MIRAI: Evaluating LLM Agents for International Event Forecasting

Chenchen Ye*
University of California, Los Angeles
`ccye@cs.ucla.edu`

Ziniu Hu*
University of California, Los Angeles
`acgbull@gmail.com`

Yihe Deng*
University of California, Los Angeles
`yihedeng@ucla.edu`

Zijie Huang
University of California, Los Angeles
`zijiehjj@gmail.com`

Mingyu Derek Ma
University of California, Los Angeles
`ma@cs.ucla.edu`

Yanqiao Zhu
University of California, Los Angeles
`yzhu@cs.ucla.edu`

Wei Wang
University of California, Los Angeles
`weiwang@cs.ucla.edu`

Abstract

We present MIRAI, a **benchmark** designed to systematically evaluate LLM agents as temporal forecasters to predict international events. Our benchmark features an agentic environment with APIs to access an extensive database of historical, structured events and textual news articles. We refine the GDELT[2] event database with careful cleaning and parsing to curate a series of relational prediction tasks with varying forecasting horizons, assessing LLM agents' abilities from short-term to long-term forecasting. Notably, MIRAI features a dynamic data construction pipeline that supports periodically downloading recent news and events, and automatically generates the most recent test split. This allows us to evaluate any newly released model in a contamination-free manner as we can always construct a test split later than its knowledge cutoff date. We implement several Tool-Use pipelines, including RAG baseline and ReAct Agent (with Single-Function or Code-Block). We evaluate different open-source and commercial LLMs, and find stronger base models are able to utilize diverse knowledge sources, able to write comprehensive and correct tool-use codes, and able to correctly reasonable temporal events to make prediction. We believe MIRAI can provide a good testbed for future development of LLM Search Agents[3].

1 Introduction

Accurate forecasting of international events is essential for stakeholders to navigate the complexities of an interconnected world, enabling informed decision-making, risk mitigation, and opportunity identification [Brown and Lee, 2018]. Researchers have developed numerous AI-driven approaches to tackle this challenge, leveraging structured knowledge graphs [Mahdisoltani et al., 2015, Jin et al., 2020a, Li et al., 2021a] or textual datasets [Zou et al., 2022, Reddy et al., 2023] to predict geopolitical

---

[*] Equal Contribution.

[2] GDELT: https://www.gdeltproject.org/

[3] We released our code repository here, constructed benchmark here, and an interactive notebook here.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: LAW 2025.

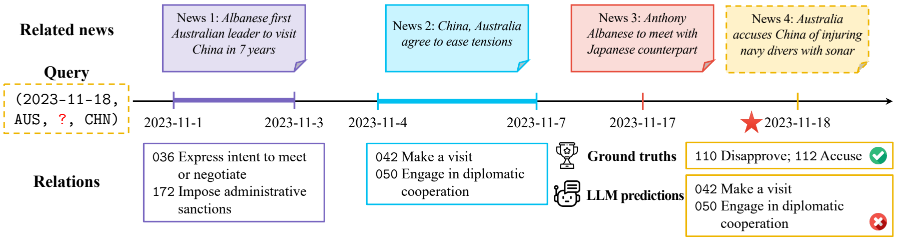

Related news
News 1: Albanese first Australian leader to visit China in 7 years
News 2: China, Australia agree to ease tensions
News 3: Anthony Albanese to meet with Japanese counterpart
News 4: Australia accuses China of injuring navy divers with sonar
Query
(2023-11-18, AUS, ?, CHN)
2023-11-1
2023-11-3
2023-11-4
2023-11-7
2023-11-17
2023-11-18
Relations
036 Express intent to meet or negotiate
172 Impose administrative sanctions
042 Make a visit
050 Engage in diplomatic cooperation
Ground truths
110 Disapprove; 112 Accuse
LLM predictions
042 Make a visit
050 Engage in diplomatic cooperation

Figure 1: **An example of forecasting the relations between Australia and China on 2023-11-18.** The agent fails to predict the change of relation and makes a wrong forecast.

developments. However, generalizability in making reliable forecasts remains a challenge for these task-specific models.

Recent advancements in Large Language Model (LLM) agents, especially systems like DeepResearch [dee, 2024, goo, 2024, per, 2025] and DeepSearch [gro, 2025], offer a promising solution. These agents can autonomously search the web, integrate diverse information sources, and use long-CoT reasoning to solve complex tasks. While their potential for forecasting has been demonstrated in exploratory use cases, the absence of standardized benchmarks hinders a systematic evaluation of LLM Agents' forecasting capabilities. Developing such a benchmark requires: 1) accurate and challenging prediction tasks tied to international events; 2) diverse, queryable knowledge sources; and 3) most importantly, the prediction task must be set in the future relative to both the provided knowledge sources and the LLMs' internal knowledge cutoff to prevent information leakage.

To address these needs, we present **MIRAI** (Multi-Information FoRecasting Agent Interface), the first forecasting benchmark designed as an agentic environment with rich structured and textual data. Built on the continuously updated Global Database of Events, Language, and Tone (GDELT) [Leetaru and Schrodt, 2013a], MIRAI transforms real-world event data into forecasting tasks across multiple timeframes and horizons. Our dynamic data pipeline periodically incorporates recent news and events, generating contamination-free test sets aligned with the latest developments. This ensures that evaluations remain robust, testing LLMs against data postdating their knowledge cutoffs—a feature absent in prior benchmarks with static datasets.

We assessed both open- and closed-source LLMs on MIRAI using RAG and prompting baselines, as well as ReAct-style agents [Yao et al., 2023a] with "Single Function" and "Code Block" action types. We also implement a single-turn multi-function-call agent for long-cot thinking model. Experiments across test splits (2023-11, 2023-12, 2024-01, 2024-02) revealed key insights: 1) Temporal forecasting poses significant challenges, with the top-performing GPT-4o-mini agent achieving a 30.3 F1 score on second-level relation prediction, while long-term and fine-grained tasks proved even harder; 2) The "Code Block" strategy, enabling flexible interactions, benefits models like GPT-4o-mini with strong code generation skills more than others. These findings underscore the need for improved temporal reasoning and tool-use capabilities in LLM agents.

In summary, we present MIRAI as **a benchmark for evaluating LLM forecasting agents'** with: 1) An agentic environment with APIs to access diverse data from structured events to textual news to support Agent Tool-Use. 2) A dynamic pipeline to automatically construct contamination-free test splits beyond model knowledge cutoffs.

2 The MIRAI Benchmark

2.1 Task and Data

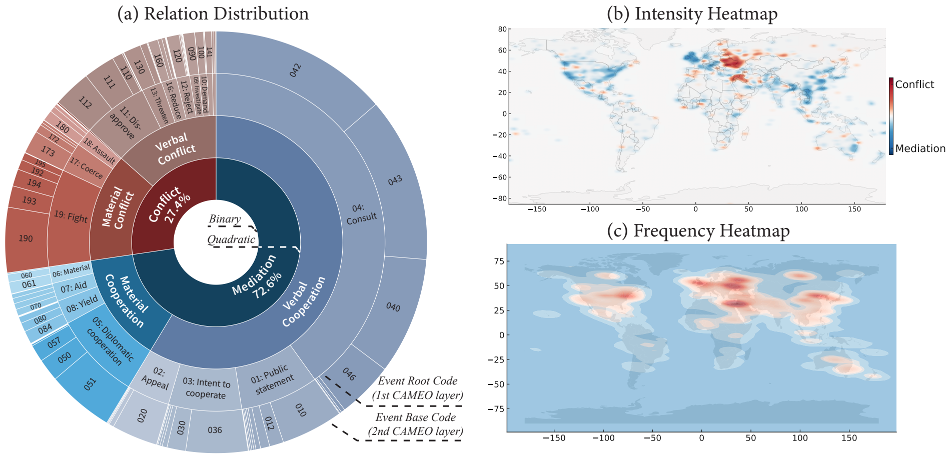
We consider *forecasting* as the process of collecting essential historical data and performing temporal reasoning to anticipate the outcomes of future events, as illustrated in Figure 1.

Structured and Textual Event Representations. We represent an event as $e^t = (t, s, r, o)$, and all news articles mentioning this event e^t at the same day as $\boldsymbol{D}^{t'}_{e^t}$, such that $\boldsymbol{D}[t'] = e[t]$. We denote t as the daily timestamp, formatted in `"YYYY-MM-DD"`; $s, o \in \mathcal{C}$ are subject and object countries[4] from the country pool $\mathcal{C}$, $r \in \mathcal{R}$ denotes the relation type defined by CAMEO (Conflict and Mediation Event Observations) ontology[5] [Boschee et al., 2015a]. Events at timestamp t form a set $\boldsymbol{E}^t = \{e^t_1, ..., e^t_{|\boldsymbol{E}^t|}\}$ with $\{\boldsymbol{E}^t\}_{t=1}^{T}$ being able to organize the temporal graphs where countries are nodes and relations are edges. Correspondingly, $\boldsymbol{D}^t = \{d^t_1, ..., d^t_{|\boldsymbol{D}^t|}\}$ is the set of all news

---

[4] In this paper, the term "country" includes all countries, dependent territories, special geographic areas, and their subdivisions. We use the standardized ISO-3166 Alpha-3 codes for country names, e.g. `"AUS"` for Australia.

[5] CAMEO is a well-established ontology meticulously developed by domain experts over years, for categorizing international political events across multiple levels of granularity.



(a) Relation Distribution

(b) Intensity Heatmap

(c) Frequency Heatmap

Figure 2: **MIRAI comprehensive global event coverage.** (a) Circular chart: Relation hierarchy and distribution in MIRAI. (b) Heatmap: Global events intensity, including areas of conflict (red) and mediation (blue). (c) Heatmap: Event frequency by region.

articles published at day t. This dual representation of structured events and their associated textual information allows for a comprehensive analysis of international events, leveraging both the concise, categorized nature of the event tuples and the rich contextual details provided by the news articles.

Event Forecasting Task. The task of event forecasting $(t+l, s, ?, o)$ is to predict all events between a pair of countries s and o, happening l days ahead from the current date t.[6] Formally, given historical events $\boldsymbol{E}^{\leq t}$ and associated news articles $\boldsymbol{D}^{\leq t}$ up to the current time t, our goal is to forecast future relationships $\boldsymbol{E}_{s,o}^{t+l}$. This requires agent utilizing both structured and textual information, considering interactions not only between the target countries but also involving third parties, such as their mutual neighbors.

Hierarchical Event Categories. As shown in Figure 2a, we incorporate two hierarchical relation levels from the CAMEO ontology to facilitate a detailed and comprehensive spectrum of geopolitical dynamics. The first level includes 20 broad categories, represented by a two-digit code (e.g., "`01: Public Statement`" or "`04: Consult`"), which are subdivided into second-level categories identified by a three-digit code (e.g., "`03: Express intent to cooperate`" is a first-level category that includes 10 different second-level relations such as "`036: Express intent to meet`"). Subsequently, the quadruple "(`2023-11-03`, `AUS`, `036`, `CHN`)" denotes that on 3 November 2023, the Australian leader announces a planned visit to China. These relations are also organized along two dimensions, from `Verbal` to `Material` and from `Conflict` to `Cooperation`, to form a quadratic categorization in the inner circle of Figure 2a.

2.2 Dataset Construction

We carefully curate and clean our database to consider critical aspects such as preventing test information leakage, ensuring label accuracy, verifying source reliability, and addressing ethical concerns.

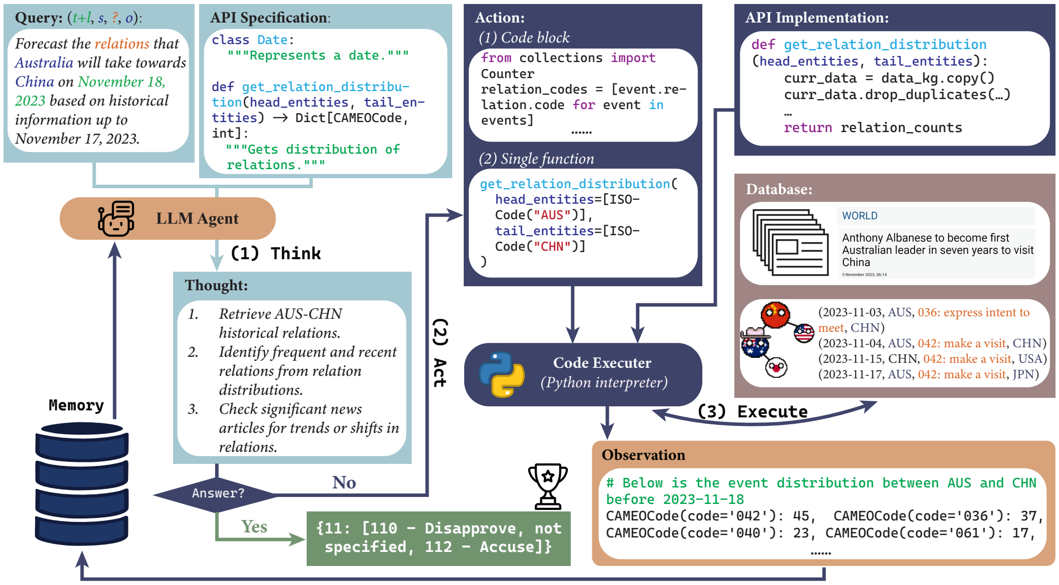
Raw Data Collection. We construct the database based on the GDELT project[7], which crawls global news media and extracts event information every 15 minutes. Each event contains date, actor, action (relation), geography, and news source. The dataset used in this paper spans from January 1, 2023, upto February 29, 2024, based on which we can create multiple time-split test sets. Noted that each published news might mention past events, but the extracted date can be noisy (e.g., hard to predict exact date for news referring "one month ago"). To ensure date correctness and prevent potential information leakage, we only keep those events for which their sourced news explicitly mention it happens "today". Detailed standardization is listed in Appendix D.1.

Textual Data Processing. For associated news articles, we implement the following steps: (1) *Source reliability threshold*: We retain events with at least 50 daily news mentions. This reduces

---

[6] MIRAI naturally support other tasks like object prediction $(t+l, s, r, ?)$. We focus on event (edge) prediction in this paper as: 1) event distribution is less biased compared to countries, which often dominated by a few countries only; 2) predicting the relation dynamic between a pair of countries over time require Agent to capture key shift points, and thus more challenging as an eval task.

[7] `https://www.gdeltproject.org/`



Forecast the relations that Australia will take towards China on November 18, 2023 based on historical information up to November 17, 2023.

```
class Date:
  """Represents a date."""

def get_relation_distribu-
tion(head_entities, tail_en-
tities) → Dict[CAMEOCode,
int]:
  """Gets distribution of
  relations."""
```

```
from collections import
Counter
relation_codes = [event.re-
lation.code for event in
events]
        ......
```

```
get_relation_distribution(
  head_entities=[ISO-
  Code("AUS")],
  tail_entities=[ISO-
  Code("CHN")]
)
```

```
def get_relation_distribution
(head_entities, tail_entities):
    curr_data = data_kg.copy()
    curr_data.drop_duplicates(…)
    …
    return relation_counts
```

Anthony Albanese to become first Australian leader in seven years to visit China

(2023-11-03, AUS, 036: express intent to meet, CHN)
(2023-11-04, AUS, 042: make a visit, CHN)
(2023-11-15, CHN, 042: make a visit, USA)
(2023-11-17, AUS, 042: make a visit, JPN)

1. *Retrieve AUS-CHN historical relations.*
2. *Identify frequent and recent relations from relation distributions.*
3. *Check significant news articles for trends or shifts in relations.*

```
# Below is the event distribution between AUS and CHN
before 2023-11-18
CAMEOCode(code='042'): 45,  CAMEOCode(code='036'): 37,
CAMEOCode(code='040'): 23, CAMEOCode(code='061'): 17,
                     ......
```

```
{11: [110 - Disapprove, not
 specified, 112 - Accuse]}
```

Figure 3: **Overview of the LLM agent's forecasting process using the ReAct strategy**. The framework consists of three main steps: (1) Think: Agent analyzes status and plans next action based on query and API specs. (2) Act: The agent generates a "Single Function" call or a "Code Block" to retrieve and analyze relevant data. (3) Execute: Python interpreter runs generated code, producing observations. These steps are repeated performed until the agent reaches final forecast.

the influence of less reliable sources such as personal blogs and decreases erroneously extracted events. (2) *cleaning for reliability and ethical integrity*: We apply rigorous text cleaning following the OBELICS protocol [Laurençon et al., 2023]. This process operates at both paragraph and document levels, filtering low-value content with low word counts or high character/word repetition ratios and removing excessive special characters. Importantly, we employ a list of flagged words to identify and eliminate potentially sensitive or inappropriate content, aligning our data collection with ethical standards.

Test Splits Construction. We construct multiple test splits (2023-11, 2023-12, 2024-01, 2024-02), each covering all events in the one-month period. For each test split: (1) *Enhanced filtering*: We apply higher thresholds (100 daily mentions, 5 news articles) to ensure higher data quality and reliability for test set. (2) *Balanced sampling*: For each month, we sample 100 queries to form balanced test splits, ensuring more uniform distribution across dates, countries, and CAMEO code types. Using the same processing script, we can generate test split for any following month, keep MIRAI a contamination-free and forecasting test set for any LLMs to ensure their knowledge cutoff date is before all predicted events' dates.

Statistics and Documentation. The resulting dataset contains 1,296,991 GDELT event records, corresponding to 75,341 unique (t, s, r, o) events and 401,013 unique news articles. Figures 2b and 2c illustrate the global distribution of our curated events, highlighting the varying intensities of conflict and mediation between regions. We provide additional details of human evaluation of the data quality in Appendix D.2. A standardized datasheet [Gebru et al., 2021] for MIRAI is in Appendix G, clearly documenting its motivation, collection process, distribution and maintenance.

