electronics

MDPI

Article

Document-Level Future Event Prediction Integrating Event Knowledge Graph and LLM Temporal Reasoning

Shaonian Huang [1], Huanran Wang [1], Peilin Li [1] and Zhixin Chen [2,*]

[1] School of Computer, Hunan University of Technology and Business, Changsha 410205, China; snhuang@hutb.edu.cn (S.H.); 230720854034@stu.hutb.edu.cn (H.W.); 230720854024@stu.hutb.edu.cn (P.L.)
[2] Information Technology Center, Hunan Normal University, Changsha 410081, China
* Correspondence: zhixin1007@hunnu.edu.cn

Abstract

Predicting future events is crucial for temporal reasoning, providing valuable insights for decision-making across diverse domains. However, the intricate global interactions and temporal–causal relationships at the document level event present significant challenges. This study introduces a novel document-level future event prediction method that integrates an event knowledge graph and a large language model (LLM) reasoning framework based on metacognitive theory. Initially, an event knowledge graph is constructed by extracting event chains from the original document-level event texts. An LLM-based approach is then used to generate diverse and rational positive and negative training samples. Subsequently, a future event reasoning framework based on metacognitive theory is introduced. This framework enhances the model's reasoning capabilities through a cyclic process of task understanding, reasoning strategy planning, strategy execution, and strategy reflection. Experimental results demonstrate that the proposed approach outperforms baseline models. Notably, the incorporation of the event knowledge graph significantly enhances the performance of different reasoning methods, while the proposed reasoning framework achieves superior performance in document-level future event prediction tasks. Furthermore, the interpretability analysis of the prediction results validates the effectiveness of the proposed method. This study advances research on document-level future event prediction, highlighting the critical role of event knowledge graphs and large language models in temporal reasoning. It offers a more sophisticated future event prediction framework for government management departments, facilitating the enhancement of government safety management strategies.

Keywords: future event prediction; event knowledge graph; LLM temporal reasoning; metacognitive theory

Academic Editor: Stefano Ferilli

Received: 27 August 2025
Revised: 19 September 2025
Accepted: 22 September 2025
Published: 26 September 2025

Citation: Huang, S.; Wang, H.; Li, P.; Chen, Z. Document-Level Future Event Prediction Integrating Event Knowledge Graph and LLM Temporal Reasoning. *Electronics* **2025**, *14*, 3827. https://doi.org/10.3390/electronics14193827

1. Introduction

Temporal reasoning plays a crucial role in natural language processing by facilitating the prediction of future trends and supporting informed decision-making through the comprehension of event sequences [1–4]. Its application spans various domains including legal analysis, financial forecasting, medical decision-making, and crisis management [5–8]. One of the most challenging applications of temporal reasoning is the future event prediction, which requires not only an understanding of current event characteristics but also the inference of plausible future scenarios from intricate temporal and causal

relationships. In any given context, the goal of future event prediction is to forecast likely subsequent events based on historical events and their causal interconnections.

Previous studies have primarily focused on script-level event prediction, aiming to forecast subsequent events from individual sentences or short text segments [9,10]. Although these methods effectively capture local event coherence, they often fail to model the macro-level narrative structure in lengthy documents, where events can extend across multiple paragraphs and entail intricate temporal and causal connections [11]. Unlike script-level prediction, document-level future event prediction necessitates a profound understanding of long-range dependencies and global event interactions within documents. This entails a thorough comprehension of temporal and causal relationships between events and the capacity to incorporate contextual details from various paragraphs or complete documents.

Large language models (LLMs) such as GPT-4, Llama 2, and GLM-4 have shown remarkable capabilities in temporal reasoning tasks [12,13], encompassing temporal question answering, event prediction, and knowledge inference. This success stems from their extensive training on large datasets, facilitating the capture of intricate text patterns. Particularly noteworthy is the effectiveness of LLMs enhanced with temporal knowledge graphs in comprehending structured temporal relationships. GenTKG [14] introduces an innovative framework for temporal knowledge graph prediction by merging temporal logic rule-based retrieval with efficient instruction tuning. Similarly, LLM-DA [15] utilizes LLMs to extract temporal rules, thereby improving the accuracy of temporal knowledge graph reasoning through adaptive rule adjustments. These investigations highlight the substantial potential of integrating LLMs with structured knowledge representations for advanced temporal reasoning.

Despite recent progress, two primary challenges remain in document-level future event prediction. Firstly, current methods mainly focus on script-level event prediction, involving temporal reasoning by analyzing event attributes and causal connections within sentences. However, the effectiveness of LLMs in document-level future event prediction remains uncertain. This task necessitates the model's comprehension of dispersed events across long texts and its ability to engage in multi-hop reasoning. Secondly, while LLM has exhibited promising capabilities in temporal knowledge graph reasoning tasks, the effectiveness of event knowledge graphs in document-level future event prediction require further investigation. Specifically, does a structured event knowledge graph enhance prediction accuracy and offer interpretability evidence? This question is critical. Given these gaps, we propose the following research questions to guide our study.

Q1: Can LLMs effectively predict future events by capturing the intricate evolution of document-level events and how does its performance compare to traditional methods?

Q2: Can event knowledge graphs enhance the interpretability of future event prediction and effectively reduce event redundancy while preserving essential event features?

To address these challenges, this study proposes a document-level future event prediction method integrating event knowledge graphs with LLM temporal reasoning. Specifically, TimeEchain, an instruction fine-tuning dataset for the emergency domain, is first constructed, which is used to validate the performance of LLM on the task of document-level future event prediction.

The TimeEchain dataset encompasses a diverse range of event types, with an average document length of 2415 words. It contains 2058 event chains derived from both original event texts and those generated with LLM guidance. Event features such as trigger words, times, entities, and relationships are extracted to construct an event knowledge graph, with the aim of enhancing document-level future event prediction accuracy. Furthermore, a Prompt reasoning framework based on metacognitive theory (MPF) is proposed. MPF

framework employs a structured reasoning process comprising a four-stage cycle: task understanding, planning strategy, execution, and reflection. This cyclical approach is implemented to facilitate interpretable document-level event prediction. Additionally, this study uses the LLaMA-Factory framework combined with LoRA technology to fine-tune two small language models, EChainQwen-7B and EChainQwen-14B. Experimental results show that, compared with other LLMs, EChainQwen has significant advantages in document-level future event prediction and interpretability.

The main contributions of this paper are as follows:

The instruction fine-tuning dataset TimeEchain was constructed for evaluating the ability of LLMs to predict future events at document-level. The event knowledge graph of the dataset is extracted and enhances the performance of improving future event prediction

The MPF Framework for document-level future event prediction was proposed. The MPF framework introduced a structured reasoning process with a four-stage cycle to realize interpretable future event prediction, verifying the effectiveness of LLMs in document-level event reasoning.

An experimental comparison of future event prediction was carried out between the proposed MPF framework and mainstream LLM reasoning framework. The experimental results show that the MPF framework exhibits better prediction performance while having lower token requirements and higher reasoning efficiency.

2. Related Work

This study focuses on document-level prediction of future events and the impact of large language models on temporal reasoning. In Section 2 we review the literature on prediction of future events and LLM temporal reasoning.

2.1. Future Event Prediction

There are two primary methods for future event prediction methods: scripted event prediction and temporal knowledge graph event prediction.

Script-level event prediction aims to predict future events by leveraging historical background information within the event script. Previous research has focused on utilizing event representation and script modeling to predict subsequent events [16,17]. Chambers and Jurafsky [18] first defined the scripted event prediction task and represent events as a tuple. Subsequently, Balasub ramanian et al. [19] proposed the event representation as a triple, while Pichotta and Mooney [20] proposed quadruples to represent an event. Although the quadruples-based representation has been widely adopted in subsequent studies, it may suffer from semantic information sparsity. Consequently, researchers have explored alternative event embedding representation techniques based on neural networks. For instance, Granroth-Wilding and Clark [21] developed a neural network model capable of learning embedded event representations and coherence functions to facilitate event prediction. Similarly, Pichotta and Mooney [22] proposed a recurrent neural network based on Long Short-Term Memory (LSTM) to capture event sequences and enable script event prediction. Furthermore, Various LSTM models have since been proposed for application in scripted event prediction tasks [23–25].

In addition, Bai et al. [26] observed that the contextual relationships between event arguments can provide a more comprehensive event semantic representation. Therefore, they developed the Transformer-based model MCPredictor, which integrates event-level and script-level information for script-level event prediction. Zhou et al. [27] proposed a hierarchical network model for script event prediction, which uses stacked Transformers and attention mechanisms to model local and global features. Recently, many researchers have been dedicated to constructing event chain/graph patterns to model the complex

interactions between script events. Lv et al. [28] proposed an event chain attention model to model the relationships between events. Wang et al. [29] further proposed a multi-granularity learning framework at the argument level, event level, and chain level to improve event representation learning. Zheng et al. [30] proposed a heterogeneous event graph network to model the heterogeneous relationships between events and adopted a contrastive learning framework to enhance the robustness of graph training. Du et al. [31] used the Bert model to automatically construct event graphs and predicted the relationships between events through structured variables. Islam et al. [32] proposed a dynamic graph contrastive learning method for event prediction, which uses a local view encoder to learn the evolving node features and effectively captures the local dynamic structure of the input graph. Rong et al. [33] proposed the Pred-ID model, which realizes event prediction by constructing an event graph pattern based on core events.

Temporal knowledge graph prediction aims to model entities, relationships, and their interaction patterns based on historical data and structural information, thereby inferring possible future events. Existing research methods can be broadly classified into three categories: embedding learning-based methods, graph neural network-based methods, and multi-source information fusion-based methods. Embedding learning-based methods map entities, relationships, and timestamps in the temporal knowledge graph to a low-dimensional vector space and capture the structural and semantic information of the temporal knowledge graph based on the embedded vector representation. Typical methods include TTransE [34], RotatE [35], and RE-GCN [36]. Nguyen et al. [37] proposed the BiCoTime model, which uses bicomplex embeddings to represent entities, relationships, and time. Yang et al. [38] proposed the TIE model, which models the tripartite interaction among entities, relationships, and time through cross-convolutional layers and tensor neural networks, significantly improving the prediction accuracy.

Methods based on graph neural networks combine GNNs with deep neural networks and their variants to model the graph structure information and temporal dependencies in TKGs. Typical methods include RE-NET [39], REGCN [40], TiRGN [41], etc., refs. [42,43]. Huai et al. [44] proposed the STKGN model to construct a cross-regional spatio-temporal event graph for effectively predicting multiple concurrent events. Tang et al. [45] proposed the DHyper model, which models the high-order correlations between entity hypergraphs and relation hypergraphs by introducing a dual hypergraph neural network. Methods based on multi-source information fusion enhance the representation ability of temporal knowledge graphs through multi-information fusion. Jia et al. [46] proposed the SFTe model, which captures the evolutionary relationships in temporal knowledge graphs through structural embedding, fact embedding, and temporal embedding to achieve interaction prediction tasks in the social Internet of Things.

2.2. Temporal Reasoning in LLM

LLMs have shown significant promise in temporal reasoning tasks, particularly in event prediction, temporal knowledge graph reasoning, and complex temporal logic modeling. Current research can be categorized into temporal reasoning using in-context learning, temporal knowledge graph reasoning with knowledge enhancement, and prediction via multi-modal temporal fusion.

