



# ONSEP: A Novel Online Neural-Symbolic Framework for Event Prediction Based on Large Language Model

Xuanqing Yu<sup>1,2,3</sup>, Wangtao Sun<sup>1,2,3</sup>, Jingwei Li<sup>1,2</sup>, Kang Liu<sup>1,2</sup>, Chengbao Liu<sup>1,2†</sup>, Jie Tan<sup>1,2</sup>

<sup>1</sup>*Institute of Automation, Chinese Academy of Sciences, Beijing, China*

<sup>2</sup>*School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China*

<sup>3</sup>*AI Lab, AIGility Cloud Innovation, Beijing, China*

{yuxuanqing2021, sunwangtao2021, lijingwei2019, liuchengbao2016, jie.tan}@ia.ac.cn  
{kliu}@nlpr.ia.ac.cn

## Abstract

In the realm of event prediction, temporal knowledge graph forecasting (TKGF) stands as a pivotal technique. Previous approaches face the challenges of not utilizing experience during testing and relying on a single short-term history, which limits adaptation to evolving data. In this paper, we introduce the Online Neural-Symbolic Event Prediction (ONSEP) framework, which innovates by integrating dynamic causal rule mining (DCRM) and dual history augmented generation (DHAG). DCRM dynamically constructs causal rules from real-time data, allowing for swift adaptation to new causal relationships. In parallel, DHAG merges short-term and long-term historical contexts, leveraging a bi-branch approach to enrich event prediction. Our framework demonstrates notable performance enhancements across diverse datasets, with significant Hit@k (k=1,3,10) improvements, showcasing its ability to augment large language models (LLMs) for event prediction without necessitating extensive retraining. The ONSEP framework not only advances the field of TKGF but also underscores the potential of neural-symbolic approaches in adapting to dynamic data environments.

## 1 Introduction

Event prediction is a widely researched topic (Zhao, 2021; Benzin and Rinderle-Ma, 2023) since accurate prediction of future events allows one to minimize losses associated with certain future events. To model large amounts of real-world event data that represent complex interactions between entities over time, the temporal knowledge graph (TKG) has been introduced (Ding et al., 2023; Yuan et al., 2023). TKG is used to represent structural relationships among entities through timestamped quadruples  $(s, r, o, t)$ , where  $s$  and  $o$  are entities,  $r$  is a binary relation

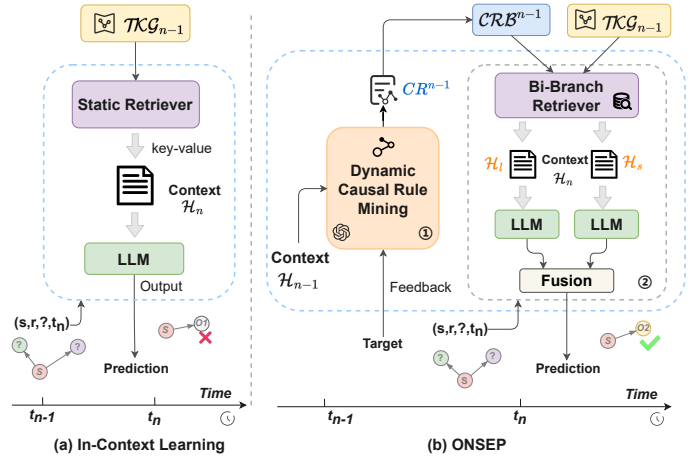


Figure 1: Comparison of ONSEP and ICL Frameworks for Event Prediction with Schematic Overview of ONSEP’s Core Components and Operational Processes.

between them, and  $t$  specifies the time when the event  $(s, r, o)$  occurs. For example, the quadruple  $(Angela\ Merkel, visit, China, 2014/07/04)$  indicates that Angela Merkel visited China on July 4, 2014. In this task, temporal knowledge graph forecasting (TKGF) aims to predict future events of the graph by inferring missing entities in a quadruple for a future time. This involves generating predictions for either the object entity  $(s, r, ?, t+k)$  or the subject entity  $(?, r, o, t+k)$  by utilizing historical data from previous snapshots, where  $k$  represents the number of time steps or intervals into the future beyond the current time  $t$ . Existing research studies have provided theoretical methodologies for time-sensitive applications such as recommendation systems (Wang et al., 2022; Zhao et al., 2022), financial analysis (Li, 2023), and social crisis early warning systems (Gastinger et al., 2023).

Traditional approaches (Jin et al., 2020; Zhu et al., 2021; Li et al., 2021; Han et al., 2020; Sun et al., 2021; Li et al., 2022) involve converting event data into TKGs and combining graph neural networks (GNNs) and recurrent neural net-

<sup>†</sup>Corresponding author.

\*The data and code of this paper can be found at <https://github.com/aqSeabiscuit/ONSEP>.

works (RNNs) capture evolving entity relationships through embeddings. However, these methods must perform model training on specific datasets, which is resource-intensive. With the proven understanding and generative capabilities of large language models (LLMs), recent studies have explored methods on LLMs (Tao et al., 2023; Lee et al., 2023; Shi et al., 2023b). The method proposed by (Lee et al., 2023) offers a more adaptable method for TKGf with in-context learning (ICL) on LLMs, allowing LLMs to adapt to TKGf by using examples in the context, without fine-tuning.

Nevertheless, due to limitations in the length of history of LLM inputs, this approach may not fully capture long-term trends among events and cannot effectively leverage past insights, such as causal relationships between events. Imagine a TKG scenario that relates to everyday life. An example quadruple could be: ‘(Sarah, Consult, Dr. Smith, 2022/04/10)’, indicating that Sarah consulted Dr. Smith on April 10, 2022. In this case, the ICL method may use static key values like ‘Sarah’ and ‘(Sarah, Consult)’ to extract historical events. However, this method may not account for Sarah’s tendency to begin consultations with phone calls months before meeting in person. This is evident in events such as ‘Discuss by telephone’ being a preceding cause, for example, ‘(Sarah, Discuss by telephone, Dr. Smith, 2022/01/08)’. This long-standing pattern of initiating consultations with a call is a crucial piece of historical context that short-term data analysis could miss. Without considering these longer-term causal interactions, the LLM-based method may not make accurate future predictions. Besides, in datasets like ICEWS (Boschee et al., 2015), the distribution of data containing entities and relations is dynamically changing. The emergence of new relations during test time poses challenges for traditional ICL methods, which rely on fixed keyword-key value matching and cannot adapt over time. This highlights the need for methods that can dynamically capture and apply updates in real-time, enhancing adaptability in event prediction.

To overcome these limitations, this paper proposes the **Online Neural-Symbolic Event Prediction (ONSEP)** framework. It enhances both accuracy and adaptability in event prediction by addressing inadequate long-term causal relationship capture and enabling real-time adaptability

for self-improvement without fine-tuning. Figure 1 illustrates a comparison of the ONSep and ICL approaches. By contrast, ONSep mainly has two novel components: 1) **Dynamic causal rule mining (DCRM)**: Utilizing LLMs’ external knowledge, DCRM semantically detects cause-effect links and dynamically constructs causal rules. This enables ONSep to quickly adapt to new data and causal relationships, facilitating real-time updates and leveraging past experiences without extensive retraining. 2) **Dual History Augmented Generation (DHAG)**: DHAG employs the *long short-term bi-branch retriever (LSBBR)* and a *hybrid model inference (HMI)* strategy to merge short-term and long-term historical contexts. The latter benefits from causal rules derived during the DCRM phase, enabling a broader event to be captured within the limits of historical input length. Inspired by the multi-branch fusion inference technology described in (Shi et al., 2023a), the HMI strategy applies weighted fusion to balance the contributions from both dual historical contexts. These innovations address the limitations of context length constraints in LLMs and improve the retrieval of relevant historical events over extended periods.

ONSEP shows significant performance gains on various datasets using InternLM2-7B model, achieving Hit@1 improvements over ICL of 9.63%, 9.35%, and 16.28% at history length 100, and 8.14%, 8.64%, and 15.25% at 200, and achieved competitive performance of embedding-based models trained on specific datasets. To summarize, our main contributions include:

- We introduce DCRM, an innovative real-time adaptive causal learning module for LLMs that automatically updates the rule base at the snapshot level during testing.
- We develop the DHAG module, which uses LSBBR and HMI strategies, allowing LLMs to effectively use historical data from different time scales for causal analysis.
- Our framework includes an adaptive RAG solution that improves historical event retrieval and achieves self-improvement.
- We demonstrate ONSep’s effectiveness across various models and datasets, showcasing its ability to enhance black-box LLM inference without the need for fine-tuning or manual annotations, thereby providing a robust and adaptable solution for diverse event prediction tasks.

