Reasoning with a Few Good Cross-Questions Greatly Enhances Causal Event Attribution in LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we evaluate and enhance causal reasoning in LLMs for a novel task — discovering real-world events that cause anomalies in time-varying indicators. Our evaluation on three diverse datasets show 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 classifier, improve the accuracy of causal event attributation from an average of 65% to 90%. Our approach generalizes across different datasets, serving as a meta-layer for temporal causal reasoning on event-anomaly pairs.

1 Introduction

Our goal is to 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 [19]. 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. Figure 1 presents two examples of anomalies in two time-varying indicators, and the LLM extracted events that our model reasoned to have caused these anomalies. A formal definition of our task is as follows:

Problem Formulation We are given the sequence Y of values of a time-varying indicator, and one or more marked anomalies in Y. Many different methods exist for spotting anomalies in time-series [20]. Our method is agnostic to the method used, and just requires each anomaly A to be a 3-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. Let $\mathcal{L}$ denote a large language model that has real-world knowledge about the indicator. Our goal is to harness the LLM to extract a real-world event that could have *caused* the anomaly A. For each event E we extract a 4-tuple comprising of (1) N: Event name (2) L: Location of the event (3) t_s: Start time of the event (4) t_e: End time of the event. Thus, for each input anomaly $A : (v, t, p)$ we wish to return an event $E : (N, L, t_s, t_e)$ which could have caused the anomaly A. We have no supervision in the form of any labeled data for this task.

A simple way to solve the above problem is to just ask the LLM to return a list of events via a direct prompt as shown in Figure 4. We evaluated several latest LLMs in this default setting and found that almost all LLMs exhibited poor judgement on cause-effect reasoning in these direct extractions. They instead favored popular events such as COVID-19 pandemic or dot-com bubble burst as in the example shown in Figure 5. While several recent studies have also evaluated the commonsense causal

Submitted to NeurIPS 2024 Workshop on Causality and Large Models (CaLM). Do not distribute.

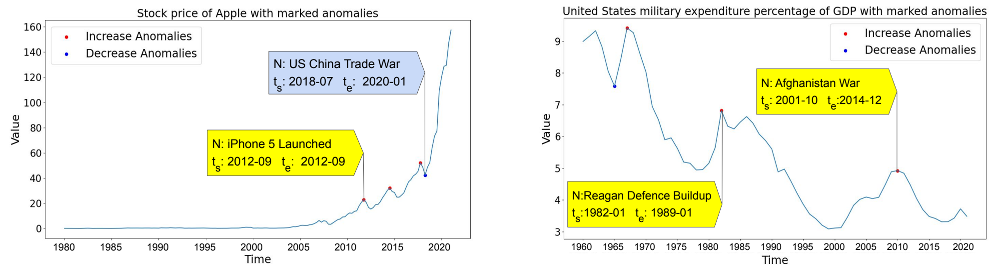

Figure 1: Example of time series. We show for two anomalies, the extracted real-world event that CauseExam attributes to the anomaly based on its LLM-based causal reasoning.

reasoning capabilities of LLMs [7, 22, 5], our scenario is different since we are provided an entire time series of values, and the causes we attribute have to be temporally consistent.

Contributions: We build a causal reasoning framework CauseExam to more accurately infer if an event E causes an anomaly A. CauseExam reasons on responses of four cross-questions carefully designed to check adherence to fundamental assumptions of temporal causal inference. To account for noise in the LLM response, the reasoning is cast as a feature-based classification task, where the features are derived from LLM responses to these four questions. Since we do not assume availability of labeled data, we propose a mechanism of harvesting labeled data for training the classifier from the LLM using a novel counterfactual prompt to generate negative labeled examples. We designed the numerical features to roughly capture the degree of adherence to basic assumption of causal inference. This results in the same trained classifier to generalize across datasets. Thus CauseExam can be thought as a meta-reasoning layer.

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 [7, 22, 5, 9, 10, 21] on commonsense variables is an emerging topic of interest. Some studies [4, 14] 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 [3] or which LLM can infer via a chain of reasoning [6]. 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 [11, 12, 15, 1]. A standard technique here is to sample multiple answers and promote the answer that has maximum consensus (SelfCheckGPT [11]). Other techniques including detecting contradictions in generated outputs [12, 15], quantifying uncertainty [1] 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. [8] 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. [17] 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.