Evaluation metrics. We instruct the agent to predict both first-level and second-level CAMEO codes in a JSON dictionary. Evaluation involves calculating *precision*, *recall*, and *F1 score* between the predicted and ground-truth lists. Moreover, we map each predicted relations to their respective binary and quadratic classes (as shown in Figure 2a), and aggregate to get a histogram. To measure prediction-ground truth alignment at class-level, we employ the *empirical Kullback-Leibler (KL) divergence*: $D_{\text{KL}}(P||Q) = \sum_i P(i) \log\left(P(i)/Q(i)\right)$, where P and Q represent the frequencies of ground-truth and predicted relations respectively. A lower KL divergence indicates a better alignment of the model's predictions with the ground-truth list. Appendix D.4.2 provides further discussion on the selection of metrics.

2.3 Agents and Environments

Similar to human political analysts, LLM agents must leverage a variety of information sources to make reliable predictions. We provide an environment with coding APIs to facilitate flexible access

Table 1: Evaluation results on the 2024-02 test split using different base LLMs and action types. The best-performing score is highlighted in **bold** and the second-best is underlined.

Base LLM	Training Data Cutoff Date	Action Type	Binary KL ($\Downarrow$)	Quad KL ($\Downarrow$)	First-level Relation (%)			Second-level Relation (%)		
					Pre. ($\Uparrow$)	Rec. ($\Uparrow$)	F1 ($\Uparrow$)	Pre. ($\Uparrow$)	Rec. ($\Uparrow$)	F1 ($\Uparrow$)
Mistral-7B-Instruct-v0.2	2023-12	Single Func	$10.3_{\pm1.7}$	$14.2_{\pm1.9}$	$38.1_{\pm0.5}$	$19.2_{\pm4.2}$	$18.9_{\pm1.1}$	$21.9_{\pm4.1}$	$9.8_{\pm3.5}$	$9.3_{\pm0.6}$
		Code Block	$9.1_{\pm2.3}$	$14.3_{\pm1.6}$	$31.3_{\pm5.1}$	$12.5_{\pm2.6}$	$15.1_{\pm3.6}$	$13.1_{\pm1.2}$	$9.4_{\pm1.6}$	$8.4_{\pm1.9}$
Llama-3-8B-Instruct	2023-03	Single Func	$9.0_{\pm2.4}$	$14.1_{\pm1.6}$	$39.8_{\pm1.6}$	$15.6_{\pm1.2}$	$18.6_{\pm0.2}$	$15.8_{\pm0.5}$	$11.8_{\pm0.6}$	$10.3_{\pm0.1}$
		Code Block	$9.4_{\pm2.6}$	$14.5_{\pm1.5}$	$39.5_{\pm0.7}$	$12.2_{\pm2.4}$	$15.9_{\pm2.5}$	$18.9_{\pm2.0}$	$8.9_{\pm1.7}$	$9.2_{\pm0.9}$
Llama-3.1-8B-Instruct	2023-12	Single Func	$6.8_{\pm1.2}$	$11.8_{\pm2.4}$	$55.5_{\pm7.8}$	$23.7_{\pm0.1}$	$28.3_{\pm2.5}$	$26.3_{\pm5.4}$	$20.9_{\pm1.1}$	$17.0_{\pm1.5}$
		Code Block	$8.8_{\pm2.8}$	$13.6_{\pm2.7}$	$36.3_{\pm0.1}$	$15.0_{\pm3.7}$	$18.3_{\pm2.5}$	$18.5_{\pm0.9}$	$12.0_{\pm2.1}$	$11.3_{\pm0.5}$
DeepSeek-R1-Distill-Llama-8B	2023-12	Single Func	$6.3_{\pm1.5}$	$12.4_{\pm1.8}$	$50.5_{\pm2.1}$	$18.9_{\pm3.7}$	$23.6_{\pm2.9}$	$26.4_{\pm0.9}$	$11.2_{\pm1.8}$	$12.8_{\pm1.8}$
		Code Block	$8.1_{\pm1.3}$	$12.6_{\pm2.1}$	$45.9_{\pm5.3}$	$17.9_{\pm1.1}$	$22.0_{\pm2.2}$	$24.0_{\pm3.4}$	$12.3_{\pm2.7}$	$12.2_{\pm2.1}$
GPT-3.5-Turbo	2021-09	Single Func	$\underline{3.5}_{\pm1.3}$	$\mathbf{7.5}_{\pm2.7}$	$55.7_{\pm5.8}$	$\mathbf{40.9}_{\pm2.6}$	$38.3_{\pm4.2}$	$\underline{42.4}_{\pm4.4}$	$\mathbf{34.3}_{\pm5.0}$	$28.3_{\pm3.9}$
		Code Block	$5.3_{\pm1.4}$	$9.3_{\pm2.3}$	$34.8_{\pm11.7}$	$34.9_{\pm0.9}$	$26.7_{\pm3.8}$	$16.8_{\pm5.9}$	$26.4_{\pm0.5}$	$15.1_{\pm2.6}$
GPT-4o-mini	2023-10	Single Func	$4.0_{\pm0.9}$	$8.1_{\pm1.3}$	$\mathbf{61.3}_{\pm11.4}$	$34.7_{\pm1.9}$	$\underline{39.0}_{\pm6.0}$	$40.0_{\pm5.5}$	$\underline{32.6}_{\pm1.6}$	$\underline{29.7}_{\pm3.8}$
		Code Block	$\mathbf{3.2}_{\pm0.8}$	$\underline{7.7}_{\pm1.9}$	$\underline{59.8}_{\pm5.1}$	$\underline{37.1}_{\pm0.4}$	$\mathbf{40.0}_{\pm3.1}$	$\mathbf{46.5}_{\pm2.1}$	$29.7_{\pm0.6}$	$\mathbf{30.3}_{\pm1.0}$
Human Performance	—	—	0.04	1.37	62.73	88.70	68.29	54.54	74.53	56.78

to various knowledge sources. Our LLM agent uses these APIs to interact with the environment for forecasting through the ReAct pipeline [Yao et al., 2023b], characterized by the iterative process of *think*, *act*, and *observe*.

APIs. We provide the LLM agent with a comprehensive set of APIs including essential *functions* designed for the various types of information within the database. The function design is characterized in two aspects:

- **Diverse Information Types** we provide functions that can retrieve diverse types of information including news articles (e.g. `get_news_article`) and knowledge graph (e.g. `get_events`). We also support auxiliary helping functions to access relation mappings, hierarchies, and events/articles statistics (counts, listings, and distributions).
- **Search conditions.** Additionally, the API functions for events and news articles offer optional parameters for tailored searches based on different criteria. For instance, `get_event` allows searches specifying conditions like `date_range`, `head_entities`, `tail_entities`, `relations`, and `text_description` to retrieve specific events from the database.

The data classes and functions provided in the API are shown in Appendix E.

Interactions with Environment. The environment is equipped with a Python code sandbox with full API and database access. Mirai informs agents how to interact with the environment through API Specifications, including detailed Python docstrings for all data classes and functions that abstract implementation details. Agents use the ReAct [Yao et al., 2023a] strategy to iteratively alternate between *think*, *act*, and *observe* to gather information and forecast. The pipeline is illustrated in Figure 3. Spcifically, we put the query to LLM in USER turn, and expect all the tool-use and execution results all put in the output ASSISTANT turn. Everytime we prompt model to **Think** about next tool-call via appending a Think suffix, and agent can analyze current retrieved knowledge to decide whether to stop or make next tool-call, which parameters to put in, etc. Then, model **Act** by outputing executiable json codes in a wrapper. We allow model to output "Single Function" for straightforward data retrieval or "Code Block" for complex operations including loops and conditionals[8]. We then parse the json codes and send to a sandbox for execution. Next, we append the execution results back to the ASSISTANT turn as **Observe**. This single-turn ReAct pipeline is naturally applicable to long-cot reasoning models like O1, O3 and Deepseek R1.

3 Experiments

We evaluate LLM Agents with different base models and code-execution types (Sec. 3.1), and compare with non-Agent baselines (Sec. 3.2), followed by extensive analysis (Sec. 3.3).

3.1 Evaluate LLM Agent Forecasting with Different Base LLMs

We evaluate both open-sourced LLMs, including Mistral-7B-Instruct-v0.2 [Jiang et al., 2023], Llama-3-8B-Instruct, and Llama-3.1-8B-Instruct [Meta, 2024], and the reasoning model, DeepSeek-R1-Distill-Llama-8B [DeepSeek-AI, 2025], as well as commercial LLMs including GPT-3.5-Turbo [gpt, 2023] and GPT-4o-mini [gpt, 2024]. Comparisons are done on the **2024-02** test split that is after all models' training data cutoff date, with 100 balanced queries. All models use ReAct framework

---

[8] Examples of agents performing forecasting using "Code Block" and "Single Function" are shown in Appendix F. Example for Thinking Model in Appendix F.3.

Table 2: Evaluation results of different forecasting methods on the 2024-02 test split.

	Method	Info Sources		Binary KL (⇓)	Quad KL (⇓)	First-level Relation (%)			Second-level Relation (%)		
		Events	News			Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)
Non-LLM Baselines	Recurrency	—	—	2.1	2.5	29.8	86.0	41.6	14.2	80.1	23.0
	REGCN	✓	—	**0.3**	2.5	23.9	74.4	31.3	5.5	28.4	7.9
	ForecastQA	—	✓	9.7	13.8	55.0	16.2	22.1	40.0	8.8	12.0
Direct Prompt	IO `(Llama-3.1-8B-Instruct)`	—	—	$18.4_{\pm1.2}$	$19.5_{\pm1.9}$	$11.5_{\pm6.4}$	$6.2_{\pm1.5}$	$6.7_{\pm2.2}$	$5.2_{\pm2.5}$	$3.7_{\pm0.4}$	$3.4_{\pm1.1}$
	IO + CAMEO `(Llama-3.1-8B-Instruct)`	—	—	$8.2_{\pm4.2}$	$9.9_{\pm3.7}$	$25.6_{\pm0.8}$	$34.3_{\pm13.4}$	$20.6_{\pm2.7}$	$8.7_{\pm2.4}$	$18.8_{\pm13.9}$	$8.6_{\pm2.6}$
	ZS-CoT `(Llama-3.1-8B-Instruct)`	—	—	$7.5_{\pm2.0}$	$8.1_{\pm2.3}$	$20.4_{\pm0.6}$	$12.8_{\pm0.2}$	$15.3_{\pm0.8}$	$6.9_{\pm2.0}$	$8.2_{\pm0.4}$	$7.1_{\pm0.7}$
	IO `(DeepSeek-R1-Distill-Llama-8B)`	—	—	$9.3_{\pm2.3}$	$13.4_{\pm2.7}$	$45.8_{\pm1.1}$	$14.0_{\pm2.3}$	$18.2_{\pm0.9}$	$17.6_{\pm3.3}$	$8.8_{\pm1.3}$	$9.4_{\pm1.8}$
RAG	KG + BM25 `(Llama-3.1-8B-Instruct)`	✓	—	$16.0_{\pm1.3}$	$17.2_{\pm1.0}$	$23.4_{\pm10.8}$	$12.9_{\pm4.6}$	$14.3_{\pm5.8}$	$15.2_{\pm7.5}$	$10.0_{\pm2.9}$	$9.6_{\pm3.9}$
		—	✓	$16.8_{\pm1.4}$	$18.2_{\pm1.1}$	$17.8_{\pm1.6}$	$8.0_{\pm0.5}$	$9.3_{\pm0.7}$	$7.6_{\pm0.6}$	$6.1_{\pm0.3}$	$5.2_{\pm1.4}$
		✓	✓	$18.1_{\pm0.9}$	$19.4_{\pm1.3}$	$14.7_{\pm2.2}$	$6.6_{\pm0.8}$	$8.0_{\pm0.3}$	$7.3_{\pm1.7}$	$4.5_{\pm1.5}$	$4.3_{\pm0.2}$
	KG + BM25 `(DeepSeek-R1-Distill-Llama-8B)`	✓	—	$15.4_{\pm2.4}$	$17.0_{\pm2.2}$	$24.2_{\pm0.1}$	$15.0_{\pm0.8}$	$15.7_{\pm0.7}$	$12.6_{\pm2.7}$	$11.2_{\pm3.3}$	$10.2_{\pm2.6}$
		—	✓	$9.8_{\pm2.4}$	$13.3_{\pm2.3}$	$44.3_{\pm0.5}$	$11.9_{\pm0.2}$	$17.0_{\pm0.4}$	$19.1_{\pm0.6}$	$7.6_{\pm0.4}$	$9.2_{\pm0.0}$
		✓	✓	$12.8_{\pm1.9}$	$15.3_{\pm1.4}$	$29.9_{\pm3.3}$	$15.3_{\pm7.7}$	$16.8_{\pm6.8}$	$16.3_{\pm0.3}$	$10.6_{\pm6.7}$	$11.0_{\pm5.4}$
Agents	ReAct + Single-Function-Call `(Llama-3.1-8B-Instruct)`	✓	—	$6.5_{\pm1.6}$	$10.9_{\pm2.4}$	$57.6_{\pm10.6}$	$27.6_{\pm4.5}$	$31.2_{\pm7.4}$	$26.1_{\pm7.9}$	$23.9_{\pm1.0}$	$17.3_{\pm4.7}$
		—	✓	$8.7_{\pm0.5}$	$14.7_{\pm1.8}$	$44.5_{\pm0.7}$	$12.7_{\pm0.0}$	$17.0_{\pm0.2}$	$12.9_{\pm0.1}$	$9.7_{\pm2.3}$	$8.1_{\pm0.7}$
		✓	✓	$6.8_{\pm1.2}$	$11.8_{\pm2.4}$	$55.5_{\pm7.8}$	$23.7_{\pm0.1}$	$28.3_{\pm2.5}$	$26.3_{\pm5.4}$	$20.9_{\pm1.1}$	$17.0_{\pm1.5}$
	ReAct + Single-Function-Call `(DeepSeek-R1-Distill-Llama-8B)`	✓	—	$8.0_{\pm2.1}$	$12.8_{\pm2.2}$	$49.9_{\pm1.3}$	$18.7_{\pm1.7}$	$24.5_{\pm0.6}$	$28.2_{\pm2.0}$	$14.0_{\pm2.9}$	$14.9_{\pm0.4}$
		—	✓	$6.5_{\pm0.5}$	$12.4_{\pm0.8}$	$44.6_{\pm7.3}$	$17.0_{\pm3.0}$	$20.5_{\pm3.3}$	$21.8_{\pm1.8}$	$10.3_{\pm1.9}$	$10.4_{\pm0.6}$
		✓	✓	$6.3_{\pm1.5}$	$12.4_{\pm1.8}$	$50.5_{\pm2.1}$	$18.9_{\pm3.7}$	$23.6_{\pm2.9}$	$26.4_{\pm0.9}$	$11.2_{\pm1.8}$	$12.8_{\pm1.8}$

with access to all APIs. The action types can be either "Single Function" or "Code Block" with a maximum tool call limit as 20 steps. The same prompt is used across all models for fair comparison, as detailed in Appendix F. The experimental results are presented in Table 1, and we observe the following findings:

1) MIRAI presents a challenging task for LLM agents. The top performer, GPT-4o-mini ReAct with "Code Block," scored 46.5 precision and 30.3 F1 on second-level relations—well below human forecasting at 56.78 (Appendix). Fine-grained relation prediction proved even harder, emphasizing the task's complexity and LLMs' room for growth in event forecasting.

2) Code Block benefits stronger LLMs but hurts weaker models: The "Code Block" action type provides greater flexibility than the "Single Function". However, its benefits are not uniformly achieved across all models. Small open-sourced models and GPT-3.5-Turbo show reduced performance with Code Block, while GPT-4o-mini as a strong model gains improvements from it. This indicates that the ability to generate effective long code is a key factor that determines LLMs as reliable forecasting agents.

3.2 Evaluate Forecasting with Different Forecasting Methods and Information Sources

For non-LLM baselines, we selected: 1) **Recurrency** [Gastinger et al., 2024], a statistics-based method predicting via historical subject-object relation frequencies; 2) **REGCN** [Li et al., 2021b], a Temporal Knowledge Graph (TKG) approach using graph networks for temporal event embeddings and link prediction; and 3) **ForecastQA** [Jin et al., 2021a], a BERT-based method encoding queries and articles for relation classification. All were trained/fitted on data up to December 2023.

For LLM baselines, we used `Llama-3.1-8B-Instruct` as the non-thinking model and `DeepSeek-R1-Distill-Llama-8B` as the thinking model, both built on the `Llama-3.1-8B base model` base with a knowledge cutoff of December 2023. Experiments set a temperature of 0.4, reporting means and standard deviations over 5 runs. Prompting baselines without tools included: **Direct IO**, where LLMs answer using internal knowledge; **IO + CAMEO**, adding event ontology mapping codes to names; and **Zero-Shot Chain-of-Thought (ZS-CoT)** [Kojima et al., 2022, Wei et al., 2023], appending "Please think step by step" (with "<think>" for the thinking model to trigger reasoning). We also implemented **Retrieval-Augmented Generation (RAG)** baselines with three setups: event-only, news-only, and combined. Event retrieval followed GPT-NeoX-ICL [Lee et al., 2023], using rule-based retrieval for historical events with shared actors; news retrieval used TCELongBench [Zhang et al., 2024], applying BM25 to fetch query-relevant articles.

Table 2 reveal several key insights: **1) MIRAI naturally supports the evaluation of various forecasting methods:** While designed for LLM agents, MIRAI 's rich data supports testing non-LLM approaches like Recurrency and REGCN. These models excel in recall by capturing temporal patterns and relational embeddings but struggle with precision, especially for fine-grained second-level relations. This highlights limitations in relying solely on simple heuristics or global graph modeling. Baseline details and LLM comparisons are in Appendices D.3 and Appendix D.4.