## 2 Preliminaries

### 2.1 Temporal Knowledge Graph Forecasting

A temporal knowledge graph (TKG) is structured as a time-sequenced series of multi-relational directed graphs. The TKG up to time  $t$  is represented as  $\mathcal{TKG}_t = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_t\}$ , where each  $\mathcal{G}_t = (\mathcal{V}, \mathcal{R}, \mathcal{E}_t)$  represents a snapshot of the graph at time  $t$ . Here,  $\mathcal{V}$  denotes the set of entities,  $\mathcal{R}$  the set of relations, and  $\mathcal{E}_t$  comprises timestamped facts as quadruples  $(s, r, o, t)$ , with  $s, o \in \mathcal{V}$  and  $r \in \mathcal{R}$ . TKGF aims to predict future states of the graph by inferring missing entities in a quadruple for a future time. This involves generating predictions for either the object entity  $(s, r, ?, t + k)$  or the subject entity  $(?, r, o, t + k)$  using historical data from previous snapshots  $\mathcal{TKG}_t$ .

### 2.2 ICL for Temporal Knowledge Graph Forecasting

ICL enables LLMs to adjust to new tasks through contextual examples, without the need for fine-tuning. Specifically, in TKGF, ICL harnesses the adaptability of LLMs for forecasting by leveraging historical data. For a query of future event  $q = (s_q, r_q, ?, t_n)$ , where  $s_q$  is an entity and  $r_q$  is a relation at timestamp  $t_n$ , this method employs static keys, such as entity  $s_q$  or the pair  $(s_q, r_q)$ , to retrieve the historical event chain  $H_n(q)$ , a set of quadruples, from previous snapshots  $\mathcal{TKG}_{n-1} = \mathcal{G}_{1:n-1}$ . Then the method constructs a prompt  $\theta_q$  based on  $H_n(q)$ . The prediction  $y_q$  is generated by leveraging the LLM’s probability distribution  $y_q \sim P_{\text{LLM}}(y_q | \theta_q)$ , employing ICL to generate forecasts without further training.

To effectively handle multi-word entity and relation names, a numeric mapping  $\mathcal{M}_{\text{ent}}$  and  $\mathcal{M}_{\text{rel}}$  assigns unique labels to entities and relations. For example, candidate entities like [South Africa, China, New England] are mapped to numerical values [0, 1, 2] to align the outputs from the LLM with these candidates. During in-context learning, LLMs perform a forward pass to produce logits  $s$  for next-token predictions. These logits are then transformed into a probability distribution  $\mathbf{D}$  using the softmax function, representing the likelihood of each candidate being the target entity or relation.

### 2.3 Causal Rules in TKG

Causal rules<sup>1</sup> in TKG capture cause-and-effect links between events, denoted as  $CR(r_e, r_c) : (X, r_e, Y, T_2) \leftarrow (X, r_c, Y, T_1)$ , where  $X$  and  $Y$  represent anonymized entities, and  $T_1$  and  $T_2$  are timestamps, with  $T_1 < T_2$  ensuring the correct temporal sequence from cause to effect.

Extending this, we define a causal rule base ( $\mathcal{CRB}$ ) as a set of tuples, each comprising a causal rule (CR) and its confidence score (conf). The  $\mathcal{CRB}$  for effect  $r_e$  at timestamp  $t_n$  is denoted as:  $\mathcal{CRB}^n(r_e) = \{((X, r_e, Y, T_2) \leftarrow (X, r_{c_i}, Y, T_1), \text{conf}_i^n) \mid 1 \leq i \leq m\}$ , where  $r_{c_i}$  indicates the cause relation,  $r_e$  denotes the resulting relation influenced by  $r_c$ , and  $\text{conf}_i^n$  is the confidence score for the  $i$ -th causal rule at timestamp  $t_n$ , a real number within  $[0, 1]$ . Here,  $m$  represents the total number of causal rules considered in the rule base.

## 3 Method

The ONSEP framework (Figure 2) aims for event prediction via two key modules: (1) DCRM, which adaptively adjusts to changing data distributions during single-step prediction testing without requiring extensive training data, and (2) DHAG, which integrates patterns from short-term historical events with causality from long-term event developments. Detailed explanations are provided in the following sections.

### 3.1 Dynamic Causal Rule Mining

The dynamic causal rule mining (DCRM) phase is crucial for identifying causal relationships within temporal knowledge graph. This phase utilizes a semantic-driven rule learning algorithm to discover causal rules. It is followed by a dynamic update module that updates the causal rule base ( $\mathcal{CRB}$ ), which is closely followed by a mechanism for rule filtering and sorting by confidence.

#### 3.1.1 Semantic-Driven Rule Learning

This module is designed for the reflective learning of causal rules, structured around three core steps: candidate causes filter, causality assessment, and causal rule construction. Initially, it retrieves historical context  $\mathcal{H}_n$  for all queries from  $\mathcal{G}_n$  and real-time feedback  $O_y$ , which is the verified outcome for a query obtained when the system receives new

<sup>1</sup>In this work, we use "causal" to intuitively convey the role these rules play in identifying and retrieving events with potential causative relations during real-time analysis, differing from the strict definition in statistical causal inference.

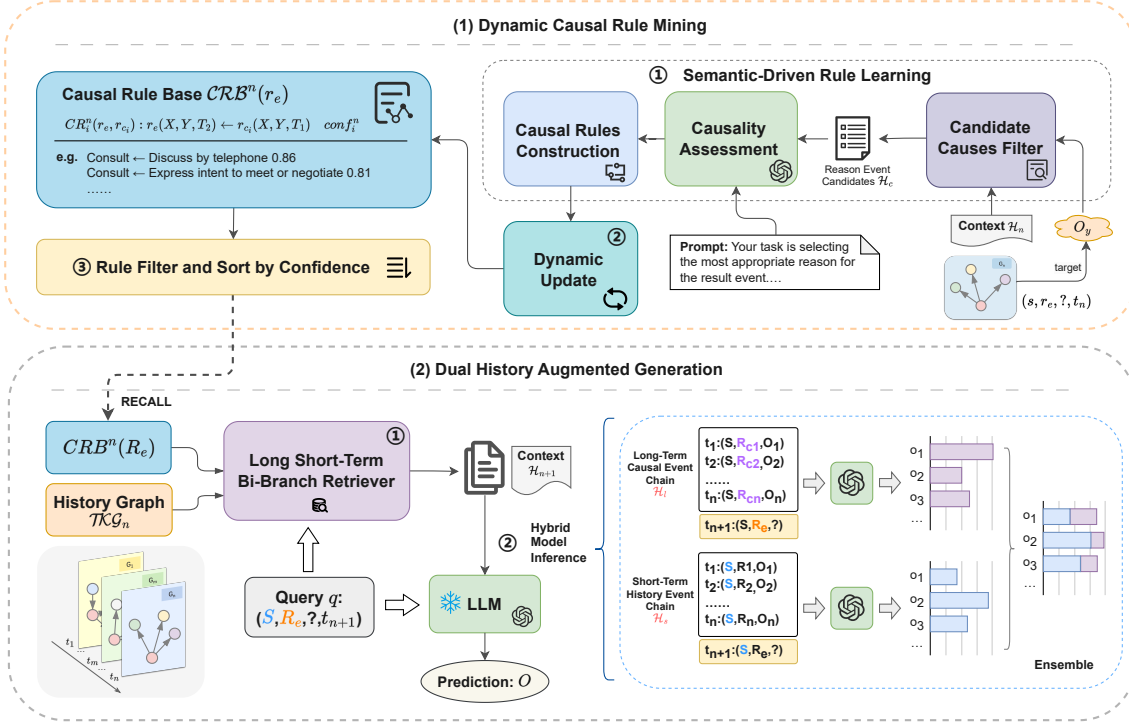


Figure 2: Detailed structure of ONSEP framework with two phases: (1) Dynamic causal rule mining (§ 3.1) and (2) dual history augmented generation (§ 3.2). Specially, the DCRM phase employs a semantic-driven algorithm (§ 3.1.1) to identify causal rules and dynamically updates the rule base (§ 3.1.2), incorporating a filtering and sorting mechanism (§ 3.1.3). (2) The DHAG phase utilizes a long short-term bi-branch retriever (§ 3.2.1) alongside a hybrid model inference (§ 3.2.2) strategy to improve prediction accuracy.

data, to filter potential reason events. In the causality assessment phase, these candidates are then evaluated using a LLM to determine the most plausible cause event  $r_c$ . Finally, the identified cause event, along with the query’s effect event  $r_e$  forms the basis for constructing a causal rule  $CR(r_e, r_c)$ . The algorithm of semantic-driven rule learning is outlined in Appendix A.

