Integrating Natural Language Events into Time Series Forecasting through Agentic LLM Orchestration

Astrid Atle

David Perntoft

Department of Mathematical Statistics / Industrial Engineering and Management

Lund University

May 19, 2026

Abstract

This thesis investigates whether Large Language Models (LLMs) can meaningfully reason over natural language event context to produces calibrated adjustments to time series forecasts, and under what conditions that reasoning produces reliable signal. We design an agentic forecasting system in which the LLM acts as a strategic orchestrator rather than a direct numerical predictor: all numerical computation is delegated to a library of validated statistical implementations, while the LLM reasons over textual event descriptions, historical analogues from the target series' own history, and cross-domain precedent cases. This architectural separation isolates the LLM's contribution to the reasoning layer.

The system is evaluated through a controlled ablation study on three simulated datasets spanning primary care, parcel logistics, and music streaming. Six conditions systematically vary access to grounding sources, and the pipeline is benchmarked against a numerical foundation model and assessed by both standard accuracy metrics and an independent LLM judge of reasoning quality.

The full pipeline reduces sMAPE by 59% (health center) and 66% (logistics) relative to a no-augmentation baseline, at a runtime overhead of 1.5× and an average API cost of $0.07 per forecast, and outperforms the foundation model by a factor of 2.0 to 2.5 in MASE on event-driven test

windows. The ablation shows that LLM event reasoning requires at least one grounding source to function reliably, and that the dominant source is determined by the structural match between available evidence and the test-period event. The findings characterise LLM event reasoning as a viable and transparent capability for hybrid forecasting where contextual events drive material deviations from baseline patterns.

Acknowledgments

We would like to express our sincere gratitude to our supervisor Magnus Wiktorsson at the Department of Mathematical Statistics, Lund University, for his guidance, thoughtful feedback, and continuous support throughout the development of this thesis. We are equally grateful to Marcus Zethraeus at Predli Consulting AB for the opportunity to pursue this work in collaboration with the company, and for his engagement and valuable input throughout the project.

Lund, May 2026
Astrid Atle and David Perntoft

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Time series forecasting is a foundational challenge in diverse domains, ranging from finance, energy management, supply chain logistics, and public health. For decades, classical statistical methods such as SARIMA and exponential smoothing have dominated the space by offering interpretable, mathematically grounded frameworks tuned for structured time series. However, these traditional approaches carry inherent limitations. They operate on numerical signals alone, require expert judgment for model selection and parameter tuning, and cannot naturally incorporate contextual information. Taking event information such as policy announcements, market events, or promotional schedules into consideration when making predictions requires high domain knowledge and multivariate statistical models.

The recent emergence of Large Language Models (LLMs) has introduced a fundamentally different modality of reasoning into forecasting. LLMs demonstrate sophisticated pattern recognition and zero-shot transfer capabilities, suggesting potential for settings where historical data is sparse or where contextual interpretation is essential (Jin et al. 2024). Their

capacity to process natural language suggests a path toward forecasting systems that can jointly reason over quantitative signals and qualitative context. However, effectively leveraging these capabilities for rigorous forecasting requires careful architectural design. LLMs are known to reason inconsistently over raw numerical sequences, and the alignment between numerical signals and natural language descriptions remain a non-trivial challenge that straightforward prompting strategies do not resolve (Zhou and Yu 2025).

Recent research has begun to explore different approaches to bridge this gap. Work on data-centric AI for time series shows that LLMs can reason over metadata to formulate effective data enrichment strategies (Yeh et al. 2025), and surveys of agentic systems document that multi-agent frameworks with specialized roles can provide interpretable root-cause explanations for anomaly detection (Chang et al. 2025). However, only a few works demonstrate how LLMs can be used for time-series reasoning in natural language with systematic evaluation of their reasoning processes (Chow et al. 2024).

This thesis addresses that gap directly. It investigates whether an agentic LLM system can meaningfully reason over natural language event context to produce calibrated adjustments to time series forecasts, and under what conditions that reasoning produces reliable signal. Rather than treating the LLM as a direct numerical predictor, the system delegates all numerical computation to a library of validated statistical implementations, isolating the LLM's contribution to the reasoning layer. The architecture is not evaluated primarily as a forecasting improvement but as an instrument for testing a specific capability, whether structured retrieval of historical analogues and domain knowledge, combined with LLM reasoning over natural language event descriptions, produces principled and transferable signal across domains with structurally different event types.