2) Reasoning and diverse data sourcing are key to temporal forecasting. ZS-CoT and Direct-IO, relying only on LLM internal knowledge, lag behind ReAct agents with full API access. Thinking

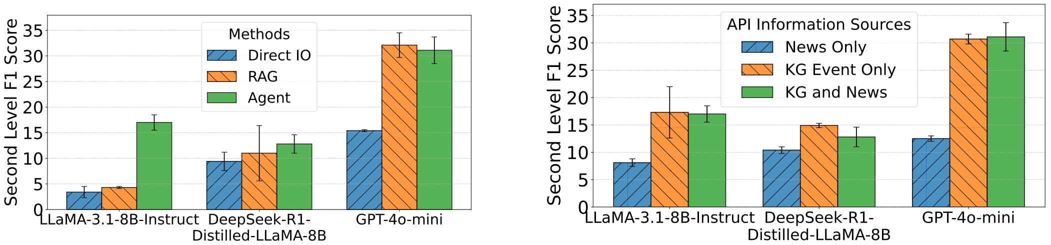


Figure 5: a) Different forecasting methods on different base model. b) Available API information types affects agent performance differently.

models and ZS-CoT outperform IO baselines, underscoring the need for historical data retrieval and analytical reasoning over pretrained knowledge alone.

3) RAG vs. Agent Performance across LLMs. Figure 5(a) shows weaker models like `Llama-3.1-8B-Instruct` perform better with ReAct than RAG, likely due to dynamic analysis and iterative function calls. Smaller reasoning models (e.g., `Deepseel-R1-Distill-Llama-8B`) show less improvement. We observe that reasoning models invokes fewer function calls, and more often hallucinate function results instead of retrieving from knowledge base. Stronger models like `GPT-4o-mini` excel with both RAG and ReAct, reflecting robust post-training for multi-document processing and temporal reasoning, though API utilization remains imperfect (see Appendix D.5 for detailed results).

As shown in Figure 5(b), all models struggle with News Only APIs, prone to overconfidence and hallucination rather than ontology verification. `LLaMA-3.1-8B-Instruct` and `DeepSeek-R1-Distilled-LLaMA-8B` has higher performance with Event Only APIs compared with using both, while GPT-4o-mini achieves best performance with combined Events and News APIs, showcasing its ability in utilizing diverse tools for heterogeneous data.

3.3 Analyzing and Understanding LLM Agent Behaviors

Data Contamination of LLM Knowledge-Cutoff over Test-Time Splits

Model	Training Data Cutoff Date	Test-Month Splits			
		2023-11	2023-12	2024-01	2024-02
Llama-3-8B-Instruct	2023-03	$6.1_{\pm 1.5}$	$8.7_{\pm 1.8}$	$8.7_{\pm 0.1}$	$10.3_{\pm 0.1}$
Llama-3.1-8B-Instruct	2023-12	$15.8_{\pm 5.7}$	$15.9_{\pm 3.6}$	$16.3_{\pm 2.0}$	$14.8_{\pm 0.7}$
GPT-4-Turbo	2023-12	$\underline{30.0}_{\pm 1.9}$	$\underline{25.8}_{\pm 3.1}$	$\underline{32.2}_{\pm 2.8}$	$\underline{28.9}_{\pm 3.2}$
GPT-4o-mini	2023-10	$\mathbf{32.8}_{\pm 2.6}$	$\mathbf{25.9}_{\pm 3.2}$	$\mathbf{33.2}_{\pm 0.7}$	$\mathbf{29.7}_{\pm 3.8}$

Table 3: F1 (⇑) scores of second-level relation forecasting on different test splits, using "Single Function". The best-performing score is highlighted in **bold** and the second-best is underlined. More results in Appendix D.6.

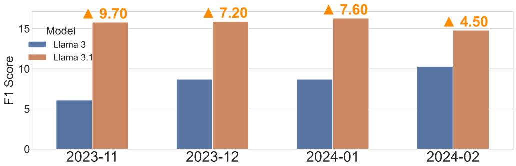


Figure 4: Llama-3.1 vs Llama-3 over four test splits. Llama-3.1 is much higher on 2023-11 (before its training data cutoff date). The gap is smaller afterwards. This shows we need to evaluate model using time-split after its cutoff date.

Table 3 compares the forecasting performance of two open-source Llama3 models [Meta, 2024] with different cutoff dates but similar training processes across multiple test splits. Figure 4 shows Llama-3.1 outperforming Llama-3 most significantly in the 2023-11 split—post-Llama-3's cutoff but pre-Llama-3.1's—with the gap narrowing by 2024-02. This suggests possible data contamination favoring models with more recent training data, underscoring the need for test splits beyond all models' cutoffs for robust evaluation. Our benchmark design ensures contamination-free test sets, preserving the integrity of forecasting assessments by testing true predictive ability, not memorized knowledge.

Impact of Temporal Distance on Forecasting Targets Our defined event forecasting task varies by temporal distance l, which specifies how far into the future we want to predict. We thus conduct an ablation study with l set to 1, 7, 30, and 90 days. For each, we fix the query event date and restrict data access to l days prior. Figure 6 shows that as l increases, F1 scores drop and KL-divergence rises, indicating reduced prediction accuracy for distant events. Short horizons (1 or 7 days) benefit from recent, relevant data (e.g., expert analyses), enabling precise forecasts. Longer horizons (30 or 90 days) demand capturing complex trends and dependencies, making them critical for benchmarking LLM agents' true forecasting abilities.

Forecasting accuracy across relation types. We categorize test events into quadratic relation classes and calculate F1 scores for each, as shown in Figure 8b. All models perform notably

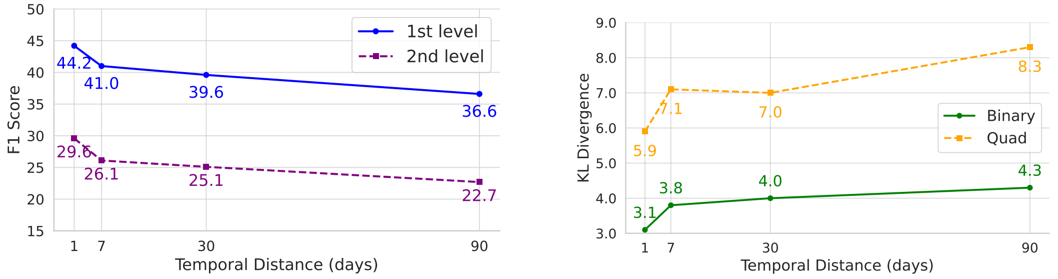


Figure 6: Evaluation of LLM Agents in different temporal distances of the forecasting event.

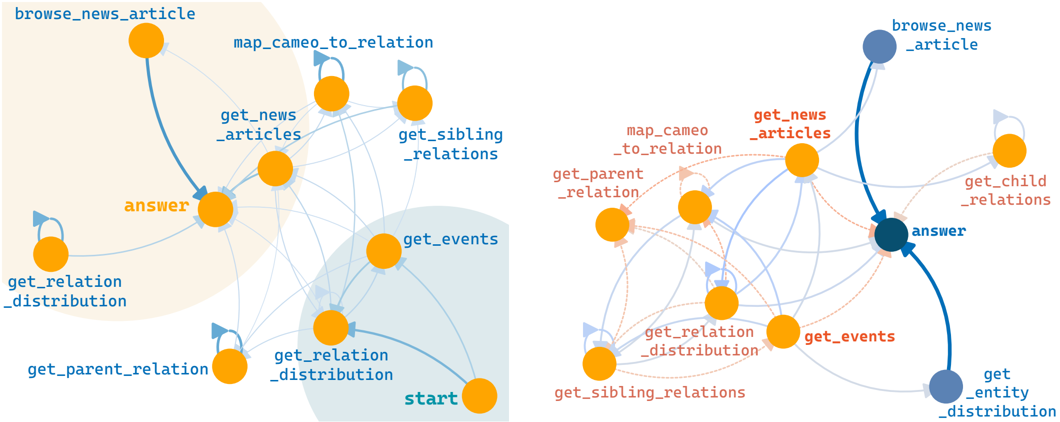


Figure 7: **Action order analysis in LLM agents.** a) Tool-Use Transition Graph of called API functions. Edge thickness indicates transition frequency. b) Freq.(correct) - Freq.(incorrect), in which (blue / red) edges indicate (positive / negative) contributions.

better on verbal cooperation" and material conflict" than on other types. This stems from verbal cooperation" events being more frequent, aiding historical recall, and material conflicts" exhibiting consistent, prolonged patterns among specific countries. In contrast, "material cooperation" and "verbal conflicts"—like 057: `Sign formal agreement` or 084: `Return or release`—are abrupt and unpredictable, requiring nuanced trend analysis and context, resulting in lower accuracy. This underscores the need for LLMs to grasp the subtleties of diverse event types.

How tool-use ordering influences forecasting. We examine the effect of action sequence on an agent in "Single Function" mode. Figure 7 depicts a transition graph from query to correct answer, with thicker edges showing frequent paths. Typically, the agent starts with `get_relation_distribution` or `get_event` for recent event data, often ending with `browse_news_article` and `get_news_articles` for news-based forecasts. To assess each function's impact, Figure 7b subtracts incorrect prediction frequencies from correct ones: blue edges mark paths to accurate outcomes, red edges highlight error-prone ones. Functions like `browse_news_article` and `get_entity_distribution` frequently yield correct answers, while `get_news_articles`—linked directly to answers in red—often errs due to vague titles. Pairing it with `browse_news_article` boosts accuracy. Similarly, `get_event` shifts from negative to positive when followed by `get_entity_distribution`. Figure 9a shows `get_child/sibling_relation` excel in initial predictions. These findings highlight the need for strategic tool sequencing in LLM agents for effective temporal forecasting..

Can we make a small LM stronger via inference-time scaling? Larger LLMs typically excel in agent performance, but can a weaker LLM match them using inference-time computation? We test this with `Mistral-7B-Instruct-v0.2`, employing ReAct in "Single Function" mode. For each query, we sample multiple times at a temperature of 0.4, applying a self-consistency variant that retains entries appearing more than twice and calculating F1 (Max@K)—the highest F1 score per instance across rounds. Figure 8a shows that performance rises with more samples: a single sample yields an F1 of 9.3, far below larger models, but by the 10th sample, F1 (Max@K) hits 38.1, surpassing GPT-4o-mini. This demonstrates that inference-time methods self-consistency can significantly enhance smaller LMs for event forecasting and potential to improve LLM Agents via Reinforcement Learning.

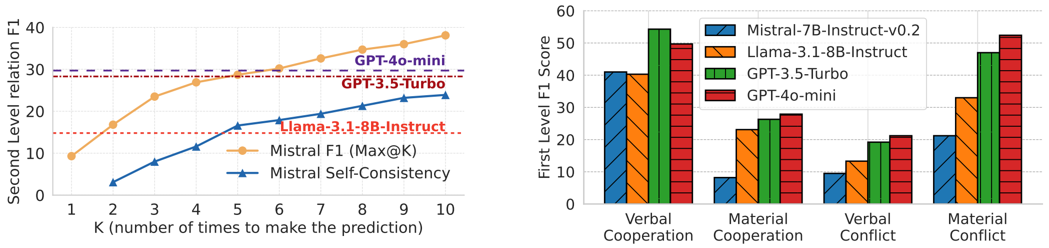


Figure 8: a) Mistral-7B-Instruct performance increases with more inference-time compute. b) F1 scores of different LLMs on forecasting by event type's quadratic category.

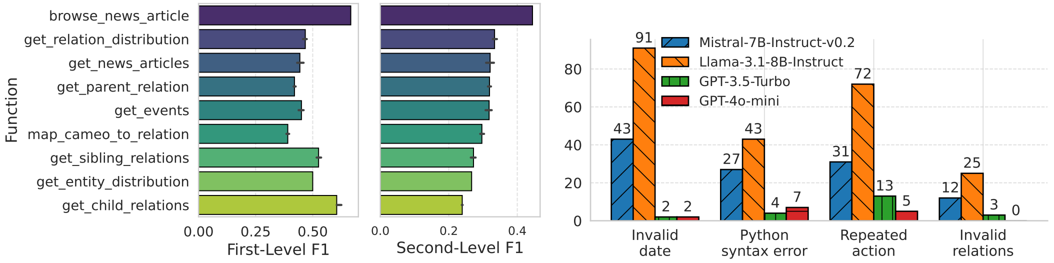


Figure 9: a) F1 Accuracy for each API function. b) Code execution error analysis.

Code execution error analysis. Our agents use code to interact with tools but frequently face execution errors. Figure 9b outlines the primary error types across LLMs. Smaller models most often encounter invalid date errors, struggling to grasp time constraints on historical data (set before the query event date). Even larger models repeat actions from prior ReAct steps, hindering effective reasoning. Llama-3.1-8B-Instruct exhibits more errors than Mistral-7B-Instruct-v0.2 yet outperforms it in forecasting. GPT-4o-mini, however, shows far fewer errors, leveraging superior code generation to enhance its event forecasting performance.

4 Related Work

Recent benchmarks for temporal reasoning in AI have two general directions: temporal understanding and temporal forecasting. Temporal understanding benchmarks [Jia et al., 2018, Saxena et al., 2021, Mavromatis et al., 2021, Tan et al., 2023a, Wang and Zhao, 2024] evaluate models' grasp of temporal relations in existing data, while temporal forecasting benchmarks, like our MIRAI, predict future events from historical data. Existing forecasting benchmarks adopt either QA [Jin et al., 2021b, Zou et al., 2022, Zhang et al., 2024, Halawi et al., 2024, Schoenegger et al., 2024] or link prediction [Boschee et al., 2015b, Leetaru and Schrodt, 2013a] formats, with QA relying on text and link prediction on temporal knowledge graphs (TKGs). MIRAI stands out by integrating diverse data sources, using a multi-relation prediction task, and introducing an agent-based approach with intermediate reasoning and a pipeline for dynamic updates, highlited in Table 4 in Appendix C.

5 Conclusion and Limitation

we present MIRAI, a new benchmark for assessing LLM agents in temporal forecasting of international events. Our main contributions are: 1)An agentic environment with APIs enabling thorough evaluation of diverse information sourcing, code-based tool use, and forecasting reasoning. 2) A dynamic data pipeline for monthly updates, ensuring contamination-free test splits for new models. 3) Comprehensive benchmarking across agent methods, prediction horizons, and test splits, with detailed analysis of factors affecting performance.

Our findings expose LLM agents' struggles with generating accurate code and handling complex temporal reasoning, pointing to significant research opportunities. By offering a standardized, adaptable, and robust evaluation platform, MIRAI seeks to advance the creation of reliable forecasting models for informed decision-making in international relations. Despite addressing key challenges, limitations persist, including limited model coverage, API functionality, and data diversity. See Appendix B for a full discussion.

References

Thomas Brown and Susan Lee. Predictive analytics in economic sanctions and international policy. *Journal of International Economics*, 26(4):311–330, 2018.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. January 2015. CIDR 2015.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs, October 2020a. EMNLP 2020.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning, April 2021a. SIGIR 2021.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. Forecasting Future World Events with Neural Networks. arXiv, October 2022. NeurIPS 2022.

Revanth Gangi Reddy, Yi R. Fung, Qi Zeng, Manling Li, Ziqi Wang, Paul Sullivan, and Heng Ji. SmartBook: AI-Assisted Situation Report Generation, March 2023. arXiv.

Introducing deep research, 2024. URL `https://openai.com/index/introducing-deep-research/`.

Gemini deep research, 2024. URL `https://gemini.google/overview/deep-research/?hl=en`.

Introducing perplexity deep research, 2025. URL `https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research`.

Grok 3 beta — the age of reasoning agents, 2025. URL `https://x.ai/news/grok-3`.

Kalev Leetaru and Philip A Schrodt. GDELT: Global Data on Events, Location and Tone,. 2013a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. Cameo.cdb.09b5.pdf. In *ICEWS Coded Event Data*. Harvard Dataverse, 2015a.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. OBELICS: an open web-scale filtered dataset of interleaved image-text documents. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets, December 2021. URL `http://arxiv.org/abs/1803.09010`. arXiv:1803.09010 [cs].

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models, March 2023b. arXiv:2210.03629 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Meta. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL `https://doi.org/10.48550/arXiv.2407.21783`.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

GPT-3.5-Turbo, `https://platform.openai.com/docs/models/gpt-3-5-turbo`, 2023.

Gpt-4o contributions. 2024.

Julia Gastinger, Christian Meilicke, Federico Errica, Timo Sztyler, Anett Schuelke, and Heiner Stuckenschmidt. History repeats Itself: A Baseline for Temporal Knowledge Graph Forecasting, April 2024. URL http://arxiv.org/abs/2404.16726.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *SIGIR*, pages 408–417. ACM, 2021b.

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. Forecastqa: A question answering challenge for event forecasting with temporal text data. In *ACL/IJCNLP (1)*, pages 4636–4650. Association for Computational Linguistics, 2021a.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023. arXiv:2201.11903 [cs].

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. Temporal knowledge graph forecasting without knowledge using in-context learning. In *EMNLP*, pages 544–557. Association for Computational Linguistics, 2023.

Zhihan Zhang, Yixin Cao, Chenchen Ye, Yunshan Ma, Lizi Liao, and Tat-Seng Chua. Analyzing Temporal Complex Events with Large Language Models? A Benchmark towards Temporal, Long Context Understanding, June 2024. arXiv:2406.02472 [cs].