Context learning-based methods enhance the temporal reasoning capabilities of LLMs through the development of efficient Prompt templates or context learning strategies [47,48]. Xiao et al. [49] demonstrated the zero-shot capability of LLMs in predicting financial events, showing that even without fine-tuning, these models can achieve accuracy comparable to specialized models. Shi et al. [50] proposed the LAMP framework, leveraging the few-shot learning ability of large language models to transform event sequence prediction into a causal

reasoning task, thereby improving prediction performance. Wei et al. [51] presented an analogical reasoning approach that optimizes context learning by analyzing event evolution patterns to boost the performance of script event prediction. Wang et al. [12] proposed a novel method that combines large language models and generative agents to enhance temporal prediction through the integration of textual and time series data reasoning. Nako and Jatowt [52] evaluated the performance of various large language models in tasks related to future prediction.

In temporal knowledge graph reasoning, researchers aim to enhance the reasoning process of TKGs by utilizing LLMs for knowledge extraction and reasoning. Wang et al. [15] proposed the LLM-DA framework, which uses LLMs to extract temporal logical rules from historical data and adapts to the evolution of TKGs through a dynamic update mechanism. Xia et al. [53] proposed the chain of history (CoH) reasoning, which utilizes high-order historical information to enhance the prediction performance of TKGs. Xiong et al. [54] proposed the TG-LLM framework, which translates text into the temporal graph space and incorporates chain-of-thought (CoT) reasoning for cross-task transfer learning. Jiang et al. [55] presented the GETER framework and introduced a structure–text prefix adapter to map graph structural features into the text embedding space, enabling LLMs to produce more interpretable reasoning outcomes.

Multimodal temporal prediction research focuses on integrating data from different modalities to improve prediction accuracy. Wang et al. [12] introduced the use of LLM agents to facilitate collaborative analysis of news events and time series, thereby improving prediction accuracy by leveraging text and time-series data. The TimeCAP designed by Lee et al. [56] adopts a dual-agent architecture. One LLM agent generates text summaries that capture the context of the time series, and the other uses this rich summary to make more informed predictions. Ye et al. [57] constructed the MIRAI benchmark to evaluate the event prediction ability of LLM agents. This benchmark includes a massive database of historical, structured events and text news articles.

2.3. Metacognitive Theory and Its Application in LLMs

Metacognitive theory, a concept from cognitive psychology, describes the ability to "think about thinking"—that is, to monitor, evaluate, and regulate one's own cognitive processes. This typically involves a cycle of planning, execution, checking, and reflection to improve performance and correct errors. As large language models (LLMs) tackle more complex reasoning tasks, they exhibit challenges analogous to human cognitive biases, such as logical fallacies, factual inconsistencies, and "hallucinations." Consequently, researchers are drawing upon metacognitive theory to enhance the self-correction and deep reasoning capabilities of these models.

The application of these principles to LLMs is primarily achieved through sophisticated Prompt Engineering or structured reasoning frameworks. For instance, by introducing "self-reflection" or "self-critique" steps, a model is Prompted to evaluate and revise its initial output, thereby simulating a human-like metacognitive loop. This approach has proven effective in reducing errors that arise from direct answer generation and significantly enhances the accuracy and logical coherence of the final response. These self-correction mechanisms have been successfully applied in diverse fields, including code generation [58], mathematical problem-solving [59], and common sense reasoning [60], yielding remarkable results.

3. Methodology

This study aims to evaluate and enhance the ability of LLM to predict future events within lengthy and intricate textual contexts. To this end, Section 3 defines the future event

prediction task for complex temporal reasoning, and constructs a future event prediction dataset based on real-world emergency cases. Then, a future event prediction framework is introduced that integrates event knowledge graphs and metacognitive theory. Additionally, a series of small language models are also fine-tuned to achieve the effect of a high-parameter model. Figure 1 illustrates the outlined methodology of the study.

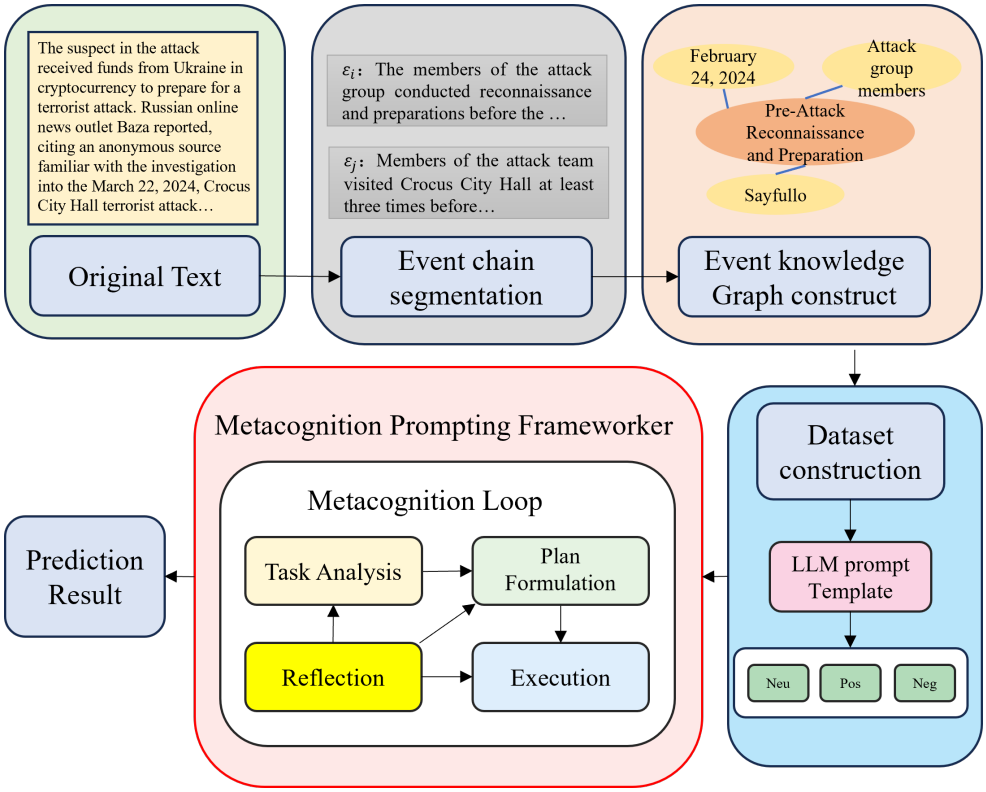

Figure 1. Overall flow of the proposed method.

3.1. Problem Definition

An interpretable future event prediction is defined as follows: Given an input document d containing a sequence of event chains, the ith event chain is denoted as $\varepsilon_i = (e_i^1, e_i^2, \ldots, e_i^{m-1})$. For an event chain pair $(\varepsilon_i, \varepsilon_j)$ in d, future event prediction is to estimate the likelihood p_i of event ε_i^m occurrence based on the information from the event chain pair $(\varepsilon_i, \varepsilon_j)$. The MPF framework based on LLMs is employed for prediction inference, with a focus on providing explainable reasoning. The probabilities p_i are categorized as *positive*, *neutral*, and *negative*. The training examples for fine-tuning the LLM model in the experiment comprises quadruples: the input question q_i, the event chain pair $(\varepsilon_i, \varepsilon_j)$, the predicted answer p_i, and the inference process r_i. A training instance can be represented as $x_i = \{q_i, (\varepsilon_i, \varepsilon_j), p_i, r_i\}$, where the inference process r_i is utilized to refine the LLM model to facilitate the generation of inference steps and explanations.

3.2. TimeEchain Dataset

In this paper, a document-level future event prediction dataset TimeEchain is constructed for the emergency event domain. TimeEchain exhibits a variety of data types, long-range dependencies, inter-paragraph inference, and structured annotations, presenting unique challenges for document-level event prediction. Table 1 highlights the differences between the existing datasets MCNC and DocSEP.

Table 1. Comparison of TimeEchain with MCNC and DocSEP dataset.

Dataset	MCNC	DocSEP-Contracts	DocSEP-MavenWiki	TimeEchain
Doc level	Sentence-level	Multiple Pages	Multiple Paragraphs	Multiple Paragraphs
Temporal Logic	×	√	√	√
Causal Logic	×	√	√	√
Cohesion	×	×	×	√
Source	News	Legal Contracts	Maven-ERE	Wikipedia-events
Domain	General News	Legal and Contract	General	Emergency
Language	English	English	English	Chinese

3.2.1. Data Collection and Preprocessing

The TimeEchain dataset contains textual data from 623 emergency events sourced from Wikipedia. Initially, the event text underwent preprocessing steps, including the removal of URL links, reference marks, common stop words, redundant spaces, and newlines. This preprocessing was conducted to standardize the text and minimize interference from irrelevant elements, thereby producing a refined dataset suitable for analysis. In addition, we have translated the texts from the English data source into Chinese to make it easier for the team to perform annotation and filtering. Nonetheless, both the paper and its presentation will remain in English. The TimeEchain corpus comprises 6 event categories: 255 accidents and disasters, 235 violent conflicts, 56 environmental pollutions, 48 safety and health, 16 political, and 13 other events. The corpus exhibits an average document length of 2415 words and an average event chain length of 437. Details of the dataset are shown in Table 2.

Table 2. Data statistics comparing TimeEchain with MCNC, DocSEP-Contracts and MavenWiki.

Dataset	MCNC	DocSEP-Contracts	MavenWiki	TimeEchain
Documents	1.03 M	98	2310	623
Avg Doc Length	31	2480	250	2415
Events	522 K	27 K	49 K	35 K
Event Chains	160 K	128 K	5.64 M	2058
Event Times	–	1628	12,046	16,462
Avg Event Chain Length	31	2479.23	232.79	437
Time Relationship	–	20 K	316 K	11 K
Causal Relationship	–	1874	28 K	8 K

3.2.2. Event Knowledge Graph Construction

An event knowledge graph differs from a traditional knowledge graph, which typically represents facts about entities and their interrelations. Instead, event knowledge graphs focus on dynamic representations, positioning events as central nodes and highlighting temporal and causal links between them. Unlike the static facts in traditional knowledge graphs, event knowledge graphs capture the evolution of events, enabling coherent narratives from unstructured text. This structure is crucial for event reasoning, as it converts unstructured text into a coherent narrative that reflects the progression of events. By explicitly representing temporal and causal dependencies, an event knowledge graph enhances a models' capacity to comprehend the progression of complex events.

To facilitate event prediction by large language models when dealing with complex and lengthy texts, the incremental Prompting technique is employed. This technique enables LLMs to concentrate on pivotal event nodes, thereby refining event descriptions and minimizing token usage. Initially, original event documents are segmented into multiple

event chains, guided by temporal and causal relationships among events. Each event chain is centered around specific time points and causal links, maintaining the temporal and causal coherence within the chain. This segmentation process yields a considerable quantity of event chains. The template for dividing event chains is delineated below.

> *Please reorganize the content of the following event knowledge graph text into an event chain with event timelines and causal relationships. Ensure compliance with the following rules: 1. Divide the event sets based on event stages, time spans, and text length. 2. Do not add content that does not exist in the original text and preserve all details. 3. Be careful not to include summarizing statements.*

The construction of an event knowledge graph involves the identification and detailed analysis of each sub-event ε_i^m within the input event chain ε_i^1. Each sub-event contains multiple key features, such as participants P, location L, time T , trigger word C, etc. By extracting these features, the core elements of each sub-event are identified, and a clear connection is established for the temporal order and causal relationship between sub-events. This ensures the event knowledge graph ε_i^G accurately reflects the original event text. The algorithm framework for constructing the event knowledge graph is detailed in Algorithm 1.

Algorithm 1 The algorithm framework for constructing the event knowledge graph.