1.1 Purpose

The purpose of this thesis is to investigate whether Large Language Models can meaningfully reason over natural language event context to produce calibrated adjustments to time series forecasts, and under what conditions that reasoning produces reliable signal. Two interrelated purposes follow from this question.

1.1.1 Agentic Orchestration as an Evaluation Instrument

The agentic framework is designed so that measurable differences in forecast quality between conditions can be attributed to the LLM's reasoning over event context rather than to its numerical computation, which is delegated entirely to a library of validated statistical implementations.

1.1.2 Conditions for Reliable Event Reasoning

A secondary purpose is to characterize when LLM event reasoning produces reliable signal, through a controlled ablation design that systematically varies access to own-history knowledge and external precedent knowledge.

1.2 Scope and Delimitations

This thesis examines a specific paradigm for integrating external event information into time series forecasts, representing events as natural language descriptions and reasoning over their quantitative imprints in the target series' own history and in analogous external precedents. Covariate-based modelling approaches, such as SARIMAX or transfer function models, which encode external drivers as structured numerical input series, are outside the scope of this work. This exclusion is not a dismissal of that paradigm, which is mature and well-characterised in the literature, but a deliberate choice to examine a complementary approach applicable in settings where structured numerical covariates are unavailable and the relevant event information exists only as unstructured text.

The choice to treat the qualitative and quantitative information sources as architecturally separate, rather than jointly trained, is itself a delimitation. End-to-end approaches that learn a shared representation across modalities are an alternative design space that this thesis does not investigate. The architectural separation adopted here trades end-to-end optimality for interpretability and modularity, with the qualitative-to-quantitative translation localised to a single specifiable component rather than distributed across learned parameters.

Several further delimitations apply. The evaluation is conducted on three simulated datasets with controlled data generating processes, which enables ground truth for event effects but does not capture the full noise structure of production time series. The forecast pipeline uses a single LLM family for all reasoning steps and a single different LLM family as judge, so the findings cannot cleanly separate architectural effects from model-specific capabilities. The numerical foundation model included as an external reference point is univariate by construction, and the focus throughout is on evaluating LLM-based forecasting systems under

controlled conditions rather than on establishing absolute performance against the full landscape of forecasting methods.

Chapter 2

Background

2.1 The Evolution of LLMs in Time Series Analysis

Time series LLMs can serve as agents that adapt based on user preference, history, or context to provide personalized predictions and decisions, acting as intermediaries to integrate with various systems and data sources (Jiang et al. 2024). This agentic capability represents a paradigm shift from traditional forecasting where models operate as isolated prediction engines.

The progression of LLM applications to time series has followed several distinct phases. Initial approaches focused on reprogramming techniques, where input time series are converted into text prototype representations naturally suited to the capabilities of language models' (Jin et al. 2024). More recent work has emphasized reasoning capabilities, with models fine-tuned using chain-of-thought augmented tasks to generate reasoning

paths that explain how time-series features like frequency and magnitude inform predictions (Chow et al. 2024).

More recently, the field has evolved toward multimodal integration, where LLMs process both numerical time series and contextual information from text, images or tabular data. Kim et al. (2024) demonstrate that jointly modelling time series alongside textual event data exploits the complementary nature of these modalities for improved forecasts. Recent approaches employ iterative refinement mechanisms where LLM-based agents filter relevant events and continuously adapt their selection logic based on forecast performance (X. Wang et al. 2024). In parallel, foundation models for time series have emerged: large transformers pretrained on diverse corpora to enable zero shot forecasting without fine-tuning on any specific task. Chronos (Ansari et al. 2024) tokenizes scaled time series values into a fixed vocabulary and trains a transformer based on the T5 architecture to predict future tokens through a categorical distribution over the vocabulary. Probabilistic forecasts are generated by autoregressive sampling, from which quantile estimates can be derived for previously unseen series directly. Such models recover seasonal and trend structure from numerical history alone, but by construction cannot ingest contextual information encoded as text, such as planned promotions, public health advisories, or product release schedules. This limitation motivates the direct comparison conducted in this thesis between LLM reasoning that incorporates events and a purely numerical foundation model.

2.2 Reasoning Topologies and Architectural Structure

Tree-of-Thoughts (ToT) enables deliberate decision making by considering multiple reasoning paths and self-evaluating choices, allowing models to look ahead or backtrack when necessary (Yao et al. 2023). In the system developed in this thesis, this topology is instantiated specifically at the hypothesis generation stage, where the LLM proposes multiple competing model specifications that are evaluated in parallel before being pruned and aggregated. The broader pipeline architecture, which includes sequential stages for data analysis, domain reasoning, and scenario generation, is better characterised as an orchestrated multi-agent system in which ToT reasoning is one component rather than the organising principle of the entire system.

