Improving Causal Event Attribution in LLMs using Cross-Questions to Validate Causal Inference Assumptions

Anonymous ACL submission

Abstract

In this paper, we address the challenge of identifying real-world events that could have caused observed anomalies in time-series data of public indicators. Previously, this was a daunting task in a data analysis pipeline due to the open-ended nature of the answer domain. However, with the advent of modern large language models (LLMs), this task appears within reach. Our experiments on three diverse public time-series datasets shows that while LLMs can retrieve meaningful events with a single prompt, they often struggle with establishing the causal validity of these events.

To enhance causal validity, we design a set of carefully crafted cross-questions that check adherence to fundamental assumptions of causal inference in a temporal setting. The responses when combined through a simple feature-based classifier, improve the accuracy of causal event attributation from average of 65% to 90%. Our approach, including the questions, features, and classifier, generalizes across different datasets, serving as a meta-layer for temporal causal reasoning on event-anomaly pairs.

We release our code[1] and three datasets, which include time-series data with tagged anomalies and corresponding real-world events.

1 Introduction

Enterprise data analytics systems have long been dependent on tedious extraction, transformation, and linking processes to incorporate external world knowledge with structured databases to enrich data analysis (Zaharia et al., 2021; Farhan et al., 2024). With the advent of LLMs that are already pre-trained on huge amounts of external knowledge, it is time to rethink how data analysis systems can directly harness LLMs for external knowledge that earlier required extensive planning and processing.

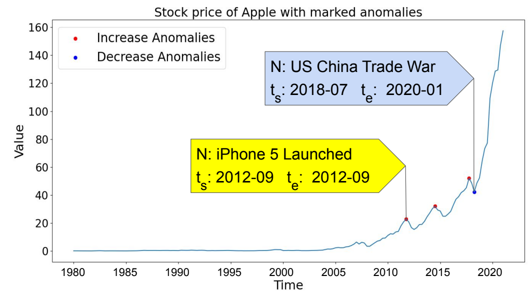

Figure 1: We show for two anomalies of a time series, the extracted real-world event that CauseExam attributes to the anomaly based on its LLM-based causal reasoning.

In this paper, we present one compelling scenario where we harness LLMs to extract attributing real-world events to explain observed patterns of anomalies in time series data. Time series are commonplace in any data analysis system, and a large part of data analysis revolves around discovering surprising changes along time, and digging out reasons to explain the changes (Sarawagi, 1999). In this paper we propose to enrich the analysis by linking to real-world events extracted from LLMs that could have plausibly caused the observed anomalies.

We work with two types of database systems: a worldbank database of various socio-economic indicators of countries, and two finance datasets of stock prices of companies. In Figure 1 we show a time series from financial system with marked anomalies that an analyst wishes to explain, and events that our system extracted by harnessing an LLM. Figure 10 in Appendix shows an example from worldbank system.

Accurate extraction of such structured events from an LLM is noisy since they are prone to hallucinations, and often confuse correlation with causation. We found that default LLM extractions tended to favor popular events such as COVID-19 or dot-

[1]Code & dataset repository: https://anonymous.4open.science/r/CauseExam. Link to dataset is provided in readme

com bubble burst to attribute to all and sundry anomalies. Recent evaluation of the commonsense causal reasoning capabilities of LLMs (Kıcıman et al., 2023; Zhang et al., 2023; Jin et al., 2023b) have shown promising results on logical reasoning based causal discovery given a pair of variable names, for example "smoking" and "cancer". Our scenario is more challenging for two reasons: (1) we need to extract candidate reasons for an observed anomaly in structured data instead of reasoning on a fixed set of variables, and (2) in addition to the variable name, we are provided an entire time series of values, and the causes we attribute have to be temporally consistent.

In this paper we show that the accuracy of cause-effect inference between an event-anomaly pair can be greatly enhanced with reasoning on responses of four cross-questions carefully designed to check adherence to fundamental assumptions of temporal causal inference. We convert LLM responses to these questions into numerical features each capturing the degree of adherence to the assumption of causal inference. Thereafter, we employ a light-weight Bayesian classifier to combine the features into binary decision variables. We propose a simple mechanism of harvesting labeled data for training the classifier from LLM using a novel counterfactual prompt to generate negative labeled examples. Since our features are generic, we show that the trained classifier generalizes across datasets.

Contributions.

- We present CauseExam a framework for extracting from an LLM, events that causally explain observed anomalies in time-series of public indicator. To the best of our knowledge, no prior work has proposed such a mechanism of enriching structured data analysis systems using LLMs.
- We enhance the accuracy of cause-effect inference on an event-anomaly pair using a set of cross-examination prompts specifically designed to check adherence to assumptions of temporal causal inference.
- We combine responses from multiple prompts using a light-weight model that can be trained using noisily extracted labeled data from the LLMs. To extract negative examples, we propose a novel method of harnessing counter-factual anomalies.
- We compare our method of calibrating correctness with other methods of checking LLM hallucinations, and show that our method, tailored for the task of extracting structured causal events provides significantly higher quality extractions. Starting from an accuracy of 65% from a single prompt, CauseExam's reasoning layer boosted accuracy to above 90%, significantly surpassing the accuracy of even GPT4 reranked events. Also, we show that our reasoning model transfers across datasets.
- We release three datasets on anomalies of public indicators along with real-world events.

2 Related Work

Causal reasoning with LLMs The investigation of an LLM's causal reasoning capabilities (Kıcıman et al., 2023; Zhang et al., 2023; Jin et al., 2023b; Liu et al., 2024; Long et al., 2024; Veljanovski and Wood-Doughty, 2024) on commonsense variables is an emerging topic of interest. Some studies (Jin et al., 2023a; Nie et al., 2023) attempt to assess if LLMs can do causal reasoning in accordance with a set of well-defined formal rules in hypothetical worlds. In constrast, we depend on causal knowledge of real world phenomenon that may have been expressed in the training data either explicitly (Hendrickx et al., 2010) or which LLM can infer via a chain of reasoning (Kosoy et al., 2022). Unlike in our case, most of these focus, on variables without any temporal context. Further, we are not aware of any prior work that combines responses from multiple diverse prompts for temporal causal reasoning.

Self-consistency checks in LLMs Many recent work propose to enhance the accuracy of facts extracted from LLMs based on self-consistency and cross-examination (Manakul et al., 2023; Mündler et al., 2024; Pacchiardi et al., 2024; Chen and Mueller, 2024). One category harness external information to verify LLM responses, whereas a second category relies on the LLM itself for correctness. Our work belongs to the second category. A standard technique here is to sample multiple answers and promote the answer that has maximum consensus (SelfCheckGPT (Manakul et al., 2023)). Other techniques including detecting contradictions in generated outputs (Mündler et al., 2024; Pacchiardi et al., 2024), quantifying uncertainty (Chen and Mueller, 2024) using simple cross-questioning along with consistency across multiple samples. Our method is also based on cross questioning the LLM but our questions are motivated to check validity of diverse assumptions of causal inference. We bypass the expensive sampling step of earlier work.

Cause-effect for Events Liu et al. (2023) propose to train a custom model to extract cause-effect relationships among events. Given the scarcity of labeled data, our focus is prompt-based extraction using LLMs. Romanou et al. (2023) contributes a dataset of events extracted from documents, and provides preliminary results on the use of LLMs to reason about the causal relations among the events. Our problem is different since we start from a structured time series of values, and extract real-world events from the LLM to explain observed anomalies in the series.

Causal discovery in time-series data For causal discovery among many time series, a common approach is Granger causality that infers that a time series X causes another time series Y if X values can predict Y values (Nauta et al., 2019; Cheng et al., 2023). A high Granger causality does not imply that X *causes* Y. More general causal discovery algorithms have been extended for time series data (Pamfil et al., 2020). Given lack of identifiability based on observation data, and the major challenge of integrating structured real-world events with time-series databases, the commonsense logic-based approach with LLMs provides a promising choice to standard data-driven causal reasoning.

3 Our Approach

In this section, we first formulate the problem we are trying to solve followed by an overview of our approach. Then, we present our cross-examination layer for reasoning about causality and method to combine different components of causality.