Input: Raw event document d
Output: Event knowledge graph G = (V, E)
1. Data preprocessing to obtain a cleaned version T = remove (d, url, citations, stop words. . .)
2. Parse cleaned text T to identify and segment event chains $\varepsilon_i = (\varepsilon_i^1, \varepsilon_i^2, \ldots, \varepsilon_i^{m-1})$, where each event chain
   is decomposed into a series of constituent sub-events e.
3. for each event chain ε_i, do
   $G_{\varepsilon_i} = []$
4. for each sub-event ε_i^1, do
   $G_{\varepsilon_i^1} = []$
   Extract event knowledge from the text, including: Subject, Object, Time, and Trigger Word, forming
   event nodes V.
   $G_{\varepsilon_i^1}$.add(V and corresponding knowledge edges E)
   Determine relationships between sub-events and create relational edges.
5. G_{ε_i}.add($G_{\varepsilon_i^1}$, relationships between sub-events)

The construction of the event knowledge graph enables the extraction of key events, event features, and the relationships between them from the original text. Based on this, a concise event description rich in structural information is generated. Figure 2 illustrates the process and results of event knowledge graph construction.

3.2.3. Data Sample Generation

To construct training samples, event documents are divided into multiple event chains based on the temporal and causal development relationships between events. Each event chain revolves around distinct time points and causal connections, preserving the integrity of the temporal and causal structure in the event chain. This process generates a substantial number of positive sample $(\varepsilon_i, \varepsilon_j, p_{pos})$, which serve as training data for future event reasoning and prediction. The template for splitting event chains is outlined below.

In this study, negative sample pairs are generated by replacing events in positive sample pairs with events of different types. The aim is to break the original causal relationship, such that the connection between the event pairs no longer exists. Specifically, events

from different temporal and contextual contexts are selected from event chain instance repository to replace the original events. Subsequently, LLMs are utilized to confirm and assess the lack of relevance between the newly paired events. For example, given a positive sample event chain $(\varepsilon_i^1, \varepsilon_j^1, p_{pos})$, where event ε_i^1 is a natural disaster event and event ε_j^1 is a violent conflict event, negative samples are generated through techniques such as time replacement and background replacement. This process ensures that the resulting event pairs cannot form a reasonable causal chain. It is specifically expressed as

$$(\varepsilon_i^1, \varepsilon_j^1, p_{pos}) \rightarrow (\varepsilon_i^1, \varepsilon_j^n, p_{neg}) \tag{1}$$

A large number of negative sample instances labeled as p_{neg} can be obtained through simple rule replacement of existing positive sample instances. To ensure the irrelevance of the contextual relationships in the generated negative samples, this study employs a large-scale model to design the following Prompt, which is used to test and evaluate the null hypothesis that the event sequences ε_i and ε_j are statistically independent.

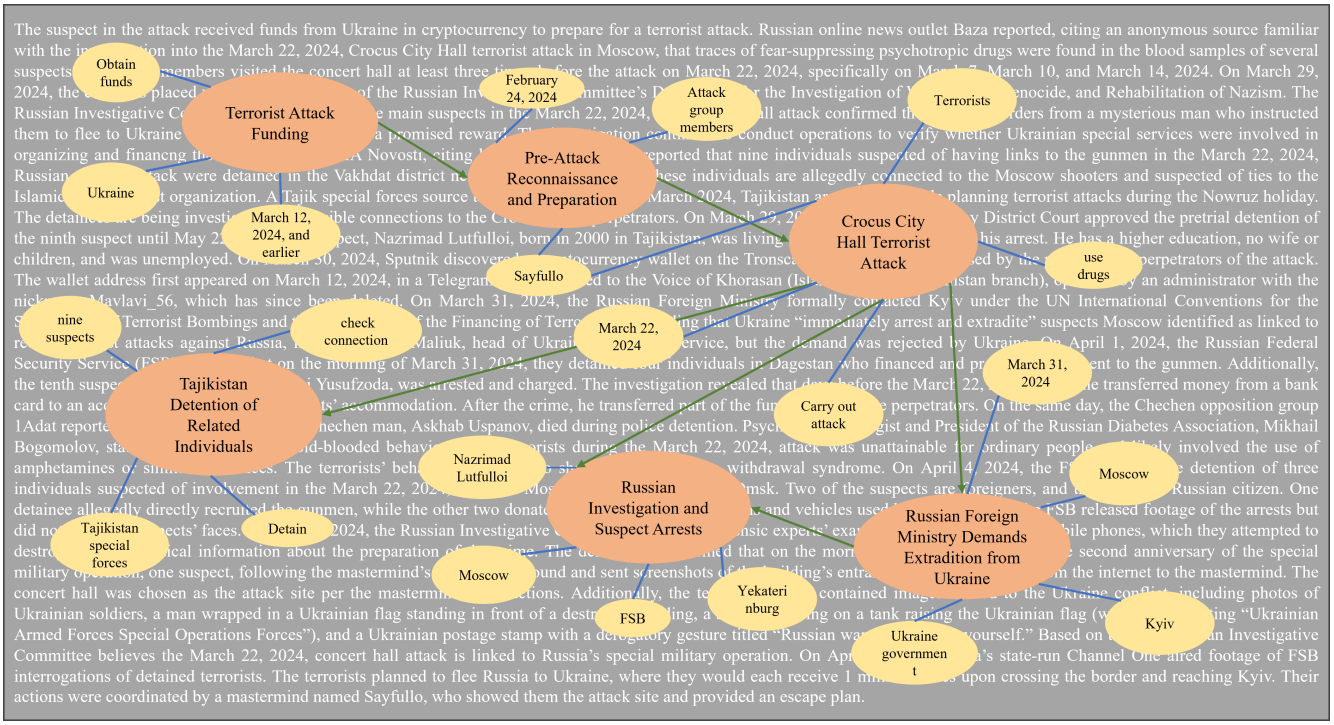

Figure 2. The main event (orange), the event features (yellow), and their corresponding relationships (the relationship between events and features is a blue line, and the relationship between events is connected by green arrows) are extracted from the document-level event text (white text). The black font is a brief description of the document.

> *Please determine whether the pair* $(\varepsilon_i, \varepsilon_j, p_{pos})$ *is a pair of event chains that have no relationship.*

For neutral samples $(\varepsilon_i^1, \varepsilon_j^1, p_{\text{neu}})$, the different events appearing in event chain pairs are interference factors generated by normal development. Therefore, this study modifies certain event features and event relationships in the event chain ε_i^1, such as time, location, participants, etc. Neutral samples required for the experiment are generated by disrupting the certainty of the temporal or causal relationships in the event chain. Different from directly replacing other events in negative sample generation, the generation of neutral samples relies on the powerful context understanding ability of LLM to ensure that the generated event chain pairs conform to natural language logic. The Prompt template for generating neutral samples is as follows.

Based on the input positive sample event chain pair $(\varepsilon_i, \varepsilon_j)$, please generate a neutral sample by randomly replacing key event features in ε_j to alter the original event development trend, thereby creating a neutral sample such that the model cannot clearly determine whether there is a relationship between ε_i and ε_j.

3.2.4. Data Validation and Quality Assessment

The TimeEchain dataset was rigorously validated to ensure data accuracy. A Prompt ensemble refinement method was employed to verify the quality of the dataset and remove ambiguous or incorrect classifications. Data evaluation employed several language models with moderate reasoning capabilities, such as Grok2. Samples that exhibited robust consistency across multiple assessments were retained. For sample data whose quality could not be directly judged, more advanced language models with enhanced reasoning abilities, such as O1, Grok3, and Gemini 2.5 Pro, were employed for further verification. The assessment framework depicted in Figure 3 illustrates a data validation framework based on a large language model.

In this framework, event chain pairs $(\varepsilon_i, \varepsilon_j)$ are input into a structured process to determine whether the event chain pairs meet the corresponding labels according to the Prompts. Inference models 1 to k independently process the data and validate the labels. The final judgment is based on the consistency of the results of these models, and the final answer is determined based on the principle of stronger consistency. Based on the above data validation principle, 2058 event chains were extracted, generating 358 positive samples, 400 negative samples, and 271 neutral samples.

The dataset's quality was further evaluated by employing two seasoned annotators to independently assess a sample of the data. The annotators scored the labels of each sample based on the following three criteria:

Accuracy. Evaluate whether the event chain pairs $(\varepsilon_i, \varepsilon_j)$ and their labels correctly reflect the data content and event relationships.

Completeness. Assess whether the event chain pairs $(\varepsilon_i, \varepsilon_j)$ provide the necessary background for future event prediction, enabling a well-founded judgment of the labels.

Reasoning difficulty. Evaluate the degree of uncertainty in the label judgment of the event chain pairs $(\varepsilon_i, \varepsilon_j)$. It is used to measure the possibility that the model may misjudge the target label as other labels while correctly identifying it.

The architecture of our data validation framework, which leverages multiple LLMs to ensure consistency, is depicted in Figure 3.

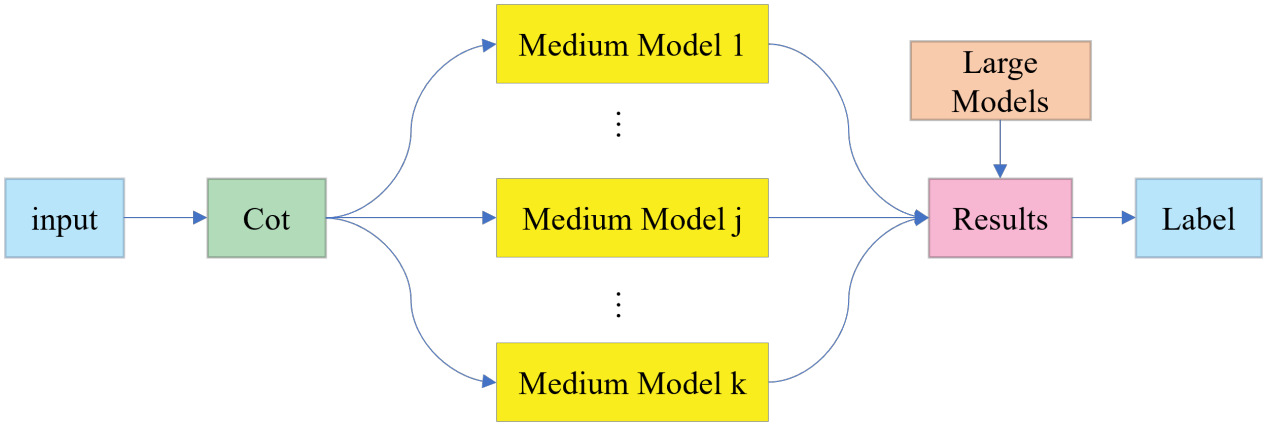

Figure 3. Data consistency validation framework based on multiple LLMs.

The statistical distribution of the evaluation results is shown in Figure 4. As can be seen from Figure 4, positive samples consistently garnered high scores across all criteria,

indicating substantial agreement among evaluators. Negative samples, while also receiving relatively high scores, exhibited narrower score dispersion compared to positive samples. In contrast, neutral samples displayed more varied scores and lower evaluator consistency. The analysis of Cohen's Kappa coefficient in Table 3 corroborates these findings.

