Augur: Modeling Covariate Causal Associations in Time Series via Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLM) have emerged as a promising avenue for time series forecasting, offering the potential to integrate multimodal data. However, existing LLM-based approaches face notable limitations—such as marginalized role in model architectures, reliance on coarse statistical text prompts, and lack of interpretability. In this work, we introduce Augur, a fully LLM driven time series forecasting framework that exploits LLM causal reasoning to discover and use directed causal associations among covariates. Augur uses a two stage teacher student architecture where a powerful teacher LLM infers a directed causal graph from time series using heuristic search together with pairwise causality testing. A lightweight student agent then refines the graph and fine tune on high confidence causal associations that are encoded as rich textual prompts to perform forecasting. This design improves predictive accuracy while yielding transparent, traceable reasoning about variable interactions. Extensive experiments on real-world datasets with **25** baselines demonstrate that Augur achieves competitive performance and robust zero-shot generalization.

1 Introduction

Time series forecasting serves as a critical task for analyzing complex dynamic systems across various domains (Wang et al., 2024b; Liang et al., 2024). The objective is to predict future time series values by leveraging historical observations collected from target systems and simultaneously observed auxiliary covariate features (Wang et al., 2024c; Chen et al., 2025; Wang et al., 2024c). In recent years, the emergence of Large Language Models (LLMs) has brought transformative opportunities to time series forecasting (Jin et al., 2024; Kong et al., 2025; Liu et al., 2025), utilizing their powerful representational capabilities to integrate multimodal data such as textual information.

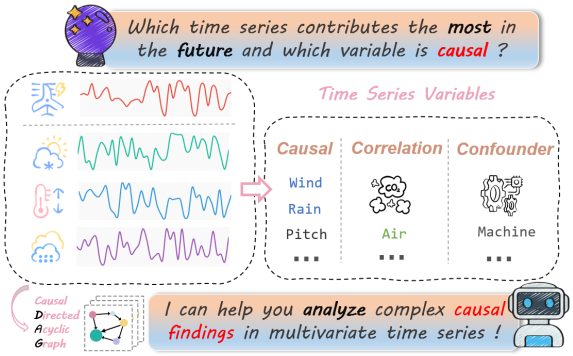

Figure 1: Motivation and problem illustration. Breathing new life into time series tasks with our Augur.

However, current LLM-based methods are hindered by several fundamental limitations: ❶ **Marginalized Role**. LLMs are typically relegated to a peripheral role, serving merely as auxiliary modules that post-process or refine representations generated by a primary forecasting model—rather than acting as the central reasoning engine. ❷ **Text Prompts**. The prompts provided to LLMs convey only coarse-grained statistical summaries (e.g., global means and variances) without encoding structured knowledge of causal among covariates. This restricts the LLM's ability to apply its native reasoning capabilities to uncover and model complex dynamic interdependencies in the data. ❸ **Interpretability**. Existing approaches generally lack transparent, systematic mechanisms to reason about variable interactions or trace how specific covariates influence final predictions. This interpretability deficit critically undermines trust and usability in high-stakes domains such as finance and healthcare (Jiang et al., 2025).

In this paper, we propose Augur, a novel framework that relies exclusively on LLM for time series forecasting. As illustrated in Figure 1, Augur uniquely leverages the causal reasoning capabilities of LLM to uncover latent causal associations among covariates in the time series. This approach

not only improves the generalization performance of forecasts but also enhances model interpretability by enabling explicit, traceable reasoning about covariate interactions.

Specifically, Augur employs a two-stage teacher–student architecture. In the first stage, a powerful pre-trained LLM acts as the teacher, identifying potential causal relationships in the time series and encoding them as a directed graph. This process combines a heuristic search-space reduction algorithm with pairwise causality tests, iteratively pruning spurious edges. In the second stage, a lightweight LLM serves as the student, refining the teacher's graph by retaining only high-confidence causal links. These validated associations, along with their textual summary rather than mere data summaries, are then converted into structured prompts to guide the student's forecasting. Extensive experiments on real-world datasets show that Augur achieves competitive forecasting accuracy and zero-shot generalization.

Contribution. ❶ To the best of our knowledge, this work presents the first exploration of LLMs' potential for analyzing causal associations among time series covariates. ❷ We propose Augur, a purely LLM-driven time series forecasting framework that leverages a teacher-student dual-stage architecture to refine causal associations and incorporate them as textual prompts, thereby enhancing both predictive accuracy and interpretability. ❸ Extensive experiments on real-world datasets with 25 baselines demonstrate that Augur achieves dominant performance.

2 Related work

Time Series Forecasting Time series forecasting is a fundamental data analysis task with broad applications across various domains (Liang et al., 2024; Huang et al., 2023; Wang et al., 2023; Ma et al., 2025b). Early approaches relied on recurrent models such as Long Short-Term Memory (LSTM) networks and TCN. Recently, Transformer, originally successful in natural language processing and computer vision, is later introduced to time series forecasting (Zhou et al., 2021, 2022b; Nie et al., 2022a). Furthermore, MLP-based architectures have emerged as lightweight alternatives (Zeng et al., 2023; Lin et al., 2024b). For instance, TimeMixer (Wang et al., 2024a) achieves competitive performance and remarkable efficiency by combining MLPs with multi-scale modeling. However, these models primarily focus on unimodal temporal dynamics and remain limited in effectively leveraging rich auxiliary modalities such as text.

LLM for Time Series Forecasting Recent efforts in time series analysis have increasingly focused on developing general-purpose foundation models, giving rise to two distinct research directions. The first direction aims to build native time series foundation models. This line of work originated with pioneering efforts such as TimeGPT-1 (Garza et al., 2023) and has since advanced rapidly, yielding significant innovations—including Chronos's novel time series tokenization scheme (Ansari et al., 2024), Lag-Llama's probabilistic forecasting framework (Rasul et al., 2023), and massively scaled architectures like TimesFM (Das et al., 2024).

The second direction explores repurposing existing large language models (LLMs) for time series forecasting by bridging the modality gap between numerical sequences and textual representations (Tan et al., 2024; Gruver et al., 2023). This stream has evolved from early fine-tuning approaches such as GPT4MTS (Jia et al., 2024) to more sophisticated, non-invasive alignment strategies—including the reprogramming framework of Time-LLM (Jin et al., 2023) and the instruction-based paradigm of UniTime (Liu et al., 2024c)—which harness the power of LLMs without modifying their core parameters.

3 Problem Formulation

Time Series. In this work, we focus on the challenge of multimodal time series. Each data instance is represented by a multimodal input pair (x, s), where $x = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^{T \times N}$ constitutes the historical sequential observations over a lookback window of length T, and N denotes the number of time series covariates. The accompanying component s encapsulates textual data that provides contextual, real-world information pertinent to the numerical observations.

Causal Explanation. Causal Explanation is denoted as a tuple (G, S), where $G = (V, E)$ is a Causal Directed Acyclic Graph (DAG) used to describe the causal associations between variable pairs, and $E \in \mathbb{R}^{N \times N}$ represents the set of edges. S means Causal Summary that describes the mechanisms encoded in G.

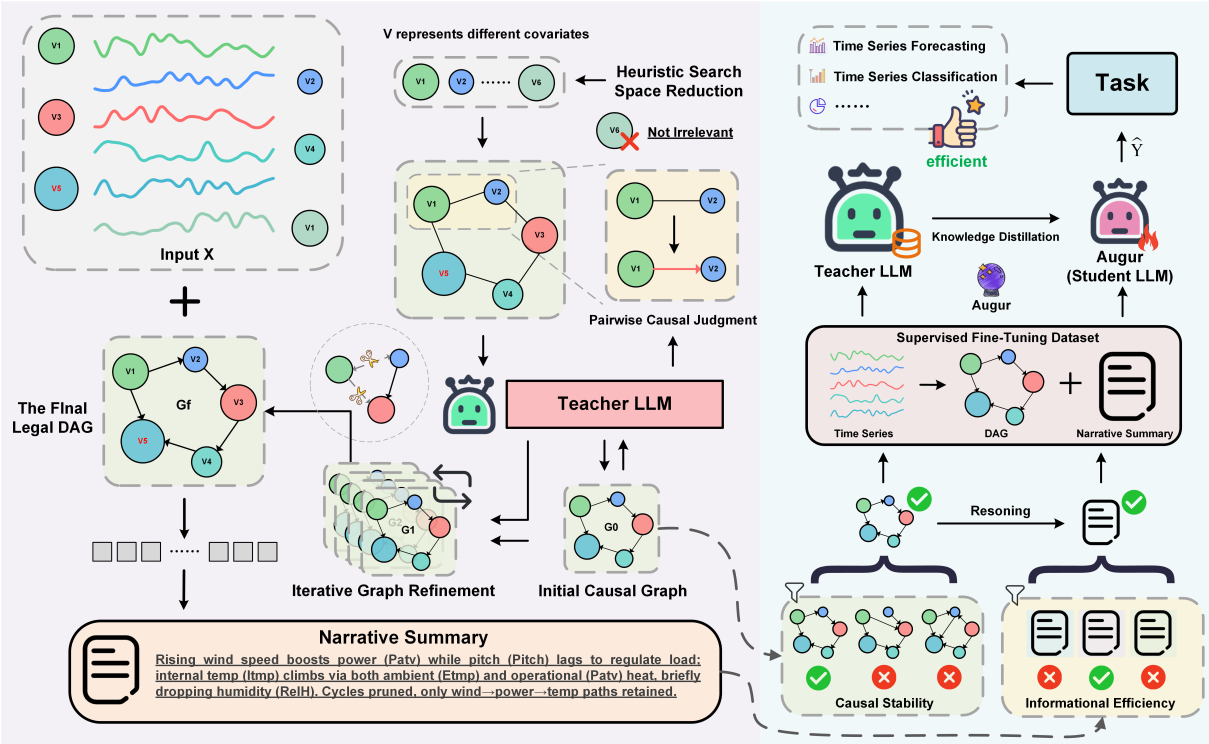

Figure 2: Overview of the Augur framework. Including causal explanation generation and student agent distillation for efficient downstream time series tasks.