3.1 Problem Formulation

We start with a set of observed anomalies in a time series Y of values of a known indicator variable. Many different methods exist for spotting anomalies in time-series (Schmidl et al., 2022). Our method is agnostic to the method used, and just require each anomaly A to be a 4-tuple:

1. v: denoting the name of the public indicator whose values along time form the time series where the anomaly is observed.
2. t denoting the time when the anomaly occurred.
3. p denoting the pattern type of the anomaly. We focus on two patterns — a sharp increase or a sharp drop in the values along time.
4. L: optional location attribute of the time series

Let $\mathcal{L}$ denote a large language model, like OpenAI's ChatGPT. We assume $\mathcal{L}$ has real-world knowledge about the indicator. Our goal is to harness the LLM to extract a real-world event that could have caused an anomaly A. We impose structure in the extracted events by viewing them as instances of event categories from a well-known event ontology such as Wikidata. For each event E we extract a 5-tuple comprising of

1. N: Event name
2. t_s: Start time of the event
3. t_e: End time of the event
4. C: Category of the event. We assume event categories are nodes in a given ontology.
5. L: Location attached with the event.

Thus, for each input anomaly $A : (v, t, p)$ we wish to return an event E which could have caused the anomaly A. Figure 1 shows examples of two anomalies and corresponding extracted events. We have no supervision in the form of any labeled data for this task. We next present an overview of our method of performing such extractions using LLM.

3.2 Overview

Our framework comprises of three steps. Figure 2 presents an overview of our method. Our first step is to query the LLM to extract a ranked list of real-world events $E_1, \ldots, E_k$ to which an observed anomaly A can be attributed. We design a prompt that instructs the LLM to return the events as a structured tuple. The prompt used for such an extraction from LLM is present in Figure 4, and a sample response is shown in Figure 5. If the LLM was perfect, we could have stopped after this first step. But we observed several cases of errors in the extracted events using this single prompt. While in most cases the attributes of the events were factual, the LLM exhibited poor judgement on cause-effect reasoning. The LLM tended to favor popular events such as COVID-19 pandemic or dot-com bubble burst to attribute to all and sundry anomalies. Figure 5 shows one example. While several prior work have proposed techniques to correct mistakes and hallucinations in LLMs (Manakul et al., 2023; Mündler et al., 2024; Pacchiardi et al., 2024; Chen and Mueller, 2024), most of these are designed for factuality checks, whereas our task entails a more nuanced temporal causal reasoning. This led us to design a separate causal reasoning layer to rerank and prune the list of events returned in the first step. In the second step we issue a set of carefully designed cross-examination questions for testing

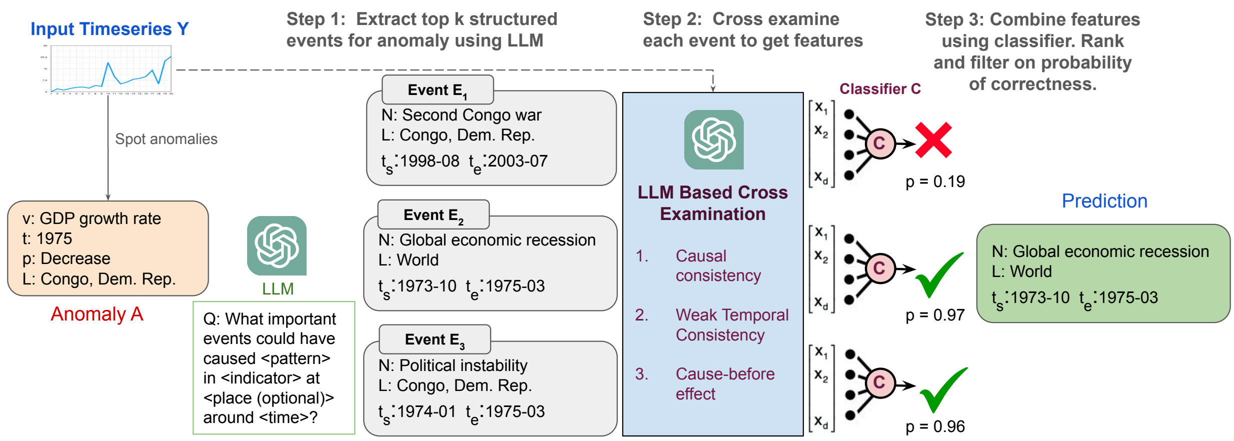

Figure 2: Overview of CauseExam inference framework for extracting real-world events to attribute to observed anomalies in time-series databases. The training of the classifier C is discussed in Section 3.4. Pseudocode of entire pipeline is present in Algorithm 1 in Appendix.

diverse aspects of what constitutes a valid temporal causality relationship between each anomaly A and candidate extracted event E_j. The set of questions and how we converted these into a feature vector is presented in Section 3.3. In the third step, we combine evidences from these features to output the final decision. We present details in Section 3.4.

3.3 Cross-Examination Prompts and Features

In the causal reasoning layer, we decide if an event E could have caused the anomaly A in the values of a series Y at time t. In causal inference terminology, E is a Boolean random treatment variable, and we are reasoning on its effect on Y which is continuous. Our reasoning is based on the following assumptions about causal inference:

1. Consistency: We follow the Neyman-Rubin potential outcomes framework (Rubin, 2005) and assume that the effect of E on Y is consistent. This implies that the observed anomaly A in values of Y at t is the same as the potential outcome if E were to re-occur in a parallel world.
2. Weak temporal consistency: If E is recurring e.g. financial crisis and it occurred at other points within the time-span of the series Y, its effect on Y would be mostly the same.
3. Cause-before-effect: The time of event occurrence has to be before the anomaly time t.

In the cross-examination phase, we ask questions to the LLM to check in diverse ways how well these assumptions hold. We assume the LLM's training data expresses in textual form the cause-effect relationship among real-world phenomenon after adjusting for confounders. The response to various questions provides a noisy peak into such documents. The questions are templatized and we process the response in conjunction with the time series Y such that the output of this phase is a vector of features where each feature quantifies adherence to one of the above assumption. Pseudocode in Algorithm 1 describes the process of feature creation in detail. We will later present ways to combine the response across multiple questions.

3.3.1 Causal consistency features

We first check for causal consistency by asking the LLM two Boolean questions with opposite effects of E on Y. The first question $\mathcal{R}(I)$ asks if E could cause a significant increase in the value of Y at t, and the second question $\mathcal{R}(D)$ asks the opposite question, if E could cause a drop. The exact prompt appears in Figure 6. We view the response as a verbalization of the potential outcome of E on Y at t, and we check consistency by matching with observed anomaly in Y. If the pattern p associated with the observed anomaly A is I (for "increase") then a consistent response would be a "Yes" for $\mathcal{R}(I)$ and a "No" for $\mathcal{R}(D)$, and equivalently for the case where p is a "drop". Since LLM responses are noisy, the response may not be consistent. We therefore treat the responses to these questions as noisy evidence of consistency or lack of it. Accordingly, we create two features: x_c, x_o. The feature is 1 iff response to the question $\mathcal{R}(p)$ matches the observed pattern p is "Yes", and second feature is 1 iff response to the other question is a "Yes". We call this set of features Boolean Consistency features.

An alternative to the above questions is a prompt that probes the LLM for the exact direction and magnitude of change that the event will have on Y. We ask the LLM to output the change direction (increase, decrease, or no change) along with a score between 0 and 100 indicating the strength of the change. The exact prompt $\mathcal{R}_M$ appears in Figure 7. Following this we obtain a set of three features which we call Effect Consistency features:

1. x_{d} that measures if the LLM response on change pattern matches the observed anomaly pattern p and takes value +1,-1,0 depending on whether they agree, disagree, or LLM response is no-change respectively.
2. x_m: This feature is the strength score chosen by LLM scaled to be between 0 and 1.
3. x_s: This feature is a product of the x_{d} and x_m.