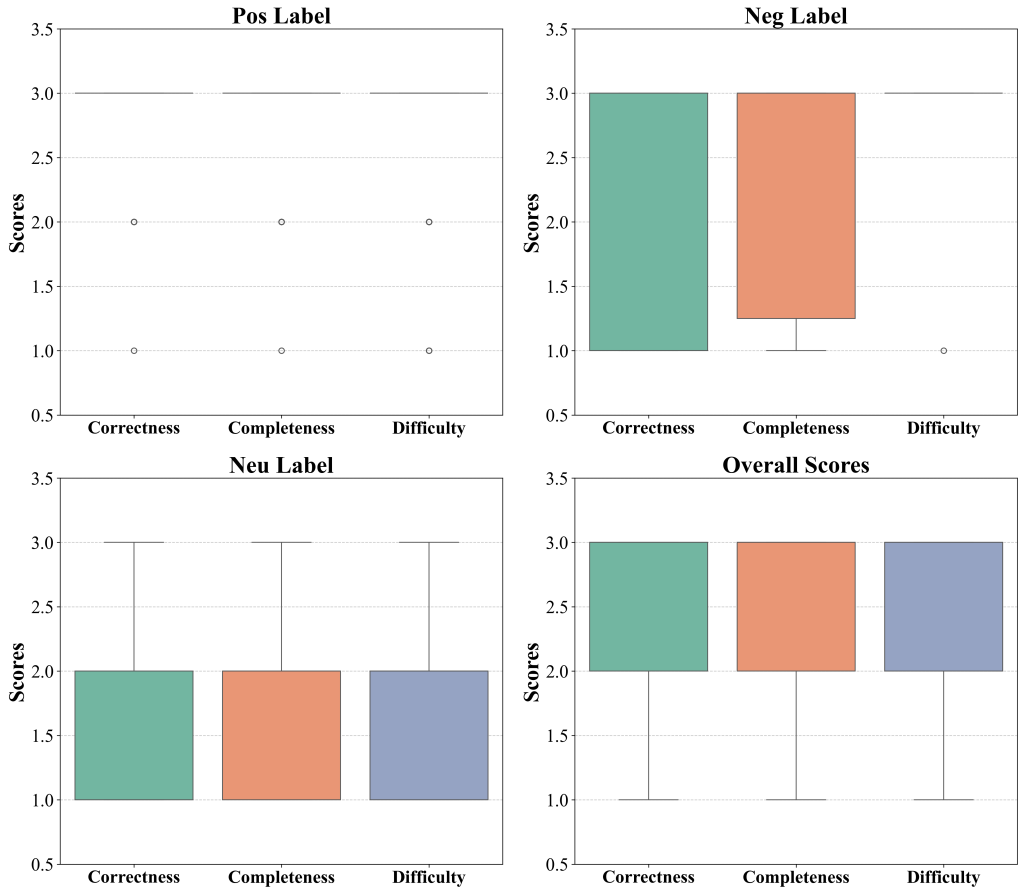

Figure 4. Results of manual evaluation on the quality of the TimeEchain dataset: Box plots of correctness, completeness, and inference difficulty scores classified by label type.

Table 3. Cohen's Kappa score of human annotation.

Metric	Positive	Negative	Neutral	Overall
Correctness	0.70	0.62	0.45	0.60
Completeness	0.68	0.60	0.50	0.59
Reasoning Difficulty	0.75	0.65	0.55	0.60

Table 3 reveals high Kappa values for positive and negative samples, contrasting with notably lower values for neutral samples, indicating substantial discrepancies among evaluators in assessing neutral judgments. The comprehensive analysis demonstrates that a majority of samples scored highly across the three criteria. Particularly noteworthy is the substantial consensus among evaluators for positive and negative samples. Samples with low scores on the three criteria were excluded to uphold the dataset's quality and standardization.

3.3. Reasoning Framework Based on Metacognitive Theory

The MPF integrates metacognitive theory into LLM reasoning tasks to enhance their predictive capabilities. Originating from Flavell's work, metacognitive theory delineates cognition into four key components: knowledge cognition, goals and tasks, metacogni-

tive experiences, and strategies and actions. This study incorporates this metacognitive model into the reasoning tasks of LLMs. The proposed framework employs a four-stage cyclical reasoning process to improve the LLM's capacity for document-level future event prediction. This process encompasses task understanding, strategy planning, execution, and reflection. By identifying sub-events and leveraging relevant knowledge domains, this structured approach aims to enhance the predictive capability. The reasoning framework is depicted in Figure 5.

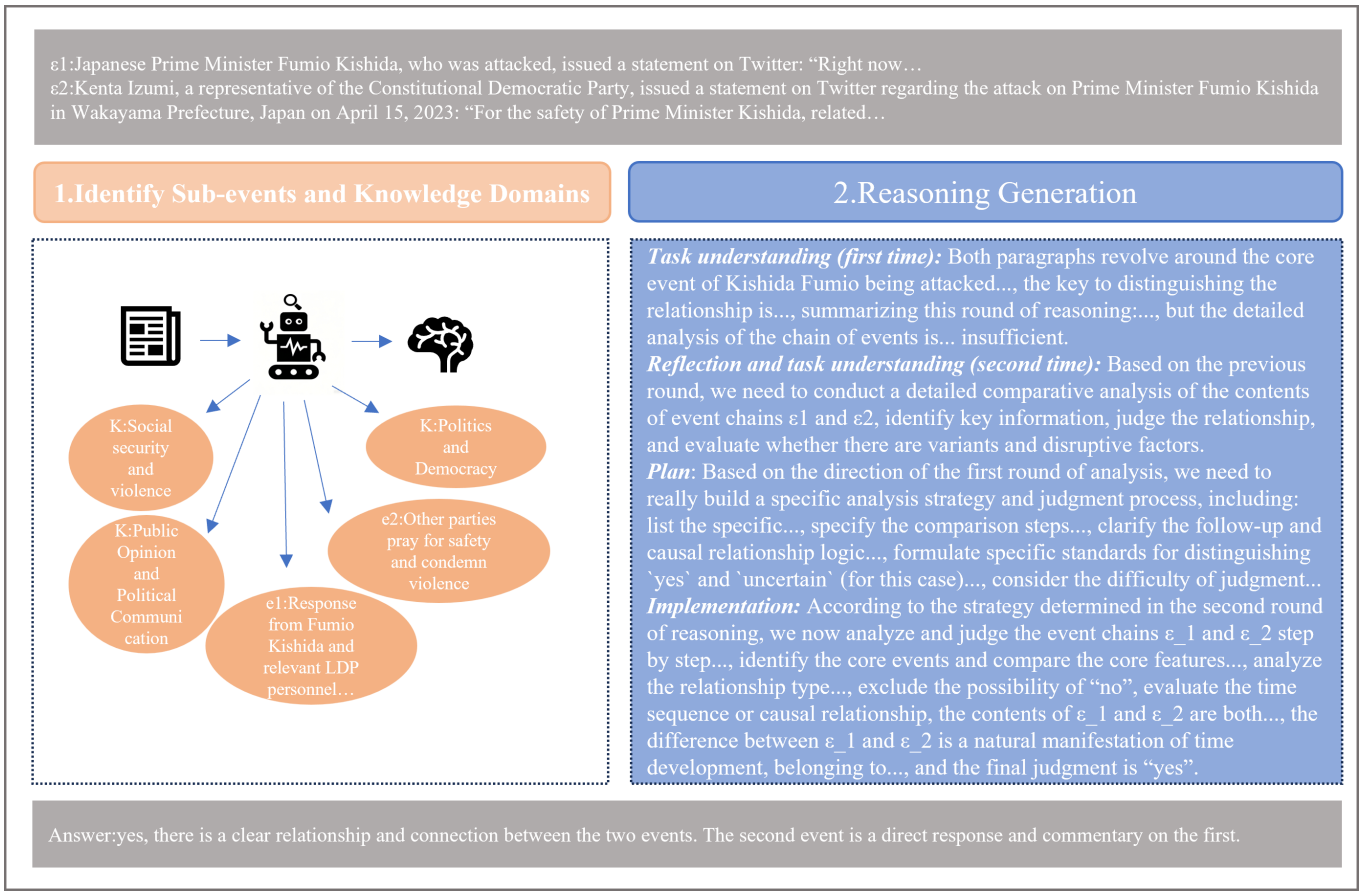

Figure 5. A metacognitive Prompt framework for future event prediction. This framework integrates the metacognitive model into the reasoning process of large language models, with the aim of enhancing their capacity for predicting future events. The framework comprises four key stages: input processing, sub-event and knowledge domain identification (orange box), reasoning generation (blue box), and result output. The example at the bottom illustrates the actual output during the reasoning generation stage, showcasing iterative processes like "Task Understanding," "Reflection," and "Implementation" to assess event relationships and facilitate reflection and adjustment.

3.3.1. Data Preprocessing

In the task analysis phase, we utilize the extensive prior knowledge of large language models to analyze the sub-events within the overall event chain and identify the relevant knowledge domains. The task analysis Prompt is structured as follows:

Data definition:

- pos: Event chain pairs $(\varepsilon_i, \varepsilon_j)$ *originate from the same real-world event and have a clear temporal sequence or causal relationship.*

- neg: There is no association between event chain pairs $(\varepsilon_i, \varepsilon_j)$*. Their themes or backgrounds are completely different, and there is no overlap in core features.*

- neu: The event chain i is a variant of the subsequent events of the event chain j. Some features are similar, but the other half are significantly different. There is a correlation between the features,

but there are obvious other disturbing factors. Pay attention to distinguishing the disturbing factors from the characteristic factors that may arise during event development.

Event chain: $\varepsilon_i = (\varepsilon_i^1, \varepsilon_i^2, \ldots \varepsilon_i^{m-1})$

Event chain: $\varepsilon_i = (\varepsilon_j^1, \varepsilon_j^2, \ldots \varepsilon_j^{m-1})$

Task: Determine the relationship between event chain i and event chain j, and preliminarily identify the occurrence of sub-event e in event chain and event chain, as well as the event knowledge domain.

3.3.2. Inference Generation

This stage aims to guide LLMs in deeply understanding input data and tasks, planning suitable reasoning strategies, executing selected strategies, and reflecting the results of strategy execution.

Through step-by-step analytical reasoning, the temporal relationship of the given event chains pair $(\varepsilon_i, \varepsilon_j)$ within the same context can be obtained. To better achieve this goal, this paper seeks to foster the model's active cognitive capacity, enabling it to analyze problems and weigh options prudently like human experts, rather than passively accepting instructions or directly giving answers. As previously noted, merely instructing the model to "think, judge, and determine the temporal relationship of ε_i and ε_j" is insufficient. This assertion is empirically validated by our experiments , which frame the comparison between our MPF framework and the CoT baseline as an ablation-style analysis. Specifically, the CoT baseline represents a powerful linear reasoning approach that lacks the final metacognitive stage. The significant performance gap between CoT and our complete MPF framework highlights the critical contribution of self-reflection. Therefore, this study encourages the model to reflect on its own reasoning results by adding an additional "rethink" Prompt.

The MPF framework first analyzes the input event chain data to accurately interpret its structure and meaning, which includes timestamps, descriptive text, and contextual event details. The Prompt template is outlined as follows:

> *Please distinguish the key points of data types based on the input event chain data, the definition of the data, and the task understanding.*

Based on a comprehensive understanding of the task and data, the model must select appropriate reasoning strategies and formulate detailed execution plans. Determining the temporal relationships in event chains may rely on various methods, such as direct timestamp comparison, causal logic inference, or contextual event sequence analysis. This stage guides the model to select appropriate strategies and analyze key points according to the data characteristics, thereby providing a clear roadmap for subsequent reasoning. The Prompt template is as follows:

> *Select an appropriate strategy and plan how to determine the relationship between event chains* ε_i *and* ε_j*.*

Subsequently the model should systematically analyze the data in accordance with the predetermined plan, engaging in logical reasoning and making informed judgments to derive the ultimate outcome.

> *According to the plan, gradually conduct analysis to judge the relationship between event chain pairs.*

To ensure the rigor of reasoning, the model should review its analysis process to check for omissions, incorrect assumptions, or logical loopholes. Upon identifying deficiencies, the model should engage in iterative reflection to refine its reasoning. To avoid infinite loops, a predefined limit is imposed on the number of iterations. During this phase, Prompt words are designed to stimulate the model's reflective capabilities and enhance judgment accuracy. The Prompt template is as follows:

Please determine whether there are any deficiencies in the previous round of analysis and whether further reasoning is required on this basis.