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. TempQuestions: A Benchmark for Temporal Question Answering. In *Companion Proceedings of the The Web Conference 2018*, WWW ’18, pages 1057–1062, Republic and Canton of Geneva, CHE, April 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5640-4. WWW 2018.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question Answering Over Temporal Knowledge Graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676, Online, August 2021. Association for Computational Linguistics. ACL 2021.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N. Ioannidis, Soji Adeshina, Phillip R. Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. TempoQR: Temporal Question Reasoning over Knowledge Graphs. arXiv, December 2021. AAAI 2022.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada, July 2023a. Association for Computational Linguistics. ACL 2023.

Yuqing Wang and Yun Zhao. TRAM: Benchmarking Temporal Reasoning for Large Language Models, May 2024. arXiv:2310.00835 [cs].

Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan, and Xiang Ren. ForecastQA: A Question Answering Challenge for Event Forecasting with Temporal Text Data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4636–4650, Online, August 2021b. Association for Computational Linguistics. ACL 2021.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. Approaching Human-Level Forecasting with Language Models, February 2024. URL http://arxiv.org/abs/2402.18563. arXiv:2402.18563 [cs].

Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, and Philip E. Tetlock. Wisdom of the Silicon Crowd: LLM Ensemble Prediction Capabilities Rival Human Crowd Accuracy, May 2024. URL http://arxiv.org/abs/2402.19379. arXiv:2402.19379 [cs].

Elizabeth Boschee, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data, 2015b.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. TORQUE: A Reading Comprehension Dataset of Temporal Ordering Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online, November 2020a. Association for Computational Linguistics. EMNLP 2020.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip E. Tetlock. ForecastBench: A Dynamic Benchmark of AI Forecasting Capabilities, September 2024. URL `http://arxiv.org/abs/2409.19839`. arXiv:2409.19839 [cs].

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In *ACL*, pages 14820–14835. Association for Computational Linguistics, 2023b.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. TORQUE: A reading comprehension dataset of temporal ordering questions. In *EMNLP*, pages 1158–1172, 2020b.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *EMNLP*, pages 3363–3369, 2019.

Michael Zhang and Eunsol Choi. Situatedqa: Incorporating extra-linguistic contexts into qa. In *EMNLP*, 2021.

Yuqing Wang and Yun Zhao. Tram: Benchmarking temporal reasoning for large language models. 2023.

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

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *EMNLP (1)*, pages 6669–6683. Association for Computational Linguistics, 2020b.

Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *WSDM*, pages 794–803. ACM, 2022.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI Conference on Artificial Intelligence*, 2020.

Haohai Sun, Jialu Zhong, Yunpu Ma, Zhen Han, and Kun He. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *EMNLP*, 2021.

Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *ACL*, 2021c.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In *ICML*, page 3462–3471, 2017.

Zifeng Ding, Zhen Han, Yunpu Ma, and Volker Tresp. Temporal knowledge graph forecasting with neural ode. abs/2101.05151, 2021.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *KDD*, pages 1585–1595. ACM, 2020.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Understanding event predictions via contextualized multilevel feature learning. In *CIKM*, pages 342–351. ACM, 2021.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, and Tat-Seng Chua. Context-aware event forecasting via graph disentanglement. In *KDD*, pages 1643–1652. ACM, 2023a.

Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, Liang Pang, and Tat-Seng Chua. Structured, complex and time-complete temporal event forecasting. *CoRR*, abs/2312.01052, 2023b.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. Pre-trained language model with prompts for temporal knowledge graph completion. In *ACL (Findings)*, pages 7790–7803. Association for Computational Linguistics, 2023a.

Ruotong Liao, Xu Jia, Yunpu Ma, and Volker Tresp. Gentkg: Generative forecasting on temporal knowledge graph. *CoRR*, abs/2310.07793, 2023.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, JUN ZHOU, Chenhao Tan, and Hongyuan Mei. Language models can improve event prediction by few-shot abductive reasoning. In *NeurIPS*, 2023.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of EMNLP*, 2015.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of NAACL*, 2021.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of EMNLP*, 2023a.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. On the tool manipulation capability of open-source large language models. *arXiv preprint arXiv:2305.16504*, 2023b.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. ToolQA: A dataset for LLM question answering with external tools. In *Proceedings of NeurIPS*, 2023.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *Proceedings of NeurIPS*, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. In *Proceedings of ICLR*, 2024.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *Proceedings of ICLR*, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A Benchmark for Real-World Planning with Language Agents, February 2024. arXiv:2402.01622 [cs].

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html`.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. Augmented language models: a survey. In *arXiv preprint arXiv:2302.07842*, 2023.

Aaron Parisi, Yao Zhao, and Noah Fiedel. Talm: Tool augmented language models. In *arXiv preprint arXiv:2205.12255*, 2022.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Proceedings of NeurIPS*, 2023.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Proceedings of NeurIPS*, 2023.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. AVIS: autonomous visual information seeking with large language model agent. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/029df12a9363313c3e41047844ecad94-Abstract-Conference.html`.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html`.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. *CoRR*, abs/2307.16789, 2023. doi: 10.48550/ARXIV.2307.16789. URL `https://doi.org/10.48550/arXiv.2307.16789`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8696–8708, 2021.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023b.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf`.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. xERTE: Explainable Reasoning on Temporal Knowledge Graphs for Forecasting Future Links, April 2021. ICLR 2021.

Appendix

A Reproducibility Statement 16

B Limitations 16

C Additional Related Work 16
- C.1 Temporal Reasoning Benchmarks . . . 16
- C.2 Temporal Event Forecasting Benchmarks and Methods . . . 17
- C.3 Evaluation of Language Agents . . . 18
- C.4 LLMs for Tool-Use . . . 18
- C.5 LLMs for Code Generation . . . 18

D Additional Experimental Results and Analysis 19
- D.1 Data Standardization and Cleaning Pipeline . . . 19
- D.2 Human Forecasting Performance and Dataset Evaluation . . . 19
- D.3 Analysis with TKG and NLP Forecasting Methods . . . 19
- D.4 Analysis with Heuristic-based and TKG methods . . . 20
  - D.4.1 Experimental Setup . . . 20
  - D.4.2 Metric Selection and Justification . . . 21
  - D.4.3 Results Analysis and Implications . . . 21
- D.5 Analysis with Retrieve-Augmented Generation (RAG) Methods . . . 22
  - D.5.1 Methods and Experimental Setup . . . 22
  - D.5.2 Comparison of Retrieval Strategies with Agent . . . 23
  - D.5.3 Key Findings and Analysis . . . 23
- D.6 Forecasting Performance on Different Test Splits . . . 24

E Additional Information about API 24

F Additional Forecasting Examples of LLM Agent 26
- F.1 GPT-4o-mini-based Agent with ReAct and "Single Function" Action . . . 26
- F.2 GPT-4o-mini-based Agent with ReAct and "Code Block" Action . . . 29
- F.3 Deepseek-r1-distill-llama-8b-based Agent with Reasoning and Function Call . . . 32

G Datasheet for MIRAI 36
- G.1 Motivation . . . 36
- G.2 Composition . . . 37
- G.3 Collection Process . . . 37
- G.4 Uses . . . 38
- G.5 Distribution . . . 38
- G.6 Maintenance . . . 39

G.7 Term of Use for GDELT 39

A Reproducibility Statement

We provide detailed information and explanation of our experimental setup, dataset, and evaluation metrics, ensuring reproducibility. Specifically, we describe the following:

- Experimental setup (Sec. 3): Detailed descriptions of the hardware and software configurations, including libraries and tools used.
- Dataset (Sec. 2: Comprehensive information on the dataset construction, including sources, preprocessing steps, and any techniques applied.
- Evaluation Metrics (Sec. 2.2): Clear definitions for the metrics chosen to evaluate performance.

Additionally, we commit to releasing the following resources for the replication of our results:

- Codebase: The complete codebase, including scripts for dataset construction, model serving, and evaluation. This is currently available on an anonymous repository here.
- Dataset: The processed dataset, along with detailed instructions on how to construct the dataset here.

By providing these resources, we aim to ensure that our work is fully reproducible and can be independently verified by the research community.

B Limitations

Our benchmark addresses several key challenges in evaluating LLM agents for event forecasting, including the integration of diverse information sources, the construction of an agentic interactive environment, and the contamination-free forecasting data and task formulation through its dynamic design. Despite this significant advantage, we acknowledge the following limitations:

1. **Model Coverage:** While we have tested representative open-source and closed-source LLMs, our experiments do not exhaustively cover all available models. Future work could expand to include a wider range of LLMs for more comprehensive evaluation.
2. **API Functionality:** The current API, while functional, has room for expansion. Future iterations could incorporate more sophisticated analytical tools, such as time series analysis functions, to encourage deeper temporal reasoning. Allowing agents to generate and add custom functions during their reasoning process could also lead to more diverse problem-solving approaches.
3. **Experimental Robustness:** The current experiments, while informative, are limited in scope due to cost and time constraints. Increasing the number of experimental rounds and adjusting parameters like model temperature could provide more statistically robust results and insights into model performance variability.
4. **Geopolitical Bias:** Reliance on GDELT as the primary data source may introduce biases in event coverage and interpretation, potentially skewing towards Western or English-language media perspectives. Future work could explore integrating multiple diverse data sources and languages to mitigate this limitation.

Addressing these limitations in future iterations will further enhance the benchmark's robustness and relevance in the rapidly evolving field of AI-driven event forecasting.

C Additional Related Work

C.1 Temporal Reasoning Benchmarks

Many benchmarks sensing the temporal reasoning ability of AI models have been constructed, but they have different focuses and settings with MIRAI, particularly in terms of task, information, and method, as shown in Table 4. One line of benchmarks focuses on the *temporal understanding* ability of the model [Jia et al., 2018, Saxena et al., 2021, Mavromatis et al., 2021, Ning et al., 2020a, Tan et al., 2023a, Wang and Zhao, 2024], such as understanding the temporal relations between available facts in knowledge graphs (KGs) or text, either a short piece of text or a document corpus. While the *temporal forecasting* task largely differs from understanding, where the reasoning target is **unseen**

Table 4: **Comparison of MIRAI with other temporal reasoning benchmarks.** "Method" refers to the methodology of original and recent models evaluated on the benchmark. Column "Underst." stands for Understanding. ICL stands for LLM w/ In-Context Learning. FT stands for fine-tuning.

Benchmark	Temporal Reasoning		Task Format	Information				Method
	Underst.	Forecast		Time Series	KG	Textual	API	
TempQuestions [Jia et al., 2018]	✓		KGQA		✓			KGQA Systems
CRONQuestions [Saxena et al., 2021]	✓		KGQA		✓			Bert-based FT
TempoQR [Mavromatis et al., 2021]	✓		KGQA		✓			Bert-based FT
TORQUE [Ning et al., 2020a]	✓		QA			✓		Bert-based FT
TempReason [Tan et al., 2023a]	✓		QA		✓	✓		ICL + Task FT
TRAM [Wang and Zhao, 2024]	✓		MCQ			✓		ICL + Bert-based FT
TCELongBench [Zhang et al., 2024]	✓	✓	QA/MCQ			✓		ICL
ForecastQA [Jin et al., 2021b]		✓	MCQ			✓		Bert-based FT
IntervalQA [Zou et al., 2022]		✓	QA/MCQ	✓		✓		ICL
Approach [Halawi et al., 2024]		✓	QA			✓		ICL
ForecastBench [Karger et al., 2024]		✓	QA			✓		ICL
GDELT [Leetaru and Schrodt, 2013a]		✓	Link Prediction		✓			Graph FT + ICL + Task FT
ICEWS [Boschee et al., 2015b]		✓	Link Prediction		✓			Graph FT + ICL + Task FT
MIRAI		✓	Relation List		✓	✓	✓	LLM Agent

in the database for the model, and as such, the model has to not only understand but to reason. For forecasting, there are two main task formulations among previous benchmarks: the QA task format for benchmarks with history information represented in textual format [Zou et al., 2022, Zhang et al., 2024, Jin et al., 2021b, Halawi et al., 2024, Schoenegger et al., 2024, Karger et al., 2024], and graph link prediction task format for temporal knowledge graph (TKG)-based benchmarks [Boschee et al., 2015b, Leetaru and Schrodt, 2013a]. However, the uniformat of information sources either lacks of support to clearly structural facts or contextual detail for the model to perform advanced reasoning, while in MIRAI, we provide both information sources. Additionally, we provide carefully constructed API with various data classes and functions that access to various part of the data. With the flexibility provided by code generation, the model is exposed to a broader and more flexible range of information. More importantly, MIRAI distinguishes itself by introducing an agentic environment specifically designed to evaluate LLM agents in the forecasting task. This represents a significant departure from previous work, which has not explored or even considered the potential of performing temporal forecasting tasks using LLM agents.

C.2 Temporal Event Forecasting Benchmarks and Methods

Existing Forecasting Benchmarks. LLMs have been tested for their *temporal understanding* through tasks such as temporal event ordering or storyline comprehension [Tan et al., 2023b, Ning et al., 2020b, Zhou et al., 2019, Zhang and Choi, 2021, Wang and Zhao, 2023]. In the context of *temporal forecasting*, LLMs have been evaluated on traditional structured-event-only benchmarks, such as ICEWS (2014) [Boschee et al., 2015b] and GDELT (2018) [Leetaru and Schrodt, 2013b]; and also been evaluated on recent text-based temporal forecasting benchmarks, such as IntervalQA (2022) [Zou et al., 2022] and TCELongBench (2022) [Zhang et al., 2024]. However, these evaluations typically involve providing LLMs with retrieved-context for in-context learning and then directly answering the forecast question, lacking intermediate reasoning steps and interaction between the LLM and the database crucial for accurate forecasting. Moreover, although experimental results on these benchmarks show that significant challenges and research value remain in forecasting, even when models encounter events before their training cutoff date, these benchmarks still only provide a fixed timeframe for the testing data, earlier than most recent LLMs.

Traditional TKG and NLP Methods. Significant research has been conducted in the field of structured event temporal forecasting. Various methods have been proposed, including aggregating temporal and relational information among entities [Jin et al., 2020b, Li et al., 2021b, Park et al., 2022], retrieving relevant historical events [Zhu et al., 2020, Sun et al., 2021, Li et al., 2021c], and modeling the continuous time evolution of events [Trivedi et al., 2017, Ding et al., 2021]. Efforts have also been made to incorporate textual event information into Temporal Knowledge Graphs (TKGs). Glean [Deng et al., 2020] and CMF [Deng et al., 2021] integrate textual embeddings into graph edges, while SeCoGD [Ma et al., 2023a] employs textual topic modeling to separate subgraphs. The MidEast-TE dataset and LoGo model [Ma et al., 2023b] utilize text clustering to construct complex events for forecasting with local and global contexts. However, these methods still perform forecast reasoning solely on graphs using graph-based techniques. Traditional NLP methods form the event forecasting task as MCQ, for example, the method in ForecastQA [Jin et al., 2021a] use text embedding models for retrieving related event news articles and appending them to the forecasting question for a Bert-based classification over answer candidates. Notably, MIRAI contains both structured and textual event data, supporting the test for both traditional TKG and NLP methods. We show more experimental results and illustrations in Appendix D.3.

LLMs for TKG and NLP Methods. Recent studies have explored the use of LLMs for temporal event forecasting by transforming the TKG formulation into text sequences and converting missing object prediction into next token prediction [Xu et al., 2023a]. GPT-NeoX-ICL [Lee et al., 2023] employs in-context learning of LLMs and constructs prompts as a list of historical events in quadruplet format. GENTKG [Liao et al., 2023] enhances the selection of historical event inputs using a temporal logical rule-based retrieval strategy, while LAMP [Shi et al., 2023] applies LLMs to perform abductive reasoning to assist the retrieval process. However, these works only investigate LLMs with in-context learning or simple task-specific fine-tuning. In contrast, MIRAI explores forecasting with an LLM agent that supports explicit information gathering and reasoning steps, enabling a hybrid approach that leverages both text and graph data.

C.3 Evaluation of Language Agents

Previous research has investigated the performance of LLM agents in a variety of domains, including arithmetic reasoning focused on obtaining correct solutions [Roy and Roth, 2015, Cobbe et al., 2021, Patel et al., 2021], proficiency assessment in utilizing tools and reporting results [Li et al., 2023a, Xu et al., 2023b, Zhuang et al., 2023], evaluation of web navigation skills to find specific websites [Deng et al., 2023, Zhou et al., 2024, Liu et al., 2024], and planning travel itineraries under given constraints [Xie et al., 2024]. However, these evaluations do not fully address the challenges posed by tasks involving complex international events with diverse information formats and temporal attributes. MIRAI presents a unique task in this context, where the agent must navigate and reason over the structured events and textual news articles with temporal information. This setup requires the agent to effectively handle multilateral relationships and information spanning different time periods.

C.4 LLMs for Tool-Use

Large Language Models (LLMs) have demonstrated remarkable language understanding [Radford et al., 2018] and reasoning capabilities [Wei et al., 2022]. However, they also possess inherent limitations, such as their inability to provide up-to-date responses based on external knowledge or to perform complex mathematical reasoning. In response to these challenges, recent advancements have seen the integration of LLMs with various external tools [Mialon et al., 2023]. Notable examples include TALM [Parisi et al., 2022] and ToolFormer [Schick et al., 2023], which utilize in-context learning to enhance the model's ability to leverage different tools in tasks like question answering and mathematical reasoning. Chameleon [Lu et al., 2023] employs an LLM as a natural language planner to deduce the optimal sequence of tools to be used, subsequently executing these tools to generate the final output. AVIS [Hu et al., 2023] employs dynamic tree search to synthesize the most effective tool-use sequence. ToolkenGPT [Hao et al., 2023] integrates tool-use operators as special tokens and trains the model through sequence-to-sequence training. ToolLLM [Qin et al., 2023] introduces an instruction tuning dataset encompassing over 16,000 real-world APIs, significantly enhancing the model's capability to utilize these tools effectively. These features, summarized in Table 4, position MIRAI as a comprehensive and unique benchmark for evaluating temporal forecasting capabilities.