**Candidate Event Filter** In the candidate event filtering phase, we utilize historical context and real-time feedback related to the query to filter potential causal events. Given a query  $q = (S_q, R_e, ?, t_n)$  at time  $t_n$ , we define the context  $\mathcal{H}_n(S_q)$  as a historical sequence of events associated with subject  $S_q$ , expressed as  $\mathcal{H}_n = \{(S_q, r, o, t) \in \mathcal{G}_n\}$ . We select candidates based on the criterion that the object  $o$  is the same as the target  $O_y$ , thereby forming a set of candidate reason events  $\mathcal{H}_c$ , formalized as:

$$\mathcal{H}_c(q) = \{(S, r_c, \hat{o}, \hat{t}) \in \mathcal{H}_n(S_q) \mid \hat{o} = O_y\}.$$

Here,  $r_c$  represents a potential causal relation,  $\hat{o}$  is the filtered object that matches the target  $O_y$ , and

$\hat{t}$  is the timestamp associated with the event. This methodology aims to pinpoint potential causative relations corresponding to  $r_e$  within the event sequence that links the query’s subject  $S$  to the target entity.

For each distinct causative event  $r_c$  within the candidates of reason events, we first extract the set of quadruples from  $\mathcal{H}_c$  that contain  $r_c$ , which we denote as  $\mathcal{T}_{r_c}$ :

$$\mathcal{T}_{r_c} = \{(s, r, o, t) \mid r = r_c, (s, r, o, t) \in \mathcal{H}_c(q)\}.$$

We then compute the support number  $supp(r_c)$  as the count of quadruples of this set:  $supp(r_c) = |\mathcal{T}_{r_c}|$ . Subsequently, we calculate the coverage rate  $cove(r_c)$  for each event  $r_c$  using:

$$cove(r_c) = \frac{supp(r_c)}{|\mathcal{H}_c(q)|}. \quad (1)$$

This coverage rate aids in assessing the confidence level of causal rules and supports the review and selection of causal event candidates by the LLM.



### LLM-based Causal Event Selection

Your task is selecting the most appropriate reason for the result event. The result event is  $\{ r_e \}$ . Below is a list of possible reasons:  
 $\{ \text{candidate causal events: } [label]. [candidate] \}$   
 The most appropriate reason is:

Table 1: Causal Event Selection Prompt Template.

**Causality Assessment** In this step, the causal evaluation utilizes the powerful semantic understanding capabilities of the LLM to assess the degree of causal association between a set of potentially causal events. This evaluation aims to select the causal rules that are most relevant to the current query and feedback goals. A pivotal aspect of our methodology involves the formulation of a structured prompt  $\theta_1$  (Table 1), designed to direct the LLM towards discerning the most pertinent causal link between a given result event ( $r_e$ ) and potential causes within the candidate reason events set  $\mathcal{H}_c(q)$ . The selection mechanism utilizes LLM to obtain the logits  $L$  for numerically mapped candidate reasons, as same as the way how to generate an output for each test query. These logits represent the preliminary evaluation of each candidate’s likelihood to be the true causal event. To convert these logits into normalized probabilities, we apply the softmax function, yielding the probability  $p_{r_{c_i}}$  for each candidate reason  $r_{c_i}$ , mapped to label  $id_i$  using  $\mathcal{M}_{rel}$ , reprinted as  $s = LLM(\theta_1(r_e, \mathcal{H}_c(q)))$ , where  $s$  are the logits produced by LLM, The probability of each candidate reason  $r_{c_i}$  is then computed by:

$$p(r_{c_i}) = \frac{e^{s[id_i]}}{\sum_{j \in \mathcal{M}_{rel}} e^{s[id_j]}}. \quad (2)$$

Based on these probabilities, the top- $k$  candidates are selected. The confidence  $conf_i^t$  for each causal rule at timestamp  $t$  is determined by combining the probability  $p(r_{c_i})$  with the coverage score  $cove(r_{c_i})$ , can obtained as:

$$conf^t(r_{c_i}) = \alpha \cdot p(r_{c_i}) + (1 - \alpha) \cdot cove(r_{c_i}) \quad (3)$$

where  $\alpha$  is a tuning parameter that balances the contribution of the probability  $p(r_{c_i})$  and the coverage score  $cove(r_{c_i})$ , integrating the causal assessment with historical occurrence insights.

**Causal Rule Construction** Following the causality assessment step, a set of selected causal events ( $r_c$ ) and their corresponding confidence scores at the current time step ( $t_n$ ) are obtained. These causal rules are formalized in a structured format, with each rule at timestamp  $t$  represented as  $CR^t(r_e, r_c) = (X, r_e, Y, T_2) \leftarrow (X, r_c, Y, T_1)$ , accompanied by a confidence value  $conf^t(CR(r_e, r_c))$ , can be obtained by  $conf^t(r_c)$ .

### 3.1.2 Dynamic Update

The dynamic update module for the causal rule base adds new rules directly and updates existing ones by adjusting their confidence levels. It assigns a confidence score  $c_t$  to each causal rule  $CR$ , linking a pair of relations, and this score is dynamically updated at each time step  $t$ . This score merges the previous confidence  $c_{t-1}$  with the current confidence  $conf$  calculated at time  $t$ , effectively updating the confidence for each pair of relations within the causal rule.

It uses a smoothing factor ( $\theta$ ) to stabilize confidence adjustments, and a growth factor ( $\beta$ ) to incrementally evaluate the reliability of the circular verification rules, but the historical confidence cannot exceed 1. The update formula of the existing rules is:

$$c_t = \theta \cdot f_g(c_{t-1}) + (1 - \theta) \cdot conf \quad (4)$$

$$f_g(c_{t-1}) = \min(c_{t-1} \cdot (1 + \beta), 1) \quad (5)$$

where  $c_t$  is the updated confidence,  $conf$  the current evaluated confidence of  $CR$ , and  $c_{t-1}$  the previous confidence,  $f_g$  represent the grow function with the growth factor  $\beta$ .

This approach ensures continuous optimization of the rule base, allowing for adaptability in light of distribution changes over time.

### 3.1.3 Rule Filter and Sort by Confidence

In the final module of the DCRM, rules falling below a predefined confidence threshold, denoted as  $conf_{min}$ , are filtered out to ensure the usability of the causal rule base. Subsequently, the remaining rules are sorted based on their confidence levels, allowing rules with higher confidence to take precedence in the subsequent reasoning phase.

## 3.2 Dual History Augmented Generation

DHAG, an innovative RAG variant, introduces a long short-term bi-branch retriever (LSTBBR) coupled with a hybrid model inference (HMI) strategy.

DHAG synergizes short-term historical event patterns with long-term causal trajectories, ensuring a comprehensive historical context. Unlike previous methods that model long and short-term histories through data associations, DHAG provides a more robust handling of low-frequency events, unaffected by irrelevant noise in long histories.

### 3.2.1 Long Short-Term Bi-Branch Retriever

The long short-term bi-branch retriever (LSTBBR) within the DHAG framework incorporates a dual retrieval strategy to enhance the predictive model with a rich historical context, comprising both short-term and long-term historical events. For each prediction query  $q = (S_q, R_e, ?, t_{n+1})$ , LSTBBR extracts two distinct historical contexts:  $\mathcal{H}_s$  representing the short-term history event chain and  $\mathcal{H}_l$  representing the long-term causal event chain. This dual approach ensures a comprehensive understanding of the immediate events as well as the underlying long-term causal influences. The short-term history event chain ( $\mathcal{H}_s$ ) focuses on capturing the most recent events that are temporally proximate to the prediction query, providing an immediate context that reflects the latest developments. To construct  $\mathcal{H}_s$ , the system retrieves events from  $\mathcal{H}_n(S_q)$ , sorting them by timestamp and truncating to include only the  $L$  most recent events. This truncated list of events forms  $\mathcal{H}_s$ , which is directly associated with the query’s subject  $S_q$ , thereby providing a snapshot of the most immediate historical backdrop relevant to the query.

On the other hand, the long-term causal event chain ( $\mathcal{H}_l$ ) is designed to uncover the broader causal dynamics that have shaped the subject  $S$  over a more extended period. This is achieved by initially retrieving cause rules from  $\mathcal{CRB}(r_e)$  using a RECALL mechanism, which employs a key-value approach where the effect relation  $r_e$  is used as the key to efficiently extract associated causal rules.