3.3.3. Model Fine-Tuning Based on TimeEchain

In fine-tuning the TimeEChain model, Grok 3 is employed for inference generation on event chain data. For the fine-tuning, we partitioned the complete TimeEchain dataset into training, validation, and test sets based on a strict 8:1:1 ratio. We then used high-quality, inference-generated data to fine-tune the Qwen2.5-7B and Qwen2.5-14B models, creating the EChainQwen-7B and EChainQwen-14B variants. This fine-tuning process utilizes the LLaMA - Factory framework in conjunction with LoRA (Low-Rank Adaptation) technology, enabling the models to adapt to the event inference task based on event chain pairs.

In the experiment, the hyper-parameters were set as follows: the learning rate was 0.00001, the number of training epochs was 5, the batch size was 8, the LoRA Rank was 8, the LoRA Alpha was 32, the LoRA Dropout was 0.05, the maximum number of tokens was set to 32,768, and the model was trained utilizing the NVIDIA H800 PCIe (NVIDIA Corporation, Santa Clara, CA, USA).

Specifically, the inference output generated by Grok3 contains logical relationships between events, causal inferences, and time-series information. These data are utilized as pseudo-labels and integrated with the original dataset to enhance the training process. During fine-tuning, the model parameters are efficiently adjusted through LoRA to optimize performance on the future event prediction task. Experimental results indicate that Grok3's inference output, based on the TimeEChain event chain, markedly enhances the training of smaller models. Specifically, the comprehensive metrics for EChainQwen-14B and EChainQwen-7B in future event prediction tasks improve by approximately 7.58% and 7.64%, respectively.

4. Experiments

4.1. Benchmark Model

In this study, the following models were used for tasks such as data construction, data validation, and inference prediction. These language models employed can be broadly classified into three main categories: small, medium, and large models. Small language models, such as Qwen2.5-7B and Qwen2.5-14B, have fewer parameters relative to the DeepSeek-V3 model. Medium language models, like Grok2, offer moderate reasoning capabilities and are well-suited for a range of data processing and analysis tasks. Large models encompass Grok3, OpenAI's o1 and GPT-4.1, and Google's Gemini 2.5 Pro, considered flagship or cutting-edge models by their developers due to their advanced reasoning capabilities for intricate applications.

Qwen2.5-7B and Qwen2.5-14B [61]: The language models with 7 billion and 14 billion parameters, respectively, developed by the Qwen team at Alibaba Cloud. It supports multiple languages and handles extended contexts up to 128 K tokens. The model excels in executing instructions, generating long texts, processing structured data, and producing structured outputs.

Grok2 [62]: Grok2 is a model with moderate reasoning ability launched by xAI, demonstrating excellent performance in data processing and analysis. It efficiently mines and interprets diverse data types, aiding users in extracting valuable insights from large datasets. Grok3 [63]: Grok3 is positioned as the latest and more powerful model from xAI, excelling in enterprise applications like data extraction, encoding, and text summarization, with comprehensive expertise across various professional domains.

o1 [64]: The latest and strongest model family from OpenAI, o1 is designed to spend more time thinking before responding. The o1 model series is trained with large-scale reinforcement learning to reason using chain of thought.

GPT-4.1 [65]: OpenAI's flagship general artificial intelligence model, hypothetically released on 14 April 2025. It excels at solving complex tasks and is particularly well-suited for cross-domain problem-solving.

Gemini 2.5 Pro [66]: Google's state-of-the-art AI model engineered for sophisticated reasoning, programming, mathematics, and scientific applications. Endowed with robust cognitive capabilities, it delivers more accurate and contextually nuanced responses, enabling advanced reasoning and problem-solving across a wide range of academic and technical domains.

Mistral-small-3.1 [67]: Mistral AI's highly efficient model designed for high-volume, low-latency tasks. It offers a strong balance between performance and cost, making it ideal for applications such as classification, summarization, and function calling that require rapid response times.

claude-3.5-haiku [68]: As part of Anthropic's Claude 3.5 model family, Haiku is the fastest and most compact model, optimized for near-instant responsiveness. It is designed for simple queries, content moderation, and customer service applications where speed is paramount, while benefiting from the advanced reasoning and vision capabilities of the Claude 3.5 generation.

The aforementioned models were chosen to encompass a broad spectrum of performance and parameter sizes, facilitating a thorough assessment of their strengths and weaknesses.

4.2. Benchmark Method

Prompt [69]: The most basic zero-shot Prompting method. Directly issue a task instruction to the model, requesting it to immediately output a final judgment based on its existing knowledge reserves, without guiding it to think through intermediate steps.

> *Determine the relationship between event chain i and event chain j?*

CoT [70]: Chain-of-Thought (CoT) is a Prompting strategy that guides large language models to solve complex reasoning tasks. By adding instructions like "Think step-by-step" to the Prompt, it encourages the model to output a coherent, step-by-step reasoning process before giving the final answer.

> *Please reason step-by-step and determine the relationship between event chain i and event chain j?*

MPF: The Metacognitive Prompting Framework (MPF) is a structured reasoning method proposed in this study based on metacognitive theory. It does not rely on a single Prompt but guides the model to conduct deeper, human-expert-like thinking through a four-stage cyclical process that includes task understanding, strategy planning, execution, and reflection. For details, refer to Section 3.3.

4.3. Evaluation Indicators

This study employed standardized metrics to assess the model's performance in predicting future events, focusing on accuracy, macro-mean accuracy, recall, and F1 score for each category. These metrics were independently calculated for each labeling type—neutral, negative, and positive—to offer a detailed evaluation of the model's efficacy across different categories.

For each category c, the specific definitions of the metrics are as follows:

$$\text{Precision}_c = \frac{\text{True Positive}_c}{\text{True Positive}_c + \text{False Positive}_c} \tag{2}$$

$$\text{Recall}_c = \frac{\text{True Positive}_c}{\text{True Positive}_c + \text{False Negative}_c} \tag{3}$$

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{4}$$

To comprehensively evaluate the overall performance of the model in the classification task, this study further calculated the overall indicator. Among them, Accuracy measures the overall prediction accuracy rate. Its definition is as follows:

$$Accuracy = \frac{Numberof\ Correct\ Predictions}{Total\ Predictions} \tag{5}$$

It should be noted that in this study, the precision, recall, and F1 metrics in the overall indicators are macro-averaged metrics, rather than simply calculated based on the total number of all samples. The category metrics of Neu, Neg, and Pos are calculated separately without weighting or averaging. The three macro-average metrics, namely Macro-Precision, Macro-Recall, and Macro-F1 Score, refer to calculating the metrics for each category separately and then averaging them. This approach mitigates biases from category imbalances.

4.4. Experimental Results

Section 4.4 systematically examines the impact of different reasoning approaches, event knowledge graphs, and model architectures on future event prediction performance. The experimental results demonstrate that the proposed MPF framework outperforms other methods in document-level future event prediction. The incorporation of event knowledge graphs notably enhances the performance of simple Prompts, chain-of-thought reasoning, and MPF framework reasoning. Furthermore, fine-tuning small language models with event chain knowledge can substantially improve their prediction accuracy, enabling them to approach or even surpass the performance of un-tuned large language models in event prediction tasks.

4.4.1. The Effect of MPF Reasoning Framework on the Performance of Future Event Prediction

The study evaluated the performance of various inference methods, including Prompt, CoT, and the proposed MPF, on the document-level future event prediction task using the Qwen 7B dataset. As shown in Table 4, the results demonstrate a clear trend of performance improvement across the different inference approaches. The Prompt method exhibited the lowest overall performance, while the CoT approach showed improved results, and the MPF framework achieved the best performance in terms of precision, recall, and F1 score. Specifically, prediction accuracy increased from 0.6352 with Prompt reasoning to 0.6817 with CoT reasoning, and further to 0.7362 with the MPF framework. This represents a significant improvement in the overall accuracy of the MPF framework for the future event prediction task.

Table 4. Comparison of prediction results based on simple Prompts, CoT, and the MPF framework in a document-level future event prediction task. The F1 score, recall, accuracy, and precision metrics are utilized to assess the performance.

Methods	Overall				Neu			Pos			Neg		
	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Prompt	0.6352	0.6182	0.5834	0.5659	0.3793	0.2075	0.2683	0.6174	**0.9975**	0.7627	0.8578	0.5452	0.6667
CoT	0.6817	0.6585	0.6318	0.6189	0.4276	0.2444	0.3110	0.6562	0.9975	0.7917	**0.8915**	0.6534	0.7541
MPF	**0.7362**	**0.6941**	**0.6967**	**0.6854**	**0.5301**	**0.3333**	**0.4093**	**0.7321**	0.8793	**0.7990**	0.8201	**0.8775**	**0.8478**

The horizontal comparison of classification indicators in Table 4, where the best performance for each metric is highlighted in bold, reveals that all three inference methods exhibit limited predictive capability for neutral event categories. The MPF framework, despite being the most effective, achieves an F1 score of only 0.4093 for neutral events, significantly lower than its scores of 0.7990 for positive samples and 0.8478 for negative samples. This finding suggests that existing methods still have great challenges in distinguishing and accurately predicting neutral events.

4.4.2. The Effect of Event Knowledge Graph on the Performance of Future Event Prediction

This study assesses the impact of integrating an event knowledge graph into three distinct reasoning frameworks for predicting future events. As illustrated in Table 5, the inclusion of the knowledge graph consistently enhances performance across the Prompt, CoT, and MPF framework. Specifically, the overall F1 score for predictions increases from 0.6033 with the Prompt* to 0.6854 with CoT*, and further to 0.7161 with the MPF framework. Tables 4 and 5 visually confirm these improvements, demonstrating a marked enhancement in prediction accuracy subsequent to the knowledge graph's integration. Notably, the overall F1 score for the MPF framework rises from 0.6854 to 0.7161 following the incorporation of the knowledge graph.

Table 5. The effect of event knowledge graph on the performance of future event prediction. The superscript * denotes that the data has been transformed into an event knowledge graph format.

Methods	Overall				Neu			Pos			Neg		
	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Prompt *	0.6709	0.6491	0.6200	0.6033	0.4505	0.2041	0.2809	0.6527	**0.9972**	0.7890	0.8442	0.6588	0.7401
CoT *	0.7362	0.6941	0.6967	0.6854	0.5301	0.3333	0.4093	0.7321	0.8793	**0.7990**	0.8201	0.8775	0.8478
MPF *	**0.7520**	**0.7173**	**0.7187**	**0.7161**	**0.5434**	**0.4508**	**0.4928**	**0.7370**	0.7730	0.7546	**0.8715**	**0.9325**	**0.9010**

An in-depth analysis of the results in Table 5 reveals a key advantage of the MPF* framework that transcends the overall metrics: its ability to effectively mitigate the "positive prediction bias" inherent in the baseline models. Although baselines like Prompt* and CoT* achieve high F1 scores on the "Pos" category, this is largely attributable to their tendency to default to positive predictions for ambiguous or uncertain samples. This is strongly evidenced by the Prompt* model's near-perfect recall of 0.9972 on positive examples. While this bias inflates the metric for a single category, it severely compromises the models' ability to accurately identify neutral and negative samples, rendering them unreliable in a holistic evaluation. In contrast, the MPF* framework demonstrates a significantly more balanced and robust predictive capability. Instead of maximizing performance on a single category, it enhances recognition accuracy across all classes, particularly the more challenging ones. Specifically, the MPF* framework not only boosts the F1 score for the "Neu" category to 0.4928—a relative increase of over 20% compared to CoT*—but also achieves a class-leading F1 score of 0.9010 on the "Neg" category. This balanced performance across categories indicates that the judgments of MPF* are based on effective evidence-based reasoning

rather than an inherent predictive bias. Consequently, while its recall on positive samples is slightly lower than that of the biased baselines, the substantial performance gains in other categories underscore its more reliable and generalizable reasoning process.