C.5 LLMs for Code Generation

Early advancements in LLMs have greatly impacted the field of code generation. Notable early models such as GPT-3 [Brown et al., 2020] and PALM [Chowdhery et al., 2023] have demonstrated the potential of LLMs to assist in code completion and generation. Subsequent models have built upon the foundation of pre-trained LLMs, further refining capabilities specific to code generation. This has led to the development of code-specific LLMs such as Codex [Chen et al., 2021], Code T5 [Wang et al., 2021], CodeGen [Nijkamp et al., 2022], AlphaCode [Li et al., 2022], StarCoder [Li et al., 2023b], WizardCoder [Luo et al., 2023], and phi-1 [Gunasekar et al., 2023]. Moreover, code from open-source platforms such as GitHub has increasingly been incorporated into the pre-training data for recent LLMs [Touvron et al., 2023, Achiam et al., 2023]. This integration has led to improved performance of LLMs such as Code Llama [Roziere et al., 2023], Code-Qwen [Bai et al., 2023], and DeepSeek-Coder [Guo et al., 2024] on popular code-related tasks [Chen et al., 2021, Austin et al., 2021].

D Additional Experimental Results and Analysis

D.1 Data Standardization and Cleaning Pipeline

We refine the structured event data through four key steps: (1) *Textual context extraction*: News titles and content are sourced from the corresponding URLs of each event, retaining only textual information. (2) *Event actor standardization*: Country codes are standardized to ISO-3166 norms, excluding missing or outdated codes to ensure consistency and eliminate ambiguity. (3) *Event scope delimitation*: We filter out all domestic events where the subject is identical to the object at the country level, focusing exclusively on international events to maintain relevance to our research objectives. (4) *Event relation standardization*: Event codes are standardized to the second level of the CAMEO ontology, balancing specificity with consistency and reliability. Third-level relations are omitted due to inconsistent hierarchical depth and increased risk of extraction errors at this granular level.

D.2 Human Forecasting Performance and Dataset Evaluation

To establish a reference point for our LLM agent evaluations, we conducted a human forecasting task with the following parameters:

- **Scope:** 10 queries $(t, s, ?, o)$ covering 51 distinct events.
- **Participants:** 2 college students (non-political science majors).
- **Task:** Given an event query $(t + 1, s, ?, o)$, generate forecasts for all possible relations.
- **APIs:** Participants used the same API library as the LLM agents.
- **Evaluation:** Applied the same metrics as those used for LLM agents.

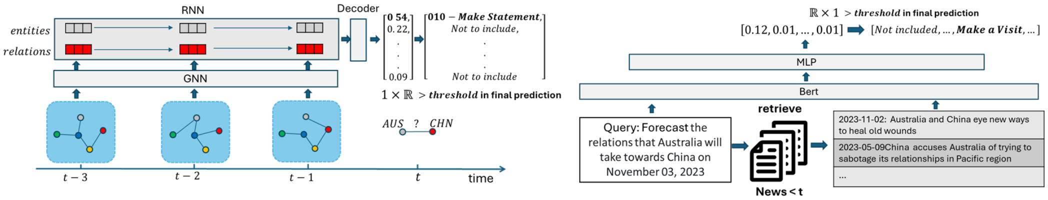
As shown in Table 2 and Table 1, compared to different forecasting methods and different LLM base models, human performance surpassed that of LLM agents in most metrics, especially in recall. This highlights significant room for improvement in LLM performance.

It is important to note that our evaluators do not specialize in political science and are likely to fall short of what domain experts could achieve. This suggests that human forecasting, even at current performance levels, has room for improvement.

We also acknowledge that event forecasting is an inherently challenging task for both humans and AI. The superior performance of non-expert human evaluators underscores the complexity of this task and the potential for further advancements in both human and AI forecasting capabilities.

To further assess the quality of the data set, we performed a human evaluation on this subset of tests. Two human annotators evaluated whether the ground-truth events were correctly extracted based on their source news articles, scoring each event as 0 (incorrect) or 1 (correct). The average score across all evaluated events is 0.82, indicating high accuracy in the sampled test events. This evaluation confirms the reliability of our data cleaning process and the overall quality of our dataset.

D.3 Analysis with TKG and NLP Forecasting Methods


RNN
Decoder
entities
relations
GNN
$\begin{bmatrix}0.54,\\0.22,\\ \vdots \\0.09\end{bmatrix}$
$\begin{bmatrix}010 - \textit{Make Statement},\\ \textit{Not to include}, \\ \vdots \\ \textit{Not to include}\end{bmatrix}$
$1 \times \mathbb{R} > \textit{threshold}$ in final prediction
AUS ? CHN
$t-3$
$t-2$
$t-1$
t
time
$\mathbb{R} \times 1 > \textit{threshold}$ in final prediction
$[0.12, 0.01, \ldots, 0.01]$
$[\textit{Not included}, \ldots, \textit{Make a Visit}, \ldots]$
MLP
Bert
retrieve
Query: Forecast the relations that Australia will take towards China on November 03, 2023
News < t
2023-11-02: Australia and China eye new ways to heal old wounds
2023-05-09China accuses Australia of trying to sabotage its relationships in Pacific region
...

(a) REGCN (TKG)

(b) ForecastQA (NLP)

Figure 10: Illustration of RE-GCN and the adapted ForecastQA aggregate historical information and output forecasting probability (0-1) for each relation type. In comparison, examples using LLM agent in Appendix F provide explicit reasoning traces with better interpretability.

We include the following additional traditional baselines that are fine-tuned:

- **Structured TKG baseline**, we implemented RE-GCN [Li et al., 2021b], which learns relation in their graph edge embeddings and supports relation prediction.
- **Traditional textual baseline**, we adopted ForecastQA [Jin et al., 2021a], implementing a Bert-based classification model with BM25 to retrieve relevant news articles as additional encoder input.

Figure 10 illustrates the proposed frameworks of REGCN and ForecastQA, along with their threshold-based prediction processes. REGCN learns entity and relation representations in the event knowledge graph at each timestamp, modeling temporal development through a recurrent neural network. The learned representations are then used by a decoder to compute probabilities for each relation type. In contrast, ForecastQA employs a BERT encoder to process both the original query and retrieved news articles, utilizing a classification network for relation type probability computation.

We trained these traditional baselines on data prior to 2023-12 and evaluated them on the 2024-02 test split. Optimal thresholds were determined through a search over 0.3, 0.5, 0.7, and 0.9, resulting in 0.5 for REGCN and 0.3 for ForecastQA. The experimental results of these traditional models are shown in Table 2, along with LLMs' performance.

REGCN demonstrates superior performance in predicting higher-level relation types, particularly at the binary and quadratic levels, suggesting a good level of relation representation learning ability. However, it struggles with accurate forecasting at the more granular second level. ForecastQA achieves higher precision but significantly lower recall compared to REGCN, due to only predicting the relation type "Make a Visit" in most queries.

Notably, data-specific fine-tuning allows these traditional methods to achieve competitive performance with zero-shot smaller LLM agents, though they still largely underperform the most advanced LLMs. Overall, these experiments and findings highlight MIRAI's value as a comprehensive benchmark for comparing diverse forecasting methods.

D.4 Analysis with Heuristic-based and TKG methods

Table 5: Evaluation results on the 2024-02 test split for relation prediction using heuristic-based and TKG-based methods and LLM agents based on GPT-4o-mini. The best-performing score is highlighted in **bold** and the second-best is underlined.

Method	Training Data Cutoff Date	Prompt	MRR (%)(⇑)	Hit@10 (%)(⇑)	Binary KL (⇓)	Quad KL (⇓)	First-level Relation (%)			Second-level Relation (%)		
							Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)
RE-GCN	2023-06	—	1.6	2.2	0.4	0.8	24.4	90.6	34.3	4.4	**83.9**	7.9
	2023-08		1.9	2.8	0.4	1.1	23.9	86.1	32.9	4.6	40.0	7.0
	2023-10		1.7	2.5	**0.3**	1.0	24.8	78.2	32.4	3.9	25.7	5.6
	2023-12		2.9	5.7	**0.3**	2.5	23.9	74.4	31.3	5.5	28.4	7.9
Recurrency (Strict)	2023-06	—	17.4	45.0	3.2	3.6	32.8	77.1	42.9	18.7	67.8	27.2
	2023-08		17.1	**45.3**	3.2	3.6	32.3	78.2	42.7	18.0	69.9	26.9
	2023-10		15.8	41.0	2.4	3.1	29.7	83.5	41.3	14.3	76.8	23.0
	2023-12		**17.8**	43.2	2.1	2.5	29.8	86.0	41.6	14.2	80.1	23.0
ReAct	2023-10	**Set Prediction**	—	—	3.6	8.0	**61.7**	38.6	40.7	**46.3**	32.9	31.1
		Rank (k=10)	—	25.7	0.6	1.4	47.5	70.2	**48.9**	38.1	61.8	**38.2**
		Rank (k=30)	—	12.0	**0.3**	0.8	34.9	**91.2**	45.8	22.5	82.8	31.7
		Rank (all)	13.9	14.1	2.1	2.8	27.0	86.2	37.9	12.5	81.4	20.2
		Rank w.Prob (k=10)	—	26.8	1.1	2.5	47.3	67.7	48.3	37.9	59.2	**38.2**
		Rank w.Prob (k=30)	—	10.8	**0.3**	**0.6**	34.8	86.6	45.3	22.2	76.4	31.0
		Rank w.Prob (all)	12.6	14.9	2.4	2.7	28.5	83.0	38.3	12.7	78.6	20.6

D.4.1 Experimental Setup

We evaluate three approaches in detail:

RE-GCN [Li et al., 2021b]:

- **Model Architecture**: we follow the original method, where the model combines relation-aware graph convolutional layers with recurrent neural networks to jointly model structural dependencies and temporal dynamics in TKGs.
- **Data Cutoffs**: We experiment with four cutoff times (2023-06, 2023-08, 2023-10, 2023-12). For example, with 2023-10 cutoff, the model trains on data until 2023-10, validates on 2023-11, and tests on the 2024-02 split.
- **Test Input**: Uses single-step prediction with a 7-day historical window (t-7 to t-1), aligning with our agent experiments' one-day forecasting horizon.

Recurrency (Strict) [Gastinger et al., 2024]:

- **Model**: The original work introduces three baselines that place strong inductive bias on fact recurrence over time: strict recurrency, relaxed recurrency, and their combination. While the original work and its scoring functions are specifically designed for link prediction, we adapt the strict recurrency variant for relation prediction. For a query event $(s, ?, o, t)$, we compute scores for all relations $r \in \mathcal{R}$ using:

$$\phi_{\Delta}((s,r,o,t),G) = \begin{cases} \Delta(t, \max\{k \mid (s,r,o,k) \in G\}) & \text{if } \exists k \text{ with } (s,r,o,k) \in G \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\Delta(t,k) = k/t$ measures temporal proximity.

- **Data Access**: Cutoff dates determine available historical events, e.g., 2023-10 means using only events before 2023-10-31 for score computation.

ReAct Agent:

- **Set Prediction**: Uses original prompt and output format, with the output directly used for set prediction metrics.
- **Ranking**: Modified prompts for ordered relation lists (k=10, 30, all). Only rank=all configurations are used for MRR calculation, while all configurations support Hit@10 evaluation.
- **Probabilistic Ranking**: Additional prompts requesting probability scores for ranked relations.
- **Evaluation**: All ranking evaluations use time-aware filtering to maintain consistency with other methods.

D.4.2 Metric Selection and Justification

Set-based Metrics: For set prediction tasks, models output a discrete set of predicted relations for each query. We evaluate these predictions using:

- Precision: Proportion of predicted relations that are correct
- Recall: Proportion of actual relations that are predicted
- F1 Score: Harmonic mean of precision and recall

Ranking-based Metrics: For ranking tasks, models output an ordered list of relations with associated scores. Following TKG conventions of time-aware filtering [Bordes et al., 2013, Han et al., 2021], we implement:

- Mean Reciprocal Rank (MRR): Average reciprocal of the first correct relation's rank
- Hit@10: Proportion of queries where at least one correct relation appears in top-10 predictions

Choice of Primary Metrics: We prioritize set-based metrics as our primary evaluation criteria for Model Capability: The primary consideration is that current LLMs are better suited to generating discrete predictions through natural language reasoning than producing comprehensive ranked lists. This is evidenced by our experimental findings:

- **List Length Sensitivity**: ReAct agent's performance deteriorates with longer list requirements, with k=30 and k=all performing worse than k=10 in ranking metrics (Hit@10: 25.7% for k=10 vs 12.0% for k=30).
- **Prompt Sensitivity**: Performance varies between pure ranking and probability-weighted ranking (Hit@10: 25.7% vs 26.8% for k=10, and MRR: 13.9% vs 12.6% for k=all), suggesting that ranking outputs are sensitive to the prompt formulation and output format.

Given these challenges, we opted for metrics that more directly and reliably assess the agents' ability to predict discrete events without the confounding factors introduced by list generation and ranking.

D.4.3 Results Analysis and Implications

Our comprehensive evaluation reveals several significant insights about different forecasting approaches:

Performance of the Recurrency Model and Insights for Agents: The experiment results of the Recurrency (strict) model are shown in row 5-8 of the table. We observe that the Recurrency baseline demonstrates strong performance in ranking metrics (17.8% MRR and 43.2% Hit@10 with 2023-12 cutoff), leading other models; it also shows consistently high recall (86.0% Recall at first-level relation, and 80.1% recall at second-level relations with 2023-12 cutoff), suggesting that international events indeed often follow repetitive patterns.

Comparing with the ReAct agent (Set Prediction, the setting we used in the main paper) in row 9, we find that although the LLM agent could achieve much higher forecasting precision, it obtains much lower recall than the Recurrency baseline. We conducted a detailed analysis of this behavior by manually going through the reasoning traces generated by the LLM agent in the test set. One possible reason is that the agent has a strong tendency to select only a subset of the most frequent historical events in its prediction. For example, for the query $(2024\text{-}02\text{-}01, \mathrm{PSE}, ?, \mathrm{EGY})$, it uses function calls like `get_relation_distribution(date_range=DateRange(start_date=Date("2023-01-31"), end_date=Date("2024-01-31")), head_entities=[ISOCode("PSE")], tail_entities=[ISOCode("EGY")])`, and obtained a full frequency list as:

```
{CAMEOCode(code=’042’): 32, CAMEOCode(code=’192’): 18,
CAMEOCode(code=’040’): 13, CAMEOCode(code=’043’): 12,
CAMEOCode(code=’046’): 8, CAMEOCode(code=’080’): 6, CAMEOCode(code=’036’):
4, CAMEOCode(code=’010’): 4, CAMEOCode(code=’190’): 3,
CAMEOCode(code=’073’): 3, CAMEOCode(code=’030’): 3, CAMEOCode(code=’084’):
3, CAMEOCode(code=’020’): 3, CAMEOCode(code=’172’): 2,
CAMEOCode(code=’014’): 2, CAMEOCode(code=’070’): 2, CAMEOCode(code=’044’):
2, CAMEOCode(code=’086’): 1, CAMEOCode(code=’013’): 1,
CAMEOCode(code=’051’): 1}.
```

It then has a further step of checking recent news articles and obtains its final prediction as 040, 042, and 192, which are the top three frequent relations.

The effectiveness of simple temporal recurrency heuristics underscores the importance of incorporating more historical pattern analysis in the future development of forecasting agents, in particular, improving their recall of capturing a greater proportion of true relationships between countries.

Performance of the TKG Baselines and Insights for Agents: The experiment results of the RE-GCN are shown in rows 1-4 of the table. We observe that the RE-GCN demonstrates strong performance in high-level relation prediction, resulting in 0.3 for binary-level relation KL (cooperation or conflict) and 0.8 for quadratic-level relation KL (verbal/material cooperation/conflict), reflecting its advantage in capturing the high-level dynamics over bilateral relationships. It also shows consistently high recall in more fine-grained relation levels (90.6% Recall at first-level relation, and 83.9% recall at second-level relations with 2023-06 cutoff), suggesting its effectiveness in modeling positive correlation between query and multiple ground-truth relations.

Comparing with the ReAct agent (Set Prediction, the setting we used in the main paper) in row 9, we find that although the LLM agent also could achieve much higher forecasting precision, it obtains much lower recall than the RE-GCN baseline. We manually go through the reasoning traces generated by the LLM agent in the test set, and conclude the following possible insights and future directions of improvement compared with TKG baselines:

One major possible reason is that the current agent mostly focuses on analyzing only the bilateral events between the query entities s and o. For instance, it typically sets the function parameter `head_entities` to the query subject, and `tail_entities` to the query object only, obtaining only events and news directly between the two. However, this analysis largely oversimplifies real international relationships where countries have engaged in multi-party and complex interactions. Events between two countries could be affected by regional or global events. In contrast, TKG methods excel in capturing this multi-party and multi-relational history by leveraging multi-layer graph convolutions, where neighboring information is aggregated to enhance the modeling of each node (entity embedding) and edge (relation embedding). Therefore, when making predictions between two countries, the TKG models consider a much broader relation network than the current LLM agents, leading to higher recall and better generalization, especially when the bilateral history is sparse.

Another problem we observed from the current LLM agent behavior is its tendency to hallucinate, particularly in listing the existence of relations and interpreting the meanings of relations in the CAMEO ontology, which leads to lower precision and recall. For example, in the example we show in Appendix F.3, in its trajectory step 3, the agent attempts to explain and conclude its final prediction: `'042'` Make a public statement (high frequency in historical data); `'036'` Negotiate (also high frequency); `'057'`: Express intent to cooperate (indicated by recent news context). However, the correct meanings are *Make a visit* for `'042'` and *Sign formal agreement* for `'057'`. This example highlights two issues: firstly, the LLM agent's overconfidence in its understanding of the CAMEO ontology without verifying the relation meanings through function calls (such as `map_cameo_to_relation` and `map_description_to_cameo`); secondly, its over-reliance on the semantic meaning of relations rather than their structural context. In contrast, TKG models learn relations by leveraging the historical graph structure, which inherently learns to capture the contextual meaning of each relation.