2.3 The Challenge of Multimodal Input Time Series Forecasting

Real-world forecasting often requires integrating numerical time series with contextual information: financial news affecting stock prices, weather events impacting energy consumption, or policy announcements influencing economic indicators. Recent approaches leverage LLMs and Generative Agents to reason across training, validation, and test and time series data, adaptively integrating social events into forecasting models by aligning news content with time series fluctuations (X. Wang et al. 2024). Despite this progress, several fundamental challenges remain. The two modalities operate at different levels of abstraction, making direct translation between numerical patterns and natural-language de-

scriptions non-trivial. Beyond representation, systems must distinguish relevant events from background noise, correctly synchronize event timing with forecast horizons, and produce reasoning that is transparent enough to be audited. These challenges motivate the design choices explored in this thesis.

2.4 The Gap This Research Addresses

Despite growing research on LLM applications in time series analysis, a critical methodological gap persists. The literature lacks systematic comparative studies quantifying when agentic architectures provide substantive advantages over simpler alternatives. While surveys document approaches ranging from direct reasoning to complex multi-agent frameworks (Chang et al. 2025), rigorous controlled comparison between architectural paradigms remains scarce. Three interrelated gaps emerge from this review, and together they motivate the design and evaluation approach taken in this thesis.

2.4.1 Controlled Architecture Evaluation

Existing studies evaluate LLM forecasting systems in isolation rather than establishing controlled comparisons between architectural paradigms while holding the underlying model constant. This thesis addresses that gap by comparing the full agentic pipeline against a no-augmentation statistical baseline and a numerical foundation model, with grounding sources systematically varied through a controlled ablation design.

2.4.2 Grounding Conditions for Reliable Event Reasoning

While recent work demonstrates that integrating time series with textual data can improve predictions (Kim et al. 2024), the conditions under which LLM event reasoning produces reliable signal remain poorly characterized. In particular, it is unclear how sensitive LLM reasoning is to the availability of own-history analogues and external precedent knowledge, and whether reasoning degrades gracefully or catastrophically when those grounding sources are removed. This gap motivates a controlled ablation design that systematically varies access to own-history knowledge and external precedent knowledge across otherwise identical experimental conditions.

2.4.3 Complexity-Benefit Analysis

LLMs have shifted time series modelling toward interactive and explanatory processes requiring human-interpretable insights (Chang et al. 2025), but critical work argues that LLMs' abilities to understand and reason about numerical time series are considerably limited (Zhou and Yu 2025). The marginal utility of increasing architectural complexity across accuracy, robustness, interpretability and efficiency remains an open empirical question.

This will be addressed by employing multi-dimensional evaluation across accuracy, robustness, interpretability, latency and computational cost.

Chapter 3

Theory

3.1 Statistical Forecasting Foundations

The system is built on the principle that LLMs should not perform numerical computation directly. Instead, they act as strategic orchestrators, selecting from and parametrizing a library of validated statistical implementations that cover three complementary model families: autoregressive models of the SARIMA family, state-space models, and decomposition-based models built on STL. Together these families cover the principal axes of variation that the hypothesis generation agent must navigate, including the degree of autocorrelation, the stability of the seasonal component, and the presence of multiplicative versus additive dynamics.

3.1.1 Autoregressive Models

The most general model class used in the system is the Seasonal Autoregressive Integrated Moving Average model, $SARIMA(p, d, q)(P, D, Q)_s$. Using the backshift operator, a SARIMA model is defined by the equation

$$\Phi_P(B^s)\phi_P(B)(1-B)^d(1-B^s)^D y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t, \qquad (3.1)$$

where $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ is a white noise process (Panjala et al. 2025). The non-seasonal autoregressive and moving average polynomials are $\phi_P(B) = 1 - \phi_1 B - ... - \phi_P B^P$ and $\theta_q(B) = 1 + \theta_1 B + ... + \theta_q B^q$ respectively, while their seasonal counterparts $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ operate at multiples of the seasonal period s. The differencing operators $(1-B)^d$ and $(1-B^s)^D$ induce stationarity in the non-seasonal and seasonal components respectively (Panjala et al. 2025).

The parameters (p, d, q, P, D, Q, s) control the structural complexity of the model. In practice, the differencing orders d and D are determined by stationarity tests, the autoregressive and moving average orders (p, q, P, Q) are informed by the autocorrelation and partial autocorrelation functions (ACF and PACF) of the differentiated series, and the seasonal period s is identified by spectral or STL analysis.