4.4.3. The Effect of Model Fine-Tuning Based on TimeEchain

This study also evaluated the effect of model fine-tuning on future event prediction performance. Two models, EChainQwen 7B and EChainQwen 14B, were fine-tuned using the event prediction knowledge from the Grok3 model, building upon the smaller language models Qwen7B and Qwen14B, respectively. Table 6 presents a comparative analysis of the models' overall prediction performance before and after the fine-tuning process.

Table 6. Performance improvement of EChainQwen models after fine-tuning with event chain knowledge.

Model	Accuracy	Precision	Recall	F1
Qwen 7B/EChainQwen 7B	0.8752 (+11.62% ↑)	0.7476 (+2.52% ↑)	0.7889 (+6.59% ↑)	0.7618 (+4.23% ↑)
Qwen 14B/EChainQwen 14B	0.9149 (+12.84% ↑)	0.7998 (+4.1% ↑)	0.8121 (+7.3% ↑)	0.8057 (+8.42% ↑)

The fine-tuned EChainQwen model demonstrates substantial performance enhancements, as evident from the data in Table 6, where the symbol ↑ denotes a performance improvement over the baseline model. The macro-averaged F1 of the EChainQwen-7B model reaches 0.7618, representing a 4.23% improvement over the Qwen-2.5-7B model. Similarly, the Accuracy score of the EChainQwen-14B model improved by 8.42%. Notably, this performance boost is not limited to the Accuracy metric but is also reflected in the simultaneous enhancement of event prediction across the neutral, positive, and negative categories.

4.4.4. Comprehensive Performance Comparison Among Various Language Models

Section 4.4.4 evaluates the performance of various language models in the task of predicting future events. Table 7 presents macro-average F1 scores and F1 scores for neutral, positive, and negative categories across various language models in the task of predicting future events. The results reveal notable performance discrepancies among the models, with larger-scale models and those with higher pre-training levels generally demonstrating superior overall performance. For example, the Qwen-7B model, featuring a relatively modest parameter scale and no fine-tuning, achieves an overall F1 score of approximately 0.7161, whereas the Qwen-14B model, with double the parameters, enhances the overall F1 score to around 0.73. Google's Gemini-2.5 model attains an overall F1 score of nearly 0.79, while the xAI Grok-3 model exhibits the most remarkable performance, with an overall F1 score of approximately 0.85. In contrast, the GPT-4.1 model achieves a slightly lower F1 score of about 0.74.

Table 7. Performance comparison among various large language models.

Models	Overall				Neu			Pos			Neg		
	Accuracy	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Qwen-2.5-7b	0.7520	0.7173	0.7187	0.7161	0.5434	0.4508	0.4928	0.737	0.773	0.7546	0.8715	0.9325	0.9010
Qwen-2.5-14b	0.7648	0.7368	0.7329	0.7299	0.6089	0.4539	0.5201	0.7171	0.8073	0.7595	0.8844	0.9375	0.9102
mistral-small-3.1	0.7988	0.7793	0.7661	0.7637	0.7065	0.4797	0.5714	0.7665	0.8436	0.8032	0.8647	0.975	0.9166
Gemini-2.5	0.8189	0.8125	0.7852	0.7834	0.7771	0.476	0.5904	0.7275	0.9048	0.8065	0.9329	**0.9749**	0.9534
deepseek-r1	0.8358	0.82	**0.8173**	0.8176	0.716	**0.679**	0.697	0.8448	0.7905	0.8167	0.8993	0.9825	0.9391
GPT-4.1	0.7931	0.8186	0.753	0.7373	**0.8333**	0.3148	0.457	0.6621	**0.9658**	0.7856	0.9602	0.9784	0.9692
Grok-3	0.8649	**0.852**	0.8432	**0.8453**	0.7848	0.6654	**0.7202**	0.8109	0.8917	0.8494	0.9605	0.9725	0.9665
deepseek-r1	0.8358	0.82	**0.8173**	0.8176	0.716	**0.679**	0.697	0.8448	0.7905	0.8167	0.8993	0.9825	0.9391
EChainQwen 7B	0.8752	0.7476	0.7889	0.7618	0.3944	0.5833	0.4706	**0.8636**	0.8636	0.8636	0.9847	0.9198	0.9511
EChainQwen 14B	**0.9149**	0.7998	0.8121	0.8057	0.549	0.5833	0.5657	0.8593	0.8788	**0.8689**	**0.9913**	0.9742	**0.9827**

It is also worth noting the performance of the models varies significantly across different event categories. Accurate prediction is most evident for negative events, whereas neutral events present a considerable challenge. Models consistently demonstrate superior F1 scores for negative events, with the Grok-3 model notably achieving an F1 score of 0.9665 in the negative category. This heightened accuracy may stem from the higher frequency of negative events in the dataset or their distinct linguistic attributes, which facilitate clearer differentiation by the model. In contrast, the prediction accuracy for neutral events is notably lower compared to positive and negative events, primarily due to the absence of pronounced emotional or outcome orientations. For instance, the Qwen-7B model exhibits an F1 score of 0.4928 for neutral events, in stark contrast to its performance of 0.901 for negative events. Even the top-performing Grok-3 model displays a modest F1 score of 0.7202 in the neutral category, significantly below its performance in the negative category at 0.9665. Figure 6 provides a visual comparison of these findings.

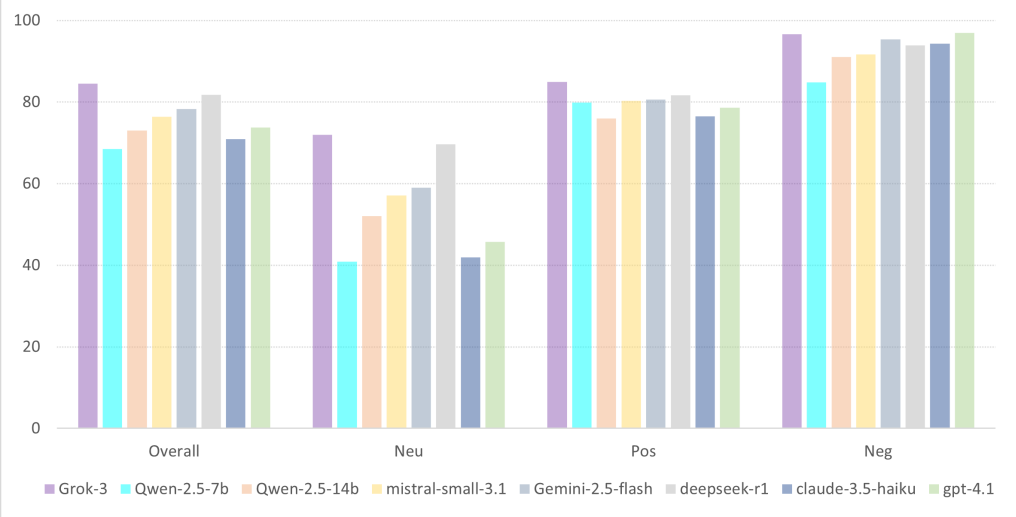

Figure 6. Comparison of F1 scores across different large language models.

Additionaly, the EChainQwen series of small models demonstrate strong competitiveness compared to other large models in the cross-model comparison. EChainQwen-14B ranks among the top in open-source models with a macro-average F1 of 0.8057, significantly outperforming the GPT-4.1 model without specialized training. This suggests that small or medium-sized language models enhanced with event chain knowledge can surpass traditional large-scale pre-trained models in specific reasoning tasks. However, some of the most advanced closed-source or semi-open-source models in the industry still lead in this task. For instance, the Grok-3 model from xAI achieves a macro F1 of 0.8453, and the DeepSeek-R1 model scores 0.8176, slightly surpassing EChainQwen-14B. These top models may benefit from extensive domain knowledge absorption during pre-training or larger parameter scales. Nonetheless, EChainQwen-14B has narrowed the performance gap to single-digit percentage points by incorporating reasoning knowledge from high-parameter models, highlighting the effectiveness of the MPF framework's reasoning fine-tuning. Additionally, EChainQwen-7B achieves a notable macro-average F1 score of 76.18%, surpassing baseline models like general-purpose GPT-4.1, which emphasizes code and mathematics.

4.5. Interpretability Evaluation

To assess the quality and reliability of the explanations produced by the MPF framework, a subjective evaluation system utilizing a 5-point Likert scale was devised. This scale facilitated scoring, with evaluators assigning ratings ranging from 1 to 5 to gauge the explanation quality. Such an approach enables the quantification of subjective assessments

regarding the explanations' quality. The evaluation was conducted from the following three dimensions:

Logical Reasoning (L): Assess the clarity and reasonableness of the reasoning process underlying the generated explanations. A high score indicates that the reasoning process is coherent, rigorous, and in line with common sense and the event context.

Explanation Correctness (E): Evaluate the clarity and rationality of the reasoning process supporting the provided explanations. A high score signifies that the reasoning is cohesive, rigorous, and consistent with both common sense and the context of the event.

Information Relevance (I): This metric assesses the effectiveness of referencing key information in events, such as the presence of "hallucinations" and the integration of actual event content into the reasoning process. A high rating suggests that the explanations are closely aligned with the event data and devoid of inaccuracies.

To enhance result objectivity, a dual evaluation approach was implemented. Specifically, GPT-4.1 conducted automated scoring of model outputs, while several human evaluators individually assessed 50 samples from the test set. The mean score was calculated to minimize personal biases. Moreover, to establish a gold standard for evaluation, we introduced a set of high-quality explanations written by human experts to serve as an ideal benchmark, referred to as "Human-written Benchmark" in Table 8. This benchmark represents the highest achievable score across the evaluation dimensions. The efficacy of the MPF framework was evaluated against several techniques across various dimensions to verify its advantages. The results of the interpretability evaluation are shown in Table 8.

To enhance result objectivity, a dual evaluation approach was implemented. Specifically, GPT-4.1 conducted automated scoring of model outputs, while several human evaluators individually assessed 50 samples from the test set. The mean score was calculated to minimize personal biases. Moreover, high-quality human-written explanations served as a benchmark. The efficacy of the MPF framework was evaluated against multiple techniques across various dimensions to verify its advantages. The results of the interpretability evaluation are shown in Table 8.

Table 8. Evaluation results of the reasoning explanations for 50 randomly sampled samples from the TimeEchain dataset using a 5-point Likert scale. The evaluators include two human annotators and GPT-4.1. The scores given by the evaluators were averaged.

Methods	**GPT-4.1**			**Human**		
	L	**E**	**I**	**L**	**E**	**I**
Prompt	3.84	4.10	3.52	2.87	4.16	3.10
CoT	4.27	4.37	4.83	3.03	3.40	4.24
MPF	4.85	4.67	4.83	4.66	4.70	4.93
human-written	5.00	5.00	5.00	4.90	5.00	4.97

Table 8 demonstrates that the explanatory capabilities of the MPF framework significantly outperform those of the baseline methods across all dimensions. Specifically, the MPF framework achieves scores of 4.85, 4.67, and 4.83 in dimensions L, E, and I, respectively, closely approaching the high-quality manual explanations score of 5. In contrast, the Prompt and CoT methods exhibit notably lower scores, particularly in dimensions L and I. Remarkably, MPF attains scores of 4.66, 4.7, and 4.93 in these dimensions, closely resembling the Oracle scores of 4.9, 5, and 4.97. Conversely, the performance of the baseline methods in manual evaluation diminishes significantly. For instance, Prompt scores 2.87 in dimension L, while CoT scores 3.4 in dimension E, indicating that automated assessment may overstate the efficacy of the baseline methods. This highlights the critical role of human judgment in evaluating the explanatory quality. In summary, the MPF framework

excels in generating logically coherent, factually precise, and highly pertinent explanations concerning event data, showcasing its resilience and efficacy.