3.3.2 Weak Temporal Consistency feature

If an event $E(n, t_s, t_e)$ is attributed to have caused an anomaly $A(v, p, t)$, then in an ideal setting where there are no other confounding variables, all other time intervals where the event n occurred should also result in the same pattern p of the indicator v at other times. Since we have the value of the indicator as a time-series, we can test whether this property holds. In real-life, we cannot assume that there are no confounders, so we can only measure weak compliance to such requirements. In order to quantify such temporal consistency we first question the LLM for the list of all time-intervals when the event of the same name n appeared. The prompt used to get this list is shown in Figure 8. The result is a list of time intervals: $\{(t_{s1}, t_{e1}), \ldots, (t_{sk}, t_{ek})\}$. On these intervals we measure the degree of consistency as the sum of the anomaly score in the time series at each time within the event interval

$$x_{\text{do}} = \text{sign}(p) \sum_{j=1}^{k} \sum_{t=t_{sj}}^{t<t_{ej}} \text{anomaly\_score}(v, t) \quad (1)$$

where the anomaly_score can be any measure of how different the value of series v at t is as compared to the expected value, and $\text{sign}(p) = 1$ if the pattern of anomaly p in A is increase, else -1.

3.3.3 Cause-before effect feature

This feature is used to find the time gap between the event and anomaly time. We observed that the LLM sometimes returned events with time-stamps *after* the anomaly time-stamps, and sometimes too soon before the anomaly. This feature helps down-score such extractions. We use the start time and end time of the event along with the anomaly time and give this feature value in the following manner:

$$x_{\text{gap}} = \begin{cases} \delta(t \geq t_s) & \text{if } t \leq t_e \\ \max(0, 1 - \frac{(t-t_e)}{5}) & \text{else.} \end{cases} \quad (2)$$

3.4 Learning to combine features

Each of the above features provide an indication on how much the extracted event (cause) adheres to the assumptions of causal inference. A baseline is to then just rank order extracted events based on the sum of these scores. We wanted to go a bit further and also filter away bogus events that could not have caused the anomaly. Let $O_{E \to A}$ denote the binary decision whether E causes A. We train a light-weight classifier $C : \mathbf{x} \mapsto O_{E \to A}$ for this task. To train the model C we depend on noisily labeled datasets constructed from the LLM.

Training data creation Given a set of anomalies $\{A_1, \ldots, A_n\}$, for each anomaly A_j, we extract a ranked list of events $E_{j1}, \ldots, E_{jk}$ from the LLM using the first prompt described in Section 3.2. Each $(A_j, E_{j,r})$ pair forms a noisy positive labeled example ($O_{E \to A} = 1$) for our dataset. To create negative examples, we use two sources. First, for each anomaly A_j, we create a counterfactual anomaly by inverting the pattern to create a new anomaly A_{n+j}. For example, if the pattern in anomaly A_j is "increase", pattern of A_{n+j} will be "decrease". We then probe the LLM to extract events $E_{n+j,1}, \ldots, E_{n+j,k}$ using prompt in Figure 4 corresponding to A_{n+j}. The $(A_j, E_{n+j,r})$ pair is treated as a negative example ($O_{E \to A} = 0$) since the event was not obtained as the reason for anomaly. Second, we randomly pair an anomaly A_j with an arbitrary other event $E_{i,r}$ to also serve as a negative example. We provide pseudocode in Algorithm 2 to describe the dataset creation and training of the classifier in detail.

Model selection and training Since we have only a small number of features (seven) and these were designed to test basic assumptions of causal inference, we found that simple models such as Naive Bayes were adequate for combining the evidence from these features. We also experimented with several classifier architectures coupled with noise tolerant noise functions such as generalized cross entropy (Zhang and Sabuncu, 2018) and found that a simple naive Bayes classifier performed the best under this noisy feature setting. Since our features are generic designed to check the satisfaction of the assumption of causal inference, the trained models generalize easily across datasets as we will show in the empirical section.

4 Experiments and Evaluation

We present an evaluation of the efficacy of state-of-the-art LLMs on the causal event extraction task. We compare our reasoning layer CauseExam of checking the correctness of event extraction with existing methods for self-checking responses. We also evaluate the sensitivity of various features and model choices, and show the generalization of CauseExam across datasets.

4.1 Datasets

We experiment with multiple time series selected from three datasets.

1. Worldbank dataset[2]: This contains annual values of socio-economic indicators for several countries. We create a dataset of top 20 countries by area and choose list of 5 important indicators: death rate, electric power consumption, GDP growth rate, military expenditure percentage of GDP and unemployment percentage. Each country, indicator pair defines a time-series. We chose the time 1960 to 2021 and dropped series with more than 50% missing values.
2. US Stock Exchange dataset: This contains historical data for stock prices of popular companies listed on NasdaqGS and NYSE. We aggregate them to a quarterly level for this analysis. We choose 5 companies each for the following 7 major categories of companies: Technology, Healthcare, Finance, Consumer Goods, Communication Services, Energy and Industrials.
3. London Stock Exchange dataset: It is similar to previous dataset but contains data for stock prices of companies listed on LSE. We choose two companies per category. Source for both stock exchange datasets is Yahoo Finance[3].

For these datasets the event types are restricted to be from 'war and conflicts', 'economic', 'political and diplomatic', 'health related' or 'natural disaster'. We manually mark anomalies in these time series. Number of anomalies is 254 in Worldbank dataset, 137 in US SE and 58 in London SE dataset.

We split the Worldbank and US SE data across train (40%), validation (20%) and test (40%). The splits are performed along country for the worldbank data, and along industry-type for the financial data so there is no overlap in the time-series across train and test. We use the entire London SE data in the test split to show generalization of our technique across datasets.

After we get the anomalies, we move on to the step of extracting events corresponding to each of these anomalies. We create train and validation data using data creation method described in Section 3.4. Extractions ar done for k=3 and k=5 using GPT 3.5 for each anomaly.

Labeling test data. For the anomalies and the set of extracted events we ask a group of human labellers to mark the events that are irrelevant to the anomaly.

Evaluation. We evaluate different methods of re-ranking and filtering the k extracted events. Accuracy is based on whether their top-1 predicted event is relevant to the anomaly as per the above gold labeling of the test data. When an anomaly has no relevant event, then a method that also does not return any event is considered correct.

4.2 Baselines

We compare our technique against these baselines:

Single extraction prompt. We use the ranking of events $E_1, \ldots, E_k$ extracted in order from the extraction prompt in Figure 4 using just GPT 3.5.

Single Extraction prompt reranked by GPT4. We ask GPT4 to rerank events $E_1, \ldots, E_k$ returned by GPT 3.5.

SelfCheckGPT methods. We rescore each event E_j using the top three methods reported in SelfCheckGPT (Manakul et al., 2023). All the variants first sample multiple ($M = 20$ in our experiments) stochastic responses to the prompt in Figure 9 using GPT 3.5, and measure the similarity of each candidate event E_j to sampled M events. These are 3 method variants used for measuring similarity: prompt-based technique, NLI (natural language inference), and unigram(max).

CauseExam. We report performance of CauseExam under various choice of classifiers for training $P(O_{E \to A}|\mathbf{x})$ models, various training data and different LLMs (GPT 3.5, GPT 4 and Llama3-70b) for cross-examination. Our model uses seven features as described in Section 3.3. The default classifier is Naive Bayes but we also compare with a logistic regression classifier and two-layer neural

[2] https://data.worldbank.org/
[3] https://finance.yahoo.com/

network.

4.3 Overall Results

In Table 1 we present an overall comparison of various methods. First observe that using just a single extraction prompt, GPT-3.5 is able to yield an accuracy around 60% for reasoning about anomalies in companies stock prices, and around 70% for various socio-economic phenomenon of the world. These numbers are encouraging, and show the promise of replacing elaborate ETL pipelines of data warehouses for integrating raw textual documents, to an LLM-based conversational integration.

Next we go over different methods of boosting the accuracy of initial extraction by reranking extracted events. SelfCheckGPT methods that rerank based on consensus with multiple sampled extractions, do help. The accuracy on the US SE dataset jumps from 62% to 72% with the best of these methods. When we use GPT-4 to rerank events generated from GPT-3.5, we get a much bigger boost and the Top-1 accuracy is now 87% for US SE and around 80% for Worldbank.