Subsequently, cause events are filtered to construct the long-term cause event chain  $\mathcal{H}_l$  from  $\mathcal{H}_n(S_q)$ . The events in  $\mathcal{H}_l$  are determined by the criteria:  $(S_q, R_{c_i}, o_i, t_i)$  where  $CR(R_e, R_{c_i}) \in \mathcal{CRB}(R_e)$  and  $(S, R_{c_i}, o_i, t_i) \in \mathcal{H}_n(S_q)$ , with  $t_i < t_{n+1}$ . Similar to  $\mathcal{H}_s$ ,  $\mathcal{H}_l$  is truncated to include only the most recent  $L$  events since limited by the max length of model input.

### 3.2.2 Hybrid Model Inference

Before inference, a numerical label mapping technique preprocesses multi-word entities to prepare them for integration into the model, similar to the causality assessment in DCRM. The essence of the hybrid model inference (HMI) strategy involves merging the query  $q$  with the short-term history event chain  $\mathcal{H}_s$  and the long-term causal event chain  $\mathcal{H}_l$ . This allows the LLM to produce distinct probabilities for each context, which are then combined using a weighted ensemble approach.

For the short-term context, the logits  $s_1$  are obtained by  $s_1 = p(y|q \oplus \mathcal{H}_s)$ , where  $\oplus$  symbolizes concatenation. The logits  $s_1$  are then normalized using the softmax function to produce a probability distribution  $D_1 = \text{softmax}(s_1)$ . Similarly, for the long-term context, the logits  $s_2$  are derived by  $s_2 = p(y|q \oplus \mathcal{H}_l)$ , and the corresponding probability distribution  $D_2$  is obtained by applying the softmax function to  $s_2$ , resulting in  $D_2 = \text{softmax}(s_2)$ . The integration strategy involves a weighted ensemble of  $D_1$  and  $D_2$ , formulated as:

$$D = D_1 \cdot (1 - \lambda) + D_2 \cdot \lambda. \quad (6)$$

Here,  $\lambda$  is a tuning parameter that balances the contributions of short-term and long-term contexts. This unified distribution  $D$  ranks candidate entities by their relevance to  $q$ , leveraging the nuanced insights from both  $\mathcal{H}_s$  and  $\mathcal{H}_l$ . By doing so, the HMI strategy enhances the LLM’s accuracy in entity predictions, grounded in a comprehensive understanding of dual historical contexts.

## 4 Experiments

### 4.1 Experimental Settings

**Datasets** Our experimental evaluation is carried out on a subset of the integrated crisis early warning system (ICEWS) dataset, which includes versions such as ICEWS14 (García-Durán et al., 2018), ICEWS05-15 (García-Durán et al., 2018), and ICEWS18 (Jin et al., 2020). These datasets are composed of timestamped records of political events, making them highly suitable for conducting temporal analysis. They are widely recognized as benchmark datasets on TKGf. Each event is represented as a tuple, such as (Barack Obama, visit, Malaysia, 2014/02/19), capturing diverse political activities across various time periods.

**Evaluation Metrics** To evaluate our method’s efficiency in ranking event candidates, we use link

prediction metrics like Hit@ $k$  (where  $k = 1, 3, 10$ ). This measures the precision of our model in forecasting future events within the top  $k$  predictions. Higher Hit@ $k$  values indicate more accurate rankings, which, in product recommendations, translate to better purchase predictions and greater economic benefits.

**Baselines** We primarily compare ONSEP with the ICL method. Additionally, we have selected several traditional supervised models based on embedding methods for performance comparison, including RE-NET (Jin et al., 2020), CyGNet (Zhu et al., 2021), RE-GCN (Li et al., 2021), xERTE (Han et al., 2020), and TITer (Sun et al., 2021) in TKG. We also compare our method with a variety of LLMs.

Further information about the datasets, evaluation metrics, LLMs used and implementation specifics can be found in the Appendix B.

## 4.2 Performance Comparison

To evaluate whether ONSEP surpasses the previous best event prediction method based on LLMs, ICL, we conduct experiments with historical inputs of 100 and 200, using InternLM2-7B for all methods. As Table 2 shows, our method outperforms ICL across all three datasets. Specifically, with a history length of 100, ONSEP achieves Hit@1 improvements of 9.63%, 9.35%, and 16.28%, and with 200, the gains are 8.14%, 8.64%, and 15.25%. These results confirm ONSEP’s capability to exceed the performance of the previous ICL method. These results underscore ONSEP’s superior performance over ICL, particularly in Hit@1 accuracy, indicating a notable enhancement in precision.

While ONSEP trails behind some trained embedding models in Hit@10 due to the LLM’s reliance on ranking candidate entities from the input history rather than all entities, it achieves top performance in Hit@1 for the ICEWS14 and ICEWS05-15 datasets. Additionally, it shows closely competitive results in other metrics. This highlights ONSEP’s capability to significantly enhance accuracy. For a detailed exploration of ONSEP’s operational dynamics and its real-world applicability, see the ICEWS14 case study in Appendix C.

## 4.3 Inductive Setting and the Effectiveness of DCRM

To assess the transferability of causal rules mined during test-time iterations, we conduct experiments where rules learned from ICEWS14 are applied to

predictions on ICEWS18, with findings presented in Table 3. There are significant temporal spans and differences in data distribution between ICEWS14 and ICEWS18. For example, ICEWS18 includes entities such as Donald Trump, who served as the President of the United States from 2017 to 2021, which are not present in ICEWS14.

With only a 20.3% overlap (see Table 5) in entities between the two datasets, our inductive experimental setup demonstrates performance improvements. Comparing the ICL method without pre-loaded rules (i) to the approach using ICEWS14-derived rules (iv), we observe a significant performance boost in the latter, indicating that the DCRM-mined causal rules are generalizable across datasets with similar relational structures, thereby improving inference.

Further analysis on the DCRM module’s impact shows that incorporating DCRM (iii) versus not incorporating it (iv) leads to enhanced performance across all metrics, including a notable 9.52% increase in Hit@1. This underscores DCRM’s effectiveness in improving the accuracy and recall of inference.

Interestingly, real-time rule mining with DCRM without pre-loaded rules (ii) is slightly higher than that of the inductive setting (iii) (i.e., with pre-loading the rules learned from  $G_{test_1}$ ), suggesting that relying on potentially outdated rules may hinder adaptation to new data. Due to the smoothing setup in dynamic updates, the old rule set may introduce some interference, hindering rapid adaptation to the new test set. This emphasizes the need for dynamic rule updates to ensure model relevance and effectiveness across varying datasets with dynamically changing distributions.

## 4.4 Effectiveness of DHAG

To assess the DHAG module’s impact, we explored how blending short-term and long-term history contexts affects performance on the ICEWS14 dataset, with similar findings observed on other datasets. We vary the Weighted Fusion Ratio from 0 to 1, as shown in Figure 3, with details in Appendix D.2.

Our findings indicate a clear pattern: a  $\lambda$  value of 0, which effectively uses only short-term history chain akin to the baseline ICL method, leads to lower performance. Conversely, a  $\lambda$  of 1, relying solely on long-term reasoning, also underperforms compared to a balanced approach. The optimal performance at a fusion ratio of 0.1 indicates that the model primarily utilizes short-term dynamic his-

Model	Train	ICEWS14			ICEWS05-15			ICEWS18		
		Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
RE-NET	✓	0.301	0.440	0.582	0.336	0.488	0.627	0.197	0.326	0.485
CyGNet	✓	0.274	0.426	0.579	0.294	0.461	0.616	0.172	0.310	0.469
RE-GCN	✓	0.316	<b>0.472</b>	<b>0.617</b>	0.373	<b>0.539</b>	<b>0.685</b>	<b>0.224</b>	<b>0.368</b>	<b>0.527</b>
xERTE	✓	0.327	0.457	0.573	0.378	0.523	0.639	0.210	0.335	0.465
TITer	✓	<b>0.328</b>	0.465	0.584	<b>0.383</b>	0.528	0.649	0.221	0.335	0.448
InternLM2-7B-ICL (L=100)	✗	0.301	0.432	0.560	0.353	0.507	0.647	0.172	0.289	0.434
InternLM2-7B-ONSEP (L=100)	✗	0.330	0.464	0.570	0.386	0.546	0.662	0.200	0.324	0.443
$\Delta Improve$		9.63%	7.41%	1.79%	9.35%	7.69%	2.32%	16.28%	12.11%	2.07%
InternLM2-7B-ICL (L=200)	✗	0.307	0.443	0.567	0.359	0.520	0.659	0.177	0.300	0.447
InternLM2-7B-ONSEP (L=200)	✗	<b>0.332</b>	<b>0.465</b>	<b>0.577</b>	<b>0.390</b>	<b>0.551</b>	<b>0.668</b>	<b>0.204</b>	<b>0.333</b>	<b>0.453</b>
$\Delta Improve$		8.14%	4.97%	1.76%	8.64%	5.96%	1.37%	15.25%	11.00%	1.34%

Table 2: Performance comparison among LLM-based ICL and traditional embedding-based methods on three datasets with time-aware metrics (Hit@k). The highest performance is highlighted in **bold**.  $\Delta Improve$  represents the percentage improvement of ONSEP over ICL. The results of the embedding-based models are excerpted from (Li et al., 2022).