4.6. Case Study

To validate the inference effectiveness and interpretability of the MPF framework, a specific case from the dataset was selected for analysis. This case involves the complex relationship between two event chains. One chain includes both direct subsequent developments and unrelated historical data, presenting a clear challenge for prediction. The analysis focuses on a multi-year timespan encompassing various event types, such as civil disputes, diplomatic developments, and media reactions. The relationship of the event chains in this case was classified as "neutral."

The original text of the case contains 882 tokens. The case describes a dispute between a Chinese tourist, Mr. Zeng, and a Swedish hostel in September 2018, which escalated into a conflict involving the police. The incident prompted a diplomatic response from the Chinese Embassy in Sweden. The case encompasses two interrelated event chains: the initial civil dispute over hotel check-in ε_1, and the subsequent reactions ε_2 from various parties , including media coverage and diplomatic representations. The case spans multiple years, from 2015 to 2018, adding complexity to the event reasoning. Overall, the case illustrates the evolution of a localized incident into a diplomatic issue between China and Sweden.

The primary challenge in this case lies in the fact that event chain ε_2 is not a straightforward continuation of event chain ε_1. It includes misleading historical information, resulting in the relationship between the two event chains being classified as "neutral." This intertwining of past and present events places significant demands on the model's ability to discern associations and filter out interfering information, thereby highlighting the complexity and challenges inherent in real-world event prediction. By evaluating the relationship between the event chain pair $(\varepsilon_1, \varepsilon_2)$, the predictive and interpretative effectiveness of various methods is demonstrated.

Section 4.6 evaluates the prediction results and reasoning mechanisms of three methods: Prompt, CoT, and MPF. It further examines the theoretical foundations and logical underpinnings of the reasoning outputs generated by each method, and discusses potential limitations of the respective reasoning frameworks. Figure 7 provides a comparative overview of the reasoning processes and outcomes across the three methods.

Figure 7 reveals that the inaccuracies in the Prompt and CoT methods stem from their inability to process complex mixed information. The Prompt method does not provide any reasoning process and only gives a "positive" prediction result, completely ignoring the interfering event in 2015 mentioned in ε_2. While the CoT method outlines a reasoning process and identifies the causal link between subsequent media and diplomatic reports in ε_1 and ε_2, it fails to recognize the 2015 event as independent and irrelevant. Instead, it erroneously considers the entirety of ε_2 as a direct subsequent development of ε_1, leading to a flawed "positive" conclusion, as illustrated by the red text in Figure 7.

The success of the MPF method hinges on its self-reflection and cognitive strategies. Utilizing task understanding and analysis, MPF employs an "elimination method combined with feature comparison and interference factor identification", as shown in the green text in Figure 7. This approach effectively isolates the irrelevant "2015 event" from ε_2. By examining and identifying the structure of this "information confounding," MPF accurately discerns that the relationship is neither purely "positive" nor "negative," leading to the correct conclusion of "neutral."

Inuput: $\varepsilon_1, \varepsilon_2$
Output Prediction: Neu

Method: Prompt **Prediction:** Pos

Reasoning: Only the pos result is given, no reasoning.

Method: Cot **Prediction:** Pos

Reasoning:
This analysis focuses on the core information of ε1, which details the course of the 2015 incident involving the Zeng family, and ε2, which covers both the 2015 incident and its follow-up reports in 2018. The Global Times report within ε2 aligns with the time, location, and description of events in ε1. Notably, specific wording like "a family of three was thrown into a cemetery" and "the Ministry of Foreign Affairs lodged solemn representations" directly responds to ε1. Consequently, the subsequent actions taken by the embassy and the Ministry of Foreign Affairs were evidently a part of the diplomatic handling triggered by the event described in ε1, establishing a clear causal link.
Interpretation: The analysis explicitly states that ε2 is a subsequent evolution of ε1, an observation particularly evident in the realms of media dissemination and official responses, which together form a temporal continuation and a causal chain. This conclusion is supported by a logical framework; for instance, it posits that the media coverage and diplomatic negotiations in ε2 are direct consequences of ε1. The final determination of "yes" is made by ruling out the alternatives of "no" and "uncertain."

Method: MPF **Prediction:** Neu

Reasoning:
Extracting features: In ε1, Chinese tourists "were brutally treated by Swedish officials," while the Swedish prosecution authority believed that "leaving people at the Woodland Cemetery station is a standard procedure for the Swedish police." In ε2, "the Embassy of the People's Republic of China in Sweden informed the Swedish Ministry for Foreign Affairs of the situation and lodged solemn representations, which mentioned the incident where Chinese tourists had been brutally treated by Swedish officials," and prosecutor Eriksson stated that the relevant investigation had been concluded, with the assessment showing that the Swedish police had committed no wrongdoing in this incident.
Interpretation: Adopting the method of Process of Elimination Combined with Feature Comparison and Identification of Confounding Factors, the strategy is executed by analyzing the content of ε2 in detail. Although the latter part of ε2 mentions subsequent media reports and diplomatic negotiations concerning the 2015 Zeng family incident, showing a thematic similarity to the developments post-ε1, it also identifies other independent events as confounding factors, leading to a preliminary confirmation of "uncertain." Through multiple rounds of analysis to ensure no key information is missed, it is clearly identified that ε2 contains two information blocks: one comprises the media and diplomatic reactions related to ε1, and the other is an unrelated old incident from 2015. It is pointed out that such a "mixed information" structure cannot constitute a pure causal or temporal sequence relationship, for instance, by explicitly identifying the interfering effect of the 2015 incident. It is clarified that while an association exists between ε1 and ε2, there are also clear confounding factors; thus, "no" and "yes" are systematically eliminated, and the features of "uncertain" are verified item by item

Figure 7. Comparison of reasoning processes and results of Prompt, CoT, and MPF.

5. Conclusions

This study proposes a document-level future event prediction method that integrates event knowledge graph feature representation and LLM temporal reasoning. Initially, a dataset named TimeEchain is developed for document-level future event prediction, employing an LLM-based strategy to generate extensive event chains from the original text. To better understand the intricate progression of document-level events, an event knowledge graph is constructed using LLMs to extract key event features, including event type, time, location, trigger words, and causal relationships. Furthermore, a metacognitive theory-based LLM reasoning framework is proposed to guide LLMs through a reasoning process that includes knowledge acquisition, task comprehension, strategy planning, strategy execution, and strategy reflection. The framework leverages the event knowledge graph to enhance the predictive capabilities of LLMs for future events.

Extensive experimentation and analysis were conducted utilizing a range of evaluation metrics, including Accuracy, macro-average precision, macro-average recall, and macro-average F1 score. The results illustrate a significant improvement in the performance of basic Prompts, chain-of-thought reasoning, and the MPF framework reasoning through the integration of the event knowledge graph. A key insight from this investigation pertains to the efficacy of employing various LLMs for predicting future events. The experimental results indicate that the proposed MPF framework remarkably enhances the ability of LLMs in future event prediction. This approach holds promise for assisting policymakers in the development of real-time event prediction systems, such as public opinion monitoring

and early disaster warnings, as well as in exploring paradigms for collaborative decision-making between humans and machines.

For future work, the proposed model's generalizability can be further enhanced by validation on a broader range of datasets. We acknowledge that the current study has certain limitations. Specifically, our data sources are restricted to text and were not extended to multimodal data. Furthermore, in our experiments, the construction of the event graph heavily relied on LLMs, and we lacked a quantitative analysis of potential error propagation, with no manual verification or filtering.

In the future, to address these issues, we can specifically improve the model's reasoning in scenarios with complex causal relationships and long sequences by integrating dynamic temporal modeling with multimodal features from images, videos, audio, and structured data. This will also allow us to add error propagation blocking and manual verification using smaller models for the modeling of event schemas. Additionally, incorporating insights from cognitive science and causal inference theory would enhance the model's interpretability, fairness, and robustness. This presents a promising direction for future research, poised to increase the utility of LLMs for intricate temporal reasoning tasks.

Author Contributions: Conceptualization, S.H.; methodology, S.H. and H.W.; formal analysis, H.W.; investigation, P.L.; resources, H.W.; data curation, P.L.; writing original draft preparation, S.H. and H.W.; writing review and editing, P.L. and Z.C.; visualization, H.W.; supervision, Z.C.; project administration, Z.C.; funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the The National Social Science Fund of China (Grant No. 21BTJ026).

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yuan, C.; Xie, Q.; Huang, J.; Ananiadou, S. Back to the future: Towards explainable temporal reasoning with large language models. In Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 1963–1974. [CrossRef]
2. Xu, X.H.; Du, Z.J.; Chen, X.H.; Cai, C.G. Confidence consensus-based model for large-scale group decision making: A novel approach to managing non-cooperative behaviors. *Inf. Sci.* **2019**, *477*, 410–427. [CrossRef]
3. Liu, D.; Liu, Y.; Chen, X. The new similarity measure and distance measure between hesitant fuzzy linguistic term sets and their application in multi-criteria decision making. *J. Intell. Fuzzy Syst.* **2019**, *37*, 995–1006. [CrossRef]
4. Chen, Z.S.; Yang, Y.; Wang, X.J.; Chin, K.S.; Tsui, K.L. Fostering linguistic decision-making under uncertainty: a proportional interval type-2 hesitant fuzzy TOPSIS approach based on Hamacher aggregation operators and andness optimization models. *Inf. Sci.* **2019**, *500*, 229–258. [CrossRef]
5. Liang, W.; Chen, X.; Huang, S.; Xiong, G.; Yan, K.; Zhou, X. Federal learning edge network based sentiment analysis combating global COVID-19. *Comput. Commun.* **2023**, *204*, 33–42. [CrossRef] [PubMed]
6. Yang, S. A novel study on deep learning framework to predict and analyze the financial time series information. *Future Gener. Comput. Syst.* **2021**, *125*, 812–819. [CrossRef]
7. Zhou, X.; Li, Y.; Liang, W. CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. *Ieee/Acm Trans. Comput. Biol. Bioinform.* **2020**, *18*, 912–921. [CrossRef]
8. Shi, D.; Zheng, H. A mortality risk assessment approach on ICU patients clinical medication events using deep learning. *Comput. Model. Eng. Sci.* **2021**, *128*, 161–181. [CrossRef]
9. Lim, B.; Zohren, S. Time-series forecasting with deep learning: a survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [CrossRef]
10. Li, Z.; Zhang, M.; Ma, Y.; Wu, S. Script Event Prediction Based on Large Model Reflection Mechanism. In Proceedings of the 2024 2nd International Conference on Computer, Vision and Intelligent Technology (ICCVIT), Huaibei, China, 24–27 November 2024; IEEE: New York, NY, USA, 2024; pp. 1–6. [CrossRef]
11. Jiang, W.; Lv, S.; Wang, Y.; Chen, J.; Liu, X.; Sun, Y. Computational experimental study on social organization behavior prediction problems. *IEEE Trans. Comput. Soc. Syst.* **2020**, *8*, 148–160. [CrossRef]