Compared to all these methods, CauseExam provides the largest boost with all LLMs improving the performance significantly. For example, CauseExam with GPT 3.5 gives an accuracy of 94% for US SE , 91% for London SE and 89% for Worldbank. Other LLMs give similar gains showing that most of the work is done by our classifier and feature aggregation technique. This shows the impact of our carefully designed cross-questions, the extracted featurization of the response, and classifier to implement sound temporal causal reasoning using LLMs as tools.

4.4 Role of different components

To understand the importance of each group of features we extracted in Section 3.3, we perform ablations where we drop one group of features at a time and record accuracy of the classifier for deciding $O_{E \to A}$ value based on the reduced feature. Table 2 shows the results. The first column of numbers are with no ablation. When we drop the Boolean Consistency feature of Section 3.3.1, we find a drop of up to 4% accuracy across both datasets. When we drop the Effect Consistency features of Section 3.3.1, the accuracy drops by as much as 9% for the US SE dataset. This group of feature turned out to be the most useful among the features we considered. By dropping the Cause-Before Effect feature accuracy dropped for the Worldbank dataset. For the US SE dataset it did not have much impact because for the initial extracted events they always had a value of 1. Finally, our Weak Temporal Consistency feature also boosted accuracy by as much as 4% for the US SE dataset. This establishes that our features motivated from the three causal inference assumptions had non-trivial mutual information with the class label, and they each provided a different important signal for the final causal decision.

The accuracy decreases significantly across all datasets and LLMs when only random negatives are used in training the classifier instead of combination of counterfactual negatives and random negatives with a drop of 5–25% across datasets and LLMs. This shows the importance of our novel method of generating counterfactual negatives described in Section 3.4 for training of classifier.

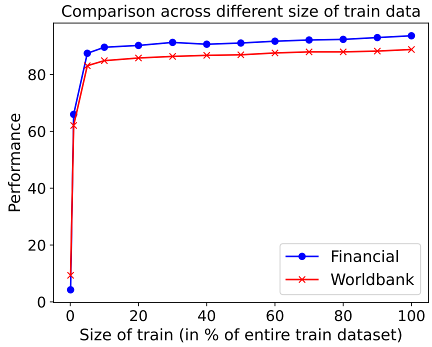

Figure 3: Accuracy with increasing size of training set for k=3 averaged over 10 random splits (100% train is 1120 samples).

4.5 Ablations on CauseExam classifier

In this section we show that the classifier used by CauseExam is robust to changing datasets and sizes, and a simple naive Bayes classifier works best for noisy labeled data. First in Table 3 we show a comparison of various choice of models for the binary classification task $P(O_{E \to A}|\mathbf{x})$ and note how Naive Bayes is significantly better, possibly because it is more robust to noisy labeled data. Next, we show that a very small amount of labeled data suffices in Figure 3. We find that even with 10% of the total training set which is about 100 noisy instances, we reach close to the peak accuracy.

In the above experiments, the training data was a union of instances from both US SE and Worldbank datasets. To establish generalization of these models to new datasets, we present another study where we train a classifier using labeled instances from

Dataset	k	Only Extract	SelfCheckGPT (GPT3.5)			GPT4 Re-Ranked	CauseExam		
			NLI	N-Gram	Prompt		GPT3.5	GPT4	Llama3
Worldbank	3	70.0	72.8	71.9	70.0	79.4	88.7	86.9	87.8
Worldbank	5	71.6	75.4	72.6	71.6	83.0	89.6	91.5	90.5
US SE	3	61.7	70.2	68.0	72.3	87.2	93.6	87.2	84.6
US SE	5	57.4	63.8	61.7	68.0	87.2	91.4	91.4	87.2
London SE	3	62.0	63.7	63.7	65.5	72.4	87.9	86.2	94.8
London SE	5	62.9	66.6	66.6	66.6	77.7	90.7	90.7	92.5

Table 1: Top-1 Accuracy of baselines against CauseExam . CauseExam outperforms all baselines across all datasets for each LLM. Only Extract method uses GPT 3.5. Table 5 in the appendix reports statistical significance over multiple runs. Samples where CauseExam beats GPT4 Re-ranked are shown in Figure 11 in Appendix.

Dataset	LLM	Without Ablation	Without features				No Counter factual Neg
			Boolean	Effect	Temporal	Cause-Before	
Worldbank	GPT3.5	88.7	85.9	83.1	85.9	82.2	83.1
Worldbank	GPT4	86.9	86.9	86.9	87.8	79.4	76.6
Worldbank	Llama3	87.8	89.7	86.9	88.7	77.5	79.4
US SE	GPT3.5	93.6	89.3	85.1	89.3	93.6	89.3
US SE	GPT4	87.2	87.2	87.2	85.1	87.2	63.8
US SE	Llama3	84.6	84.6	82.0	87.1	82.0	76.9

Table 2: Ablations on performance of the causal decision model $P(O_{E\to A}|\text{features})$ for k=3. Each feature set is important for performance and counterfactual negatives help train a more discriminating classifier.

Dataset	LLM	Logi-stic	2 Layer NN	Naive Bayes
Worldbank	GPT3.5	82.2	84.1	88.7
Worldbank	GPT4	82.2	79.4	86.9
Worldbank	Llama3	78.5	80.3	87.8
US SE	GPT3.5	85.1	89.3	93.6
US SE	GPT4	85.1	82.9	87.2
US SE	Llama3	76.9	84.6	84.6

Table 3: Comparison of performance across different training-based techniques trained on combined dataset for each LLM and k=3. Naive Bayes works best.

Dataset	LLM	Union dataset	Exchanged dataset
Worldbank	GPT3.5	88.7	87.8
Worldbank	GPT4	86.9	85.0
Worldbank	Llama3	87.8	88.7
US SE	GPT3.5	93.6	93.6
US SE	GPT4	87.2	87.2
US SE	Llama3	84.6	84.6

Table 4: Evaluating OOD generalization by training on US SE dataset and testing Worldbank and vice-versa. We compare with model trained on union of 2 datasets.

one dataset and deploy it on another dataset. In Table 4, we see that the accuracy with entire dataset is only slightly better than individual dataset.

5 Conclusion

In this paper we presented CauseExam, a novel framework of harnessing modern LLMs for extracting attributing real-world events to anomalies observed in structured time series. We observe that a default single prompt set of events generated from LLMs often lack relevance from causal view-point. We then designed a set of diverse cross-examination questions to check for adherence to three basic assumptions of temporal causal inference. We convert the responses into a small set of numerical features and train a light-weight classifier with LLM extracted noisy labeled data. We show that simple naive Bayes classifier provides a robust decision model. We boost accuracy of the single prompt extract from 65% to above 90% using our causal reasoning layer. Further our model generalizes across datasets because of the generic features we extract during the cross-examination.

This study shows both the promise of LLMs for closer integration of structured data analysis with real-world knowledge. Further, it highlights the role of more nuanced reasoning for specific tasks beyond what can be achieved by a language model.

Limitations

One of the limitations of this work is that information of the domain of time series dataset should be present in the training corpus of LLM. The LLMs used for experiments in this paper include GPT 3.5, GPT 4 and Llama 3, all of which have been trained on a large corpus of general data. Thus, they work well on datasets which are public and global in nature like social indicators dataset and stock prices of companies dataset. These LLMs will not give good performance on datasets that are private and do not belong to the training corpus of these LLMs such as the internal data of a company. The solution to this limitation is incorporating Retrieval Augmented Generation in the pipeline by providing sufficient documents with information relevant to the time series and events that can affect it. We treat this as an exciting direction for future research.

Ethics Statement

We construct the dataset used in our research using publicly available data sources like Worldbank[4] and Yahoo Finance[5] strictly adhering to their Terms of Use, and ensure that there are no privacy concerns or violations. In the annotator labellings, we collect no personal or identifiable information which can be misused.

For extractions from the LLMs used in this paper, we checked some samples manually and found no obvious ethical concerns, like violent or offensive content. However, we understand that text generation from LLMs is subject to unexpected outputs to a small degree and we should be careful while using this data.