Problem Definition Given historical time series data and its accompanying textual context (x, s), our goal is to predict the future value y. This value may represent discrete categories in classification tasks or continuous quantities in regression tasks. Following prior research (Zhang et al., 2025; Jiang et al., 2025), this study focuses on "discrete trend change" prediction. This emphasis arises because, in decision-critical applications such as risk assessment and strategic planning, understanding the trend direction (e.g., increase, decrease, or drastic change) is often more practically valuable than pursuing precise but uncertain continuous values.

4 Method

As shown in Figure 2 and Algorithm 1, our Augur employs a two-stage teacher-student collaborative learning process to produce accurate and interpretable time series predictions.

❶ Causal Explanation Generation via Teacher Model. We first utilize a powerful general-purpose LLM foundation model (referred to as the "Teacher" $\mathcal{M}_t$) to perform a preliminary analysis of potential causal associations within massive multivariate time series data. These causal explanations, consisting of a causal graph G_f and a corresponding narrative summary, S, along with their associated time series, are distilled into a corpus $\mathcal{D}_{\text{SFT}}$ for supervised fine-tuning the student model. Notably, the teacher model does not undergo any additional fine-tuning stages.

❷ Supervised Fine-tuning of Student Agent. We begin by refining the corpus generated by the teacher model, eliminating any false or misleading information to ensure the causal explanations are accurate and optimized for downstream tasks. These refined explanations are then utilized for supervised fine-tuning of a smaller, more efficient "Student" agent. This process enables the student model to effectively carry out specific prediction tasks with high accuracy and efficiency.

Based on the above process, a powerful pre-trained LLM (e.g., GPT-5) can serve as the teacher model, effectively guiding lightweight student models (e.g., Qwen) designed for specific prediction tasks. This approach fully leverages the strong representation and causal reasoning capabilities of the large teacher model, while significantly reducing deployment and inference costs through the lightweight student model.

4.1 Causal Explanation Generation via Teacher Model

Heuristic Search Space Reduction To make the causal discovery process tractable, the teacher

model first prunes the combinatorial space of possible edges. Based on the heuristic that significant causal links often produce detectable statistical associations, it computes the Spearman's rank correlation (Sedgwick, 2014) for all variable pairs and forms a candidate set $\mathcal{K}$ of the top-K most correlated pairs:

$$\mathcal{K} = \text{Top-K}_{(V_a,V_b)} |\rho_s(V_a, V_b)| \tag{1}$$

where $\rho_s(\cdot)$ means the Spearman's rank correlation with the time series of node V_a and node V_b as input. This focuses the subsequent, more expensive reasoning process on a high-likelihood subspace of potential causal associations.

Pairwise Causal Judgment Next, the teacher model performs a semantic lift, translating numerical patterns into causal hypotheses. For each candidate pair $(V_a, V_b) \in \mathcal{K}$, the raw time series segments x_a and x_b are serialized into textual representations, (τ_a, τ_b), by converting each numerical vector into a comma-separated string. The teacher model then evaluates a discrete hypothesis space $\mathcal{H} = \{V_a \to V_b, V_b \to V_a, \text{Confounded}, \text{Spurious}\}$ to determine the most plausible causal link:

$$h^*_{ab} = \arg\max_{h_i \in \mathcal{H}} P_{\mathcal{M}_t}(h_i \mid \tau_a, \tau_b) \tag{2}$$

The resulting set of directed edges $\{h^*_{ab}\}$ are aggregated to construct an initial global causal graph, $G_0 = (V, E_0)$.

Iterative Causal Graph Refinement The initial graph G_0 is treated as a promising but potentially inconsistent hypothesis. The teacher model refines it in an iterative loop to ensure logical consistency. At each step k, the agent receives the current graph G_{k-1} and a set of system-generated analytical critiques C_k, which are indicators of structural violations (e.g., the presence of a cycle). The teacher then proposes a graph modification ΔG_k (e.g., an edge reversal or deletion) to resolve the critique:

$$\Delta G_k = \mathcal{M}_t(G_{k-1}, C_k) \tag{3}$$

Specifically, to resolve a cycle, the teacher model is prompted with the full set of edges forming the circular dependency. It then initiates an iterative reasoning process, evaluating the plausibility of each causal link within the context of the entire cycle and its embedded domain knowledge. The model deliberates to identify the link that represents the weakest or least plausible causal associations, designating it for removal. The resulting modification, ΔG_k, represents this reasoned, context-aware decision from the model. The new graph is formed by $G_k = G_{k-1} \oplus \Delta G_k$, where $\oplus$ denotes the application of the modification to the graph's edge set. This process continues until no critiques remain ($C_k = \emptyset$), yielding a final, validated DAG, G_f.

Algorithm 1 The model process of Augur

Require: Dataset $\mathcal{D} = \{x_i\}$; Teacher $\mathcal{M}_t$; Student $\mathcal{M}_s^{(0)}$; Parameters $K, \lambda, K_{\max}, \tau$
Ensure: Trained student model $\mathcal{M}_s$
Initialize SFT dataset: $\mathcal{D}_{\text{SFT}} \leftarrow \emptyset$
for each sample $x \in \mathcal{D}$ **do**
  $\mathcal{K} \leftarrow \text{Prune}(x, K, \tau)$
  $E_0 \leftarrow \text{JudgePairs}(x, \mathcal{K}, \mathcal{M}_t)$
  $G_0 \leftarrow (V, E_0)$
  $(G_f, \mathcal{I}) \leftarrow \text{Refine}(G_0, \mathcal{M}_t, K_{\max})$
  $S \leftarrow \text{Narrate}(\mathcal{M}_t, G_f, \mathcal{I})$
  $q \leftarrow \text{Score}(x, G_f, S, \lambda)$
  if $q \geq \alpha$ **then**
    $\mathcal{D}_{\text{SFT}} \leftarrow \mathcal{D}_{\text{SFT}} \cup \{(x, G_f, S)\}$
  end if
end for
$\mathcal{M}_s \leftarrow \text{FineTune}(\mathcal{M}_s^{(0)}, \mathcal{D}_{\text{SFT}})$
return $\mathcal{M}_s$

Grounded Narrative Synthesis Finally, the teacher model synthesizes a coherent narrative summary, S. It is conditioned on the validated graph G_f and the set of key modifications $\mathcal{I}$ made during refinement (e.g., edges that were removed to break cycles), ensuring the summary is fully grounded in the final causal structure:

$$S = \mathcal{M}_t(G_f, \mathcal{I}) \tag{4}$$

We combine the causal explanations generated by the teacher model with their corresponding time series into a corpus, which is denoted as $\mathcal{D}_{\text{SFT}}$. Please note that this corpus only involves a subset of variables from the used datasets.

4.2 Distillation and Training of Student Agent

After generating a corpus dataset from millions of time series instances, we introduce a distillation process to train a specialized student agent. However, the raw outputs from the teacher model exhibit variable quality and are not uniformly reliable for direct use in training. Therefore, a critical inter-

mediate step is to curate this dataset by scoring and filtering for the highest-quality causal explanations.

Causal Stability ($\mathcal{F}_{\text{s}}$). We adopt a consensus-based approach to identify the most robust causal structure. For a given time series x, we first generate a set of N diverse candidate DAGs, $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$ through multiple sampling. We then score each candidate graph G_k based on its structural agreement with all other candidates in the set. The graph with the highest cumulative overlap of causal edges is considered the most stable and reliable explanation. The stability score for a graph G_k with edge set E_k is defined as the sum of shared edges with all other graphs in the ensemble:

$$\mathcal{F}_{\text{s}}(G_k|\mathcal{G}) = \sum_{j=1}^{N} |E_k \cap E_j| \tag{5}$$

The final graph selected for the quality function is the one that maximizes this stability score, $G^* = \arg\max_{G_k \in \mathcal{G}} \mathcal{F}_{\text{s}}(G_k|\mathcal{G})$.