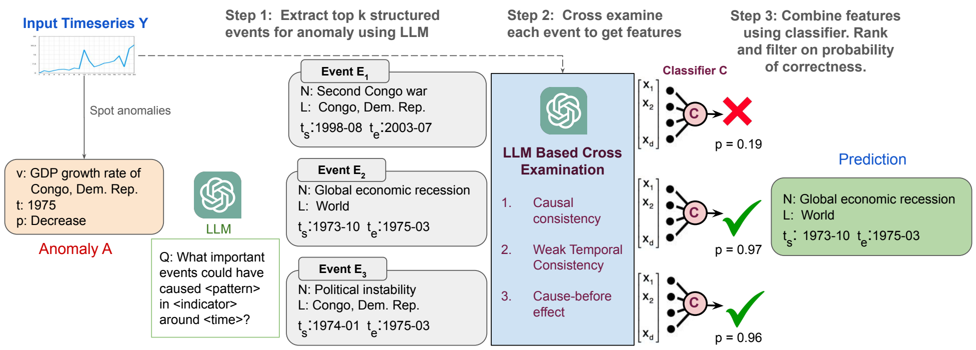

Figure 2: Overview of CauseExam inference framework for extracting real-world events to attribute to observed anomalies in time-series databases. The training of the classifier C is discussed in Section 3.2. Pseudocode of entire pipeline is present in Algorithm 1 in Appendix.

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 [13, 2]. A high Granger causality does not imply that X *causes* Y. More general causal discovery algorithms have been extended for time series data [16]. 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

Figure 2 presents an overview of our method. We first 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. For each event E, we invoke CauseExam for a more elaborate causal reasoning of if E could have caused the anomaly A in the values of the 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 [18] 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. Since the responses provides a noisy peak into such documents, we perform the final reasoning as a feature-based classification task. The features are derived from the response to the questions in conjunction with the time series Y. Next, in Section 3.1 we present the cross-questions, and in Section 3.2 we present how we combine the responses via the classifier. Feature creation is described in Algorithm 1.

3.1 Cross-Examination Questions and features

We extract three category of features from four cross-questions as described next.

3.1.1 Causal consistency

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 (described in Algorithm 1). 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.

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

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

3.2 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. 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 counter-factual 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 [23] 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.

Datasets. We experiment with multiple time series selected from three datasets. (1) **Worldbank dataset**[1](W-Bank): This contains annual values of socio-economic indicators for top 20 countries by area. We choose list of 5 important indicators. Each country, indicator pair defines a time-series. (2) **US Stock Exchange dataset** (US-SE): 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 companies from 7 major sectors. (3) **London Stock Exchange dataset** (L-SE): It is similar to previous dataset but contains data for stock prices of companies listed on LSE. Source for both stock exchange datasets is Yahoo Finance[2]. More details of datasets are present in Appendix E.

We manually mark anomalies in these time series. We split the W-Bank and US-SE data in train (40%), validation (20%) and test (40%). The splits are performed along country for the W-Bank data, and along industry-type for the US-SE data so there is no overlap across train and test. We use the entire L-SE data in the test split to show generalization of our technique across datasets. We extract events corresponding to each of these anomalies to create train and validation data using data creation method described in Section 3.2. Extractions are done 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.

Baselines. We compare our technique against these baselines: (1) **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. (2) **Single Extraction prompt reranked by GPT4**: We ask GPT4 to rerank events $E_1, \ldots, E_k$ returned by GPT 3.5. (3) **SelfCheckGPT methods**: We rescore each event E_j using the top three methods reported in SelfCheckGPT [11]. 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). (4) **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 3.1. The default classifier is Naive Bayes but we also compare with a logistic regression classifier and two-layer neural network.

Overall Results We present an overall comparison of various methods in Table 1. Using single extraction prompts, GPT-3.5 is able to yield an accuracy around 65% across datasets. Different methods of boosting the accuracy of initial extraction by reranking extracted events prove helpful. SelfCheckGPT methods increase accuracy on the US-SE dataset from 62% to 72%. Using GPT-4 to rerank events generated from GPT-3.5, gives a much bigger boost to accuracy which is now 87% for

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

Dataset	k	Only Extract	SelfCheckGPT (GPT3.5)			GPT4 Re-Ranked	CauseExam		
			NLI	N-Gram	Prompt		GPT3.5	GPT4	Llama3
W-Bank	3	70.0	72.8	71.9	70.0	79.4	88.7	86.9	87.8
W-Bank	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
L-SE	3	62.0	63.7	63.7	65.5	72.4	87.9	86.2	94.8
L-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 . Only Extract method uses GPT 3.5. Table 6 in the appendix reports statistical significance over multiple runs.

Dataset	LLM	Without Ablation	Without features				No Counter factual Neg
			Boolean	Effect	Temporal	Cause-Before	
W-Bank	GPT3.5	88.7	85.9	83.1	85.9	82.2	83.1
W-Bank	GPT4	86.9	86.9	86.9	87.8	79.4	76.6
W-Bank	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.