	Method (InternLM2-7B)	Hit@1	Hit@3	Hit@10
(i)	ICL	0.156	0.265	0.382
(ii)	ONSEP	<b>0.186</b>	<b>0.305</b>	<b>0.424</b>
(iii)	ONSEP - inductive	0.184	0.302	0.422
(iv)	ONSEP w/o DCRM (ICL w/ DHAG) - inductive	0.168	0.292	0.416

Table 3: Analysis on DCRM module under inductive setting. Utilizing causal rules derived by ONSEP from the test graph  $G_{test_1}$  (ICEWS14, with a history input length of  $L=50$ ) and applying them to a different test graph  $G_{test_2}$  (ICEWS18, with a history input length of  $L=30$ ).

tory for reasoning, while also incorporating long-term causal knowledge acquired during test time to improve its effectiveness.

#### 4.5 Performance Comparison of Different Model Scale and Series

Our analysis reveals that model performance improves with increased parameter scale, with the 20B models outperforming the 7B models, aligning with the scaling law. However, the performance gain from increasing the model size from 7B to 20B is less pronounced and comes with higher computational costs. The ONSEP method enhances performance across different model scales and series, demonstrating its adaptability and effectiveness. Detailed comparisons across model series indicate that ONSEP’s improvements are consistent, with InternLM2-7b showing the most significant gains. Further in-depth analysis and discussions are provided in Appendix D.1.

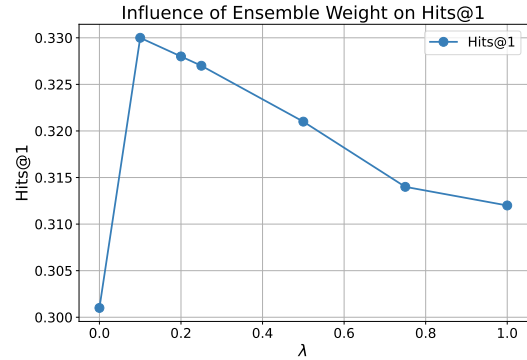


Figure 3: Performance of ONSEP in terms of Hit@1 across various DHAG ensemble weights  $\lambda$  of DHAG. The underlying LLM is InternLM2-7B, processing input histories of length 100.  $\lambda$  represents the weight given to long-term causal event chains. This illustrates how varying  $\lambda$  influences the integration of short-term and long-term reasoning contexts within ONSEP.

#### 4.6 Hyperparameter Sensitivity Analysis

Our hyperparameter sensitivity analysis highlights the critical role of selecting the right history length  $L$ , rule selection thresholds  $k$ , and fusion ratios  $\alpha$  for causal rule confidence scores. Finding the right balance between the length of historical input and computational efficiency is essential for optimal model performance. Similarly, precise calibration of rule selection thresholds and fusion ratios is vital for the effective application and updating of causal rules, taking into account factors such as the smoothing factor  $\theta$  and growth factor  $\beta$ . These parameters collectively influence the model’s ability to adapt and perform accurately over time. A more detailed discussion on these findings and their



implications is provided in the Appendix E.

## 5 Related Work

### 5.1 Temporal Knowledge Graph Forecasting

In TKGF, traditional embedding-based methods (Jin et al., 2020; Zhu et al., 2021; Li et al., 2021; Han et al., 2020; Sun et al., 2021; Li et al., 2022) learn representations of the quadruple, showing efficacy in supervised learning. Recent efforts explore LLMs for event prediction. For example, Xu et al. (Xu et al., 2023) use a masking strategy to make the forecasting task similar to predicting missing words. LAMP (Shi et al., 2023b) utilizes LLMs as a cause generator to reorder candidate outcomes, while ICL has been applied in TKGF (Lee et al., 2023), transforming forecasting into a sequence generation task. Besides, GenTKG (Liao et al., 2023) leverages few-shot parameter-efficient instruction tuning of LLMs using the training set to enhance inference capabilities. These approaches hold promise for improved generalization and contextual understanding. However, their effectiveness in dynamic real-world scenarios warrants further investigation.

On the other hand, rule-based approaches like Tlogic (Liu et al., 2022) focus on interpretability and generalization by learning symbolic rules from TKGs. Despite their strengths, these methods struggle with large search spaces, can't incorporate textual semantics of relations, and rely on static rule sets, limiting their adaptability.

### 5.2 Retrieve Augmented Generation

Retrieval-augmented generation (RAG) significantly enhances LLMs by integrating dynamic external knowledge retrieval (Lewis et al., 2020), mitigating common challenges like hallucinations and slow information updates. Advancements like REPLUG (Shi et al., 2023a) and AAR (Yu et al., 2023) further enhance RAG. REPLUG refines retrieval models with supervised feedback from language models. AAR, on the other hand, is a versatile plugin trained with a single source LLM but adaptable to various target LLMs. These adaptive mechanisms of RAG could be particularly beneficial in LLM-based TKGF, improving semantic alignment between queries and retrieved history context.

### 5.3 Self-Improving on LLMs

Researchers have recently proposed methods by using LLM's inherent knowledge as an external database to let LLMs self-improve without an-

notated datasets and parameter updates. Frameworks like ExpNote (Sun et al., 2023), HtT (Zhu et al., 2023), and MoT (Li and Qiu, 2023) facilitate learning from experience, rule induction, and high-confidence thought generation. However, their application to tasks with temporal dimensions remains to be explored. Following this idea, we proposed an ONSEP framework to enable the model to induce rules in real-time during the testing process and then use them for future predictions.

## 6 Conclusion

In this paper, we introduce a novel online neural-symbolic framework, ONSEP, that integrates LLMs with TKGs to achieve adaptive and precise event forecasting in a dynamic online environment. To overcome the challenges of not utilizing experience during testing and relying on a single short-term history, which limits adaptation to new data, we propose a dynamic causal rule mining module and a dual history augmented generation module within the ONSEP framework. This design allows LLMs to access the most recent history to identify patterns, as well as causal relationships from a broader range of past events. Extensive experiments conducted on three benchmark datasets have proven the efficacy of ONSEP in TKGF, surpassing previous methods and demonstrating its broad applicability across diverse LLMs. Our framework shows great potential for future applications in financial forecasting, public sentiment monitoring, and recommendation systems.

### Limitations

Our method requires multiple uses of large models, leading to increased inference time (see Appendix F) and higher computational costs compared to simpler models. The effectiveness of the model is influenced by the length of the input context; longer contexts are only useful for LLMs designed to handle them. The method also faces interpretability challenges due to unclear reasoning paths, which can be particularly evident in applications like campaign strategy analysis with the ICEWS dataset.

Additionally, the method's potential to improve performance is limited for data that lacks a rich semantic understanding or detailed relationships needed to extract comprehensive causal rules. Since there is an upper limit on the length, the model inference uses indexing for the cue words and does not use the lexical setting, ignoring the effect of entity semantics on the results.

## Ethics Statement

This paper presents a novel online neural-symbolic framework for event prediction, particularly tailored for dynamic real-world environments like TKGF. All experiments are conducted on publicly available datasets. Thus there is no data privacy concern. Meanwhile, this paper does not involve human annotations, and there are no related ethical concerns.

## Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3304602 and the National Nature Science Foundation of China under Grant 62003344.