References

Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness.

Yuxiao Cheng, Runzhao Yang, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. 2023. CUTS: Neural causal discovery from irregular time-series data. In *The Eleventh International Conference on Learning Representations*.

Marwa Salah Farhan, Amira Youssef, and Laila Abdelhamid. 2024. A model for enhancing unstructured big data warehouse execution time. *Big Data Cogn. Comput.*, 8:17.

[4]https://data.worldbank.org/
[5]https://finance.yahoo.com/

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng LYU, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023a. CLadder: A benchmark to assess causal reasoning capabilities of language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Scholkopf. 2023b. Can large language models infer causation from correlation? *ArXiv*, abs/2306.05836.

Eliza Kosoy, David M. Chan, Adrian Liu, Jasmine Collins, Bryanna Kaufmann, Sandy Han Huang, Jessica B. Hamrick, John Canny, Nan Rosemary Ke, and Alison Gopnik. 2022. Towards understanding how machines can learn causal overhypotheses. *Preprint*, arXiv:2206.08353.

Emre Kıcıman, Robert Osazuwa Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *ArXiv*, abs/2305.00050.

Jintao Liu, Zequn Zhang, kaiwen wei, Zhi Guo, Xian Sun, Li Jin, and Xiaoyu Li. 2023. Event causality extraction via implicit cause-effect interactions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, Julian McAuley, Wei Ai, and Furong Huang. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *Preprint*, arXiv:2403.09606.

Stephanie Long, Tibor Schuster, and Alexandre Piché. 2024. Can large language models build causal graphs? *Preprint*, arXiv:2303.05279.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

Meike Nauta, Doina Bucur, and Christin Seifert. 2019. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340.

Allen Nie, Yuhui Zhang, Atharva Amdekar, Christopher J Piech, Tatsunori Hashimoto, and Tobias Gerstenberg. 2023. Moca: Measuring human-language model alignment on causal and moral judgment tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Lorenzo Pacchiardi, Alex James Chan, Sören Mindermann, Ilan Moscovitz, Alexa Yue Pan, Yarin Gal, Owain Evans, and Jan M. Brauner. 2024. How to catch an AI liar: Lie detection in black-box LLMs by asking unrelated questions. In *The Twelfth International Conference on Learning Representations*.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Paul Beaumont, Konstantinos Georgatzis, and Bryon Aragam. 2020. Dynotears: Structure learning from time-series data. *ArXiv*, abs/2002.00498.

Angelika Romanou, Syrielle Montariol, Debjit Paul, Leo Laugier, Karl Aberer, and Antoine Bosselut. 2023. CRAB: Assessing the strength of causal relationships between real-world events. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15198–15216, Singapore. Association for Computational Linguistics.

Donald B. Rubin. 2005. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *Journal of the American Statistical Association*, 100:322–331.

S. Sarawagi. 1999. Explaining differences in multidimensional aggregates. In *Proc. of the 25th Int'l Conference on Very Large Databases (VLDB)*, pages 42–53, Scotland, UK.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797.

Marko Veljanovski and Zach Wood-Doughty. 2024. Doublelingo: Causal estimation with large language models.

Matei A. Zaharia, Ali Ghodsi, Reynold Xin, and Michael Armbrust. 2021. Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In *Conference on Innovative Data Systems Research*.

Cheng Zhang, Stefan Bauer, Paul Bennett, Jiangfeng Gao, Wenbo Gong, Agrin Hilmkil, Joel Jennings, Chao Ma, Tom Minka, Nick Pawlowski, and James Vaughan. 2023. Understanding causality with large language models: Feasibility and opportunities. *Preprint*, arXiv:2304.05524.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8792–8802, Red Hook, NY, USA. Curran Associates Inc.

A Pseudo Codes for CauseExam

We show the pseudocode for the CauseExam inference pipeline in Algorithm 1. The pseudocode for creating training data and training the classifier is shown in Algorithm 2

Algorithm 1 CauseExam Inference pipeline

Required: Time Series Y, Anomaly A_j, LLM $\mathcal{L}$, Classifier C
$E_{j1}, \ldots_{jk} \leftarrow$ query $\mathcal{L}$ with A_j using prompt in Figure 4
Initialize an empty map M
for $r \leftarrow 1$ to k **do**
    $\mathbf{x} \leftarrow \text{GETFEATURES}(Y, A_j, E_{j,r})$
    $O_{E \to A} \leftarrow C(\mathbf{x})$
    if $O_{E \to A} > 0.5$ **then** append $E_{j,r}$ to M with value $O_{E \to A}$
end for
Sort M by values in descending order
If M is not empty **then** return Top event in M as prediction **else** return None

---

function GETFEATURES(Y, A_j, $E_{j,r}$)
    Input: Time Series Y, Anomaly A_j, Event $E_{j,r}$
    Output: Feature vector $\mathbf{x}$
    x_c, x_o, x_d, x_m, $x_s \leftarrow \text{CAUSALCONSISTENCY}(A_j, E_{j,r})$
    $x_{do} \leftarrow \text{TEMPORALCONSISTENCY}(Y, A_j, E_{j,r})$
    Get x_{gap} using Equation 2
    $\mathbf{x}$:= [x_c, x_o, x_d, x_m, x_s, x_{do}, x_{gap}]
end function
function CAUSALCONSISTENCY(A_j, $E_{j,r}$)
    Input: Anomaly A_j, Event $E_{j,r}$
    Output: Features x_c, x_o, x_d, x_m, x_s
    ▷ Boolean Consistency Features
    $response(\mathcal{R}(I)) \leftarrow$ Query $\mathcal{L}$ with $\mathcal{R}(I)$ in Figure 6 and A_j, $E_{j,r}$, "increase" as arguments
    $response(\mathcal{R}(D)) \leftarrow$ Query $\mathcal{L}$ with $\mathcal{R}(D)$ in Figure 6 and A_j, $E_{j,r}$, "decrease" as arguments
    If $response(\mathcal{R}(p)$ = "Yes" **then** $x_c = 1$ **else** $x_c = 0$
    If $response(\mathcal{R}(p'))$ = "Yes" **then** $x_o = 1$ **else** $x_o = 0$ ▷ p' refers to opposite pattern of p
    ▷ Effect Consistency Features
    $res(\mathcal{R}_M) \leftarrow$ Query $\mathcal{L}$ with $\mathcal{R}_M$ in Figure 7
    $response(\mathcal{R}_M)_{change}$, $response(\mathcal{R}_M)_{mag} \leftarrow res(\mathcal{R}_M)$
    If $response(\mathcal{R}_M)_{change}$ = "no effect" **then** $x_d \leftarrow 0$
    elif $response(\mathcal{R}_M)_{change}$ = $p(A_j)$ **then** $x_d \leftarrow 1$
    else $x_d \leftarrow -1$
    $x_m \leftarrow response(\mathcal{R}_M)_{mag}/100$
    $x_d \leftarrow x_d * x_m$
end function
function TEMPORALCONSISTENCY(Y, A_j, $E_{j,r}$)
    Input: Time Series Y, Anomaly A_j, Event $E_{j,r}$
    Feature **Output:** x_{do}
    $\{(t_{s1}, t_{e1})], \ldots, (t_{sk}, t_{ek})\} \leftarrow$ Query $\mathcal{L}$ with prompt in Figure 8 and A_j $E_{j,r}$ as argument
    Get x_{do} using Equation 1
end function