US-SE. CauseExam provides the largest boost with all LLMs improving the performance significantly. CauseExam with GPT 3.5 gives an accuracy of around 90% across all datasets. Other LLMs give similar gains showing that most of the work is done by our causal reasoning layer.

Role of different components: We present ablation results in Table 2 where we drop one group of features extracted in Section 3.1 at a time and record accuracy of the classifier. Observed that all feature groups are important for the performance with the most important group being Effect Consistency. We also observe a significant drop in accuracy (5–25% across datasets and LLMs) when we drop our novel counterfactual negatives from the negative training set.

Generalization across datasets To establish generalization of these models to new datasets, we present another study in Table 4 where we train a classifier using labeled instances from one dataset and deploy it on another dataset. We see that the accuracy with entire dataset is only slightly better than individual dataset.

Ablations on CauseExam classifier: We show a comparison of various choice of models for the binary classification task $P(O_{E\to A}|\mathbf{x})$ in Table 3 and Naive Bayes comes up to be significantly better, possibly because it is more robust to noisy labeled data. In Figure 3, we show that a very small amount of labeled data (about 100 noisy instances) suffices to reach close to the peak accuracy.

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 viewpoint. 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 highlights the role of more nuanced reasoning for specific tasks beyond what can be achieved by a language model.

References

[1] J. Chen and J. Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness, 2024.

[2] Y. Cheng, R. Yang, T. Xiao, Z. Li, J. Suo, K. He, and Q. Dai. CUTS: Neural causal discovery from irregular time-series data. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In K. Erk and C. Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[4] Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. LYU, K. Blin, F. G. Adauto, M. Kleiman-Weiner, M. Sachan, and B. Schölkopf. CLadder: A benchmark to assess causal reasoning capabilities of language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[5] Z. Jin, J. Liu, Z. Lyu, S. Poff, M. Sachan, R. Mihalcea, M. T. Diab, and B. Scholkopf. Can large language models infer causation from correlation? *ArXiv*, abs/2306.05836, 2023.

[6] E. Kosoy, D. M. Chan, A. Liu, J. Collins, B. Kaufmann, S. H. Huang, J. B. Hamrick, J. Canny, N. R. Ke, and A. Gopnik. Towards understanding how machines can learn causal overhypotheses, 2022.

[7] E. Kıcıman, R. O. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality. *ArXiv*, abs/2305.00050, 2023.

[8] J. Liu, Z. Zhang, kaiwen wei, Z. Guo, X. Sun, L. Jin, and X. Li. Event causality extraction via implicit cause-effect interactions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[9] X. Liu, P. Xu, J. Wu, J. Yuan, Y. Yang, Y. Zhou, F. Liu, T. Guan, H. Wang, T. Yu, J. McAuley, W. Ai, and F. Huang. Large language models and causal inference in collaboration: A comprehensive survey, 2024.

[10] S. Long, T. Schuster, and A. Piché. Can large language models build causal graphs?, 2024.

[11] P. Manakul, A. Liusie, and M. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, Dec. 2023. Association for Computational Linguistics.

[12] N. Mündler, J. He, S. Jenko, and M. Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] M. Nauta, D. Bucur, and C. Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

[14] A. Nie, Y. Zhang, A. Amdekar, C. J. Piech, T. Hashimoto, and T. Gerstenberg. Moca: Measuring human-language model alignment on causal and moral judgment tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] L. Pacchiardi, A. J. Chan, S. Mindermann, I. Moscovitz, A. Y. Pan, Y. Gal, O. Evans, and J. M. Brauner. How to catch an AI liar: Lie detection in black-box LLMs by asking unrelated questions. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] R. Pamfil, N. Sriwattanaworachai, S. Desai, P. Pilgerstorfer, P. Beaumont, K. Georgatzis, and B. Aragam. Dynotears: Structure learning from time-series data. *ArXiv*, abs/2002.00498, 2020.

[17] A. Romanou, S. Montariol, D. Paul, L. Laugier, K. Aberer, and A. Bosselut. CRAB: Assessing the strength of causal relationships between real-world events. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15198–15216, Singapore, Dec. 2023. Association for Computational Linguistics.

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

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

[20] S. Schmidl, P. Wenig, and T. Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

[21] M. Veljanovski and Z. Wood-Doughty. Doublelingo: Causal estimation with large language models. 2024.

[22] C. Zhang, S. Bauer, P. Bennett, J. Gao, W. Gong, A. Hilmkil, J. Jennings, C. Ma, T. Minka, N. Pawlowski, and J. Vaughan. Understanding causality with large language models: Feasibility and opportunities, 2023.