Informational Efficiency ($\mathcal{F}_{\text{e}}$). This term rewards explanations that are both concise and logically grounded. It combines a precision-based Groundedness Score (S_G) with a penalty for the summary's length $|S|$. S_G is calculated as the proportion of causal claims extracted from the summary text S that have a corresponding edge in the graph G:

$$\mathcal{F}_{\text{e}} = S_G(S, G) - \lambda \cdot |S| \tag{6}$$

Finally, we evaluate the overall quality of each causal explanation by considering both causal stability and informational efficiency scores to select only the highest-quality explanations for our training corpus.

Supervised Fine-Tuning. Our training data is composed of the curated set of optimal pairs $\{(x_i, G_i^*, S_i^*)\}_{i=1}^{M}$. Each target explanation (G_i^*, S_i^*) is serialized into a single text sequence, Y_i^*. The student agent is then fine-tuned to map a given time series x_i to its target explanation Y_i^* by minimizing the standard cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i} \log P(Y_i^*|x_i; \theta_{\text{s}}) \tag{7}$$

This distillation process transfers the complex, multi-step reasoning of the teacher into a single, efficient student model.

4.3 Utility of the Causal Summary

The synthesized Causal Summary (S_g) provides critical utility by translating the formal, complex Causal DAG (G_f) into a human-readable narrative. This validated causal structure then serves as a definitive guide for feature selection, enabling the construction of sparse, robust predictive models based on the true causal drivers (the Markov Blanket) while explicitly excluding known confounders or downstream effects. Furthermore, the summary text S itself becomes a powerful asset; it can be injected back into a multi-modal forecasting model as a rule-based instruction or an informative prior. When provided alongside new numerical data, this text acts as a physics constraint, ensuring the model adheres to the known causal logic to dramatically improve its forecasting accuracy and robustness, especially in novel or out-of-distribution scenarios.

5 Experiment

In this section, we conduct extensive experiments to answer the following research questions (RQs):

- (**RQ1**) How does our Augur perform in time series forecasting tasks?
- (**RQ2**) How is the quality of the causal summaries generated by our Augur?
- (**RQ3**) Is every component of Augur efficient?
- (**RQ4**) Are there marginal effects in causal explanations?

5.1 Experiment Setup

Datasets. We employ four time series datasets from diverse real-world domains for evaluation, spanning air, transportation, energy, and finance. These datasets not only contain rich and meaningful covariate features but also exhibit clearly identifiable causal structures. For instance, in the air dataset, meteorological conditions and holiday indicators demonstrate significant and interpretable effects on air pollution dynamics. More details can be found in Appendix C.1.

Data Processing. We create a hybrid dataset using the LargeAQ air quality dataset (Ma et al., 2025a) and the SDWPF power dataset (Zhou et al., 2022a) for causal association analysis in the teacher model and fine-tuning of the student model in our Augur. Our initial training data contains over 100 billion time points and more than 10 million distinct causal events. Traffic and Finance datasets, on the other hand, are directly used to evaluate the model's zero-shot generalization performance.

Table 1: Performance comparison for Augur on multivariate time series with their pretrained counterparts. Best results are in **pink**, and second-best are underlined blue.

Model	Air		Power		Traffic		Finance	
	F1-Score	AUROC	F1-Score	AUROC	F1-Score	AUROC	F1-Score	AUROC
Time-LLM	0.826	0.907	0.744	0.796	0.617	0.686	0.609	0.708
GPT4TS	0.803	0.864	0.716	0.777	0.562	0.621	0.581	0.652
Moirai	0.876	0.928	0.797	0.858	0.706	0.787	0.676	0.767
Chronos	0.839	0.921	0.789	0.849	0.698	0.769	0.665	0.744
Time-MoE	0.891	0.941	0.818	0.879	0.718	0.803	0.681	0.762
Informer	0.846	0.903	0.764	0.844	0.676	0.748	0.646	0.717
Autoformer	0.803	0.908	0.757	0.836	0.684	0.756	0.653	0.724
FEDFormer	0.859	0.912	0.781	0.841	0.688	0.761	0.626	0.707
Crossformer	0.844	0.897	0.767	0.840	0.681	0.733	0.658	0.711
DLinear	0.745	0.841	0.642	0.748	0.516	0.596	0.523	0.615
iTransformer	0.878	0.929	0.793	0.864	0.713	0.764	0.684	0.755
PatchTST	0.885	0.936	0.823	0.881	0.724	0.803	0.696	0.748
LightTS	0.761	0.858	0.674	0.761	0.523	0.595	0.551	0.622
TimesNet	0.864	0.919	0.759	0.851	0.703	0.751	0.667	0.736
SparseTSF	0.640	0.793	0.628	0.755	0.531	0.612	0.502	0.531
PatchMixer	0.805	0.869	0.702	0.768	0.542	0.613	0.568	0.659
CycleNet	0.811	0.874	0.708	0.773	0.538	0.609	0.524	0.635
TimeMixer	0.889	0.939	0.813	0.884	0.735	0.806	0.692	0.763
TimeXer	0.873	0.924	0.795	0.855	0.706	0.785	0.672	0.744
Augur	**0.928**	**0.958**	**0.849**	**0.909**	**0.751**	**0.825**	**0.705**	**0.783**

Task Setting. Following prior work (Zhang et al., 2025; Jiang et al., 2025), we reformulate the forecasting target as a more practical, robust, and interpretable trend prediction task. Specifically, for the four datasets, our task settings are defined as follows. ❶ **Power**: Using the past 24 hours of operational data, we predict whether the wind power output over the next 24 hours will exceed its historical average. ❷ **Air**: Given 48 hours of historical air quality and weather records, we predict whether a severe-level pollution event will occur in the subsequent 24 hours. ❸ **Traffic**: Based on the past 96 hours of data, we classify the average traffic flow trend over the next 24 hours as rising, stable, or falling. ❹ **Finance**: Using the trend from the past four days, we classify whether the stock trend for the next day will be up, down, or neutral.

Evaluation Metrics. For prediction tasks, we use two widely used classification metrics: F1-Score and AUROC. To evaluate the quality of the generated causal summaries, we introduced five metrics for comprehensive assessment: BLEU(Papineni et al., 2002) and ROUGE-L(Lin, 2004) to measure lexical overlap, BERTScore (Zhang et al., 2019) to compare contextual embeddings of the text, Perplexity (PPL) to assess language fluency, and the total length of the summary (in terms of tokens) to evaluate conciseness. Finally, we also conduct human evaluations, with the detailed evaluation criteria provided in Appendix C.1.1.

Baselines. Our experiment compares **25 advanced baselines**. ❶ **For TS prediction tasks**, we use the latest *LLM-based models* such as TimeLLM (Jin et al., 2023), GPT4TS (Jia et al., 2024), Moirai (Woo et al., 2024), Chronos (Ansari et al., 2024), and Time-MoE (Shi et al., 2024), as well as *Classic Unimodal Time Series Models* including Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), FEDFormer (Zhou et al., 2022b), DLinear (Zeng et al., 2023), iTransformer (Liu et al., 2023), PatchTST (Nie et al., 2022b), LightTS (Campos et al., 2023), TimesNet (Wu et al., 2022), SparseTSF (Lin et al., 2025), PatchMixer (Gong et al., 2023), CycleNet (Lin et al., 2024a), TimeMixer (Wang et al., 2024a), and TimeXer (Wang et al., 2024c). ❷ **For the qualitative evaluation of the causal summaries**, we use powerful LLMs including LLaMA3.1-8B (Touvron et al., 2023), GPT-4o (Hurst et al., 2024), Gemini2.0-flash (Team et al., 2023), DeepSeek-v3 (Liu et al., 2024a), Qwen-3-14B (Yang et al., 2025), and ChatTS (Xie et al., 2024).

Table 2: Comprehensive automatic evaluation of summary generation, with datasets on the horizontal axis (**Power** and **Air**). Best results are in **pink**, and second-best are underlined blue.

Method	Power					Air				
	ROUGE-L	BLEU	BERTScore	PPL	Tokens	ROUGE-L	BLEU	BERTScore	PPL	Tokens
LLaMA3.1-8B	0.24	0.34	0.71	34.8	1955	0.21	0.35	0.74	34.2	1751
GPT-4o	0.29	0.37	0.74	29.8	1854	0.29	0.38	0.79	26.5	1756
Gemini2.0-Flash	0.32	0.42	0.80	25.0	1365	0.36	0.47	0.82	22.5	986
ChatTS-12B	0.32	0.46	0.83	20.5	1134	0.33	0.43	0.84	21.2	1022
DeepSeek-v3	0.34	0.51	0.87	16.5	2010	0.37	0.52	0.85	15.8	1827
Qwen3-14B	0.37	0.45	0.84	15.2	1967	0.39	0.48	0.86	14.7	1788
Augur	**0.49**	**0.56**	**0.89**	**12.3**	2317	**0.52**	**0.57**	**0.91**	**11.6**	2108

Implementation. In the causal explanation extraction stage, we use `GPT-5` as the teacher model and `gemini-2.5-flash-lite` to perform the initial causal judgment and graph refinement steps. During the fine-tuning phase, we use `Qwen3-8b` as our student agent. We used the AdamW optimizer with a cosine learning rate scheduler. We fine-tuned the model for 3 epochs with a global batch size of 64. All datasets are split chronologically into training, validation, and test sets using a 7:2:1 ratio. Text inputs available for each time step and a static setting where one description applies to the entire series. For the unimodal model, we do not use text data. All reported metrics are the mean of five independent runs. Further implementation details are provided in the Appendix C.2.