Algorithm 2 Classifier Training Algorithm

Required: Time Series Y, Anomaly Set $\{A_1, \ldots, A_n\}$, LLM $\mathcal{L}$
Initialise empty lists S_{+ve} (positive samples), S_{-ve} (negative samples), E_{all} (all events)
for $j \leftarrow 1$ to n **do**
    $E_{j,1}, \ldots E_{j,k} \leftarrow$ query $\mathcal{L}$ with A_j using prompt in Figure 4
    Create counter factual anomaly A_{n+j} by inverting change direction
    $E_{n+j,1}, \ldots E_{n+j,k} \leftarrow$ query $\mathcal{L}$ with A_{n+j} using prompt in Figure 4
    Extend E_{all} with $E_{j,1}, \ldots E_{j,k}$, $E_{n+j,1}, \ldots E_{n+j,k}$
    for $r \leftarrow 1$ to k **do**
        $\mathbf{x}_{+ve} \leftarrow \textsc{GetFeatures}(Y, A_j, E_{j,r})$
        Append $\mathbf{x}_{+ve}$ to S_{+ve}
        $\mathbf{x}_{-ve} \leftarrow \textsc{GetFeatures}(Y, A_{n+j}, E_{n+j,r})$
        Append $\mathbf{x}_{-ve}$ to S_{-ve}
    end for
end for
for $j \leftarrow 1$ to n **do**
    Get an arbitrary event $E_{i,r}$ for A_j from E_{all} following constraints mentioned in Appendix.
    $\mathbf{x}_{rand} \leftarrow \textsc{GetFeatures}(Y, A_j, E_{i,r})$
    Append $\mathbf{x}_{rand}$ to S_{-ve}
end for
Train Binary Classifier C using S_{+ve} and S_{-ve}
return C

B More Experiments

We show the consistency of CauseExam technique over 10 runs with 80% training dataset randomly sampled and report the mean and standard deviation of performance for different LLMs and datasets in Table 5. We observe that performance is consistent over splits with a very small standard deviation showing that our classifier is robust to fluctuations in training data.

Dataset	k	Cause Exam GPT3.5	Cause Exam GPT4	Cause Exam Llama3
Worldbank	3	87.9 $\pm$ 0.53	86.0 $\pm$ 0.81	88.5 $\pm$ 0.63
Worldbank	5	89.6 $\pm$ 0.44	91.4 $\pm$ 0.29	91.0 $\pm$ 0.49
US SE	3	92.3 $\pm$ 1.09	87.2 $\pm$ 0.00	84.8 $\pm$ 0.81
US SE	5	91.2 $\pm$ 0.67	91.2 $\pm$ 0.67	86.3 $\pm$ 1.09
London SE	3	87.9 $\pm$ 0.81	86.2 $\pm$ 0.00	94.8 $\pm$ 0.00
London SE	5	90.7 $\pm$ 0.00	90.3 $\pm$ 0.78	92.9 $\pm$ 0.78

Table 5: Mean Top-1 Accuracy with standard deviation (mean $\pm$ std) for the performance of CauseExam using 80 % of training dataset over 10 random splits. We see that the training is stable and performance remains consistent across all splits.

The results of different ablations on London SE dataset are present in Table 6 and Table 7.

Dataset	LLM	Without Ablation	No Boolean features	No Effect features	No Temporal feature	No Cause-Before feature	No Counter factual Negatives
London SE	GPT 3.5	87.9	86.2	84.4	87.9	86.2	79.3
London SE	GPT 4	86.2	86.2	72.4	84.4	82.7	63.7
London SE	Llama 3	94.8	94.8	82.7	93.1	89.6	74.1

Table 6: Impact of ablations on performance of the causal decision model $P(O_{E \to A}|\text{features})$ for k=3. Each feature set appears to be important for performance and counterfactual negative prove to help training of classifier.

Dataset	LLM	Logi-stic	2 Layer NN	Naive Bayes
London SE	GPT 3.5	87.9	86.2	87.9
London SE	GPT 4	75.8	82.7	86.2
London SE	Llama 3	93.1	91.3	94.8

Table 7: Comparison of performance across different training-based techniques trained on combined dataset for each LLM and k=3. Naive Bayes works best.

C Prompts to the LLM

You are a helpful assistant for causal relationship understanding. Think about the cause-and-effect relationships between the events and its effect on the timeseries.
According to you, what important events could have caused <pattern> in <indicator><place(optional)> around <time>?
Return only python list of top <k> events in descending order of relevance as answer where each event is in a json parsable dictionary form (all values should be in string format) with keys event name, location (country name or "world" if event is global), start time in format yyyy-mm, end time in format yyyy-mm and type of event (one from <event-type-list>).

Figure 4: Prompt to the LLM to generate the ranked list of structured events to attribute to an Anomaly characterized by <indicator>, <pattern>, <time> at <place(optional)>. For each dataset there is a separate list of valid event-types.

- 1 : [‘dot-com bubble burst’, ‘2000-01’, ‘2002-01’]
- 2 : [‘y2k bug’, ‘1999-12’, ‘2000-01’]
- 3 : [‘microsoft releases windows 2000’, ‘2000-02’, ‘2000-03’]

Figure 5: Three extracted events to explain the anomaly: increase in stock price of Microsoft in 2000Q1. The response is obtained using the prompt in Figure 4 with arguments <Indicator>: stock price of Microsoft Corporation, <Pattern>:increase, <Time>: 2000Q1. It can be seen that dot com bubble burst is returned as top event corresponding to this anomaly which is not correct.

You are a helpful assistant for causal relationship understanding. Think about the cause-and-effect relationships between the event and its effect on the indicator.
Event: <event name> which happened from <event start time> to <event end time> in <event location> Effect: <pattern> in <indicator> (at <place> (optional)) around <time>

Could the event create this effect? Answer from one of the following options. Yes: Event could cause this effect. No: Event cannot cause this effect.

Answer should be one of the options 'Yes', 'No'. Important Note: Return just the answer from the options and nothing else.

Figure 6: Prompt to LLM to extract Boolean consistency features

You are a helpful assistant for causal relationship understanding. Think about the cause-and-effect relationships between the event and its effect on the indicator.
Event: <event name> which happened from <event start time> to <event end time> in <event location>
Indicator: <indicator> <place (optional)> around <time>

Event's effect on the Indicator is:
Increase: Event could increase the indicator. Choose this option if event has positive impact on indicator.
Decrease: Event could decrease the indicator. Choose this option if event has negative impact on indicator.
No effect: Event could not affect the indicator. Choose this option if event has no impact on indicator.

Magnitude of this effect is measured using a strength score from 0 to 100. (In case of No Effect return 0)
Score above 80: Event is related to this indicator and will definitely affect it.
Score between 50 and 80: Event is related to this indicator and might affect it.
Score between 20 and 50: Event might be related to this indicator but is less likely to affect it.
Score below 20: Event is not related to this indicator and will not affect it.

Return your answer as a python list of strings ["Effect", "Magnitude"]. Effect must be from one of the 3 options provided. Magnitude must be a single integer score from 0 to 100. Important Note: Return just this list as answer and nothing else.

Figure 7: Prompt to LLM to extract Effect consistency features

You are a helpful assistant for causal relationship understanding. Think about the cause-and-effect relationships between the events and its effect on the timeseries.
According to you, what important events could have caused <pattern> in <indicator><place(optional)> around <time>?
Return most relevant event as a json parsable dictionary form (all values should be in string format) with keys event name, location (country name or "world" if event is global), start time in format yyyy-mm, end time in format yyyy-mm and type of event (one from <event-type-list>).

Figure 9: Prompt to the LLM for SelfCheckGPT sample generation

You are a helpful assistant who has good knowledge of history and important events. Use this knowledge to answer the following question.
Event: <event name> which happened in <event loc> Related Indicator: <indicator>(at <place> (optional)) Between <series start time> and <series end time>, return the time periods when this event happened.

Return answer as a list of these time periods in the format:

[[<start time 1>, <end time 1>], [<start time 2>, <end time 2>], [<start time 3>, <end time 3>]...]

Some sample answers are shown below (each line is a sample answer): <examples of answer format>
Give the best answer as per your knowledge.
Important Note: Return the final answer between the tags <Answer>answer</Answer>.