## References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. *Qwen technical report*. *ArXiv preprint*, abs/2309.16609.
- Janik-Vasily Benzin and Stefanie Rinderle-Ma. 2023. A survey on event prediction methods from a systems perspective: Bringing together disparate research areas. *arXiv preprint arXiv:2302.04018*.
- Sidney Black, Stella Biderman, Eric Hallahan, Quentin Gregory Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. In *Challenges & Perspectives in Creating Large Language Models*.
- E. Boschee, J. Lautenschlager, S. O’Brien, S. Shellman, J. Starz, and M. Ward. 2015. *ICEWS Coded Event Data*.
- Zifeng Ding, Zongyue Li, Ruoxia Qi, Jingpei Wu, Bailan He, Yunpu Ma, Zhao Meng, Shuo Chen, Ruotong Liao, Zhen Han, et al. 2023. Forecasttkgquestions: A benchmark for temporal question answering and forecasting over temporal knowledge graphs. In *International Semantic Web Conference*, pages 541–560. Springer.
- Alberto García-Durán, 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, Brussels, Belgium. Association for Computational Linguistics.
- Julia Gastinger, Nils Steinert, Sabine Gründer-Fahrer, and Michael Martin. 2023. Dynamic representations of global crises: Creation and analysis of a temporal knowledge graph for conflicts, trade and value networks.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. *xerte: Explainable reasoning on temporal knowledge graphs for forecasting future links*. *ArXiv preprint*, abs/2012.15537.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. *Recurrent event network: Autoregressive structure inference over temporal knowledge graphs*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, Online. Association for Computational Linguistics.
- Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. *Temporal knowledge graph forecasting without knowledge using in-context learning*. *ArXiv preprint*, abs/2305.10613.
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. *Retrieval-augmented generation for knowledge-intensive NLP tasks*. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Xiaohui Victor Li. 2023. Findkg: Dynamic knowledge graph with large language models for global finance. Available at SSRN 4608445.
- Xiaonan Li and Xipeng Qiu. 2023. Mot: Memory-of-thought enables chatgpt to self-improve. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Yujia Li, Shiliang Sun, and Jing Zhao. 2022. Tirgn: time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2152–2158. ijcai. org.
- Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 408–417.
- Ruotong Liao, Xu Jia, Yunpu Ma, and Volker Tresp. 2023. Gentkg: Generative forecasting on temporal knowledge graph. In *Temporal Graph Learning Workshop@ NeurIPS 2023*.
- Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4120–4127.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics (ACL).
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023a. [Replug: Retrieval-augmented black-box language models](#). *ArXiv preprint*, abs/2301.12652.
- Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023b. [Language models can improve event prediction by few-shot abductive reasoning](#). *ArXiv preprint*, abs/2305.16646.
- 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Wangtao Sun, Xuanqing Yu, Shizhu He, Jun Zhao, and Kang Liu. 2023. Expnote: Black-box large language models are better task solvers with experience notebook. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Zhengwei Tao, Zhi Jin, Xiaoying Bai, Haiyan Zhao, Yanlin Feng, Jia Li, and Wenpeng Hu. 2023. Eval: A comprehensive evaluation of event semantics for large language models. *arXiv preprint arXiv:2305.15268*.
- InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *ArXiv preprint*, abs/2307.09288.
- Xiaolin Wang, Guohao Sun, Xiu Fang, Jian Yang, and Shoujin Wang. 2022. Modeling spatio-temporal neighbourhood for personalized point-of-interest recommendation. In *Proceedings of IJCAI*.
- Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023. [Pre-trained language model with prompts for temporal knowledge graph completion](#). *ArXiv preprint*, abs/2305.07912.
- Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. [Baichuan 2: Open large-scale language models](#). *ArXiv preprint*, abs/2309.10305.
- Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. [Augmentation-adapted retriever improves generalization of language models as generic plug-in](#). *ArXiv preprint*, abs/2305.17331.
- Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2023. Back to the future: Towards explainable temporal reasoning with large language models. *arXiv preprint arXiv:2310.01074*.
- Liang Zhao. 2021. Event prediction in the big data era: A systematic survey. *ACM Computing Surveys (CSUR)*, 54(5):1–37.
- Yuyue Zhao, Xiang Wang, Jiawei Chen, Yashen Wang, Wei Tang, Xiangnan He, and Haiyong Xie. 2022. Time-aware path reasoning on knowledge graph for recommendation. *ACM Transactions on Information Systems*, 41(2):1–26.
- 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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4732–4740. AAAI Press.
- Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. 2023. [Large language models can learn rules](#). *ArXiv preprint*, abs/2310.07064.

## A Algorithm of Semantic-Driven Rule Learning

In this section we introduce the Semantic-Driven Rule Learning algorithm in DCRM. The pseudo code can be found at algorithm A.

## B Experimental Details

In the experiment, we used the ICEWS data sets, including ICEWS14, ICEWS05-15 and ICEWS18, we use the training and validation sets to build historical graphs, which then serve to test our model’s predictive accuracy. The specific parameters are shown in the table 4.

### B.1 Datasets

Dataset	# Ents	# Rels	Train	Valid	Test	Interval
ICEWS14	7,128	230	74,845	8,514	7,371	24 hours
ICEWS05-15	10,094	251	368,868	46,302	46,159	24 hours
ICEWS18	23,033	256	373,018	45,995	49,545	24 hours

Table 4: Statistics of the datasets.

ICEWS14 and ICEWS18 have significant differences in data distribution. Specifically, the number of entities in ICEWS18 far exceeds that in

**Algorithm 1** Semantic-Driven Rule Learning

**Require:** Historical context  $\mathcal{H}_n$  at time  $t_n$ , Query  $(S_q, R_e, ?, t_n)$ , target  $O_y$

**Ensure:** Causal Rules  $\mathcal{CRB}^n(R_e)$

- 1: **Initialize:**  $\mathcal{H}_c \leftarrow \emptyset$ ,  $\mathcal{CRB}^n(R_e) \leftarrow \mathcal{CRB}^{n-1}(R_e)$ ,  $supp \leftarrow \emptyset$ ,  $cove \leftarrow \emptyset$
- 2: **for** each event  $(S_q, r_{c_i}, o_i, t_i)$  in  $\mathcal{H}_n$  **do**
- 3:   **if**  $o_i$  aligns with  $O_y$  **then**
- 4:      $\mathcal{H}_c \leftarrow \mathcal{H}_c \cup \{(S_q, r_{c_i}, o_i, t_i)\}$
- 5:      $supp[r_{c_i}] \leftarrow supp[r_{c_i}] + 1$
- 6:   **end if**
- 7: **end for**
- 8: **for** each  $r_{c_i}$  in  $\mathcal{H}_c$  **do**
- 9:    $p_i \leftarrow \text{LLM-Causality-Assessment}(r_{c_i}, R_e)$
- 10:   Compute  $cove_i \leftarrow \frac{supp[r_{c_i}]}{|\mathcal{H}_c|}$
- 11: **end for**
- 12:  $\mathcal{H}_{c_{topk}} \leftarrow$  Select top- $k$   $r_{c_i}$  based on  $p_i$
- 13: **for** each  $r_{c_i}$  in  $\mathcal{H}_{c_{topk}}$  **do**
- 14:   Compute  $conf_i^n \leftarrow \alpha \cdot p_i + (1 - \alpha) \cdot cove_i$
- 15:    $CR^n(R_e, r_{c_i}) \leftarrow (X, R_e, Y, T_2) \Leftarrow (X, r_{c_i}, Y, T_1)$
- 16:    $\mathcal{CRB}^n(R_e) \leftarrow \mathcal{CRB}^n(R_e) \cup CR^n(R_e, r_{c_i})$
- 17: **end for**
- 18: **return**  $\mathcal{CRB}^n(R_e)$

ICEWS14. This can be clearly demonstrated in the dataset statistics:

Dataset	Entities	Relations	Time Span
ICEWS14	7,128	230	2014
ICEWS18	23,033	256	2018
Overlap	4,685	226	-
Overlap (%)	20.3%	88.3%	-

Table 5: Overlap statistics between ICEWS14 and ICEWS18. The "Overlap (%)" shows the percentage of entities and relations in ICEWS18 that overlap with ICEWS14.

## B.2 Evaluation Metrics

To assess ONSEP’s ability to rank candidates for event prediction without recalling all entities, we use link prediction metrics like Hit@k (where  $k=1, 3, 10$ ). Hit@k is an evaluation metric that measures how often the model correctly places entities within the top k positions of the ranking list for a given query. Our evaluation specifically focuses on a time-aware setting, where we apply filtering to remove valid candidates not pertinent to the specific query.

## B.3 Baselines

All large models used in the paper’s experiments are shown here, and the series and parameters are reported, as shown in Table 6.

Model Family	Model Name	# Params
Qwen (Bai et al., 2023)	qwen-7b	7B
Baichuan2 (Yang et al., 2023)	baichuan2-7b	7B
LLaMA2 (Touvron et al., 2023)	llama-2-7b	7B
InternLM2 (Team, 2023)	internlm2-7b	7B
	internlm2-20b	20B
GPT-NeoX (Black et al., 2022)	gpt-neox-20b	20B

Table 6: LLMs Used in the Study.