12. Wang, X.; Feng, M.; Qiu, J.; Gu, J.; Zhao, J. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 58118–58153. [CrossRef]
13. Shui, R.; Cao, Y.; Wang, X.; Chua, T.S. A comprehensive evaluation of large language models on legal judgment prediction. *arXiv* **2023**, arXiv:2310.11761. [CrossRef]
14. Liao, R.; Jia, X.; Li, Y.; Ma, Y.; Tresp, V. GenTKG: Generative Forecasting on Temporal Knowledge Graph with Large Language Models. In Proceedings of the NAACL-HLT (Findings), Mexico City, Mexico, 16–21 June 2024. [CrossRef]
15. Wang, J.; Kai, S.; Luo, L.; Wei, W.; Hu, Y.; Liew, A.W.C.; Pan, S.; Yin, B. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 8384–8410. [CrossRef]
16. Feng, Y.; Qin, Y.; Zhao, S. Correlation-split and Recombination-sort Interaction Networks for air quality forecasting. *Appl. Soft Comput.* **2023**, *145*, 110544. [CrossRef]
17. Shyalika, C.; Wickramarachchi, R.; Sheth, A.P. A comprehensive survey on rare event prediction. *ACM Comput. Surv.* **2024**, *57*, 1–39. [CrossRef]
18. Chambers, N.; Jurafsky, D. Unsupervised learning of narrative event chains. In Proceedings of the ACL-08: HLT, Columbus, OH, USA, 15–20 June 2008; pp. 789–797. [CrossRef]
19. Balasubramanian, N.; Soderland, S.; Etzioni, O. Generating coherent event schemas at scale. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1721–1731. [CrossRef]
20. Pichotta, K.; Mooney, R. Statistical script learning with multi-argument events. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014; pp. 220–229. [CrossRef]
21. Granroth-Wilding, M.; Clark, S. What happens next? Event prediction using a compositional neural network model. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30. [CrossRef]
22. Pichotta, K.; Mooney, R.J. Using sentence-level LSTM language models for script inference. *arXiv* **2016**, arXiv:1604.02993. [CrossRef]
23. Jiang, W.; Ye, F.; Liu, W.; Liu, X.; Liang, G.; Xu, Y.; Tan, L. Research on Prediction Methods of Prevalence Perception under Information Exposure. *Comput. Mater. Contin.* **2020**, *65*, *3* . [CrossRef]
24. Wang, Z.; Chai, Y.; Sun, C.; Rui, X.; Mi, H.; Zhang, X.; Yu, P.S. A weighted symmetric graph embedding approach for link prediction in undirected graphs. *IEEE Trans. Cybern.* **2022**, *54*, 1037–1047. [CrossRef]
25. Rostamian, A.; O'Hara, J.G. Event prediction within directional change framework using a CNN-LSTM model. *Neural Comput. Appl.* **2022**, *34*, 17193–17205. [CrossRef]
26. Bai, L.; Guan, S.; Guo, J.; Li, Z.; Jin, X.; Cheng, X. Integrating deep event-level and script-level information for script event prediction. *arXiv* **2021**, arXiv:2110.15706. [CrossRef]
27. Zhou, P.; Wu, B.; Wang, C.; He, L. An improved hierarchical neural network model with local and global feature matching for script event prediction. *Expert Syst. Appl.* **2025**, *259*, 125325. [CrossRef]
28. Lv, S.; Qian, W.; Huang, L.; Han, J.; Hu, S. Sam-net: Integrating event-level and chain-level attentions to predict what happens next. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6802–6809. [CrossRef]
29. Wang, L.; Yue, J.; Guo, S.; Sheng, J.; Mao, Q.; Chen, Z.; Zhong, S.; Li, C. Multi-level connection enhanced representation learning for script event prediction. In Proceedings of the Web Conference 2021, Online, 12–23 April 2021; pp. 3524–3533. [CrossRef]
30. Zheng, J.; Cai, F.; Ling, Y.; Chen, H. Heterogeneous graph neural networks to predict what happen next. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 328–338. [CrossRef]
31. Du, L.; Ding, X.; Zhang, Y.; Xiong, K.; Liu, T.; Qin, B. A graph enhanced BERT model for event prediction. *arXiv* **2022**, arXiv:2205.10822. [CrossRef]
32. Islam, M.I.K.; Saifuddin, K.M.; Hossain, T.; Akbas, E. Dygcl: Dynamic graph contrastive learning for event prediction. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; IEEE: New York, NY, USA, 2024; pp. 559–568. [CrossRef]
33. Rong, H.; Chen, Z.; Lu, Z.; Xu, X.k.; Huang, K.; Sheng, V.S. Pred-ID: Future event prediction based on event type schema mining by graph induction and deduction. *Inf. Fusion* **2025**, *117*, 102819. [CrossRef]
34. Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Chang, B.; Li, S.; Sui, Z. Towards time-aware knowledge graph completion. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 1715–1724. [CrossRef]
35. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197. [CrossRef]
36. De Caigny, A.; Coussement, K.; De Bock, K.W. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *Eur. J. Oper. Res.* **2018**, *269*, 760–772. [CrossRef]

37. Nguyen, N.T.; Tran, C.; Le, T. TBicomR: Event Prediction in Temporal Knowledge Graphs with Bicomplex Rotation. *Knowl.-Based Syst.* **2024**, *306*, 112711. [CrossRef]
38. Yang, J.; Yang, L.T.; Wang, H.; Gao, Y. Temporal Interaction Embedding for Link Prediction in Global News Event Graph. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 5327–5336. [CrossRef]
39. Jin, W.; Qu, M.; Jin, X.; Ren, X. Recurrent Event Network: Autoregressive Structure Inferenceover Temporal Knowledge Graphs. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6669–6683. [CrossRef]
40. Li, Z.; Jin, X.; Li, W.; Guan, S.; Guo, J.; Shen, H.; Wang, Y.; Cheng, X. Temporal knowledge graph reasoning based on evolutional representation learning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 11–15 July 2021; pp. 408–417. [CrossRef]
41. Li, Y.; Sun, S.; Zhao, J. TiRGN: Time-Guided Recurrent Graph Network with Local-Global Historical Patterns for Temporal Knowledge Graph Reasoning. In Proceedings of the IJCAI, Vienna, Austria, 23–29 July 2022; pp. 2152–2158. [CrossRef]
42. Zhou, X.; Xu, X.; Liang, W.; Zeng, Z.; Yan, Z. Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT. *IEEE Int. Things J.* **2021**, *8*, 12588–12596. [CrossRef]
43. Zhou, X.; Wu, J.; Liang, W.; Wang, K.I.K.; Yan, Z.; Yang, L.T.; Jin, Q. Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11817–11828. [CrossRef]
44. Huai, Z.; Yang, G.; Tao, J. Spatial-temporal knowledge graph network for event prediction. *Neurocomputing* **2023**, *553*, 126557. [CrossRef]
45. Tang, X.; Chen, L.; Shi, H.; Lyu, D. Dhyper: a recurrent dual hypergraph neural network for event prediction in temporal knowledge graphs. *ACM Trans. Inf. Syst.* **2024**, *42*, 1–23. [CrossRef]
46. Jia, W.; Ma, R.; Niu, W.; Yan, L.; Ma, Z. SFTe: Temporal knowledge graphs embedding for future interaction prediction. *Inf. Syst.* **2024**, *125*, 102423. [CrossRef]
47. Mao, X.; Shan, Y.; Li, F.; Chen, X.; Zhang, S. CLSpell: contrastive learning with phonological and visual knowledge for chinese spelling check. *Neurocomputing* **2023**, *554*, 126468. [CrossRef]
48. Zhou, X.; Zheng, X.; Cui, X.; Shi, J.; Liang, W.; Yan, Z.; Yang, L.T.; Shimizu, S.; Wang, K.I.K. Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3191–3211. [CrossRef]
49. Xiao, Z.; Mai, Z.; Xu, Z.; Cui, Y.; Li, J. Corporate event predictions using large language models. In Proceedings of the 2023 10th International Conference on Soft Computing & Machine Intelligence (ISCMI), Mexico City, Mexico, 25–26 November 2023; IEEE: New York, NY, USA, 2023; pp. 193–197. [CrossRef]
50. Shi, X.; Xue, S.; Wang, K.; Zhou, F.; Zhang, J.; Zhou, J.; Tan, C.; Mei, H. Language models can improve event prediction by few-shot abductive reasoning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 29532–29557. [CrossRef]
51. Wei, S.; Zang, L.; Zhang, X.; Liu, Q.; Hu, S. Leveraging Evolution Patterns to Enhance Script Event Prediction by Large Language Models. In Proceedings of the International Conference on Database Systems for Advanced Applications, Gifu, Japan, 2–5 July 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 371–380. [CrossRef]
52. Nako, P.; Jatowt, A. Navigating Tomorrow: Reliably Assessing Large Language Models Performance on Future Event Prediction. *arXiv* **2025**, arXiv:2501.05925. [CrossRef]
53. Xia, Y.; Wang, D.; Liu, Q.; Wang, L.; Wu, S.; Zhang, X. Chain-of-history reasoning for temporal knowledge graph forecasting. *arXiv* **2024**, arXiv:2402.14382. [CrossRef]
54. Xiong, S.; Payani, A.; Kompella, R.; Fekri, F. Large Language Models Can Learn Temporal Reasoning. *CoRR* **2024**, 10452–10470. [CrossRef]
55. Jiang, Z.; Liu, B.; Peng, M.; Xu, W.; Xiao, Y.; Shan, Z.; Peng, M. Towards Explainable Temporal Reasoning in Large Language Models: A Structure-Aware Generative Framework. *arXiv* **2025**, arXiv:2505.15245. [CrossRef]
56. Lee, G.; Yu, W.; Shin, K.; Cheng, W.; Chen, H. Timecap: Learning to contextualize, augment, and predict time series events with large language model agents. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 10–14 November 2025; Volume 39, pp. 18082–18090. [CrossRef]
57. Ye, C.; Hu, Z.; Deng, Y.; Huang, Z.; Ma, M.D.; Zhu, Y.; Wang, W. Mirai: Evaluating llm agents for event forecasting. *arXiv* **2024**, arXiv:2407.01231. [CrossRef]
58. Li, J.; Li, G.; Wang, L.; Zhu, H.; Jin, Z. Generating equivalent representations of code by a self-reflection approach. *arXiv* **2024**, arXiv:2410.03351. [CrossRef]
59. Yan, Y.; Jiang, J.; Liu, Y.; Cao, Y.; Xu, X.; Zhang, M.; Cai, X.; Shao, J. Sˆ 3cmath: Spontaneous step-level self-correction makes large language models better mathematical reasoners. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 10–14 November 2025; Volume 39, pp. 25588–25596. [CrossRef]
60. Liu, F.; AlDahoul, N.; Eady, G.; Zaki, Y.; Rahwan, T. Self-Reflection Makes Large Language Models Safer, Less Biased, and Ideologically Neutral. *arXiv* **2024**, arXiv:2406.10400. [CrossRef]

61. Alibaba Cloud. Qwen2.5-7B and Qwen2.5-14B. Model Released by Alibaba Cloud, Available on Hugging Face, 2025. Available online: https://huggingface.co/Qwen (accessed on 27 June 2025).
62. xAI. Grok-2. Large Language Model Developed by xAI, 2025. Available online: https://www.xai.com/grok-2 (accessed on 27 June 2025).
63. xAI. Grok-3. Large Language Model Developed by xAI, 2025. Available online: https://www.xai.com/grok-3 (accessed on 27 June 2025).
64. OpenAI. o1 System Card. Large Language Model by OpenAI. 2024. Available online: https://openai.com/zh-Hans-CN/o1/ (accessed on 27 June 2025).
65. OpenAI. GPT-4.1. Large Language Model Developed by OpenAI. 2025. Available online: https://openai.com/research/gpt-4 (accessed on 27 June 2025).
66. Google. Gemini 2.5 Pro. Large Language Model Developed by Google DeepMind. 2025. Available online: https://deepmind.google/models/gemini/pro/ (accessed on 27 June 2025).
67. Mistral AI. Mistral-Small-3.1. Large Language Model Released by Mistral AI. 2025. Available online: https://mistral.ai/news/mistral-small-3-1 (accessed on 27 June 2025).
68. Anthropic. Claude 3.5 Haiku. Large Language Model Developed by Anthropic, 2025. Available online: https://www.anthropic.com/claude/haiku (accessed on 27 June 2025).
69. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901. [CrossRef]
70. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837. [CrossRef]