3.1.2 State-Space Models

A second model class available to the system is the family of state-space models, which represents time series dynamics through a latent state vector $\alpha_t \in \mathbb{R}^m$ that evolves according to

$$\alpha_{t+1} = F_t\alpha_t + R_t\eta_t, \quad \eta_t \sim \mathcal{N}(0, Q_t), \qquad (3.2)$$

with observations related to the latent state by the measurement equation

$$y_t = Z_t \alpha_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, H_t), \tag{3.3}$$

where F_t is the $m \times m$ state transition matrix, R_t is an $m \times k$ selection matrix routing the k-dimensional disturbance vector to the state components, Z_t is the observation matrix, Q_t is the $k \times k$ positive semi-definite covariance matrix of the state disturbances, and H_t is a diagonal positive semi-definite observation noise covariance matrix (Helske 2017). The Kalman filter provides efficient recursive estimation of the conditional mean and variance of α_t given observations up to time t, and the resulting filtered states can be used to produce forecasts. Many classical models can be expressed as special cases of this state-space form by appropriate specification of the system matrices, which makes the representation both general and practically useful.

3.1.3 Decomposition-Based Models

The third model family used in the system is built on STL decomposition, an acronym for Seasonal and Trend Decomposition, using LOESS. STL decomposes the observed values into three additive components,

$$y_t = T_t + S_t + R_t, \tag{3.4}$$

where T_t is a slowly varying trend, S_t is a periodic seasonal component, and R_t is the remainder. A multiplicative variant,

$$y_t = T_t \cdot S_t \cdot (1 + R_t), \tag{3.5}$$

is more appropriate when the amplitude of the seasonal fluctuations scales with the level of the series, as is the case of the datasets used for evaluation in this thesis (Cleveland et al. 1990).

3.1.4 Foundation Models for Time Series

Foundation models for time series adapt the pretraining paradigm from language modeling to numerical sequences. Chronos (Ansari et al. 2024) is a family of pretrained models built on the T5 encoder-decoder architecture that tokenizes scaled time series values into a fixed vocabulary, with probabilistic forecasts obtained by sampling from a categorical distribution over the token vocabulary. The Chronos-Bolt variant used in this thesis (Small, approximately 48 million parameters) replaces autoregressive sampling with direct multi-step quantile forecasting: the encoder ingests patched representations of the historical context, and the decoder produces quantile estimates across multiple future steps in a single forward pass. At inference, the model accepts a univariate context series of arbitrary length and produces median plus quantile forecasts for the requested horizon directly, without sampling. Chronos-Bolt is trained to produce quantiles in the range $[0.1, 0.9]$, so 80% prediction intervals are produced natively while wider intervals are clipped. The model has no mechanism for ingesting exogenous variables, calendar information, or text. It operates purely on the numerical signal.

3.2 Limitations of LLMs in Numerical Reasoning

A central design premise of the system is that LLMs are unsuitable as direct processors of raw numerical time series. Recent benchmark evidence shows that LLMs have fundamental limitations on basic numerical operations such as arithmetic, magnitude comparison, and numerical retrieval (Li et al. 2025), limitations that extend naturally to the ordered numerical operations required for time series analysis. This formalizes the basis for that claim, drawing on the literature on LLM tokenization and numerical reasoning, and introduces the mitigation strategy of statistical grounding through pre-computed descriptors.

3.2.1 The Perception Gap

LLMs are trained to predict the next token in natural language text, and their internal representations are optimized for that task. When presented with a raw numerical sequence, the model lacks the structural inductive biases that classical statistical procedures bring to such tasks, and recent empirical work finds no evidence that LLMs can reason reliably about subtle temporal patterns beyond trivial cases (Zhou and Yu 2025). These require operations over ordered magnitudes and differences that are not naturally encoded in the representations a language model learns from text data. The misalignment between the model's capabilities and the demands of time series reasoning has been documented in recent work. LLMs show inconsistent behaviour across different model architectures when applied to time series tasks, and their accuracy degrades when input sequences are long or when the underlying patterns extend beyond trivial cases (Zhou and Yu 2025). We refer to this misalignment as the *perception gap*: the structural inability of a language

model to reason reliably from raw numerical time series in the same way a classical statistical procedure can.

3.2.2 The Tokenization Problem

The perception gap is compounded at the level of input encoding by the subword tokenization schemes used by most modern LLMs, such as Byte-Pair-Encoding (BPE). These tokenizers split numbers into chunks based on statistical patterns in training data rather than mathematical value (Li et al. 2025).