5.2 Prediction Performance Study (RQ1)

We conducted comprehensive experiments to validate the effectiveness of Augur. The experimental results, as shown in Table 1, demonstrate that Augur achieves the best predictive performance. For traditional time series forecasting models, PatchTST adheres to the principle of independent channel learning, resulting in relatively strong prediction performance. TimeMixer, leveraging a multi-scale modeling approach, effectively captures complex temporal dynamics, thereby achieving the best performance among traditional models while also ensuring competitive zero-shot performance on the Traffic and Finance datasets. Among LLM-based models, Time-MoE achieves the best performance due to its billion-parameter-scale mixture-of-experts architecture, which can effectively capture complex temporal dynamics while maintaining considerable zero-shot generalization capabilities. Our model, Augur, fully leverages the causal inference and analytical capabilities of large models. Its ability to accurately model variable dependencies significantly enhances predictive accuracy. Furthermore, the results on the Traffic and Finance datasets validate its superior generalization performance in zero-shot scenarios.

Table 3: Human evaluation results: (**EoU**) ease of understanding, (**Ins.**) insightfulness for interpretation, and (**Corr.**) causal correctness.

Dataset	Method	Evaluation Metrics			
		EoU	Ins.	Corr.	Avg.
Power	GPT-4o	4.3	3.9	4.2	4.1
	Qwen3	4.6	4.4	3.7	4.2
	Augur	**4.7**	**5.2**	**4.9**	**4.9**
Air	GPT-4o	4.5	4.0	4.3	4.3
	Qwen3	**5.1**	4.5	3.8	4.5
	Augur	4.9	**5.5**	**5.0**	**5.1**

Table 4: Ablation of Augur's core components on the Power dataset.

Variant	Time (×)	Prune	Judge	Refine	F1	AUC
w/o Prune	5.3	✗	✓	✓	0.81	0.88
w/o Judge	1.7	✓	✗	✓	0.76	0.84
w/o Refine	0.6	✓	✓	✗	0.79	0.87
Augur	**1.0**	✓	✓	✓	**0.85**	**0.91**

5.3 Quality of Causcal Summary (RQ2)

We evaluated the quality of causal summaries generated by various LLMs using the Power and Air datasets. The quantitative metrics, presented in Table 2, and the human evaluation results, detailed in Table 3, highlight significant differences in performance. Among the models, LLaMA3.1-8B produced the lowest-quality summaries, likely due to its smaller parameter size (8B), which limits its ability to capture complex semantic patterns in multivariate time series data. In contrast, DeepSeek-v3 and Qwen3-14B demonstrated superior causal

Figure 3: Impact of narrative granularity on zero-shot classification performance across four datasets.

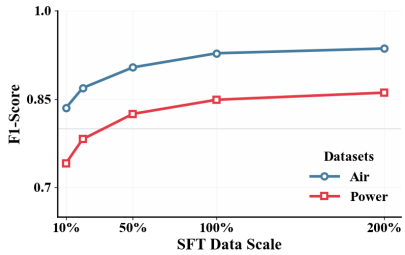

Figure 4: Impact of the SFT data scaling up.

reasoning capabilities, consistently outperforming other models in summary generation. Our model, Augur, employs a teacher-student two-stage framework to distill and summarize causal associations more effectively. This framework enables Augur to produce higher-quality causal summaries, as validated by both quantitative metrics and human evaluation results. Our model, Augur, leverages a teacher-student two-stage framework to uncover and summarize the underlying causal associations in the data more deeply, thereby generating higher-quality causal summaries.

5.4 Ablation Study (RQ3)

In this section, we conduct an ablation study to isolate the contribution of its core components. We create three ablated versions: (1) **w/o Prune**, which performs causal judgment on all variable pairs without initial filtering; (2) **w/o Judge**, which bypasses LLM-based causal reasoning and relies on a simpler correlation-based heuristic to orient edges; and (3) **w/o Refine**, which uses the initial, unrefined graph without ensuring global consistency.

As detailed in Table 4, removing the initial Prune step significantly increases computational cost, while omitting the LLM-based Judge step leads to the most substantial drop in predictive accuracy. Furthermore, disabling the Refine stage also degrades performance, highlighting the necessity of ensuring a globally coherent causal graph and validating our architectural choices.This design allows us to quantify the necessity of each step—pruning for efficiency, judgment for causal accuracy, and refinement for structural coherence.

5.5 Analysis of Narrative Granularity (RQ4)

We conduct an additional analysis to evaluate the relationship between the granularity of causal explanations and downstream task performance. In our zero-shot forecasting tasks on the Traffic and Finance datasets, we systematically varied the textual inputs, constructing variants ranging from raw time series data with only high-level summaries to progressively more detailed causal discoveries. This approach allowed us to quantify the marginal utility of each additional discovery.

As shown in Figure 3, model performance improves significantly when at least one key causal discovery is included, compared to using only high-level summaries. Further analysis reveals a clear point of diminishing returns: while the first two causal discoveries contribute substantial gains, adding a third or subsequent minor discoveries provides negligible benefits.

Figure 4 indicates that a carefully selected subset of high-quality data yields greater performance improvements than simply expanding the dataset volume. Scaling the supervised fine-tuning corpus to 200% proves disproportionately costly and, according to our analysis, fails to enhance results. This supports our hypothesis that for this task, a quality-focused "less-is-more" strategy outperforms data-intensive approaches.

6 Conclusion

In this paper, we present Augur, a framework that leverages large language models to extract explicit causal associations among covariates, thereby enhancing both forecasting capability and interpretability. The method implements a teacher-student architecture to generate high-quality causal explanations. These explanations are subsequently utilized as textual prompts to guide the student LLM in making predictions. Extensive experiments across four diverse domains demonstrate that Augur outperforms multiple SOTA time series baselines in time series forecasting tasks while exhibiting effective zero-shot generalization performance.

Limitations

Our approach fundamentally relies on the assumption of causal sufficiency (i.e., no unobserved confounders) and employs a correlation-based heuristic that may overlook complex non-linear or lagged dependencies. Second, the quality of the generated narratives is contingent on the teacher LLM's internal knowledge, which may be incomplete or biased in highly specialized domains.

Ethics Statement

All datasets and language models used in this work are publicly available and comply with relevant licensing terms. No personally identifiable information (PII) or sensitive data was collected or used.Five annotators with formal training in logic provided informed consent and were fairly compensated. Evaluation protocols included clear rubrics to minimize subjective bias. All evaluated data was anonymized.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, and 1 others. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. 2023. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27.

Wei Chen, Yuqian Wu, Yuanshao Zhu, Xixuan Hao, Shiyu Wang, and Yuxuan Liang. 2025. Select, then balance: A plug-and-play framework for exogenous-aware spatio-temporal forecasting. *arXiv preprint arXiv:2509.05779*.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Zihan Dong, Xinyu Fan, and Zhiyuan Peng. 2024. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4918–4927.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. 2023. Timegpt-1. *arXiv preprint arXiv:2310.03589*.

Zeying Gong, Yujin Tang, and Junwei Liang. 2023. Patchmixer: A patch-mixing architecture for long-term time series forecasting. *arXiv preprint arXiv:2310.00655*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36:46885–46902.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. 2024. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23343–23351.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. 2025. Explainable multi-modal time series prediction with llm-in-the-loop. *arXiv preprint arXiv:2503.01013*.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and 1 others. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. 2024. Position: What can large language models tell us about time series analysis. In *41st International Conference on Machine Learning*. MLResearchPress.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025. Time-mqa: Time series multi-task question answering with context enhancement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 29736–29753.

Bo Li, Ruotao Yu, Zijun Chen, Yingzhe Ding, Mingxia Yang, Jinghua Li, Jianxiao Wang, and Haiwang Zhong. 2024. High-resolution multi-source traffic data in new zealand. *Scientific Data*, 11(1):1216.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th*

ACM SIGKDD conference on knowledge discovery and data mining, pages 6555–6565.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. 2024a. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and CL Philip Chen. 2025. Sparsetsf: Lightweight and robust time series forecasting via sparse modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. 2024b. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Chenxi Liu, Hao Miao, Qianxiong Xu, Shaowen Zhou, Cheng Long, Yan Zhao, Ziyue Li, and Rui Zhao. 2025. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. *arXiv preprint arXiv:2505.02138*.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and 1 others. 2024b. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. 2024c. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, pages 4095–4106.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Jiaming Ma, Zhiqing Cui, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Zhe Zhao, and Yang Wang. 2025a. Causal learning meet covariates: Empowering lightweight and effective nationwide air quality forecasting.

Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. 2025b. Bist: A lightweight and efficient bi-directional model for spatiotemporal prediction. *Proceedings of the VLDB Endowment*, 18(6):1663–1676.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022a. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022b. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, and 1 others. 2023. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Philip Sedgwick. 2014. Spearman's rank correlation coefficient. *Bmj*, 349.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2024. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*.

Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. 2024. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. 2023. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2223–2232.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024a. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. 2024b. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. 2024c. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. 2024. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiyuan Zhang, Boran Han, Haoyang Fang, Abdul Fatir Ansari, Shuai Zhang, Danielle C Maddix, Cuixiong Hu, Andrew Gordon Wilson, Michael W Mahoney, Hao Wang, and 1 others. 2025. Does multimodality lead to better time series forecasting? *arXiv preprint arXiv:2506.21611*.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. 2022a. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *arXiv preprint arXiv:2208.04360*.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022b. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

Appendix

This appendix provides supplementary material organized into several sections to support the main paper. Each section is dedicated to a specific topic:

- **Appendix A** establishes the theoretical foundations with a self-contained overview of Causal Directed Acyclic Graphs (DAGs).
- **Appendix B** presents a formal proof of causal feature optimality, building upon the theoretical groundwork.
- **Appendix C** details our comprehensive methodology and implementation, including dataset and baseline descriptions, evaluation protocols, and the technical environment.
- **Appendix D** offers supplementary results and analyses, featuring a quantitative evaluation of our feature selector and an in-depth case study that illustrates the entire pipeline, from initial analysis to the final causal narrative.
- **Appendix E** concludes with a discussion on framework extensibility and, finally, presents the specific prompts used to guide the LLM-driven processes.

A Causal Directed Acyclic Graphs

This appendix provides a brief overview of the key concepts from the theory of causal inference based on Directed Acyclic Graphs (DAGs) that are used in this paper.

A.1 DAGs and Observational Distributions

A causal Directed Acyclic Graph (DAG) is a graph $\mathcal{G} = (V, E)$ where nodes $V = \{X_1, \ldots, X_n\}$ represent random variables and directed edges E represent direct causal associations, with no directed cycles. The graph structure encodes the *causal Markov property*: every variable is assumed to be

independent of its non-descendants given its direct causes (parents), denoted $\mathrm{Pa}_{\mathcal{G}}(X_i)$. This property implies that the joint observational distribution $P(V)$ factorizes according to the graph:

$$P(V) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}_{\mathcal{G}}(X_i)). \tag{8}$$

A.2 Interventions and Causal Effects

A causal effect is defined via a surgical intervention on the system, formalized by Pearl's do-operator. An intervention $do(X_k = x)$ sets the variable X_k to a constant value x, severing the influence of its natural parents. This corresponds to a modified graph where all edges into X_k are removed. The post-interventional distribution is obtained via a *truncated factorization*:

$$P(V \mid do(X_k)) = \prod_{i \neq k} P(X_i \mid \mathrm{Pa}_{\mathcal{G}}(X_i)). \tag{9}$$

A.3 Paths, d-Separation, and Confounding

Associations between variables in a DAG are transmitted along paths. A *back-door path* from a treatment X to an outcome Y is a path that begins with an edge into X (e.g., $X \leftarrow \dots$). Such paths are non-causal and can create spurious associations due to common causes (*confounders*). A node on a path is a *collider* if both edges on the path point into it (e.g., $A \rightarrow C \leftarrow B$). The concept of *d-separation* determines conditional independence: a set of nodes Z d-separates X and Y if it blocks every path between them. A path is blocked by Z if it contains either (1) a non-collider that is in Z, or (2) a collider that is not in Z and has no descendants in Z. If all paths are blocked, then $X \perp\!\!\!\perp Y \mid Z$.

A.4 Identifiability via the Back-door Criterion

The causal effect $P(y \mid do(x))$ can be identified from observational data if confounding can be appropriately controlled. The *back-door criterion* provides a sufficient condition for this. A set of variables Z satisfies the back-door criterion relative to (X, Y) if: (1) no node in Z is a descendant of X, and (2) Z blocks every back-door path between X and Y. If such a set exists, the causal effect is identifiable via the *back-door adjustment formula*:

$$P(y \mid do(x)) = \sum_{z} P(y \mid x, z)\, P(z). \tag{10}$$

A.5 A Consolidated Example

Consider a causal model represented by the DAG with edges $\{Z \rightarrow X, Z \rightarrow Y, X \rightarrow Y, X \rightarrow C \leftarrow Y\}$. Here, X is the treatment and Y is the outcome.

- **Confounding:** The path $X \leftarrow Z \rightarrow Y$ is a back-door path created by the common cause (confounder) Z. It induces a spurious association between X and Y. To estimate the causal effect of X on Y, this path must be blocked.
- **Collider:** The node C is a collider. The path $X \rightarrow C \leftarrow Y$ is naturally blocked. Conditioning on C would open this path, inducing a spurious association, and is therefore incorrect.
- **Adjustment:** The set $\{Z\}$ satisfies the back-door criterion. Z is not a descendant of X, and conditioning on Z blocks the back-door path $X \leftarrow Z \rightarrow Y$. Therefore, the causal effect is identifiable by adjusting for Z:

$$P(y|\mathrm{do}(x)) = \sum_{z} P(y|x, z)P(z). \tag{11}$$

B A Proof of Causal Feature Optimality

We provide a rigorous, first-principles proof that the mutual information between a set of features and a target variable is maximized when the feature set is the target's causal Markov Blanket.

Theorem 1.

Given a system of variables V and a target Y with a known causal DAG $\mathcal{G}$, let $X_c = \mathrm{MB}_{\mathcal{G}}(Y)$ be the causal Markov Blanket of Y. Then, the mutual information between X_c and Y is equal to the mutual information between the entire system V and Y:

$$I(X_c; Y) = I(V; Y) \tag{12}$$

Proof. The proof relies on showing the equality of the conditional entropies, $H(Y \mid V) = H(Y \mid X_c)$, from which the theorem follows directly from the definition of mutual information, $I(A; B) = H(B) - H(B \mid A)$.

The conditional entropy $H(Y \mid V)$ is defined as:

$$H(Y \mid V) = -\sum_{v \in V} p(v) \sum_{y \in Y} p(y \mid v) \log p(y \mid v) \tag{13}$$

By the causal Markov property, Y is conditionally independent of all variables in $V \setminus X_c$ given X_c. This implies that for any realization v of V, where v_c is the portion corresponding to X_c:

$$p(y \mid v) = p(y \mid v_c) \tag{14}$$

Substituting (14) into (13), we obtain:

$$H(Y \mid V) = -\sum_{v \in V} p(v) \sum_{y \in Y} p(y \mid v_c) \log p(y \mid v_c) \tag{15}$$

We can now regroup the summation over all $v \in V$ by summing over the components $v_c \in X_c$ and $v' \in V \setminus X_c$, and then apply the law of total probability to marginalize over v':

$$\begin{aligned} H(Y \mid V) &= -\sum_{v_c, v'} p(v_c, v') \cdot \left(\sum_y p(y|v_c) \log p(y|v_c) \right) \\ &= -\sum_{v_c} \left(\sum_{v'} p(v_c, v') \right) \cdot \left(\sum_y p(y|v_c) \log p(y|v_c) \right) \\ &= -\sum_{v_c} p(v_c) \left(\sum_y p(y|v_c) \log p(y|v_c) \right) \end{aligned} \tag{16}$$

The final expression in (16) is precisely the definition of the conditional entropy $H(Y \mid X_c)$. Thus, we have shown:

$$H(Y \mid V) = H(Y \mid X_c) \tag{17}$$

From this equality, it directly follows that:

$$H(Y) - H(Y \mid V) = H(Y) - H(Y \mid X_c) \tag{18}$$

which proves the theorem that $I(V;Y) = I(X_c;Y)$. □

C Experimental Setup

C.1 Description of Datasets

The **Power** dataset is sourced from **SDWPF** (Zhou et al., 2022a), which was collected over two years (2020–2021) from a wind farm comprising 134 turbines. It contains over 11 million high-resolution records (sampled every 10 minutes), integrating SCADA sensor measurements with ERA5 meteorological reanalysis data. To visualize inter-variable relationships, we compute the Spearman correlation matrix, shown in Figure 5. This matrix reveals dependency patterns among several key covariates, providing valuable guidance for our causal discovery process.

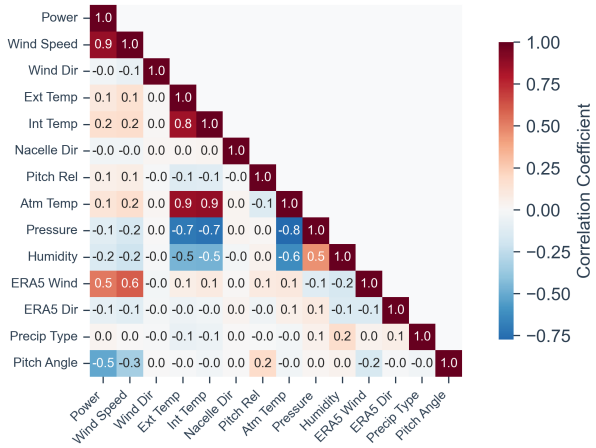

Figure 5: Spearman correlation matrix of the variables in the power dataset.

The **Air** dataset is from **LargeAQ**, a nationwide air-quality dataset spanning eight years (2015–2023). Each station provides time-stamped observations of major *criteria pollutants* together with rich *meteorological covariates*. Records are provided at (predominantly) hourly cadence, enabling long-horizon AQI research and spatiotemporal modeling.