To enhance future LLM agents, incorporating a hybrid approach that combines semantic understanding with structural learning from TKGs could help to enhance relation modeling and address certain hallucinations.

D.5 Analysis with Retrieve-Augmented Generation (RAG) Methods

D.5.1 Methods and Experimental Setup

Besides the **Direct IO / QA** and three **ReAct** agents with different tool-use that we already implemented in the paper, we add the following baselines:

- **Direct QA with Augmentation** (for comment Q4)

Table 6: Evaluation results of GPT-4o-mini on the 2024-02 test split using different non-agentic methods and the ReAct agent with Single Function action type. The best-performing score is highlighted in **bold** and the second-best is underlined.

Method	Augmented Context	Agent API	Binary KL (⇓)	Quad KL (⇓)	First-level Relation (%)			Second-level Relation (%)		
					Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)
Direct IO	—	—	3.6$_{\pm1.0}$	7.6$_{\pm1.9}$	39.5$_{\pm3.2}$	44.8$_{\pm3.2}$	34.9$_{\pm3.5}$	15.4$_{\pm0.8}$	23.9$_{\pm3.6}$	15.4$_{\pm0.2}$
	CAMEO	—	5.0$_{\pm1.5}$	7.3$_{\pm1.9}$	35.5$_{\pm4.6}$	32.6$_{\pm2.2}$	28.6$_{\pm2.7}$	10.0$_{\pm1.5}$	14.2$_{\pm0.9}$	10.1$_{\pm0.8}$
RAG	*Events-Only*	—	**2.2**$_{\pm0.9}$	**5.9**$_{\pm2.0}$	57.5$_{\pm3.5}$	**53.4**$_{\pm3.4}$	**50.5**$_{\pm3.8}$	32.4$_{\pm1.1}$	**43.9**$_{\pm2.0}$	**33.2**$_{\pm1.4}$
	News-Only	—	9.1$_{\pm2.8}$	12.7$_{\pm2.9}$	47.2$_{\pm0.8}$	23.2$_{\pm2.4}$	25.4$_{\pm0.2}$	19.5$_{\pm2.0}$	14.9$_{\pm2.1}$	13.4$_{\pm0.8}$
	All	—	2.3$_{\pm1.4}$	6.3$_{\pm2.0}$	59.0$_{\pm1.2}$	48.1$_{\pm1.2}$	46.7$_{\pm0.4}$	36.4$_{\pm5.3}$	38.8$_{\pm1.2}$	32.1$_{\pm2.4}$
ReAct	—	*Event-Only*	3.3$_{\pm0.8}$	7.7$_{\pm1.4}$	**62.8**$_{\pm10.5}$	39.0$_{\pm0.8}$	41.7$_{\pm5.3}$	44.2$_{\pm3.3}$	37.0$_{\pm0.8}$	30.7$_{\pm0.9}$
	—	*News-Only*	6.5$_{\pm1.7}$	13.0$_{\pm2.1}$	41.5$_{\pm6.1}$	16.8$_{\pm0.7}$	20.2$_{\pm1.9}$	17.8$_{\pm0.2}$	12.2$_{\pm1.0}$	12.5$_{\pm0.5}$
	—	*All*	3.6$_{\pm0.9}$	8.0$_{\pm1.5}$	61.7$_{\pm10.1}$	38.6$_{\pm1.9}$	40.7$_{\pm5.6}$	**46.3**$_{\pm4.4}$	32.9$_{\pm3.8}$	31.1$_{\pm2.6}$

  - **QA with CAMEO**: We provide the CAMEO ontology in an ordered dictionary format mapping relation codes to their names and detailed descriptions. This is closer to the QA-format the authors mention, and a more fair comparison as the model can refer to the output vocabulary without needing to memorize CAMEO codes.
- **RAG Methods**
  Following recent work, we implement three RAG variants:
  - **RAG Events-Only**: Following GPT-NeoX-ICL Lee et al. [2023], we explore rule-based approaches for retrieving historical facts. Using the 'Pair' and 'Undirectional' setting, given a query event $(s, ?, o, t)$, we retrieve historical events $(s, r \in \mathcal{R}, o, < t)$ and $(o, r \in \mathcal{R}, s, < t)$. Events are sorted by recency with a cap of 30, aligning with the default cap of the `get_events` API function.
  - **RAG News-Only**: Following TCELongBench Zhang et al. [2024], we employ BM25 retrieval to fetch the most query-relevant news articles before the query date. The top 15 news articles are retained, matching the default cap of the `get_news_articles` API function.
  - **RAG All**: Combines both retrieved structured events and textual news articles.

D.5.2 Comparison of Retrieval Strategies with Agent

Both baseline approaches—CAMEO context augmentation and RAG methods—employ static, predefined retrieval strategies that are fixed for all queries and executed only once per query.

In contrast, our agentic approach enables dynamic, multi-step information gathering and reasoning. The agent **can** replicate the baseline retrieval strategy by fixing certain API parameter values, for example, the agent can use the function call `get_events(head_entities=[s, o], tail_entities=[o,s])` to get the retrieved context as RAG Event-Only, and use the function call `get_news_articles(text_description='(t, s, ?, o)')` to get the retrieved context as RAG News-Only; its capabilities extend far beyond these static approaches through its flexible parameter settings for each function call and multiple steps per query.

This multi-step, adaptive approach represents a fundamental shift from static retrieval to dynamic information gathering and reasoning, though it introduces higher requirements for the LLM's planning ability in:

- Automatically selecting optimal information-gathering strategies
- Integrating and reasoning over information of different formats
- Adjusting strategies based on intermediate findings and current context

D.5.3 Key Findings and Analysis

Our experiment results reveal several important insights:

Performance of the RAG Baselines:

RAG demonstrates improved precision over Direct IO (15.4% Pre in second-level) when using either event (32.4% Pre in second-level) or text (19.5% Pre in second-level) information source independently, with event data contributing more significantly to recall (43.9% Rec in second-level). When combining the two information sources, RAG achieves higher precision (36.4% Pre in second-level) but with a lower recall than RAG Event Only (from 43.9% to 38.8%), leading to a lower overall F1 score (from 33.2% to 32.1%), this suggests that a simple combination of both information in the context not effectively and collaboratively contribute to a better forecasting performance.

Comparison and Insights for Agent:

ReAct agents exhibit similar performance patterns with RAG when using different information sources (changed by the type of API functions available for the agent). Specifically, event data also contributes to high precision for the agent (62.8% Pre at first-level and 44.2% at second-level), outperforming RAG baselines. The structured event data consistently provides stronger signals for forecasting across both approaches, likely due to its standardized format.

However, agent baselines generally obtain a lower recall than RAG baselines. Meanwhile, while ReAct with full API access achieves higher average F1 scores in second-level relation prediction than its partial access performance (31.1% than 30.7% and 12.5%), the benefits of combining information sources aren't consistent across all relation hierarchies and methods, similar to RAG, suggesting substantial room for exploring more effective information integration strategies.

This reveals both promises and challenges of the agent's more flexible retrieval approach: RAG's predefined and fixed retrieval strategies can often yield stable performance, while the agent's dynamic and multi-step retrieval allows for flexible information gathering and integration, yet higher requirements for planning and reasoning sometimes also lead to relatively lower performance.

These observations underscore the core purpose of our benchmark: not just to compare current methods but to encourage the development of more advanced agentic forecasting approaches. The current performance patterns suggest significant opportunities for improving agent architectures, particularly in:

- Developing more robust and automatic planning strategies for multi-step information gathering
- Improving information integration capabilities across different information sources, formats, and temporal scales

D.6 Forecasting Performance on Different Test Splits

Table 7: Evaluation results of second-level relation forecasting on the different test splits, using "Single Function" as the action type. The best-performing score is highlighted in **bold** and the second-best is underlined.

Model	Training Data Cutoff Date	2023-11			2023-12			2024-01			2024-02		
		Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)	Pre. (⇑)	Rec. (⇑)	F1 (⇑)
Llama-3-8B-Instruct	2023-03	$10.7_{\pm 4.0}$	$6.1_{\pm 2.4}$	$6.1_{\pm 1.5}$	$13.6_{\pm 3.0}$	$10.2_{\pm 2.8}$	$8.7_{\pm 1.8}$	$16.0_{\pm 1.2}$	$9.0_{\pm 0.8}$	$8.7_{\pm 0.1}$	$15.8_{\pm 0.5}$	$11.8_{\pm 0.6}$	$10.3_{\pm 0.1}$
Llama-3.1-8B-Instruct	2023-12	$20.6_{\pm 9.0}$	$22.3_{\pm 5.4}$	$15.8_{\pm 5.7}$	$21.2_{\pm 7.0}$	$18.5_{\pm 1.5}$	$15.9_{\pm 3.6}$	$23.2_{\pm 1.5}$	$22.4_{\pm 3.2}$	$16.3_{\pm 2.0}$	$22.7_{\pm 3.2}$	$16.8_{\pm 0.5}$	$14.8_{\pm 0.7}$
GPT-4-Turbo	2023-12	$33.5_{\pm 7.5}$	$43.5_{\pm 5.4}$	$30.0_{\pm 1.9}$	$31.5_{\pm 4.5}$	$33.9_{\pm 0.5}$	$25.8_{\pm 3.1}$	$36.5_{\pm 3.4}$	$41.9_{\pm 4.7}$	$32.2_{\pm 2.8}$	$33.5_{\pm 4.4}$	$41.6_{\pm 1.3}$	$28.9_{\pm 3.2}$
GPT-4o-mini	2023-10	$41.3_{\pm 9.0}$	$41.4_{\pm 1.4}$	$32.8_{\pm 2.6}$	$39.4_{\pm 7.5}$	$25.4_{\pm 2.9}$	$25.9_{\pm 3.2}$	$45.9_{\pm 3.1}$	$36.6_{\pm 1.7}$	$33.2_{\pm 0.7}$	$40.0_{\pm 5.5}$	$32.6_{\pm 1.6}$	$29.7_{\pm 3.8}$

We discuss the effect of models' knowledge cutoff on different test splits in Sec. 3.3. We show the full forecasting performance of second-level relation prediction with different base models over the four test splits in Table 7.

E Additional Information about API

Table 8: API data classes and their attributes

Class Name	**Attributes and Types**
`Date`	`date: str # 'YYYY-MM-DD'`
`DateRange`	`start_date: Optional[Date], end_date: Optional[Date]`
`ISOCode`	`code: str # 3-letter ISO code`
`Country`	`iso_code: ISOCode, name: str`
`CAMEOCode`	`code: str # CAMEO code`
`Relation`	`cameo_code: CAMEOCode, name: str, description: str`
`Event`	`date: Date, head_entity: ISOCode, relation: CAMEOCode, tail_entity: ISOCode`
`NewsArticle`	`date: Date, title: str, content: str, events: List[Event]`

Table 9: API functions categorized by functionality

Functions related to Countries and Relations
`map_country_name_to_iso(name: str) -> List[Country]`
`map_iso_to_country_name(iso_code: ISOCode) -> str`
`map_relation_description_to_cameo(description: str) -> List[Relation]`
`map_cameo_to_relation(cameo_code: CAMEOCode) -> Relation`
`get_parent_relation(cameo_code: CAMEOCode) -> Relation`
`get_child_relations(cameo_code: CAMEOCode) -> List[Relation]`
`get_sibling_relations(cameo_code: CAMEOCode) -> List[Relation]`
Functions related to Events
`count_events(date_range: Optional[DateRange], head_entities: Optional[List[ISOCode]], tail_entities: Optional[List[ISOCode]], relations: Optional[List[CAMEOCode]]) -> int`
`get_events(date_range: Optional[DateRange], head_entities: Optional[List[ISOCode]], tail_entities: Optional[List[ISOCode]], relations: Optional[List[CAMEOCode]], text_description: Optional[str]) -> List[Event]`
`get_entity_distribution(date_range: Optional[DateRange], involved_relations: Optional[List[CAMEOCode]], interacted_entities: Optional[List[ISOCode]], entity_role: Optional[str]) -> Dict[ISOCode, int]`
`get_relation_distribution(date_range: Optional[DateRange], head_entities: Optional[List[ISOCode]], tail_entities: Optional[List[ISOCode]]) -> Dict[CAMEOCode, int]`
Functions related to News
`count_news_articles(date_range: Optional[DateRange], head_entities: Optional[List[ISOCode]], tail_entities: Optional[List[ISOCode]], relations: Optional[List[CAMEOCode]], keywords: Optional[List[str]]) -> int`
`get_news_articles(date_range: Optional[DateRange], head_entities: Optional[List[ISOCode]], tail_entities: Optional[List[ISOCode]], relations: Optional[List[CAMEOCode]], keywords: Optional[List[str]], text_description: Optional[str]) -> List[Tuple[Date, str]]`
`browse_news_article(date: Date, title: str) -> str`

F Additional Forecasting Examples of LLM Agent

In this section, we show examples of how LLM agents perform reasoning for the forecasting query. The query-specific values are highlighted in yellow in the system prompt and query prompt, which follows the prompt templates shown in Appendix F. For simplicity, we use `{api_description}` as a placeholder for the API specification in the prompt, which is replaced by the actual contents in experiments.

F.1 GPT-4o-mini-based Agent with ReAct and "Single Function" Action

F.1.1 Query Details

- **Query Quadruplet**: (2024-02-08, PNG, ?, AUS)
- **Temporal Distance**: 1; therefore, the current date is 2024-02-07
- **Agent Max Steps**: 20

F.1.2 Query Prompt

Please forecast the relations that Papua New Guinea will take towards Australia on February 08, 2024 based on historical information up to February 07, 2024. I.e. forecast the relation CAMEO codes in query event Event(date=2024-02-08, head_entity=ISOCode(PNG), relation=CAMEOCode(?), tail_entity=ISOCode(AUS)).

F.1.3 System Prompt

You are an expert in forecasting future events based on historical data. The database contains news articles from January 1, 2023 to the current date February 07, 2024 and the events extracted from these articles. The events are in the form of (date, subject country, relation, object country), where the countries are represented by ISO 3166-1 alpha-3 codes and the relations are represented by the CAMEO codes defined in the "Conflict and Mediation Event Observations" ontology. The relations are hierarchical: first-level relations are general parent relations represented by two-digit CAMEO codes, while second-level relations are more specific child relations represented by three-digit CAMEO codes. Child relations have the same first two digits as their parent relations. For example, "01" is a first-level relation, and "010" and "011" are some of its second-level relations. The relations in the database are represented in the second-level form.

Your task is to forecast the future relations between two entities in a given query. You have access to a defined Python API that allows you to query the database for historical events and statistics, and to get precise information about the ISO country codes and CAMEO relation codes.

The defined API is described as follows:
```python
{api_description}
```

You will use an iterative approach, interleaving "Thought", "Action", and "Observation" steps to collect information and perform the forecast. You may perform up to 20 iterations. The steps are as follows:

- "Thought": Analyze the current information and reason about the current situation, and predicts which API you want to use (try to use different APIs to collect diverse information) or make a decision that you want to make a final answer.
- "Action": Use the API to gather more information or provide the final forecast.
    - If using the API: the action must be only one single line of exactly one function call from the API with appropriate inputs, without additional code, explanations, or natural language descriptions.
    - If making the final forecast: the action must start immediately with "Final Answer:", and follow with the results in the expected JSON format.
- "Observation": Return the output of the called function.

To make a reasonable forecast, you should collect both news and relational evidence to support your prediction. When you are fully confident that you accumulate enough information to make the final forecast, you should start the "Thought" with your reasoning using the news and structural information to make the prediction, and then start the "Action" step with "Final

Answer:" followed by the answer in the expected JSON format. The answer should be a JSON dictionary where the keys are the forecasted two-digit first-level CAMEO codes and the values are lists of forecasted three-digit second-level CAMEO codes that are child relations of the key. For example, "Action: Final Answer: {"01": ["010", "011", "012"], "02": ["020", "023"]}".

The final answer will be evaluated based on the precision and recall of the forecasted first-level and second-level relations, so only include confident first-level and second-level CAMEO codes in your final forecast.

Try to use different APIs to collect diverse information (including multi-hop relations), such as the precise meaning of CAMEO codes, insights from news content, relational data, and statistical analyses to support your forecasts. Consider not only the frequency of the relations but also the temporal aspects of the data when making your forecast.

F.1.4 Agent ReAct Process

1. **Thought**: To forecast the relations between Papua New Guinea (PNG) and Australia (AUS) for February 08, 2024, I need to gather historical data on events involving these two countries. I will start by counting the number of events that have occurred between PNG and AUS in the past year to see which relations have been most prevalent. This will help me identify potential future relations.

   Action:

   ```
   count_events(date_range=DateRange(start_date=Date("2023-02-08"), end_date=Date("
       2024-02-07")), head_entities=[ISOCode("PNG")], tail_entities=[ISOCode("AUS")])
   ```

   Observation:

   ```
   33
   ```

2. **Thought**: The count of 33 events indicates that there has been a significant number of interactions between Papua New Guinea and Australia over the past year. Next, I will analyze the distribution of relations between these two countries to understand which specific CAMEO relations are most common. This will help me identify likely future relations.