Figure 8: Prompt to LLM to extract all time periods when event occurred for weak temporal consistency features

D Additional Examples and Samples of better perfomance by CauseExam

D.1 Example of Time Series labelled with anomaly

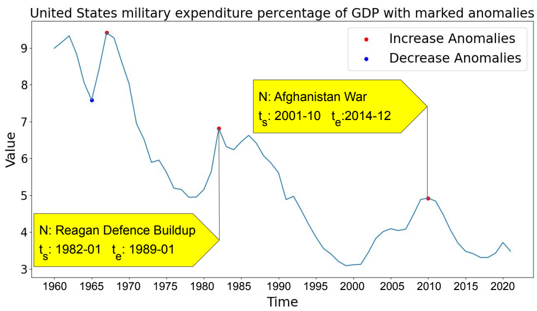

Figure 10: Example of time series from socio indicator system marked with two anomalies and the extracted real-world event that CauseExam attributes to the anomaly based on its LLM-based causal reasoning. In the first anomaly, US President Reagen significantly increased the military spending in his tenure. In the second anomaly, Afghanistan War peak happened around 2010 which caused increase in military expenditure of USA.

D.2 Examples of responses from the LLM from the first extraction prompt

Samples where GPT 3.5 fails:

1. <Popularity Problem>Pattern:increase, Indicator: stock price of Microsoft Corporation, Place: , Time: 2000Q1
   (a) Initial Event Order
      i. 1 : ['dot-com bubble burst', 'world', '2000-01', '2002-01']
      ii. 2 : ['y2k bug', 'world', '1999-12', '2000-01']
      iii. 3 : ['microsoft releases windows 2000', 'world', '2000-02', '2000-03']
   (b) Ground Truth Order
      i. 1 : ['microsoft releases windows 2000', 'world', '2000-02', '2000-03']
      ii. 2 : ['dot-com bubble burst', 'world', '2000-01', '2002-01']<IRRELEVANT>
      iii. 3 : ['y2k bug', 'world', '1999-12', '2000-01']<IRRELEVANT>
2. <Popularity Problem> Pattern:increase, Indicator: stock price of SunPower Corporation, Place: , Time: 2021Q1
   (a) Initial Event Order
      i. 1 : ['covid-19 pandemic', 'world', '2020-12', '2021-03']

        ii. 2 : ['us presidential election', 'united states', '2020-11', '2021-01']
        iii. 3 : ['renewable energy policies', 'united states', '2021-01', '2021-03']
    (b) Ground Truth Order
        i. 1 : ['renewable energy policies', 'united states', '2021-01', '2021-03']
        ii. 2 : ['us presidential election', 'united states', '2020-11', '2021-01']
        iii. 3 : ['covid-19 pandemic', 'world', '2020-12', '2021-03']<IRRELEVANT>

3. <Popularity Problem>Pattern:increase, Indicator: stock price of NVIDIA Corporation, Place: , Time: 2018Q3
    (a) Initial Event Order
        i. 1 : ['trade war between us and china', 'world', '2018-07', '2018-09']
        ii. 2 : ['strong quarterly financial results', 'world', '2018-08', '2018-08']
        iii. 3 : ['launch of new gaming gpus', 'world', '2018-08', '2018-08']
        iv. 4 : ['increased demand for ai and data center applications', 'world', '2018-07', '2018-09']
        v. 5 : ['positive industry outlook for semiconductor sector', 'world', '2018-07', '2018-09']
    (b) Ground Truth Order
        i. 1 : ['strong quarterly financial results', 'world', '2018-08', '2018-08']
        ii. 2 : ['launch of new gaming gpus', 'world', '2018-08', '2018-08']
        iii. 3 : ['increased demand for ai and data center applications', 'world', '2018-07', '2018-09']
        iv. 4 : ['positive industry outlook for semiconductor sector', 'world', '2018-07', '2018-09']
        v. 5 : ['trade war between us and china', 'world', '2018-07', '2018-09']<IRRELEVANT>
4. <Time delta and popularity problem>Pattern:decrease, Indicator: GDP growth rate, Place: Congo, Dem. Rep., Time: 1975
    (a) Initial Event Order
        i. 1 : ['second congo war', 'congo, dem. rep.', '1998-08', '2003-07']
        ii. 2 : ['global economic recession', 'world', '1973-10', '1975-03']
        iii. 3 : ['oil crisis', 'world', '1973-10', '1974-03']
        iv. 4 : ['political instability', 'congo, dem. rep.', '1975-01', '1975-12']
        v. 5 : ['drought', 'congo, dem. rep.', '1974-01', '1975-12']
    (b) Ground Truth Order
        i. 1 : ['drought', 'congo, dem. rep.', '1974-01', '1975-12']
        ii. 2 : ['oil crisis', 'world', '1973-10', '1974-03']
        iii. 3 : ['second congo war', 'congo, dem. rep.', '1998-08', '2003-07']
        iv. 4 : ['political instability', 'congo, dem. rep.', '1975-01', '1975-12']
        v. 5 : ['global economic recession', 'world', '1973-10', '1975-03']<IRRELEVANT>
5. <Fake event at top, consensus will help here because no time returned for this case> Pattern:increase, Indicator: military expenditure percentage of GDP, Place: Peru, Time: 1977
    (a) Initial Event Order
        i. 1 : ['peruvian constitutional crisis', 'peru', '1977-01', '1978-12']
        ii. 2 : ['world oil crisis', 'world', '1973-10', '1974-03']
        iii. 3 : ['shining path insurgency', 'peru', '1980-01', '1992-12']
    (b) Ground Truth Order
        i. 1 : ['world oil crisis', 'world', '1973-10', '1974-03']<IRRELEVANT>
        ii. 2 : ['peruvian constitutional crisis', 'peru', '1977-01', '1978-12']<IRRELEVANT>
        iii. 3 : ['shining path insurgency', 'peru', '1980-01', '1992-12']<IRRELEVANT>
6. <Popularity problem>Pattern:increase, Indicator: military expenditure percentage of GDP, Place: China, Time: 2009
    (a) Initial Event Order
        i. 1 : ['global financial crisis', 'world', '2008-09', '2009-12']
        ii. 2 : ['chinese economic stimulus package', 'china', '2008-11', '2009-12']
        iii. 3 : ['global recession', 'world', '2008-12', '2009-06']
    (b) Ground Truth Order
        i. 1 : ['chinese economic stimulus package', 'china', '2008-11', '2009-12']

ii. 2 : ['global financial crisis', 'world', '2008-09', '2009-12']<IRRELEVANT>
iii. 3 : ['global recession', 'world', '2008-12', '2009-06']<IRRELEVANT>

D.3 Examples where CauseExam beats GPT 4 reranking

Anomaly: increase in stock price of NVIDIA Corporation around Time: 2021Q4
Initial Order:
1 : covid-19 pandemic in world from 2020-12 to 2021-12
2 : global chip shortage in world from 2020-12 to 2022-12
3 : launch of new gaming consoles in world from 2020-11 to 2021-01
GPT4: global chip shortage in world from 2020-12 to 2022-12
CauseExam: launch of new gaming consoles in world from 2020-11 to 2021-01

Anomaly: increase in military expenditure percentage of GDP at Peru around 1977
Initial Order:
1 : Peruvian economic crisis in Peru from 1980-01 to 1985-12
2 : Falklands war in world from 1982-04 to 1982-06
3 : Debt crisis in Latin America from 1982-07 to 1989-12
GPT4: Peruvian economic crisis in Peru from 1980-01 to 1985-12
CauseExam: Falklands war in world from 1982-04 to 1982-06

Figure 11: Examples where CauseExam (GPT-3.5) beats GPT-4 Re-ranking

D.4 Examples where individual features improve performance

Figure 12 shows the examples for each of the set of features where they individually aid the performance.

E Dataset Details

E.1 Annotator Information

The annotators who marked anomalies and labeled test data for this research are 5 final-year students of the Undergraduate program who had good knowledge of the task. The average age of annotators was 21 years. They were paid for the task at par with the country's norms. Their demographic background is not disclosed to maintain anonymity. They were provided with clear instructions for both the tasks:

1. Anomaly Labelling: The definition of anomaly varied with different time series types. They were provided with sample labelings for each type of anomaly. To maintain uniformity, all time series of a particular type were given to one student.
2. Test Data Labelling: The annotators were shared a file with anomaly details and corresponding extracted. They were shared the following textual instruction "Mark the events which could not have caused this anomaly as irrelevant as per your understanding and inference. You are free to use any knowledge source to aid your decision making like web search and books.