The **Traffic** dataset is from **NZ-Traffic** (Li et al., 2024), which spans a nine-year period. It aggregates data from 2,042 sensors across New Zealand's highway network, encompassing over 600 million high-resolution records (15-min intervals). Each entry provides granular vehicle counts, distinguishing between light-duty and heavy-duty vehicles. This core traffic data is fused with rich contextual information, including key meteorological covariates (e.g., temperature, precipitation) from NOAA and extensive metadata detailing highway structure, coastlines, and public holidays.

The **finance** dataset is from **FNSPID** (Dong et al., 2024), spans nearly a quarter-century (1999–2023). It covers 4,775 companies from the SP 500 index and comprises over 29.7 million stock price records alongside 15.7 million financial news articles sourced from four major outlets.

In our experiments, we use the most important

variable as targets and the other variables as covariates; station metadata are used only for grouping and reporting and are not injected as numeric inputs unless explicitly noted.

C.1.1 Evaluation Metrics

Automatic Evaluation. Our evaluation of summary quality is comprehensive. First, to measure content similarity, we compare our generated summaries against reference texts using a suite of metrics. We employ the classic n-gram-based metrics, BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004), to assess lexical overlap. To capture deeper semantic meaning and properly handle paraphrasing, we also include BERTScore (Zhang et al., 2019), which compares the contextual embeddings of the texts. Second, we assess linguistic fluency using Perplexity (PPL). A lower perplexity score indicates that the generated text is more coherent, grammatically sound, and aligns well with the patterns of natural language. Finally, we evaluate conciseness by measuring the summary's total length in tokens.

Human Evaluation. We conduct a human study to assess how readers perceive causal summaries produced by different methods. For each dataset, we randomly sample 50 instances per method. Five annotators with formal training in logic (at least undergraduate level) independently rate each summary on a 7-point scale along three dimensions: (1) ease of understanding, (2) insightfulness for interpreting the time series, and (3) causal correctness in reflecting inter-variable relationships.

C.2 Description of Baselines

Unless otherwise specified, all baseline implementations, data pipelines, and default hyperparameters follow the open-source library **MM-TSFlib**(Liu et al., 2024b)[1]. For Time-LLM and similar models, we follow the authors' public implementations. We keep their official training/evaluation protocols for both numeric-only time-series models and LLM-aligned variants to ensure a fair and reproducible comparison.

Process. We adopt the unified preprocessing: time alignment to a single grid, forward-fill then mean-impute missing values, per-variable z-score standardization with training-split statistics with strictly non-overlapping temporal splits to avoid leakage.

Optimization. Our supervised fine-tuning (SFT) is implemented using the LLaMA-Factory framework. We used the AdamW optimizer with a cosine learning rate scheduler. The initial learning rate was set to 2×10^{-5}, with a warmup ratio of 0.1 and a weight decay of 0.01. We fine-tuned the model for 3 epochs with a global batch size of 64.

For time series forecast model, we use AdamW with cosine decay and 5% warmup, gradient clipping at 1.0, mixed precision, and early stopping on validation loss (patience 10). Learning rate is selected from $\{1\mathrm{e}{-3},\ 5\mathrm{e}{-4},\ 1\mathrm{e}{-4}\}$ for numeric-only baselines and $\{2\mathrm{e}{-4},\ 1\mathrm{e}{-4},\ 5\mathrm{e}{-5}\}$; batch size from $\{32,\ 64,\ 128\}$ subject to memory. Max epochs 100 with model selection on validation performance. Probabilistic baselines use 100 samples to form point estimates.

Other models. For the causal *judge* and *refine* stages, each iteration's decisions (edge proposals and cycle-resolution edits) are generated by a lightweight `Gemma-2B` model, which we use to produce pairwise causal labels and graph updates until convergence.

We query the vendors' official APIs. For open-source models, we run `Qwen-3-8B`, `ChatTS-14B-0801`[2] and `LLaMA-3.1-8B` locally. Unless otherwise stated, the OpenAI endpoint uses `gpt-4o-2024-08-06` (GPT-4o). We apply the same decoding configuration to `Gemini-2.5-Flash-Lite` and `gpt-5-mini`,We set `max_tokens`$=4096$ and use `temperature`$=0.5$ for analysis-style generation (self-reflection and textual refinement), and `temperature`$=0.1$ for prediction and causal judgments, which yielded the most stable empirical behavior in our preliminary tests. All prompts share identical instruction templates across providers, with only minimal schema-specific tokens adjusted for compatibility. All evaluations use greedy decoding, and we retain all other settings from the official HuggingFace configurations.

Cost. We synthesize raw explanations via API calls at an average cost of \$1–\$2 per instance and will release the full supervised fine-tuning corpus under an open license. As the student model, we fine-tune `Qwen3-8B`.

[1] https://github.com/AdityaLab/MM-TSFlib

[2] https://github.com/NetManAIOps/ChatTS

C.3 Human Evaluation

In our human evaluation, we compare GPT-4o, Qwen3, and Augur on the power and air datasets. For each dataset and system, we uniformly sample 50 instances. Three annotators—each with formal training in logic and basic knowledge of meteorology or energy systems—independently rate every summary. For each annotator, the item order is independently randomized. Ratings follow a 7-point Likert scale across three dimensions: Ease of Understanding (EoU), Insightfulness (Ins.), and Causal Correctness (Corr.).

C.4 Environment

All experiments were conducted on a TensorEX server equipped with two Intel Xeon Gold 5218R CPUs, and four NVIDIA A100 80GB GPUs.

C.5 Rationale for Supervised Fine-Tuning

Our supervised fine-tuning (SFT) dataset consists of input-target pairs serialized into single text sequences. This format trains the model to map numerical time series and a prompt to a structured causal explanation.

The input sequence contains the numerical data and task instruction, delineated by the `<|data|>` and `<|task|>` tokens. The target sequence contains the ground-truth causal graph and a summary, separated by the `<|graph|>` and `<|summary|>` tokens. The `<|EOT|>` token marks the end of both sequences.

D Supplementary Results and Analyses

D.1 Additional Experiments

To evaluate the effectiveness of our proposed LLM-driven feature selector, **Augur**, we conducted a comprehensive comparative analysis. We tested its performance against two baseline feature sets: one utilizing all available variables (All Features) and a univariate approach using only the most direct predictor (Wind Speed). These three feature sets were evaluated across four distinct forecasting architectures: MLP, LSTM, DLinear, and PatchTST. We report the Mean Squared Error (MSE) and Mean Absolute Error (MAE) for each experiment in Table 5, with lower values indicating better performance.

The results clearly demonstrate the superiority of the Augur methodology for most models. By identifying a more informative and less noisy subset of variables, Augur consistently yielded the best performance for the MLP, LSTM, and DLinear architectures.while Augur provides a significant advantage for recurrent and linear models, advanced Transformer-based architectures like PatchTST may possess powerful internal mechanisms that are already highly effective at filtering and weighting information from a larger, unfiltered set of features.

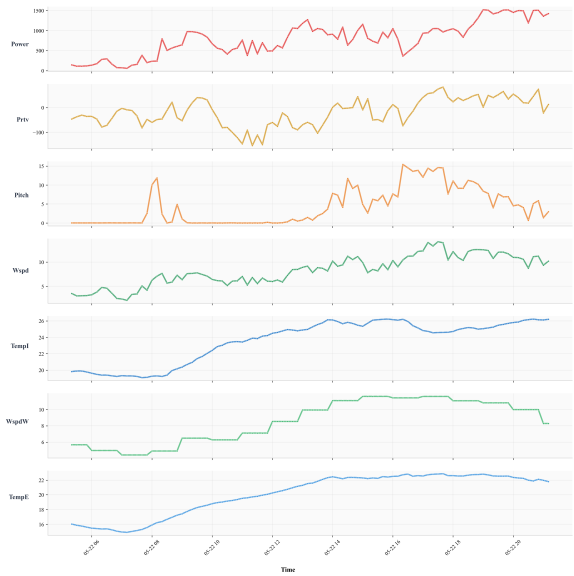

Figure 6: The multivariate time series data sample.

In summary, the experiments validate that Augur serves as a powerful and effective feature selection framework. By identifying a causally-informed subset of variables, it consistently enhances the predictive accuracy of various conventional forecasting models, making a strong case for the integration of LLM-driven causal discovery into time-series analysis pipelines.

D.2 Case Study: Wind Power Analysis

Initial graph construction We conduct a controlled comparison across correlation metrics (Pearson, Spearman, Kendall) and thresholds on per-sample windows of length T=96. For each sample, we compute the correlation matrix over all numeric variables. We instantiate a candidate undirected graph by (i) linking $Patv$ to its top-5 variables ranked by $|\rho|$, (ii) adding pairwise edges among non-$Patv$ variables whenever $|\rho| \geq \tau$, and (iii) retaining only the connected component that contains $Patv$.

We report the average number of retained edges per sample as a proxy for search-space size. Edge counts decrease monotonically with τ, and at any

Table 5: Comparison of Forecasting Performance. The best result in each column is in **bold**.