   Action:

   ```
   get_relation_distribution(date_range=DateRange(start_date=Date("2023-02-08"), end_date
       =Date("2024-02-07")), head_entities=[ISOCode("PNG")], tail_entities=[ISOCode("AUS
       ")])
   ```

   Observation:

   ```
   {CAMEOCode(code='042'): 8, CAMEOCode(code='036'): 7, CAMEOCode(code='043'): 4,
       CAMEOCode(code='057'): 4, CAMEOCode(code='030'): 2, CAMEOCode(code='010'): 2,
       CAMEOCode(code='181'): 1, CAMEOCode(code='051'): 1, CAMEOCode(code='046'): 1,
       CAMEOCode(code='055'): 1, CAMEOCode(code='130'): 1, CAMEOCode(code='160'): 1}
   ```

3. **Thought**: The distribution of relations between Papua New Guinea and Australia shows that the most common relations are as follows:
   - **CAMEO Code ‘042’**: 8 events (likely related to economic agreements or cooperation).
   - **CAMEO Code ‘036’**: 7 events (likely related to diplomatic relations).
   - **CAMEO Code ‘043’**: 4 events (possibly related to trade).
   - **CAMEO Code ‘057’**: 4 events (possibly related to military cooperation).

   Given the frequency of these relations, it is likely that Papua New Guinea will continue to engage in similar interactions with Australia. To further refine my forecast, I will analyze the specific descriptions of these relations to understand their context better and see if any recent news articles provide insights into future interactions.

   Action:

   ```
   get_relation_distribution(date_range=DateRange(start_date=Date("2023-01-01"), end_date
       =Date("2024-02-07")), head_entities=[ISOCode("PNG")], tail_entities=[ISOCode("AUS
       ")])
   ```

   Observation:

```
{CAMEOCode(code='042'): 8, CAMEOCode(code='036'): 7, CAMEOCode(code='043'): 5,
    CAMEOCode(code='057'): 4, CAMEOCode(code='030'): 2, CAMEOCode(code='010'): 2,
    CAMEOCode(code='181'): 1, CAMEOCode(code='051'): 1, CAMEOCode(code='046'): 1,
    CAMEOCode(code='055'): 1, CAMEOCode(code='130'): 1, CAMEOCode(code='160'): 1}
```

4. **Thought**: The distribution of relations remains consistent with the previous observation, indicating that the most common interactions between Papua New Guinea and Australia are primarily focused on economic agreements (CAMEOCode ‘042’) and diplomatic relations (CAMEOCode ‘036’).

   To further understand the context of these relations, I will retrieve recent news articles that mention Papua New Guinea and Australia to see if there are any specific events or agreements that could influence the relations on February 08, 2024. This will help me make a more informed forecast.

   Action:

```
get_news_articles(date_range=DateRange(start_date=Date("2023-01-01"), end_date=Date("
    2024-02-07")), head_entities=[ISOCode("PNG")], tail_entities=[ISOCode("AUS")])
```

   Observation:

```
[(Date(date='2024-02-07'), 'Papua New Guineas PM to Visit Australian Parliament,
    Address Beijings Influence'),
 (Date(date='2024-02-07'), "PNG wants to stand on its 'own two feet': PM Marape"),
 (Date(date='2024-02-07'), 'Papua New Guineas PM to address Australian parliament
    as Pacific security race with China builds'),
 (Date(date='2024-01-11'), "Travel warning issued as 'widespread public disorder'
    rages in Port Moresby"),
 (Date(date='2024-01-11'), 'Papua New Guinea declares state of emergency after deadly
    riots')]
```

5. **Thought**: The recent news articles indicate that Papua New Guinea’s Prime Minister is set to visit the Australian Parliament and address concerns regarding China’s influence in the Pacific. This suggests a focus on diplomatic relations and security cooperation between PNG and AUS. The emphasis on standing on their "own two feet" also hints at a desire for economic independence and cooperation.

   Given the historical data showing a strong presence of economic agreements (CAMEOCode ‘042’) and diplomatic relations (CAMEOCode ‘036’), along with the context provided by recent news articles, it is reasonable to forecast that Papua New Guinea will likely engage in diplomatic discussions and economic cooperation with Australia on February 08, 2024.

   Action:

```
Final Answer: {"04": ["042", "043"], "03": ["036"], "05": ["057"]}
```

F.1.5 Agent Final Status

- **End State**: Final Answer
- **Number of Steps Taken**: 5
- **Final Answer**:

```
{
  "03": ["036"],
  "04": ["042", "043"],
  "05": ["057"]
}
```

We show the corresponding relation names of the predicted codes here for a better review:

```
{
  "Express intent to cooperate": ["Express intent to meet or negotiate"],
  "Consult": ["Make a visit", "Host a visit"],
  "Engage in diplomatic cooperation": ["Sign formal agreement"]
}
```

- **Ground Truth Answer**:

```
{
  "01": ["010"],
  "05": ["051"]
}
```

We show the corresponding relation names of the ground-truth codes here for a better review:

```
{
  "Make public statement": ["Make statement, not specified"],
  "Engage in diplomatic cooperation": ["Praise or endorse"]
}
```

F.2 GPT-4o-mini-based Agent with ReAct and “Code Block” Action

F.2.1 Query Details

- **Query Quadruplet**: (2024-02-08, PNG, ?, AUS)
- **Temporal Distance**: 1; therefore, the current date is 2024-02-07
- **Agent Max Steps**: 20

F.2.2 Query Prompt

Please forecast the relations that Papua New Guinea will take towards Australia on February 08, 2024 based on historical information up to February 07, 2024. I.e. forecast the relation CAMEO codes in query event Event(date=2024-02-08, head_entity=ISOCode(PNG), relation=CAMEOCode(?), tail_entity=ISOCode(AUS)).

F.2.3 System Prompt

You are an expert in forecasting future events based on historical data. The database contains news articles from January 1, 2023 to the current date February 07, 2024 and the events extracted from these articles. The events are in the form of (date, subject country, relation, object country), where the countries are represented by ISO 3166-1 alpha-3 codes and the relations are represented by the CAMEO codes defined in the "Conflict and Mediation Event Observations" ontology. The relations are hierarchical: first-level relations are general parent relations represented by two-digit CAMEO codes, while second-level relations are more specific child relations represented by three-digit CAMEO codes. Child relations have the same first two digits as their parent relations. For example, "01" is a first-level relation, and "010" and "011" are some of its second-level relations. The relations in the database are represented in the second-level form.

Your task is to forecast the future relations between two entities in a given query. You have access to a defined Python API that allows you to query the database for historical events and statistics, and to get precise information about the ISO country codes and CAMEO relation codes. You are also authorized to utilize additional safe, well-established Python libraries such as numpy, pandas, scikit-learn, and NetworkX to enhance your data analysis and forecasting accuracy.

The defined API is described as follows:
``` python
{api_description}
```

You will use an iterative approach, interleaving "Thought", "Action", and "Observation" steps to collect information and perform the forecast. You may perform up to 20 iterations. The steps are as follows:

- "Thought": Analyze the current information and reason about the current situation, and predicts which API you want to use (try to use different APIs to collect diverse information) or make a decision that you want to make a final answer.
- "Action": Use the API to gather more information or provide the final forecast.
    - If gathering more data: the action must be an executable Python code snippet that starts with ```python and ends with ```. It can contain multiple lines of codes and function calls using the defined API or Python libraries. You must use print() to output the results, and only the printed output will be returned in the observation step.
    - If making the final forecast: the action must start immediately with "Final Answer:", and follow with the answer in the expected JSON format. This should not be enclosed within triple backticks.
- "Observation": Return the printed output of the executed code snippet.

To make a reasonable forecast, you should collect both news and relational evidence to support your prediction. When you are fully confident that you accumulate enough information to make the final forecast, you should start the "Thought" with your reasoning using the news and structural information to make the prediction, and then start the "Action" step with "Final Answer:" followed by the answer in the expected JSON format. The answer should be a JSON dictionary where the keys are the forecasted two-digit first-level CAMEO codes and the values are lists of forecasted three-digit second-level CAMEO codes that are child relations of the key. For example, "Action: Final Answer: {"01": ["010", "011", "012"], "02": ["020", "023"]}".

The final answer will be evaluated based on the precision and recall of the forecasted first-level and second-level relations, so only include confident first-level and second-level CAMEO codes in your final forecast.

Try to use different APIs and Python libraries to collect diverse information (including multi-hop relations), such as the precise meaning of CAMEO codes, insights from news content, relational data, and statistical analyses to support your forecasts. Consider not only the frequency of the relations but also the temporal aspects of the data when making your forecast.

F.2.4 Agent ReAct Process

1. **Thought**: To forecast the relations that Papua New Guinea (PNG) will take towards Australia (AUS) on February 08, 2024, I need to gather historical data on events involving these two countries. Specifically, I will look for patterns in the relations between PNG and AUS in the past events to understand the likely future interactions.

   First, I will check the distribution of relations between PNG and AUS over the past year to identify any prevalent patterns or trends. This will help me understand the types of relations that have been most common and could indicate future behavior.

   I will use the ‘get_relation_distribution’ API to get this information.

   Action:

```
# Define the date range for the past year up to February 07, 2024
date_range = DateRange(start_date=Date("2023-02-08"), end_date=Date("2024-02-07"))
head_entities = [ISOCode("PNG")]  # Papua New Guinea
tail_entities = [ISOCode("AUS")]  # Australia

# Get the distribution of relations between PNG and AUS
relation_distribution = get_relation_distribution(date_range=date_range, head_entities
    =head_entities, tail_entities=tail_entities)
print(relation_distribution)
```

   Observation:

```
{CAMEOCode(code=’042’): 8, CAMEOCode(code=’036’): 7, CAMEOCode(code=’043’): 4,
    CAMEOCode(code=’057’): 4, CAMEOCode(code=’030’): 2, CAMEOCode(code=’010’): 2,
    CAMEOCode(code=’181’): 1, CAMEOCode(code=’051’): 1, CAMEOCode(code=’046’): 1,
    CAMEOCode(code=’055’): 1, CAMEOCode(code=’130’): 1, CAMEOCode(code=’160’): 1}
```

2. **Thought**: The relation distribution between Papua New Guinea (PNG) and Australia (AUS) over the past year reveals several notable patterns:

   - **CAMEO Code ‘042’**: This relation appears 8 times, which suggests a strong tendency for PNG to engage in "Make a public statement" towards Australia.
   - **CAMEO Code ‘036’**: With 7 occurrences, this indicates a frequent pattern of "Negotiate" relations.
   - **CAMEO Code ‘043’**: This relation, which signifies "Make a public statement, not specified," has been observed 4 times, indicating a general trend in public communication.
   - **CAMEO Code ‘057’**: Also appearing 4 times, this suggests some level of "Express intent to cooperate," which could be significant for future interactions.
   - Other codes like ‘030’ (Make a statement, not specified) and ‘010’ (Make statement, not specified) also show minor interactions.

   Given the high frequency of relations involving public statements and negotiations, it is likely that PNG will continue to engage in similar relations with Australia in the near future.

   Next, I will analyze the context of news articles related to PNG and AUS to see if there are any recent developments or trends that could influence their future relations. This will provide additional insights into the nature of their interactions.

I will use the ‘get_news_articles‘ API to gather relevant news articles from the past month that mention both PNG and AUS. This will help me understand current sentiments or events that could impact their relationship on February 08, 2024.

Action:

```
# Define the date range for the past month leading up to February 07, 2024
date_range = DateRange(start_date=Date("2024-01-07"), end_date=Date("2024-02-07"))
head_entities = [ISOCode("PNG")]  # Papua New Guinea
tail_entities = [ISOCode("AUS")]  # Australia

# Get news articles that mention both PNG and AUS
news_articles = get_news_articles(date_range=date_range, head_entities=head_entities,
     tail_entities=tail_entities)
print(news_articles)
```

Observation:

```
[(Date(date=’2024-02-07’), ’Papua New Guineas PM to Visit Australian Parliament,
     Address Beijings Influence’),
(Date(date=’2024-02-07’), "PNG wants to stand on its ’own two feet’: PM Marape"),
(Date(date=’2024-02-07’), ’Papua New Guineas PM to address Australian parliament as
      Pacific security race with China builds’),
(Date(date=’2024-01-11’), "Travel warning issued as ’widespread public disorder’ rages
      in Port Moresby")]
```

3. **Thought**: The recent news articles regarding Papua New Guinea (PNG) and Australia (AUS) indicate significant developments that could influence their relations on February 08, 2024. Notably, the Prime Minister of PNG is scheduled to visit the Australian Parliament and address issues related to Beijing’s influence in the region. This suggests a proactive approach by PNG to strengthen ties with Australia, particularly in the context of regional security and cooperation.

   The repeated mention of "widespread public disorder" in Port Moresby could indicate internal challenges for PNG, which may affect its diplomatic posture. However, the focus on addressing Australian Parliament implies that PNG is keen on maintaining a dialogue with Australia and possibly negotiating support or partnerships.

   Given the historical data showing a strong inclination towards public statements and negotiations (CAMEO Codes ‘042’ and ‘036’), along with the current political context, it is reasonable to forecast that PNG will likely engage in similar relations with Australia.

   Based on the analysis of both the historical event distribution and the recent news articles, I will forecast the following CAMEO codes for the relation that Papua New Guinea will take towards Australia on February 08, 2024:

   - **CAMEO Code ‘042’**: Make a public statement (high frequency in historical data).
   - **CAMEO Code ‘036’**: Negotiate (also high frequency).
   - **CAMEO Code ‘057’**: Express intent to cooperate (indicated by recent news context).

   Action:

```
Final Answer: {"04": ["042"], "05": ["057"], "03": ["036"]}
```

F.2.5 Agent Final Status

- **End State**: Final Answer
- **Number of Steps Taken**: 3
- **Final Answer**:

```
{
  "03": ["036"],
  "04": ["042"],
  "05": ["057"]
}
```

We show the corresponding relation names of the predicted codes here for a better review:

```
{
  "Express intent to cooperate": ["Express intent to meet or negotiate"],
  "Consult": ["Make a visit"],
  "Engage in diplomatic cooperation": ["Sign formal agreement"]
}
```

- **Ground Truth Answer**:

```
{
  "01": ["010"],
  "05": ["051"]
}
```

We show the corresponding relation names of the ground-truth codes here for a better review:

```
{
  "Make public statement": ["Make statement, not specified"],
  "Engage in diplomatic cooperation": ["Praise or endorse"]
}
```

F.3 Deepseek-r1-distill-llama-8b-based Agent with Reasoning and Function Call

As recommended by DeepSeek-R1 official[9], we put the prompt for reasoning model in the user prompt, and start the assistant prompt with token '<think>'. We show our user prompt as follows:

F.3.1 Query Details

- **Query Quadruplet**: (2024-02-05, USA, ?, IRN)
- **Temporal Distance**: 1; therefore, the current date is 2024-02-04
- **Max Function Call**: 20

F.3.2 User Prompt

You are an expert to use tools for forecasting future events based on historical data.

You can use the following tools by writing a function call in this format: <function_call> xxx </function_call>, where xxx is a python function call.

The database contains news articles from January 1, 2023 to the current date {current_date_nlp} and the events extracted from these articles. The events are in the form of (date, subject country, relation, object country), where the countries are represented by ISO 3166-1 alpha-3 codes and the relations are represented by the CAMEO codes defined in the 'Conflict and Mediation Event Observations' ontology. The relations are hierarchical: first-level relations are general parent relations represented by two-digit CAMEO codes, while second-level relations are more specific child relations represented by three-digit CAMEO codes. Child relations have the same first two digits as their parent relations. For example, '01' is a first-level relation, and '010' and '011' are some of its second-level relations. The relations in the database are represented in the second-level form.

Your task is to forecast the future relations between two entities in a given query. You have access to a defined Python API that allows you to query the database for historical events and statistics, and to get precise information about the ISO country codes and CAMEO relation codes.

To call these APIs, you need to use a streamlined reasoning process to collect information and perform the forecast.
- Use multiple <function_call> tags to call APIs as needed to collect diverse information (e.g., news and relational evidence). Each <function_call> contains exactly one single-line function call from the defined API with appropriate inputs, without additional code, explanations, or natural language descriptions. End each <function_call> with </function_call>. For example, <function_call>get_news_articles(date_range=["2023-01-01", "2023-01-31"])</function_call>.
- After each <function_call>, process the output returned in the corresponding <observation> tag to refine your reasoning.
- Repeat <function_call> and <observation> steps as necessary to accumulate sufficient evidence and reasoning.
- When fully confident, conclude the <think> phase with your reasoning based on the collected news and relational evidence to predict the forecast. End the <think> phase with </think>, and start the <answer> phase.
- <answer>: Provide the final forecast answer in the <answer> tag. The answer must be a JSON dictionary where the keys are forecasted two-digit first-level CAMEO codes, and the values are lists

---

[9] https://github.com/deepseek-ai/DeepSeek-R1

of forecasted three-digit second-level CAMEO codes that are child relations of the key. For example, <answer>{{"01": ["010", "011", "012"], "02": ["020", "023"]}}</answer>.