As a result, a number such as 123.45 may be tokenized as ["12", "3", ".45"], with the exact split depending on the token frequencies in the training corpus. The model receives no guarantee that the resulting token sequence encodes the ordinal relationship between numbers nor their interval relationships.

3.3 Agentic Systems and Orchestration

The term agent in the context of LLM systems refers to a model that does not merely respond to a single prompt but instead takes a sequence of actions including selecting tools, invoking external functions, and conditioning subsequent steps on the results of earlier ones (Mathew and Rossi 2025). The transition from a single LLM call to a network of cooperating agents introduces both new capabilities and new design constraints.

A central distinction in agentic system design is between monolithic and orchestrated architectures. In a monolithic architecture, a single model is responsible for all reasoning steps, from interpreting the input to pro-

ducing the final output. In the time series forecasting context, this would mean presenting the raw numerical series alongside natural language event descriptions directly to a single LLM call, expecting it to simultaneously identify structural properties, select a modelling strategy, and produce a numerical forecast, a task that spans both linguistic and quantitative reasoning without delegation to specialised components. In an orchestrated architecture, a coordinating agent decomposes the overall task and delegates sub-tasks to specialised agents or tools, each operating within a narrower and better defined scope (Adimulam et al. 2026). This separation has a well-established theoretical basis in the principle of separation of concerns. When a complex task can be decomposed into sub-tasks with well-defined interfaces, assigning each to a component optimised for it produces more reliable outcomes than requiring a single component to handle them all (Adimulam et al. 2026). In the context of LLM-based forecasting, this principle motivates delegating numerical computation to validated statistical implementations while reserving language model reasoning for tasks that are linguistic in nature, such as interpreting domain context, selecting candidate model specifications, and evaluating structured diagnostic feedback.

A further consideration is the structure of information flow. When agents communicate through structured, typed outputs rather than free-form text, the scope for error propagation between stages is reduced and failures can be localised to a specific component rather than attributed to the system as a whole (X. Shen et al. 2025). A common pattern within such systems is the refinement loop, in which an executor agent produces an output that an evaluator agent assesses against some objective, with structured feedback cycling back to the executor until a stopping criterion is met.

3.4 Hybrid Forecasting as a Modality Alignment Problem

When a forecasting system combines numerical time series with contextual text, the two modalities operate at fundamentally different levels of abstraction. The numerical signal encodes local dynamics such as trend, seasonality and autocorrelation, while text encodes causal drivers, regime context and domain knowledge that may not be visible in the numbers alone. Aligning these modalities so that each contributes what the other lacks is a central challenge in hybrid forecasting.

3.4.1 Representation Level versus Prediction Level Fusion

Existing approaches address this alignment at different stages of the forecasting pipeline. Representation level methods project both modalities into a shared embedding space before prediction: for instance, contrastive learning can align decomposed trend and seasonal text features with their numerical counterparts (S. Wang et al. 2026). Prediction level methods instead let each modality produce an independent forecast and fuse the outputs, for instance in the frequency domain where event driven predictions dominate low frequency components and numerical predictions dominate high frequency fluctuations (S. Wang et al. 2026). A third family, which includes the system developed in this thesis, applies textual information as a structured post hoc adjustment to a purely statistical baseline, keeping the two modalities architecturally separated rather than jointly trained.

3.4.2 Event Driven Reasoning as a Special Case of Alignment

A key finding in recent work is that the value of text is concentrated in event driven dynamics, such as abrupt level shifts and event-induced movements, rather than in describing the smooth patterns that numerical models already capture well (S. Wang et al. 2026). This motivates architectures where the text branch specialises in identifying and quantifying discrete events while the numerical branch handles the underlying seasonal and trend structure. Critically, unguided LLM reasoning over event text can degrade forecast quality compared to using no text at all; structured retrieval of historical analogues is necessary to ground the reasoning in observed magnitudes and durations (S. Wang et al. 2026).

3.4.3 Alignment in the Present System

The architecture evaluated in this thesis adopts a late fusion strategy. A statistical baseline produced by SARIMA provides the numerical forecast, and a scenario generation layer translates textual event descriptions into multiplicative or additive transforms applied after the baseline is finalised. This design choice trades end to end optimality for interpretability and modularity, since each layer can be inspected, evaluated and improved independently. The alignment challenge then reduces to three sub problems studied empirically in the results chapters: matching future events to the correct historical analogues, translating qualitative event descriptions into calibrated magnitudes, and placing the resulting transforms at the correct temporal position within the forecast horizon.