	MLP		LSTM		DLinear		PatchTST	
Feature Set	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
All Features	0.2416	0.3770	0.1124	0.1743	0.1591	0.2898	**0.1144**	**0.1804**
Wind Speed	0.2140	0.3245	0.1505	0.2614	0.1579	0.2782	0.1590	0.2799
Augur	**0.2015**	**0.3110**	**0.1108**	**0.1725**	**0.1560**	**0.2755**	0.1252	0.1915

Table 6: Description of Variables

Variable Name	Description
Power	Active Power (actual generation output)
Wind Speed	Nacelle-measured wind speed
Wind Dir	Nacelle-measured wind direction
Ext Temp	External nacelle temperature
Int Temp	Internal nacelle temperature
Nacelle Dir	Nacelle direction (yaw angle)
Pitch Rel	Blade pitch relative value
Atm Temp	Atmospheric temperature at 2 meters
Pressure	Surface atmospheric pressure
Humidity	Relative humidity
ERA5 Wind	Reanalysis wind speed from ERA5
ERA5 Dir	Reanalysis wind direction from ERA5
Precip Type	Precipitation type (encoded)
Pitch Angle	Blade pitch angle

fixed τ the graphs induced by Pearson are densest, Kendall sparsest, with Spearman in between. Guided by this comparison, we fix *Spearman* with τ=0.8, yielding compact yet expressive candidate graphs for the subsequent LLM-guided causal inference stage.

Table 7: Average connections by correlation threshold

Method	0.5	0.6	0.7	0.8	0.9
Spearman	30.8	24.4	17.5	11.9	6.9
Pearson	32.6	26.1	19.5	14.5	9.2
Kendall	18.6	13.0	7.9	6.4	5.1

The analysis confirms that Active Power (`Patv`) is predominantly governed by two factors. It exhibits a very strong positive correlation with Wind Speed and a strong negative correlation with the blade `Pitch` angle. This relationship reflects the core physics of wind turbine operation: power output increases with wind speed until the pitch angle is adjusted to regulate the load. The ERA5 reanalysis wind speed shows a similar, though weaker, positive correlation.

Significant multicollinearity is evident among the environmental predictor variables. The various temperature readings (`Etmp`, `Itmp`, and `T2m`) are highly inter-correlated and share strong negative relationships with Surface Pressure (`Sp`). This indicates considerable redundancy among these atmospheric measurements.

These insights directly inform the modeling strategy. Wind Speed (`Wspd`) and Pitch Angle (`Pitch`) are confirmed as primary predictors for power forecasting. However, the redundancy observed among the temperature and pressure variables suggests that a careful selection or combination of these features is necessary to build a robust and efficient model.

Figure 6 presents the multivariate time series data sample used in our case study . It plots the dynamics of key operational variables—including Active Power (`Power`), Wind Speed (`Wspd`), Pitch Angle (`Pitch`), and multiple temperature readings—over a single day. This observational data forms the empirical basis for the analysis in the following sections, where our model explains these dynamics using the causal rules defined by the discovered DAG (as shown in Figure 8).

Detected Cycles and Resolution Strategy The causal discovery agent identified multiple cycles in the wind power generation system. Below we present the first five cycles with their proposed resolutions.

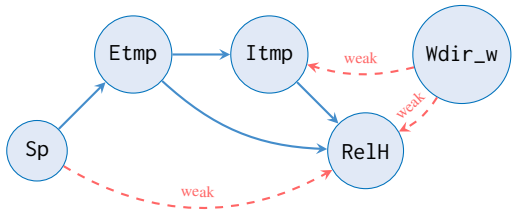

Figure 7: Example of detected cycle with weak edges marked for removal

Resolution Strategy **Primary criteria for edge removal:**

1. **Physical plausibility:** Temperature gradients

Table 8: Edge Removal Frequency in Cycles

Edge Type	Removal Frequency					
	C1	C2	C3	C4	C5	C6
Sp → RelH	✓	✓	–	✓	✓	✓
Wdir_w → RelH	✓	✓	✓	✓	–	✓
Wdir_w → Itmp	–	–	✓	✓	✓	✓

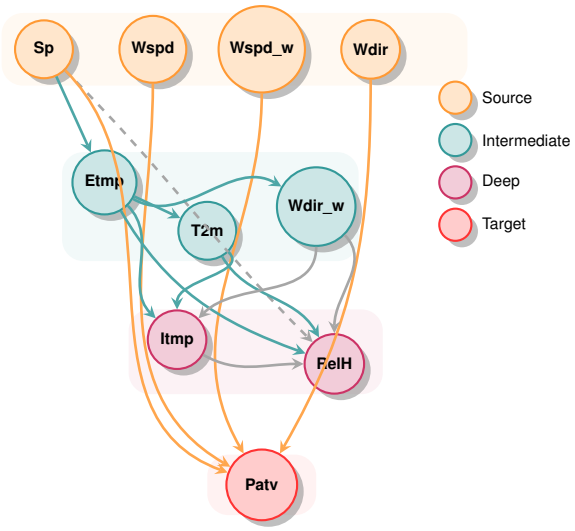

Figure 8: Final causal DAG after cycle resolution

create stronger direct effects on humidity than pressure or wind direction.

2. **Causal mediation:** Indirect effects (e.g., Sp→Etmp→RelH) are preferred over spurious direct links.

Final DAG characteristics: After removing identified weak edges, the resulting directed acyclic graph maintains wind-centric causal flow with no cycles, preserving mechanistically sound relationships essential for wind power prediction. -

Causal Narrative Summary *The data show a daytime increase in wind and temperature that drives large rises in Patv (power) and a concurrent warming (Itmp) with a midday dip in relative humidity.*

Finding 1: Power Generation Dynamics

Pattern Observed:

Patv rises sharply from early morning to afternoon/evening, tracking increases in Wspd and Wspd_w (e.g., Patv: ≈144 → >1000; Wspd: ≈3.5 → 10–13).

Causal Explanation (per DAG):

Consistent with the DAG: rising `Wspd` directly increases `Patv` (`Wspd` → `Patv`) and increases `Wspd_w` (`Wspd` → `Wspd_w`), which in turn also increases `Patv` (`Wspd_w` → `Patv`). The observed co-movement of `Pab` with `Wspd` is also expected by `Wspd` → `Pab`.

Finding 2: Internal Temperature Dynamics

Pattern Observed:

Itmp (internal temperature) increases over the day (≈19.8 → ≈26.2) following the rise in Patv and environmental temperatures.

Causal Explanation (per DAG):

Explained by DAG paths: higher `Etmp` raises `T2m` (`Etmp` → `T2m`) and `T2m` raises `Itmp` (`T2m` → `Itmp`), `Etmp` also increases `Patv` (`Etmp` → `Patv`) and `Patv` raises `Itmp` (`Patv` → `Itmp`), and rising `Wspd` increases `Wspd_w` which also raises `Itmp` (`Wspd` → `Wspd_w` → `Itmp`). These combined causal routes account for the daytime warming of `Itmp`.

Finding 3: Relative Humidity Dynamics

Pattern Observed:

Relative humidity falls through midday (≈0.22 → ≈0.18) while temperatures rise, then partially recovers later.

Causal Explanation (per DAG):

Per the DAG, `Etmp` directly affects `RelH` (`Etmp` → `RelH`); additionally `Itmp` influences `RelH` (`Itmp` → `RelH`) and `Wspd`-driven `Wspd_w` also affects `RelH` (`Wspd` → `Wspd_w` → `RelH`). Thus the midday RH drop is attributable to higher `Etmp` and `Itmp` (and concurrent `Wspd_w` changes) via the DAG-prescribed links.

This generated causal summary provides critical, actionable insights for predictive modeling. By validating the system's true causal drivers (such as `Wspd` and `TempE`) and their pathways, the analysis confirms the variables belonging to the theoretical Causal Markov Blanket. This allows for the construction of a sparse, robust, and generalizable feature set while safely excluding redundant proxy variables.

Most critically, the analysis moves beyond simple correlation to prevent modeling errors. For instance, by identifying that `Power` causes internal temperature (via the path `Power` → `TempI`), the summary explicitly instructs us to **exclude** `TempI` as a predictor for `Power`. A standard model based

Table 9: Cycle Resolution Summary

ID	Cycle Path	Edge to Remove	Justification
1	Etmp→Itmp→RelH, Sp→RelH, Sp→Etmp	Sp→RelH	Surface pressure influences humidity indirectly via temperature. Direct temperature-humidity links are mechanistically stronger.
2	Etmp→Itmp→RelH, Wdir_w→RelH, Wdir_w→Itmp, Sp→RelH, Sp→Etmp	Sp→RelH	Sp's effect on RelH is mediated by temperature and wind conditions. Direct pathways from Etmp and Wdir_w are more plausible.
3	Etmp→Itmp→RelH, Wdir_w→RelH, Wdir_w→Itmp, Etmp→RelH, Sp→RelH, Sp→Etmp	Sp→RelH	Sp primarily impacts humidity through temperature mediation. Presence of Sp→Etmp supports this indirect pathway.
4	Etmp→Itmp→RelH, Wdir_w→RelH, Wdir_w→Itmp, Etmp→RelH	Wdir_w→RelH	Wind direction's direct impact on humidity is less pronounced than temperature effects. Temperature links are stronger physical drivers.
5	Etmp→Itmp→RelH, Wdir_w→Itmp, Etmp→RelH	Wdir_w→Itmp	External temperature (Etmp) is the primary driver of internal temperature, not wind direction.

on correlation alone would likely misuse this variable, learning a spurious relationship that degrades predictive stability.