## B.4 Implementation Details

The experimental setup for the ICL method aligns with that of TKG-ICL (Lee et al., 2023), except for a change in the model. The input provided to the model is based on indexing rather than lexical content. For the 20B model, due to computational resource constraints, we employed a 4-bit quantization approach for loading the model and performing inference. All experiments were conducted on GeForce RTX 3090 GPU with 24GB Memory.

## C Case Study

This section delves into the ICEWS14 case study, demonstrating ONSEP’s innovative approach in event prediction. The aim is to showcase the method’s operational dynamics and its practical relevance.

### C.1 Operational Setup and Input Scenarios

Initially, consider an input history length limit of 5 events. For the query scenario involving a hypothetical entity **Thief** engaged in **Use unconventional violence**, ONSEP’s task is to predict potential outcomes based on past events. The ICL method, serving as a comparative baseline, identifies only a short-term history sequence. In contrast, ONSEP extends the analysis both short-term and long-term historical data as Figure 4 shows.



7968: [Thief, Use unconventional violence, 0.Citizen]
7968: [Thief, Fight with small arms and light weapons, 0.Citizen]
7992: [Thief, Use unconventional violence, 1.Men]
7992: [Thief, Use unconventional violence, 0.Citizen]
7992: [Thief, Fight with small arms and light weapons, 0.Citizen]
8016: [Thief, Use unconventional violence,

(a) the Short-Term History Event Chain

7560: [Thief, Fight with small arms and light weapons, 0.Citizen]
7824: [Thief, Fight with small arms and light weapons, 1.Employee]
7920: [Thief, Fight with small arms and light weapons, 0.Citizen]
7968: [Thief, Fight with small arms and light weapons, 0.Citizen]
7992: [Thief, Fight with small arms and light weapons, 0.Citizen]
8016: [Thief, Use unconventional violence,

(b) the Long-Term Causal Event Chain

Figure 4: Example of short-term and long-term historical event chains used by ONSEP.

## C.2 Causal Analysis and Predictive Accuracy

ONSEP employs learned causal rules to enhance prediction accuracy, for instance, inferring that **Use unconventional violence** may evolve from **Fight with small arms and light weapons** with a confidence score (*conf*: 0.76). This analysis is dynamically derived from both the LLM’s assessment and observed data frequencies.

When multiple rules apply, ONSEP retrieves relevant causative events from the historical graph for each rule and contextualizes them chronologically. This process leverages direct and precise causal relationships and enables ONSEP to integrate a broader range of historical insights, extending further back in time than baseline models. As the temporal dataset grows, ONSEP’s rule repository continuously evolves, improving the accuracy of historical context retrieval and updating the confidence levels of relational rules.

## D Additional Experiment Results

### D.1 Performance Comparison in Different Model Series

**Model Scale** To analyze the impact of the model parameter scale on performance, we selected the 7B and 20B models from the InternLM2 series as the foundation models for ONSEP, with an input length of 100. As shown in Table 7, the 20B model performs better than the 7B model on the ICL method, consistent with our expectations and in line with the scaling law. The ONSEP method demonstrates significant improvements across both model scales, but the growth on the 20B model is relatively lower than on the 7B, and it is more time-

consuming. The improvements in Hit@1 are 9.63% and 2.45% for the 7B and 20B models, respectively. This also proves that larger parameter LLMs have advantages in feature capturing, enabling better reasoning performance, but it’s important to consider the balance between performance gains and computational costs.

Model	Hit@1	Hit@3	Hit@10	Time
<b>InternLM2-7b-ICL</b>	0.301	0.432	0.560	-
<b>InternLM2-7b-ONSEP</b>	0.330	0.464	0.570	2 h 32 min
$\Delta$ Improve	9.63%	7.41%	1.79%	
<b>InternLM2-20b-ICL</b>	0.326	0.455	0.57	-
<b>InternLM2-20b-ONSEP</b>	<b>0.334</b>	<b>0.467</b>	<b>0.571</b>	7h 10 min
$\Delta$ Improve	2.45%	2.64%	0.18%	

Table 7: Comparative Analysis of Performance Enhancement Across Varied Model Sizes.

**Model Series** To compare the performance of different models under two scenarios and demonstrate ONSEP’s improvement over ICL, we use 7B models from the InternLM2, Qwen, LLaMA2, and Baichuan2 series. Figure 5 compares the performance of each model under ONSEP and ICL conditions. The comparison reveals that InternLM2-7b outperforms others with ICL and shows the most significant improvement with ONSEP, leading among models of similar size. While Qwen, LLaMA2, and Baichuan2 have similar performances with ICL, LLaMA2 and Qwen exhibit greater improvements with ONSEP than Baichuan2.

For the 20B models, we examine GPT-NeoX and InternLM2, as indicated in Table 8.

The performance differences may stem from variations in vocabulary, tokenization, training data, decoding strategies, and BPE encoding specifics (Sennrich et al., 2016) among the different model series. InternLM2, with its innovative pre-training and optimization, excels in long-context tasks by capturing long-term dependencies. InternLM2’s superior performance may be due to a progressive training approach, starting with 4k tokens and extending to 32k tokens. ONSEP consistently enhances the performance across these models, showcasing the methodology’s generalizability. Additional metrics and detailed data are available in Appendix D.1.

Table 9 shows the ICL methods for four different series of models of comparable size and the proposed ONSEP method. It is observed that InternLM2 performs the best in terms of various indicators and improvements. For Hit@1 and Hit@3,

Model	Hit@1	Hit@3	Hit@10
<b>GPT-NeoX-20b-ICL</b>	0.314	0.446	0.560
<b>GPT-NeoX-20b-ONSEP</b>	0.320	0.454	0.563
$\Delta$ Improve	1.91%	1.79%	0.54%
<b>InternLM2-20b-ICL</b>	0.326	0.455	0.57
<b>InternLM2-20b-ONSEP</b>	0.334	0.467	0.571
$\Delta$ Improve	2.45%	2.64%	0.18%

Table 8: Comparison of performance across two 20B parameter model series, highlighting the percentage improvement ONSEP achieves over ICL.

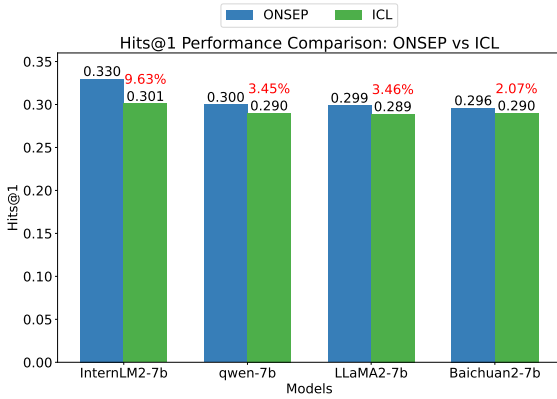


Figure 5: Performance comparison across various model series under ONSEP and ICL methods with the percentage improvement indicated in red above the green bars.

the performances of the other three models are similar, ranked in the order of Hit@1 as Qwen, Baichuan2, and Llama2. In terms of the improvement in Hit@10, the method achieved a 13.41% increase on Llama2. The pre-training data distribution and tokenization methods of these different open-source models vary, which impacts the performance.

Model	Hit@1	Hit@3	Hit@10
<b>InternLM2-7b-ICL</b>	0.301	0.432	0.560
<b>InternLM2-7b-ONSEP</b>	0.330	0.464	0.570
$\Delta$ Improve	9.63%	7.41%	1.79%
<b>Qwen-7b-ICL</b>	0.290	0.421	0.530
<b>Qwen-7b-ONSEP</b>	0.300	0.440	0.560
$\Delta$ Improve	3.4%	4.51%	5.66%
<b>LLama2-7b-ICL</b>	0.289	0.412	0.440
<b>LLama2-7b-ONSEP</b>	0.299	0.438	0.499
$\Delta$ Improve	3.5%	6.31%	13.41%
<b>Baichuan2-7b-ICL</b>	0.290	0.416	0.530
<b>Baichuan2-7b-ONSEP</b>	0.296	0.437	0.552
$\Delta$ Improve	2.06%	5.05%	4.15%

Table 9: Performance Comparison and Improvement Across Models.

## D.2 Ensemble Weight in DHAG

Table 10 demonstrates the effect of adjusting the ensemble weight  $\lambda$  within the Dual History

Augmented Generation (DHAG) of the ONSEP model on performance metrics Hit@1, Hit@3, and Hit@10. The data reveals that the optimal result for Hit@1 is achieved at  $\lambda = 0.1$ , while the highest coverage rate for Hit@10 is observed at  $\lambda = 0.5$ , indicating improvements in both metrics. Extreme values of  $\lambda$  (degenerating to the ICL method) lead to suboptimal outcomes, highlighting the dual context’s ability to enhance both accuracy and coverage, as well as the importance of balancing short-term and long-term historical contexts.