[23] Z. Zhang and M. R. Sabuncu. 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, 2018. 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
$E_{j1},\dots_{jk} \leftarrow$ query $\mathcal{L}$ with A_j using prompt in Figure 4
Initialize an empty map M
for $r \leftarrow 1$ to k **do**
  $\mathbf{x} \leftarrow \text{GETFEATURES}(Y, A_j, E_{j,r})$
  $O_{E\rightarrow A} \leftarrow C(\mathbf{x})$
  if $O_{E\rightarrow A} > 0.5$ **then** append $E_{j,r}$ to M with value $O_{E\rightarrow A}$
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 3.1.3
  $\mathbf{x}$:= $[x_c, x_o, x_d, x_m, x_s, x_{do}, x_{gap}]$
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}
  $\{(t_{s1}, t_{e1})], \dots, (t_{sk}, t_{ek})\} \leftarrow$ Query $\mathcal{L}$ with prompt in Figure 8 and A_j $E_{j,r}$ as argument
  Get x_{do} using method described in Section 3.1.2
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 Details of Experiments

B.1 More details on ablation

B.1.1 Role of different components

To understand the importance of each group of features we extracted in Section 3.1, we perform ablations where we drop one group of features at a time and record accuracy of the classifier for deciding $O_{E\rightarrow A}$ 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 3.1.1, we find a drop of up to 4% accuracy across both datasets. When we drop the Effect Consistency features of Section 3.1.1, 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 W-Bank 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 3.2 for training of classifier.

B.1.2 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 $P(O_{E\rightarrow A}|\mathbf{x})$ 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 W-Bank datasets. To establish generalization of these models to new datasets, we present another study where we train a classifier using labeled instances from 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.

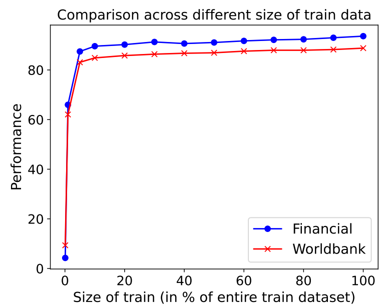

Figure 3: Accuracy with increasing size of training set for k=3 averaged over 10 random splits (100% train is 1120 samples).

Dataset	LLM	Logi-stic	2 Lay-er NN	Naive Bayes
W-Bank	GPT3.5	82.2	84.1	88.7
W-Bank	GPT4	82.2	79.4	86.9
W-Bank	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
L-SE	GPT 3.5	87.9	86.2	87.9
L-SE	GPT 4	75.8	82.7	86.2
L-SE	Llama 3	93.1	91.3	94.8

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
W-Bank	GPT3.5	88.7	87.8
W-Bank	GPT4	86.9	85.0
W-Bank	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 W-Bank and vice-versa. We compare with model trained on union of 2 datasets.

Results of ablation on L-SE dataset are shown in Table 5

B.2 Performance over multiple runs

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 6. 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	LLM	Without Ablation	Without features				No Counter factual Neg
			Boolean	Effect	Temporal	Cause-Before	
L-SE	GPT 3.5	87.9	86.2	84.4	87.9	86.2	79.3
L-SE	GPT 4	86.2	86.2	72.4	84.4	82.7	63.7
L-SE	Llama 3	94.8	94.8	82.7	93.1	89.6	74.1

Table 5: 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	k	Cause Exam GPT3.5	Cause Exam GPT4	Cause Exam Llama3
W-Bank	3	87.9 ± 0.53	86.0 ± 0.81	88.5 ± 0.63
W-Bank	5	89.6 ± 0.44	91.4 ± 0.29	91.0 ± 0.49
US-SE	3	92.3 ± 1.09	87.2 ± 0.00	84.8 ± 0.81
US-SE	5	91.2 ± 0.67	91.2 ± 0.67	86.3 ± 1.09
L-SE	3	87.9 ± 0.81	86.2 ± 0.00	94.8 ± 0.00
L-SE	5	90.7 ± 0.00	90.3 ± 0.78	92.9 ± 0.78

Table 6: 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.

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> 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’, ‘world’, ‘2000-01’, ‘2002-01’]
- 2 : [‘y2k bug’, ‘world’, ‘1999-12’, ‘2000-01’]
- 3 : [‘microsoft releases windows 2000’, ‘world’, ‘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> 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> 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> 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>
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 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 of 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 of 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 of 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.2 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 10: Examples where CauseExam (GPT-3.5) beats GPT-4 Re-ranking

D.3 Examples where individual features improve performance

Figure 11 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-SE dataset 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" ,
      v. "Communication Services": "Meta Platforms, Inc.", "Netflix Inc.", "T-Mobile US, Inc.", "Comcast Corporation", "Charter Communications, Inc." ,

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 11: Examples where individual features improve performance

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 L-SE dataset 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 L-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 L-SE , 137 US-SE , 250 worldbank
   (b) k=5: 58 L-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 [11], 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).