Finally, this summary reveals the complex, multi-path dynamics required for truly robust forecasting. The insight that `TempI` is driven by *both* environmental heat (`Etmp` → `T2m` → `TempI`) and operational heat (`Power` → `TempI`) allows a model to correctly anticipate system states—such as a high internal temperature on a cold but windy day—that a purely correlative model would fail to predict. This causal grounding directly translates to a more physically accurate and reliable forecasting system.

E Extensibility

As shown in Figure 9, Augur produces DAG-grounded explanations that identify wind speed as the dominant causal driver of power and provide causal routes for residual variability, whereas GPT-4o tends to restate correlations without consistently grounding claims in the graph.

Beyond Causality. A key advantage of our framework is the inherent modularity and extensibility of the generated narrative. The textual format of our causal explanation allows for seamless integration with other forms of time-series analysis, creating a richer auxiliary modality for any downstream task. For instance, the narrative can be programmatically augmented with other structured insights, such as automatically inserting pre-computed **statistical properties** like the Spearman correlation coefficients. Similarly, results from classical time-series decomposition can be integrated to explicitly state **temporal dynamics** like periodicities or trends. By concatenating these diverse textual representations, we can construct a holistic, multi-faceted summary that captures not only the "why" (causality) but also the "what" (statistics) and the "how" (patterns). This enriched textual modality provides a far more complete contextual understanding for any language-model-based downstream system, positioning our framework as a central component in a broader, hybrid time-series analysis ecosystem.

Efficiency and Controllability. While our framework leverages a powerful teacher model for the initial, one-time generation of the training corpus, we deliberately employ a smaller, supervised fine-tuned (SFT) student model for all downstream tasks. This strategic choice is driven by critical considerations of efficiency, economy, and controllability, which are paramount in real-world time-series applications. Direct, repeated inference with a frontier model like a hypothetical GPT-5 would be prohibitively expensive and slow for the high-throughput processing often required in time-series analysis.

By distilling the teacher's complex reasoning capabilities into a specialized student agent, we achieve a system that is not only orders of magnitude more cost-effective and faster at inference, but also more controllable. The SFT approach allows us to create a deterministic, self-contained artifact that can be deployed reliably in production environments without reliance on external APIs, ensuring stable performance and predictable behavior. This distillation process, therefore, represents a pragmatic yet powerful method to harness the reasoning power of state-of-the-art LLMs while meeting the practical constraints of operational time-series

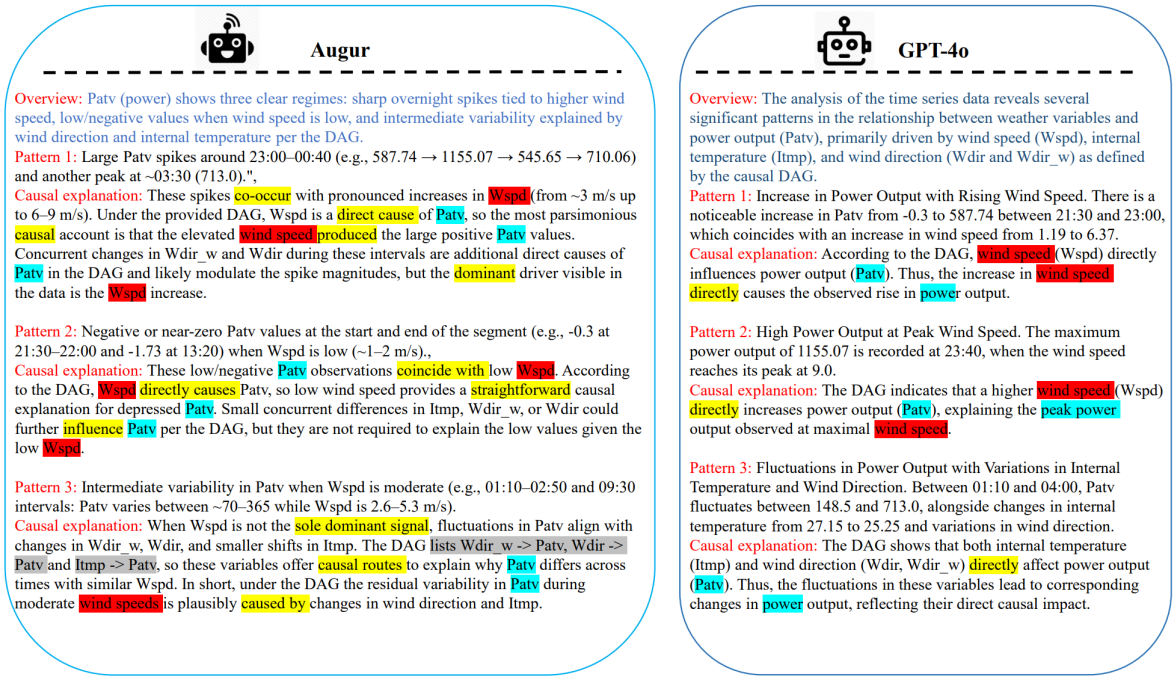

Figure 9: Side-by-side comparison of causal narrative outputs for the same segment.

analysis.

E.1 Prompt Example

Prompt 1 generates pairwise causal hypotheses for correlated variables. These hypotheses are then passed to **Prompt 2**, which assembles them into a global structure and resolves cycles to form a valid Directed Acyclic Graph (DAG). Finally, **Prompt 3** uses this validated DAG to synthesize a grounded narrative that explains key patterns observed in the time series data.

Prompt 1: Pairwise Causal Hypothesis Generation

ROLE:
You are an expert in *[Your Domain, e.g., "financial markets"]* and a specialist in causal inference.

CONTEXT:
I am analyzing data from a *[System or Process Name]* to build a Causal Directed Acyclic Graph (DAG). I have identified a significant statistical correlation between two variables and need to determine their causal associations.

VARIABLE DEFINITIONS:

Variable A: *[Variable A Name] - [Clear, concise definition...]*

Variable B: *[Variable B Name] - [Clear, concise definition...]*

INPUT DATA:
Correlation between *[Var A]* and *[Var B]*: *[e.g., "Spearman's rho = +0.85"]*

TASK:
Evaluate the following causal hypotheses based on first principles...

HYPOTHESES:

- `A -> B:` ...
- `B -> A:` ...
- `Confounder:` ...
- `Correlation Only:` ...

OUTPUT FORMAT:
Provide a JSON object with keys: "reasoning" and "conclusion".

Prompt 2: Global Graph Assembly & Cycle Resolution

ROLE:
You are an expert in systems modeling and graph theory, specializing in the validation of causal structures.

CONTEXT:
I have performed pairwise causal analysis to generate a set of directed edges representing a system's hypothesized causal structure in the domain of *[Your Domain]*. I need you to validate this structure.

INPUT: LIST OF DIRECTED EDGES
[Paste all inferred directed edges from Stage 1 here, one per line.]

```
VarA -> VarB
VarC -> VarA
VarB -> VarC
...
```

TASK:

1. **Identify Cycles:** Analyze the provided edges and explicitly identify any cycles.
2. **Propose Resolution:** For each cycle, propose which single edge is the "weakest link" and should be removed.
3. **Justify Proposal:** Provide a clear, logical justification for your choice.

OUTPUT FORMAT:
Provide a structured response listing identified cycles and your justified recommendations. If no cycles exist, state that "The graph is a valid DAG."

Prompt 3: Causal Analysis & Summary from Time Series Data

TASK:

Your task is to analyze the provided multivariate time series data to identify the 2-3 most significant patterns or events. Then, write a concise narrative summary that explains your findings using the causal associations defined in the Causal DAG.

INPUTS:

1. Causal DAG: *(This graph is the "rule book" for causation...)*

```
[Paste your DAG here, one edge per line, e.g.:]
Wspd -> Patv
Patv -> Itmp
Etmp -> Itmp
```

2. Core Variable Time Series: *(Provide a downsampled or key segment...)*

```
[Paste your time series data here, for example:]
Timestamp, Wspd, Patv, Itmp
2025-09-12 12:00, 8.1, 1.2, 45.1
2025-09-12 12:05, 15.2, 2.5, 45.5
...
```

INSTRUCTIONS:

1. **Analyze First:** Examine the raw time series to find the most important patterns...
2. **Explain with DAG:** For each significant pattern you identify, construct a causal explanation...
3. **Causal Fidelity is Crucial:** You must not infer any cause-and-effect relationship...

OUTPUT:

Produce a concise summary. Start with a one-sentence overview, followed by bullet points. Each bullet point should first **describe a key pattern** you found in the data and then **explain its cause(s)** based on the DAG.