As  $\lambda$  increases, accuracy decreases, while the answer coverage remains constant, suggesting that DHAG enhances model performance by slightly increasing the probability of selecting the target (Hit@1).

Ensemble Weight $\lambda$	Hit@1	Hit@3	Hit@10
0	0.301	0.432	0.560
0.1	0.330	0.464	0.570
0.2	0.328	0.462	0.571
0.25	0.327	0.462	0.572
0.5	0.321	0.460	0.573
0.75	0.314	0.456	0.572
1	0.312	0.445	0.552

Table 10: Results for different choices of ensemble weight of DHAG. The LLM based in ONSEP is InternLM2-7B with a history length of 100.

## E Hyperparameter Sensitivity Analysis

This section looks into how different hyperparameters affect our method. We use the ICEWS14 dataset and the InternLM2-7B model for this analysis, but we find similar patterns in other datasets too.

The set of tested hyperparameter ranges and best parameter values for ONSEP are displayed in Table 11. The best hyperparameter values are chosen based on the Hit@1.

Hyperparameter	Set	Best
Ensemble Weight $\lambda$	{0, 0.1, 0.2, 0.25, 0.5, 0.75, 1}	0.1
History Length $L$	{10, 30, 50, 100, 150, 200}	200
Select rules num $k$	{0, 1, 3, 5, 10, 20}	20
Causality Ratio $\alpha$	{0, 0.1, 0.2, 0.25, 0.5, 0.75, 1}	0.1
Smooth Factor $\theta$	{0, 0.25, 0.5, 0.75, 1}	0.25
Growth Factor $\beta$	{0, 0.1, 0.15, 0.2, 0.25, 0.3, 0.5, 0.75, 1}	0.2

Table 11: Overview of hyperparameters.

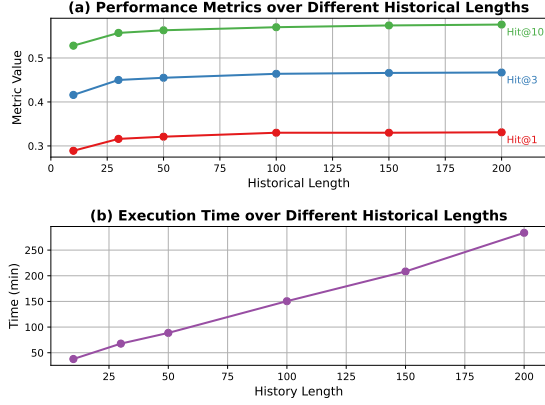


Figure 6: How History Length Affects Performance and Prediction Time.

### E.1 History Length

We assess the impact of varying historical lengths on performance metrics (Hit@N) and the model’s inference time, as shown in Figure 6. Increasing historical length generally leads to better performance but also longer inference times. The most significant improvements occur up to a history length of 100; after that, the benefits level off, indicating an optimal balance between performance gains and computational efficiency is necessary.

Table 12 demonstrates the impact of different history lengths  $L$  on model performance. As the history length increases, all performance metrics (Hit@1, Hit@3, and Hit@10) improve, but the growth saturates with longer  $L$ . If computational resources are limited or fast inference is required,  $L$  should be appropriately reduced to balance performance and efficiency.

History Length $L$	Hit@1	Hit@3	Hit@10
10	0.289	0.416	0.528
30	0.316	0.450	0.557
50	0.321	0.455	0.563
100	0.330	0.464	0.570
150	0.330	0.466	0.574
200	0.331	0.467	0.576

Table 12: Results for different choices of history length  $L$ .

### E.2 Maximum Number of Rules Selected

In the DCRM module, we experiment with changing the upper limit,  $K$ , of optimal rules selected after causality evaluation by the LLM. This parameter shows how selective we are in filtering causality semantics and influences the causal rule set size.

As  $K$  increases, Hit@N performance improves, up to a certain point. Setting  $K$  to infinity (meaning no selection) causes a slight drop in performance from the optimum, suggesting that a larger rule set helps, but keeping high causality confidence is essential to minimize noise and avoid rule application disruption.

The results of selecting the maximum number of rules  $k$ , as shown in 13, indicate that the model’s performance increases with the number of rules selected, reaching a saturation point. The last row suggests that not selecting rules after LLM evaluation results in slightly lower performance, highlighting the importance of optimal rule selection for achieving peak model effectiveness.

Select rules num $k$	Hit@1	Hit@3	Hit@10
0	0.318	0.454	0.561
1	0.327	0.461	0.569
3	0.328	0.462	0.570
5	0.329	0.462	0.570
10	0.330	0.464	0.570
20	0.329	0.462	0.570

Table 13: Results for different choices of Select rules num  $k$ .

### E.3 Fusion Ratio of Causal Rule Confidence Scores

We experiment with various fusion ratios  $\alpha$ , mixing large model predicted probabilities  $p$  and coverage  $cove$  from contextual frequencies to see their effect on performance. A balanced  $\alpha$  setting achieves optimal performance, highlighting the importance of both predicted probabilities and contextual coverage in determining causality confidence.

Regarding the fusion ratio of causal rule confidence scores  $\alpha$ , the table illustrates how adjusting  $\alpha$  affects model performance. The data in Table 14 show that the model performs best at Hit@1 with  $\alpha = 0.5$  and reaches peak performance at Hit@3 with  $\alpha = 0.75$ . This indicates that an appropriate  $\alpha$  value can balance the model’s prediction accuracy and coverage range. Too high or too low  $\alpha$  values may degrade performance on certain metrics.

### E.4 Causal Rule Confidence Update Function

In DCRM’s dynamic update process, adjusting causal rule confidence through factors like smoothing and growth is crucial for adapting to new data and trends. A balanced smoothing factor  $\theta$  setting works well to incorporate both historical and new

Causality Assessment Ratio $\alpha$	Hit@1	Hit@3	Hit@10
0	0.325	0.460	0.568
0.25	0.327	0.461	0.569
0.5	0.329	0.462	0.570
0.75	0.327	0.464	0.570
1	0.323	0.461	0.566

Table 14: Results for varying Causality Assessment Ratio  $\alpha$ .

data, with larger values improving performance on the ICEWS dataset, suggesting stability is needed in dynamic data distributions. On the other hand, a careful setting of the growth factor  $\beta$  improves performance. It shows the importance of slowly increasing focus on rules that have worked in the past without making their confidence scores too high too quickly.

**smooth factor** The results for the smooth factor  $\theta$  in the causal rule confidence update function (15) suggest that a moderate  $\theta$  value balances historical and new information effectively. With a smooth factor of 0, the confidence scores are updated solely based on new samples, while a factor of 1 retains the initial confidence assessments. A slightly higher  $\theta$  may be preferable for stable rule adaptation in dynamic data scenarios, whereas a lower value could be suitable if there are significant intervals between samples.

Smooth Factor $\theta$	Hit@1	Hit@3	Hit@10
0	0.325	0.462	0.570
0.25	0.330	0.464	0.570
0.5	0.327	0.463	0.570
0.75	0.327	0.462	0.571
1	0.324	0.461	0.566

Table 15: Results for different smooth factor  $\theta$  in the confidence score function.

**growth factor** Table 16 shows that appropriately setting the growth factor  $\beta$  can optimize model performance without being excessively high. A cautious, lower growth factor can incrementally increase attention to historically effective rules, enhancing their confidence scores. Conversely, too high a  $\beta$  may introduce more noise, adversely affecting performance.

Growth Factor $\beta$	Hit@1	Hit@3	Hit@10
0	0.328	0.465	0.571
0.1	0.328	0.464	0.570
0.15	0.328	0.464	0.571
0.2	0.330	0.464	0.570
0.25	0.325	0.461	0.568
0.3	0.325	0.461	0.568
0.5	0.324	0.458	0.565
0.75	0.320	0.457	0.567
1	0.323	0.457	0.566

Table 16: Results for different growth factor  $\beta$  in the confidence score function.

## F Inference Time

Our tests on an RTX 3090 showed that ONSEP’s inference time (see Table 17) is about twice that of ICL due to multiple LLM calls, but it’s still suitable for scenarios like the ICEWS dataset where immediate real-time responses aren’t crucial.

Model	ICEWS14	ICEWS05-15	ICEWS18
InternLM2-7B-ICL (L=100)	4.16 it/s	3.14 it/s	3.37 it/s
InternLM2-7B-ONSEP (L=100)	1.94 it/s	1.47 it/s	1.66 it/s

Table 17: Inference times comparison for different methods on ICEWS datasets.