E.2 Dataset numbers

1. Dataset details
   (a) The list of companies for US SEdataset per category:
      i. "Technology": "Apple Inc.", "Microsoft Corporation", "Amazon.com Inc.", "Alphabet Inc.", "NVIDIA Corporation" ,
      ii. "Healthcare": "Amgen Inc.", "Biogen Inc.", "Gilead Sciences Inc.", "Regeneron Pharmaceuticals Inc.", "Vertex Pharmaceuticals Incorporated" ,
      iii. "Finance": "PayPal Holdings Inc.", "The Goldman Sachs Group, Inc.", "JPMorgan Chase & Co.", "American Express Company", "Square, Inc." ,
      iv. "Consumer Goods": "Tesla, Inc.", "The Coca-Cola Company", "PepsiCo, Inc.", "Nike, Inc.", "Procter & Gamble Company" ,

Boolean consistency feature
Anomaly: Decrease in GDP growth rate at Congo, Dem. Rep. around 1975
Initial Event Order
1 : second congo war in congo, dem. rep. from 1998-08 to 2003-07
2 : global economic recession in world from 1973-10 to 1975-03
3 : political instability in congo, dem. rep. from 1974-01 to 1975-12
CauseExam prediction: global economic recession in world from 1973-10 to 1975-03
Explanation: The responses were Yes and No for this event, and for the top event of initial order, both responses were No.

Effect consistency feature
Increase in stock price of NVIDIA Corporation around 2018Q3
Initial Order:
1 : trade war between us and china in world from 2018-07 to 2018-09
2 : strong financial performance by nvidia in world from 2018-07 to 2018-09
3 : launch of new gaming gpus by nvidia in world from 2018-07 to 2018-09
CauseExam prediction: strong financial performance by nvidia in world from 2018-07 to 2018-09
Explanation: Gave the highest score to this event whereas the top of initial got negative score

---

Cause-before effect feature
Decrease in electric power consumption at Congo, Dem. Rep. around 1982
Initial Event Order
1 : second congo war in congo, dem. rep. from 1998-08 to 2003-07
2 : first congo war in congo, dem. rep. from 1996-10 to 1997-05
3 : economic crisis in congo, dem. rep. from 1982-01 to 1984-12
CauseExam prediction: economic crisis in congo, dem. rep. from 1982-01 to 1984-12
Explanation: Only 1 event was in the permitted time window. Time of top event of initial order was after the anomaly.

Weak Temporal Consistency feature
Increase in stock price of Clean Energy Fuels Corp. around 2021Q1
Initial Event Order
1 : covid-19 pandemic in world from 2020-12 to 2021-03
2 : joe biden’s inauguration united states 2021-01 2021-01
3 : renewable energy policies united states 2021-01 2021-03
CauseExam prediction: joe biden’s inauguration united states 2021-01 2021-01
Explanation: Covid-19 time was over 8 quarters, the net score came to be negative whereas for predicted event the score was positive

Figure 12: Examples where individual features improve performance

v. "Communication Services": "Meta Platforms, Inc.", "Netflix Inc.", "T-Mobile US, Inc.", "Comcast Corporation", "Charter Communications, Inc." ,
vi. "Energy": "Marathon Petroleum Corporation", "Clean Energy Fuels Corp.", "Plug Power Inc.", "Renewable Energy Group, Inc.", "SunPower Corporation" ,
vii. "Industrials": "Boeing Company", "Lockheed Martin Corporation", "FedEx Corporation", "United Parcel Service, Inc.", "Caterpillar Inc."

(b) The list of companies for London SEdataset per category:
   i. "Technology": "Rolls-Royce Holdings plc", "Informa PLC" ,
   ii. "Healthcare": "AstraZeneca PLC", "Smith & "Nephew plc" ,
   iii. "Finance": "Lloyds Banking Group plc", "Barclays PLC" ,
   iv. "Consumer Goods": "British American Tobacco plc", "Unilever PLC" ,
   v. "Communication Services": "Vodafone Group Pln", "ITV plc" ,
   vi. "Energy": "SSE plc", "BP plc" ,
   vii. "Industrials": "Babcock International Group PLC", "Melrose Industries PLC"

(c) Worldbank chosen 20 country list in descending order of area: "Russian Federation", "Canada", "China", "United States", "Brazil", "Australia", "India", "Argentina", "Kazakhstan", "Algeria", "Congo, Dem. Rep.", "Greenland", "Saudi Arabia", "Mexico", "Indonesia", "Sudan", "Libya", "Iran, Islamic Rep.", "Mongolia", "Peru"

2. As mentioned in the paper we had 254 anomalies for the worldbank dataset, 137 anomalies for the US SE dataset and 58 anomalies in London SE dataset.
   We use GPT 3.5 (gpt-35-turbo-16k) to extract events from anomalies. After we did event extraction, we had to drop a few anomalies due to parsing-related errors. After we drop these anomalies we are left with:
   (a) k=3: 54 London SE , 137 US SE , 250 worldbank
   (b) k=5: 58 London SE , 136 US SE , 247 worldbank
3. For training dataset creation, we have a positive to negative ratio of 3:4 for k=3 case and 5:6 for k=5 case. We ensured that training data is not skewed.
4. Size of training dataset creation:
   (a) k=3: 1120 samples, 480 positive, 640 negative in 100% combined dataset.
   (b) k=5: 1738 samples, 790 positive, 948 negative in 100% combined dataset.

F Experimental Details and Reproducibility

F.1 LLM details and Reproducibility

We work with 3 primary LLMs GPT 3.5, GPT 4 and Llama 3 (70 billion). Azure OpenAI was used to access GPT models and Ollama library in python was used to access Llama3 70b model. We set the temperature to 0 while generating responses for event extraction and cross-examination. The results should remain majorly reproducible barring a small fluctuation subject to variance in returned values from LLMs. We provide more details in following sections for reproducing the results.

F.2 Weak Temporal Consistency feature's Anomaly method

In this, we calculate the anomaly score using the statsmodels.tsa.seasonal.STL function. For worldbank dataset we use the timeperiod as 5 years and for the financial dataset we use the time period as 6 quarters. We find the trend in the data and then subtract this trend from the residue values to get the anomaly score. We normalize this anomaly score by dividing with the max absolute value of anomaly scores.

F.3 Constraints on Random Sampling of events

During random sampling of the event to associate with the anomaly we ensure the following conditions to avoid any misassociations:

1. Worldbank: We exclude all the events in the same country and the same indicator.
2. Financial: We exclude all the events of companies of this industry type and also the events with the similar trend. Removal of events with similar trend is essential because Global events will affect the

entire stock market as a whole and will create same effect across company types.

F.4 Training details

Naive Bayes and Logistic regression training is standard training. For training the 2 Layer NN, we use a model with 1 hidden layer of dimension 16. The training is done using Generalised cross entropy loss with noise parameter q=0.5. We choose this parameter because without gold truths we cannot estimate the noise in train data and so we cannot choose the most optimal q. Thus we take a middle value. Optimiser is Adam with lr=0.1 . We train for 100 epochs, breaking on Validation accuracy. The training time for each model training experiment is less than 1 minute on NVIDIA A100-SXM4 GPU.

G Details of SelfCheckGPT Baseline

We adapt the SelfCheckGPT methods to our case as follows:

1. In terms of the terminology used in SelfCheckGPT paper (Manakul et al., 2023), each of the k extracted events corresponding to an anomaly are treated as response R (R_1, R_2,...R_k). The objective is to rank each of these responses based on their scores. We then stochastically sample N=20 events using a prompt described in Figure 9. These 20 samples make the S for the technique as in selfcheckGPT method.
2. Since selfcheckGPT works on passages and sentences. We convert the structured event into a passage as follows:
   "Event <event name> can <pattern> <indicator><place str> around <anomaly time>. Event <event name> started in <event time start> and ended in <event time end>. Event <event name> happened in <event location>."
   This passage has 3 sentences.
3. We use different passage-level scores to rerank each event. This score is the average of the sentence level scores.
4. We compare our method against the top 3 performing methods for passage-level ranking performances in the Selfcheckgpt paper: prompt-based technique, NLI (natural language inference), and unigram(max).