3.5 Evaluation Theory

Evaluating a hybrid forecasting system requires metrics that can assess different aspects of forecast quality independently. Point accuracy metrics measure how close the central forecast is to the observed values, but they say nothing about whether the uncertainty expressed around that forecast is well-calibrated (Gneiting and Raftery 2007). A further dimension, specific to systems where the forecast is accompanied by a reasoning trace, is whether the stated reasoning is coherent and grounded in available evidence, a property that numerical metrics cannot capture at all. Three classes of evaluation are therefore distinguished: point forecast accuracy, interval forecast quality, and reasoning quality.

3.5.1 Point Forecast Accuracy

A desirable point forecast accuracy metric should be scale-independent, so that results are comparable across series with different levels and variability, and symmetric, so that over- and under-forecasting of the same magnitude are penalized equally. While symmetry is a desirable property for cross-dataset comparison, it is worth noting that in many operational settings over- and under-forecasting carry asymmetrically different costs, and practitioners should consider this when interpreting results.

The framework adopts the Symmetric Mean Absolute Percentage Error (sMAPE), proposed by (Makridakis 1993), which resolves the directional bias of standard MAPE by placing the average of actual and forecast in the denominator. The individual error at time t is

$$\text{sAPE}_t = \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2}$$

and the aggregate metric over an evaluation window of n periods is

$$\text{sMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \text{sAPE}_t.$$

sMAPE is bounded in the range $[0\%, 200\%]$ and is scale-independent, permitting direct comparison across the three evaluation datasets used in this thesis. As a secondary metric, the Mean Absolute Scaled Error (MASE) (Hyndman and Koehler 2006) scales the model's mean absolute error against that of a seasonal naïve forecast computed on the training data,

$$\text{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t|}{\frac{1}{T-s}\sum_{t=s+1}^{T} |y_t - y_{t-s}|},$$

where T is the length of the training series and s is the seasonal period. A MASE below 1 indicates that the model improves upon the seasonal naïve baseline. Because the scaling factor depends only on the training series, MASE is well-defined regardless of whether actuals in the evaluation window are zero.

3.5.2 Interval Forecast Quality

The evaluation of probabilistic forecasts requires a dual assessment of calibration and sharpness. Calibration refers to the statistical consistency between the predictive distributions and the observed realizations, while sharpness describes the concentration of the predictive intervals (Gneiting and Raftery 2007).

The hit rate measures the empirical fraction of actual observations falling within the forecast interval,

$$\text{Hit rate} = \frac{1}{n} \sum_{t=1}^{n} \mathbf{1}\{L_t \leq y_t \leq U_t\},$$

where n is the number of forecast periods. Since the scenario intervals in the present system are constructed from the range of historically observed analogue outcomes rather than from a parametric distributional assumption, the hit rate is interpreted as an empirical measure of how well past evidence bounds future realizations rather than as a calibration test against a nominal coverage level.

To jointly penalize under-coverage and excessive interval width, the Winkler score (Winkler 1972) is used. For a single forecast period it is defined as

$$W_t = \begin{cases} (U_t - L_t) + \frac{2}{\alpha}(L_t - y_t) & \text{if } y_t < L_t \\ (U_t - L_t) & \text{if } L_t \leq y_t \leq U_t \\ (U_t - L_t) + \frac{2}{\alpha}(y_t - U_t) & \text{if } y_t > U_t \end{cases}$$

where the factor $2/\alpha$ ensures that the marginal cost of a miss exceeds the marginal benefit of narrowing the interval. To permit comparison across datasets that differ substantially in scale, the Winkler score is normalized by the mean of the test series,

$$\bar{W} = \frac{1}{n} \sum_{t=1}^{n} \frac{W_t}{\bar{y}_{\text{test}}} \times 100\%.$$

3.5.3 LLM-As-A-Judge

Point and interval metrics quantify the numerical quality of a forecast but cannot assess whether the reasoning that produced it is coherent or grounded in the available evidence. The LLM-as-a-judge paradigm addresses this gap by prompting a strong language model to score the output of another model along explicitly defined criteria. Zheng et al. (2023) demonstrated that strong LLM judges such as GPT-4 can match human preferences at an agreement rate of over 80%, comparable to inter-annotator agreement between humans, establishing the approach as a scalable alternative to manual annotation.