The defined API is described as follows: ```python
{api_description}
```

The final answer will be evaluated based on the precision and recall of the forecasted first-level and second-level relations, so only include confident first-level and second-level CAMEO codes in your final forecast.

Try to use different APIs by calling <function_call>...</function_call> to collect diverse information (including multi-hop relations), such as the precise meaning of CAMEO codes, insights from news content, relational data, and statistical analyses to support your forecasts. Consider not only the frequency of the relations but also the temporal aspects of the data when making your forecast. Remember to call tools in <function_call>...</function_call> tags to call functions.

Below are a few scenarios about how to use function calls:
1. Get the events between the United States and China from December 18, 2023 to January 27, 2024, with the relation "economic trade".
<function_call>get_events(date_range=DateRange(start_date=Date("2023-12-18"), end_date=Date("2024-01-27")), head_entities=[ISOCode("USA"), ISOCode("CHN")], tail_entities=None, relations=[CAMEOCode("010")], text_description="economic trade")</function_call>
2. Get the news articles about US president election on November, 2024 between the United States and China.
<function_call>get_news_articles(date_range=DateRange(start_date=Date("2024-11-01"), end_date=Date("2024-11-30")), head_entities=[ISOCode("USA"), ISOCode("CHN")], tail_entities=[ISOCode("USA"), ISOCode("CHN")], text_description="US president election.")</function_call>
3. Get the cameo code of the relation "Fight with guns"
<function_call>map_relation_description_to_cameo("Fight with guns")</function_call>

Query: Please forecast the relations that United States will take towards Iran on February 05, 2024 based on historical invocation up to February 04, 2024. I.e. forecast the relation CAMEO codes in query event Event(date=2024-02-05, head_entity=ISOCode(USA), relation=CAMEOCode(?), tail_entity=ISOCode(IRN)). Think step by step carefully and remember to call tools in <function_call>...</function_call> tags to call functions and get results.

### F.3.3 Reasoning Process

**<think>**
Okay, I need to forecast the relations between the United States and Iran on February 5, 2024, based on historical data up to February 4, 2024. Let me break this down step by step.

First, I'll start by understanding the tools available and how to use them. The user provided a list of Python functions from the API, which I can use to query the database. The goal is to collect enough data to make an informed forecast.

I think the first step is to retrieve all historical events between the USA and Iran within the given date range. So I'll call the get_events function, specifying the date range from January 1, 2023, to February 4, 2024, with head_entity as USA and tail_entity as Iran. Also, I'll filter for specific relations, but since I don't know them yet, I'll set relations to None to get all relevant events.

**<function_call>**

```
get_events(date_range=DateRange(start_date=Date(\"2023-01-01\"), end_date=Date
    (\"2024-02-04\")), head_entities=[ISOCode(\"USA\")], tail_entities=[ISOCode(\"IRN\")
    ], relations=None)
```

</function_call>
```

<observation>

```
[Event(date=Date(date=’2024-02-04’), head_entity=ISOCode(iso_code=’USA’), relation=
    CAMEOCode(cameo_code=’190’), tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(
    date=’2024-02-04’), head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(
    cameo_code=’112’), tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’
    2024-02-03’), head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’
    130’), tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-03’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’195’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-03’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’010’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-03’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’112’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-03’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-02’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’163’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-02’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’120’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-02’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’010’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-02’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’112’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-02’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-01’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’010’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-01’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’112’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-01’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’193’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-01’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-02-01’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’195’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-31’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’040’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-31’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’112’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-31’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-31’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’120’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’030’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’112’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’173’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’120’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’010’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-30’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-29’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’020’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-29’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’010’),
    tail_entity=ISOCode(iso_code=’IRN’)), Event(date=Date(date=’2024-01-29’),
    head_entity=ISOCode(iso_code=’USA’), relation=CAMEOCode(cameo_code=’190’),
    tail_entity=ISOCode(iso_code=’IRN’))]
```

</observation>

Given the above observation, I need to think about the next step. The events show a variety of relations between the USA and Iran. The most common relations are ’190’ (Use conventional military force, not specified), ’112’ (Economic sanctions), ’010’ (Make statement, not specified), ’193’ (Fight with small arms and light weapons), and ’120’ (Terminate diplomatic relations).

But wait, the user is asking to forecast the relations for February 5, 2024. I should consider if there’s any indication of upcoming events or changes in relations around that time. Maybe I should check for any news articles or events in early February 2024 to see if there’s an upcoming significant event that could influence the relations.

So, I’ll call the get_news_articles function for February 2024, specifically around February 5. This will help me see if there are any upcoming developments that could affect the relations.

<function_call>

```
get_news_articles(date_range=DateRange(start_date=Date(\"2024-02-01\"), end_date=Date
    (\"2024-02-05\")), head_entities=[ISOCode(\"USA\"), ISOCode(\"IRN\")], tail_entities
    =[ISOCode(\"USA\"), ISOCode(\"IRN\")], relations=None, text_description=None)
```

</function_call>

<observation>

```
Illegal action: Date must be on or before the current date 2024-02-04, but received:
    2024-02-05. Please make sure your action is a valid and executable function call
    with correct arguments based on the API description.
```

</observation>

Given the above observation, I need to think about the next step. Since the current date is February 4, 2024, I can’t query beyond that. So, I’ll adjust the end_date to February 4.

<function_call>

```
get_news_articles(date_range=DateRange(start_date=Date(\"2024-02-01\"), end_date=Date
    (\"2024-02-04\")), head_entities=[ISOCode(\"USA\"), ISOCode(\"IRN\")], tail_entities
    =[ISOCode(\"USA\"), ISOCode(\"IRN\")], relations=None, text_description=None)
```

</function_call>

<observation>

```
[(Date(date=’2024-02-04’), ’Operation Prosperity Guardian (OPG)’), (Date(date=’2024-02-04
    ’), ’Iran-backed militias in Iraq hold funeral for militants killed in US strikes’),
     (Date(date=’2024-02-04’), ’Joe Biden Officially Notifies Congress of Iraq, Syria
    Strikes’), (Date(date=’2024-02-04’), ’Huthis vow response after US, UK strike Yemen
    targets’), (Date(date=’2024-02-04’), ’What to know after latest US-UK air strikes on
     Iran-backed Houthi rebels’), (Date(date=’2024-02-04’), ’Israel-Hamas war: Blinken
    to attend Gaza truce talks \u2013 DW \u2013 02/04/2024’), (Date(date=’2024-02-04’),
    ’US-UK strikes on Yemen\u2019s Houthis: What we know so far’), (Date(date=’
    2024-02-04’), ’US says it will make additional strikes against Iran-backed groups’),
     (Date(date=’2024-02-04’), ’Morning Briefing: Feb. 4, 2024’), (Date(date=’2024-02-04
    ’), \"Houthi Red Sea attacks ’must stop,’ UK says \u2013 DW \u2013 02\"), (Date(date
    =’2024-02-04’), \"Iran Condemns US, UK’s Fresh Strikes on Yemen\"), (Date(date
    =’2024-02-04’), ’What are US troops doing in Middle East and where are they?’), (
    Date(date=’2024-02-04’), ’How Iran and the US could end up in a war neither of them
    really want’), (Date(date=’2024-02-04’), ’Why are Houthis attacking Red Sea ships? \
    u2013 Caribbean Broadcasting Corporation’), (Date(date=’2024-02-04’), \"Iran
    condemns latest U.S.-British strikes on Yemen’s Houthis\")]
```

</observation>

Given the above observation, I need to think about the next step. The news articles mention US strikes on Iran-backed groups in places like Yemen and Iraq. There’s also mention of potential diplomatic talks and tensions over Red Sea attacks. This suggests that the relations could be strained, possibly leading to military actions or diplomatic efforts.

Looking back at the historical events, the USA has shown a pattern of using certain CAMEO codes when dealing with Iran. For example, ’190’ (Use conventional military force) and ’112’ (Economic sanctions) are frequently used. The recent news indicates that military actions are ongoing, so ’190’ might be more relevant.

Additionally, the events show that ’120’ (Terminate diplomatic relations) has been used before, but in recent times, the focus seems to be more on military actions rather than diplomatic termination. So, perhaps ’190’ is the primary relation to forecast.

However, I should also consider if there are any signs of new relations emerging. The news mentions potential responses from Iran, which could lead to ’193’ (Fight with small arms) if tensions escalate. But since the current events are about conventional forces, ’190’ seems more likely.

Another aspect is economic relations. If the USA imposes more sanctions, ’112’ would be appropriate. The news doesn’t explicitly mention new sanctions, but the ongoing military actions could be part of a broader strategy that includes economic measures.

I should also check if there's any indication of new types of relations, like '010' (Make statements) or '030' (Peace negotiations). The news doesn't mention peace talks, but the USA is engaging in diplomatic efforts, so maybe '010' could be relevant if they make more statements.

In summary, based on the recent events and news, the most confident relations are '190' (Use conventional military force) and possibly '112' (Economic sanctions) if additional measures are taken. '193' is a possibility if there's an escalation, but it's less certain.
</think>

<answer>

```
{\"19\": [\"190\", \"193\", \"195\"],\n    \"01\": [\"010\", \"011\", \"012\"]\n}
```

</answer>

F.3.4 Agent Final Status

- **End State**: Final Answer
- **Number of Function Call Taken**: 3
- **Final Answer**:

```
{
  "01": ["010", "011", "012"],
  "19": ["190", "193", "195"]
}
```

We show the corresponding relation names of the predicted codes here for a better review:

```
{
  "Make public statement": ["Make statement, not specified", "Decline comment", "Make pessimistic comment"],
  "Fight": ["Use conventional military force, not specified", "Fight with small arms and light weapons", "Employ aerial weapons"],
}
```

- **Ground Truth Answer**:

```
{
  "11": ["112"],
  "12": ["120"],
  "19": ["190"]
}
```

We show the corresponding relation names of the ground-truth codes here for a better review:

```
{
  "Disapprove, not specified": ["Accuse"],
  "Reject": ["Reject"],
  "Fight": ["Use conventional military force"]
}
```

G Datasheet for MIRAI

G.1 Motivation

1. **For what purpose was the dataset created?**
   The MIRAI dataset was created to evaluate different capabilities of Large Language Model (LLM) agents in forecasting international events.
2. **Who created the dataset and on behalf of which entity?**
   The dataset was developed by a group of university researchers studying LLM reasoning. All rights are held by the individuals themselves, rather than by third-party stakeholders.
3. **Who funded the creation of the dataset?**
   The academic institutions support the university reasearch lab that the authors affiliate in.

G.2 Composition

1. **What do the instances that comprise the dataset represent?**
   Each instance in the dataset represents a record of international events, including the date, involved countries, and type of event, along with associated news articles and metadata.

2. **How many instances are there in total?**
   The current database comprises 1,296,991 GDELT event records from January 2023 to February 2024, corresponding to 75,341 unique events and 401,013 news articles. Test splits span over November 2023 to February 2024, each containing 100 balanced-sampled forecasting event queries. Note these statistics reflect the dataset used in this paper's experiments. As a dynamic benchmark, we are committed to periodic updates, continuously expanding the historical database and creating new test splits.

3. **Does the dataset contain all possible instances or is it a sample of instances from a larger set?**
   The dataset represents a curated sample from the entire GDELT database. It has been created through meticulously designed data cleaning and preprocessing steps on GDELT raw data, aimed at enhancing the quality and reliability of the event data.

4. **Is there a label or target associated with each instance?**
   Yes, each instance in the dataset is an event labeled with a relation type derived from the CAMEO[10] event taxonomy.

5. **Is any information missing from individual instances?**
   No, all instances are complete with all available information.

6. **Are there recommended data splits (e.g., training, development/validation, testing)?**
   The dataset is flexible in splitting data and expanding future events to new test splits. In this paper, we provide multiple test splits, spanning from November 2023 to February 2024. Each test instance utilizes all preceding records as its historical dataset to ensure accuracy and relevance in analysis. For training and development, users can flexibly use the data prior to the test splits based on their specific research needs.

7. **Are there any errors, sources of noise, or redundancies in the dataset?**
   The dataset has undergone extensive cleaning and structuring to minimize errors and noise. However, residual noise from the original GDELT database may still be present.

8. **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?**
   While the dataset is primarily self-contained, it includes URLs to news articles, providing links to external resources for further context and verification.

9. **Does the dataset contain data that might be considered confidential?**
   No, the dataset contains publicly available data, and does not include confidential information. It follows the term of use for GDELT, which is an open platform for research and analysis of global society.

10. **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?**
    The dataset may include descriptions of sensitive events, such as global conflicts, due to its focus on international relations. However, We conduct rigorous text cleaning procedures to reduce noise from web content while enhancing the reliability and ethical integrity of the textual information, following the OBELICS protocol [Laurençon et al., 2023]. Thorough checks such as flagging word ratios are employed during the data cleaning process to identify and potentially exclude inappropriate paragraphs or entire news articles and events. This ensures the minimization of distressing content while maintaining the integrity and relevance of the dataset for academic study.

G.3 Collection Process

1. **How was the data associated with each instance acquired?**
   Data for each instance was sourced from the GDELT project, which aggregates global event data and news articles from various worldwide media. Detailed information can be found in Section 2.2 of the paper.

2. **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?**

---

[10] Conflict and Mediation Event Observations (CAMEO)

Data collection was facilitated through software programs that aggregate event data and news articles from various sources. This automated collection is followed by data cleaning processes to enhance completeness and reliability. For a detailed description of the collection and cleaning methods, please refer to Section 2.2 of the paper. All dataset construction scripts are available in Github and also described in its README.

3. **Who was involved in the data collection process? (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)**
   The data collection and preprocessing were conducted by the authors, who are university researchers. This work was performed as part of their academic research activities.

4. **Does the dataset relate to people?**
   No. The dataset does not contain personal data or directly relate to individual people.

5. **Did you collect the data from the individuals in questions directly, or obtain it via third parties or other sources (e.g., websites)?**
   The dataset does not involve data collected from individuals directly. Instead, it is curated from the GDELT project, which aggregates information from various global news media sources.

G.4 Uses

1. **Has the dataset been used for any tasks already?**
   The dataset has not been used for any tasks other than the ones proposed and examined in this current paper, specifically for benchmarking the forecasting capabilities of LLM agents in predicting international relations. The dataset has also supported evaluating forecasting performance of traditional temporal knowledge graph-based methods and natural langugage-based methods.

2. **What (other) tasks could the dataset be used for?**
   In addition to benchmarking LLM agents and traditional forecasting methods, the dataset could be valuable for research in geopolitics, the development of other event prediction algorithms, sentiment analysis of international events, and trend analysis in global political dynamics.

3. **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?**
   The dataset was meticulously collected and preprocessed to ensure high quality and reliability. However, the reliance on the GDELT project as the primary data source and the specific cleaning process applied might influence its applicability to certain tasks. Users should consider these factors when applying the dataset to different research areas or methodologies.

4. **Are there tasks for which the dataset should not be used?**
   The dataset should not be used for any tasks that violate the terms of use associated with the GDELT project. We clearlt cite the terms of use in Appendix G.7.

G.5 Distribution

1. **Will the dataset be distributed to third parties outside of the entity?**
   Yes, the dataset, evaluation codes and leaderboards are intended to be publicly available to foster future research and development.

2. **How will the dataset be distributed?**
   The database and codebase are currently available via Google Drive, and Github. To enhance the accessibility and utility, the distribution of current version of data and its future updates will be enhanced by uploading the dataset to Hugging Face, and refining the API into a more user-friendly library format in the future.

3. **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?**
   No, there are no IP-based or other restrictions on the data: MIRAI is curated based on the GDELT[11] Event Database, which is an open platform for research and analysis of global society and all datasets released by the GDELT Project are available for unlimited and unrestricted use for any academic, commercial, or governmental use of any kind without fee; Data are allowed for any redistribution, rehost, republish, and mirror of the GDELT datasets in any form, with necessary citations[12].

---

[11] GDELT Project: https://www.gdeltproject.org/

[12] GDELT Term of Use: https://www.gdeltproject.org/about.html#termsofuse

4. **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?**
   No, there are no export controls or other regulatory restrictions applied to this dataset.
5. **When will the dataset be distributed?**
   The dataset will be made publicly available after the review process is completed, with the current Google Drive, and Github, and additional release/updates of arXiv, Hugging Face, and leaderboard to facilitate a more comprehensive accessibility to the research community.
6. **Will the dataset be distributed under a copyright or other IP license, and/or under applicable terms of use (ToU)?**
   The dataset will be distributed under the CC BY-NC 4.0 license, allowing for use and distribution for non-commercial purposes with appropriate attribution.

G.6 Maintenance

1. **Who will be supporting/hosting/maintaining the dataset?**
   The dataset maintenance will be supporting/hosting/maintaining by the authors.
2. **How can the owner/curator/manager of the dataset be contacted?**
   The owner/curator/manager of the dataset can be contacted through the authors' emails.
3. **Will the dataset be updated? (e.g., to correct labeling errors, add new instances, delete instances)?**
   Yes, the dataset is designed as a dynamic benchmark with periodic updates. We are committed to regularly expanding the historical database and creating new test splits. Updates will include adding new event data, creating new test splits, and potentially correcting any identified errors. Announcements regarding updates will be made through the project's official channels.
4. **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted?)**
   NA. This dataset does not contain data related to individuals or personal identifiers. It consists of aggregated event data and news articles related to international events.
5. **Will older version of the dataset continue to be supported/hosted/maintained?**
   Our dataset is designed as a cumulative, evolving benchmark. Historical event data is continuously incorporated into the database and remains available for future event forecasting. While we strongly recommend using the latest test splits to ensure data-contamination-free benchmarking for all LLM models, older historical data remains an integral part of the dataset. For reproducibility purposes, we will maintain records of the specific data versions used in published experiments. Researchers can request access to particular historical snapshots of the dataset, subject to resource availability. However, for ongoing research and comparisons, we encourage using the most current version of the dataset and its latest test splits.
6. **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?**
   Researchers and developers interested in extending, augmenting, or contributing to the dataset are encouraged to submit their changes through GitHub pull requests. For additional inquiries or detailed discussions, contacting the authors via email is recommended.

G.7 Term of Use for GDELT

Based on `https://www.gdeltproject.org/about.html#termsofuse`, GDELT dataset "*is an open platform for research and analysis of global society and thus all datasets released by the GDELT Project are available for unlimited and unrestricted use for any academic, commercial, or governmental use of any kind without fee.*", as long as "*any use or redistribution of the data must include a citation to the GDELT Project and a link to this website (`https://www.gdeltproject.org/`).*", which we've cited in abstract.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] They reflect.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] See Section 5.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes] See Section 3.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] See Section G.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] See Section 3.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] See Section 3.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] See Section 3.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes] We have reviewed the Code of Ethics and strictly followed it.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] See Section G.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes] See Section G.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] See Section G.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] See Section G.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]