to $T_3$) in Table 3. The experiments are carried out on Tesla V100. Codes are avaliable at https://github.com/Lee-zix/CEN.

### 5.1 Experimental Results

**Results under the Offline Setting.** The results under the traditional offline setting are presented in Table 2. CEN consistently outperforms the

| Model | ICEWS14 | | | | ICEWS18 | | | | WIKI | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| CyGNet | 35.05 | 25.73 | 39.01 | 53.55 | 24.93 | 15.90 | 28.28 | 42.61 | 33.89 | 29.06 | 36.10 | 41.86 |
| RE-NET | 36.93 | 26.83 | 39.51 | 54.78 | 28.81 | 19.05 | 32.44 | 47.51 | 49.66 | 46.88 | 51.19 | 53.48 |
| xERTE | 40.02 | 32.06 | 44.63 | 56.17 | 29.31 | 21.03 | 33.51 | 46.48 | 71.14 | 68.05 | 76.11 | 79.01 |
| TG-Tucker | - | - | - | - | 28.68 | 19.35 | 32.17 | 47.04 | 50.43 | 48.52 | 51.47 | 53.58 |
| TG-DistMult | - | - | - | - | 26.75 | 17.92 | 30.08 | 44.09 | 51.15 | 49.66 | 52.16 | 53.35 |
| TITer | 40.97 | **32.28** | 45.45 | 57.10 | 29.98 | **22.05** | 33.46 | 44.83 | 75.50 | 72.96 | 77.49 | 79.02 |
| RE-GCN | 40.39 | 30.66 | 44.96 | 59.21 | 30.58 | 21.01 | 34.34 | 48.75 | 77.55 | 73.75 | 80.38 | 83.68 |
| CEN | **42.20** | 32.08 | **47.46** | **61.31** | **31.50** | 21.70 | **35.44** | **50.59** | **78.93** | **75.05** | **81.90** | **84.90** |

Table 2: Experimental results on TKG reasoning (in percentage) under the offline setting.

| Model | ICEWS14 | | | | ICEWS18 | | | | WIKI | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 | MRR | H@1 | H@3 | H@10 |
| CEN(-TR) | 39.28 | 30.05 | 43.58 | 57.01 | 31.11 | 21.41 | 35.09 | 50.27 | **81.92** | **77.93** | **85.23** | **87.63** |
| CEN | **43.34** | **33.18** | **48.49** | **62.58** | **32.66** | **22.55** | **36.81** | **52.50** | 79.67 | 75.63 | 83.00 | 85.58 |

Table 3: Experimental results on TKG reasoning (in percentage) under the online setting.

| metrics | MRR | H@1 | H@3 | H@10 |
|---|---|---|---|---|
| CEN | 42.20 | 32.08 | 47.46 | 61.31 |
| CEN(-CL) | 41.50 | 31.53 | 46.50 | 60.81 |
| CEN(-LA) | 41.52 | 31.49 | 46.74 | 60.65 |

Table 4: Ablation Study of CEN on ICEWS14.

baselines on MRR, Hits@3, and Hits@10 on all datasets, which justifies the effectiveness of modeling the evolutional patterns of different lengths. On ICEWS datasets, CEN underperforms TITer on Hits@1 because TITer retrieves the answer through explicit paths, which usually gets high Hits@1. Whereas, CEN recalls more answer entities by aggregating the information from multiple evolutional patterns, which may be the reason for its high performance on Hits@3 and Hits@10.

**Results under the Online Setting.** Under the online setting, the model is updated via historical facts at the testset. Thus, it cannot be directly compared with the baselines designed for the offline setting. As shown in Table 3, on ICEWS datasets CEN outperforms CEN(-TR) (CEN without TR unit), which implies the effectiveness of TR unit to balance the knowledge of new evolutional patterns and the existing ones. On WIKI, CEN(-TR) gets better performance. It is because that the time interval between two adjacent timestamps in WIKI (one year) is much larger than ICEWS datasets (one day) and contains more time-variable evolutional patterns. TR unit limits the model to adapt to new knowledge and is not suitable for this dataset.

**Ablation Study.** To investigate the contributions of curriculum learning strategy and the length-aware CNN, we conduct ablation studies for CEN on the test set of ICEWS14 under the traditional offline setting, which are shown in Table 4. CEN(-CL) denotes CEN without the curriculum learning strategy. The underperformance of CEN(-CL) demonstrates the effectiveness of the curriculum learning strategy. CEN(-LA) denotes the model replacing the length-aware CNN with a traditional CNN. The underperformance of CEN(-LA) implies the effectiveness of the length-aware CNN.

## 6 Conclusions

In this paper, we proposed Complex Evolutional Network (CEN) for TKG reasoning, which deals with two challenges in modeling the complex evolutional patterns: length-diversity and time-variability. For length-diversity, CEN adopts a length-aware CNN to learn evolutional patterns of different lengths and is trained under a curriculum learning strategy. For time-variability, we explored a new online setting, where the model is expected to be updated to new evolutional patterns emerging over time. Experimental results demonstrate the superiority of the proposed model under both the offline and the online settings.

## Acknowledgments

The work is supported by the National Natural Science Foundation of China under grants U1911401, 62002341 and 61772501, the GFKJ Innovation Program, Beijing Academy of Artificial Intelligence under grant BAAI2019ZD0306, and the Lenovo-CAS Joint Lab Youth Scientist Project.

## References


Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12.

Angel Daruna, Mehul Gupta, Mohan Sridharan, and Sonia Chernova. 2021. Continual learning of knowledge graph embeddings. *IEEE Robotics and Automation Letters*, 6(2):1128–1135.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595.

Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995.

Simon Gottschalk and Elena Demidova. 2018. Eventkg: A multilingual event-centric temporal knowledge graph. In *European Semantic Web Conference*, pages 272–287. Springer.

Simon Gottschalk and Elena Demidova. 2019. Eventkg–the hub of event knowledge on the web–and biographical timeline generation. *Semantic Web*, (Preprint):1–32.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020a. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020b. Graph hawkes neural network for forecasting on temporal knowledge graphs. *8th Automated Knowledge Base Construction (AKBC)*.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Encoding temporal information for time-aware link prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2350–2354.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776. International World Wide Web Conferences Steering Committee.

Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2021a. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4732–4743.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021b. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417.

M. Mccloskey and N. J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319.

Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5730–5746.

Jiapeng Wu, Yishi Xu, Yingxue Zhang, Chen Ma, Mark J Coates, and Jackie Chi Cheung. 2021. Tie: A framework for embedding-based incremental temporal knowledge graph completion.

Liang Zhao. 2020. Event prediction in big data era: A systematic survey. *arXiv preprint arXiv:2007.09815*.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4732–4740.