The paradigm is known to exhibit several systematic biases, including position bias, verbosity bias, and self-enhancement bias, the last of which arises when a judge favours outputs generated by itself (Zheng et al. 2023). Subsequent work has expanded the catalogue to twelve distinct bias types (Ye et al. 2024). These biases motivate several design constraints in any empirical application. Position bias is mitigated by randomizing the order in which outputs are presented across runs. Verbosity bias is addressed through a structured scoring rubric with explicitly defined criteria for each dimension. Self-enhancement bias is mitigated by selecting a judge model from a different model family than the one used to generate the forecasts, on the assumption that models within the same family may share stylistic traits that extend this self-favouring tendency beyond strict model identity. The degree to which these mitigations fully eliminate the documented biases remains an open empirical question, and absolute judge scores in this study should accordingly be interpreted as ordinal rankings rather than cardinal measurements.

Chapter 4

Methods

4.1 Approach

The methodology utilizes a LangGraph-based workflow that organizes the forecasting structure into a structured multiple-step orchestration. The system is built on a model-agnostic principle, where the LLM does not generate or execute arbitrary code but instead interprets computed statistics to make strategic decisions about how the data should be modelled. All forecasting is performed by a library of pre-built, validated statistical model functions. Figure 4.1 illustrates the overall pipeline architecture and the information flow between agents, which the following subsections describe in detail.

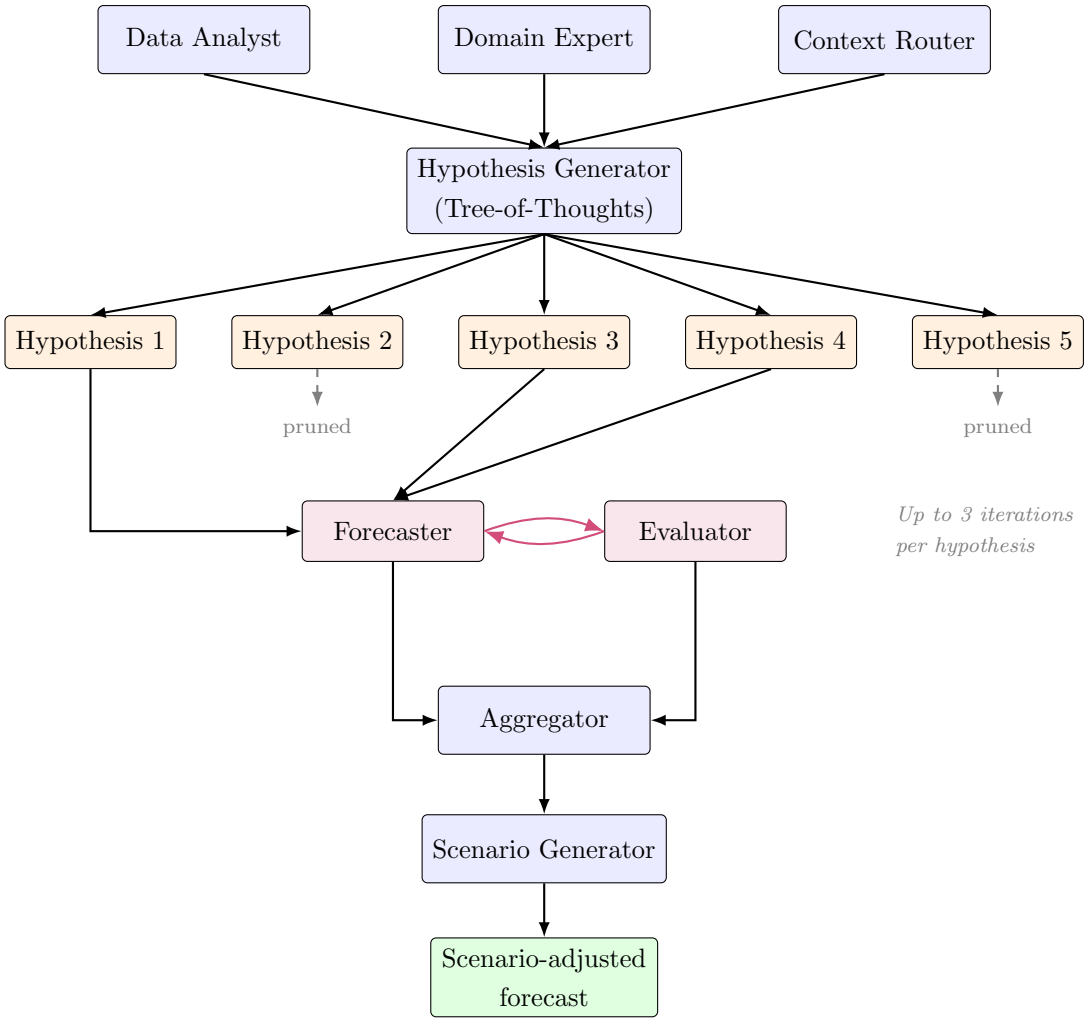

Figure 4.1: Overall pipeline architecture. The Data Analyst, Domain Expert, and Context Router agents provide statistical, qualitative, and routing inputs to the Hypothesis Generator, which branches into up to five candidate hypotheses through Tree-of-Thoughts reasoning. Hypotheses deemed unpromising are pruned (dashed); the remaining hypotheses enter the Forecaster–Evaluator refinement loop. The Aggregator selects the best-performing hypothesis as the statistical baseline, which is then adjusted by the Scenario Generator using future event information.

4.1.1 Data Analysis Agent

The initial stage of the forecasting process involves a dedicated Data Analysis agent that serves as the foundation for all subsequent strategic decisions. Rather than providing the Large Language Model with raw numerical data, which is often a source of reasoning error, this agent executes a series of pre-computed statistical tests to extract high-level descriptors of the time series. These metrics include linear regression for trend directions, STL decomposition for seasonality and period detection, and both Augmented Dickey-Fuller and KPSS test to determine stationarity and suggested differencing orders. Furthermore, the agent performs autocorrelation analysis for ARIMA order selection, identifies structural break change-points, and detects outliers through interquartile range analysis. This approach provides the LLM with a comprehensive summary of the time series' fundamental characteristics, and mitigates the known limitations of LLMs regarding direct numerical understanding by grounding the subsequent reasoning in established statistical metrics.

4.1.2 Domain Expert Agent

The Domain Expert agent acts as a specialized intermediary responsible for the interpretation and integration of qualitative external data with the numerical signals identified in the analysis phase. It provides nuanced insights into domain-specific factors and external variables that are likely to influence the data's trajectory. By identifying these qualitative drivers, the agent ensures that the forecasting process accounts for domain-specific context and potential shifts that historical numerical patterns alone might not fully reflect.

Additionally, this agent addresses the challenge of modality alignment

by synthesizing the statistical descriptors produced by the Data Analysis agent with the available domain context, such as sector-specific demand drivers, operational constraints, and structural characteristics of the series. By acting as a domain-aware interpreter, it translates this contextual understanding into strategic insights that inform the Hypothesis Generation agent, ensuring that candidate model specifications are grounded in both the statistical properties of the data and the qualitative realities of the domain.

4.1.3 Hypothesis Generation Agent

Once the statistical and contextual foundations are established, the Hypothesis Generator agent initiates a branching exploration through a Tree-of-Thoughts reasoning topology. In this stage, the LLM chooses up to five different hypotheses, each of which represents a unique strategic thought regarding the most appropriate model and parameter set for the given data. Each hypothesis specifies a locked model type from a library of given available statistical models, and includes an initial parameter guess and a core qualitative assumption about the data. This approach ensures that the system considers multiple competing interpretations simultaneously, preventing the bias inherent in committing to a single forecasting path too early in the process.

4.1.4 Forecasting-Evaluation loop

The relationship between the Forecaster and Evaluator agents is structured as a recursive, tightly coupled feedback loop designed to optimize model parameters through empirical validation. This iterative process allows the system to refine its hypotheses by moving away from heuristic

guesses toward metric-driven statistical adjustments.

4.1.4.1 Forecaster Agent

The Forecaster agent serves as the primary computational execution engine, responsible for implementing the specific forecasting model assigned to each hypothesis within the Tree-of-Thoughts framework. Rather than generating numerical predictions through the LLM itself, this agent utilizes a library of pre-built and validated statistical functions to ensure mathematical rigour. In the initial phase of the iterative refinement loop, the Forecaster runs the selected model using the parameters established during the hypothesis generation stage. In subsequent iterations, the agent acts as a recursive processor that receives structured parameter updates directly from the Evaluator, re-executing the statistical model with these refined settings to improve accuracy and reduce observed errors.

4.1.4.2 Evaluator Agent

The Evaluator agent provides critical, metric-driven feedback by assessing the output of the Forecaster agent against held-out validation data to ensure objective performance improvement. Rather than providing qualitative or vague textual feedback, the Evaluator first subjects each forecast to a validity check that rejects degenerate outputs such as constant, all-zero, over-smoothed, or out-of-range predictions before any scoring is applied. Valid forecasts are then assigned a composite confidence score bounded between 0 and 1, constructed from two components. The primary component aggregates four interval-quality scores, sharpness, calibration, horizon uniformity, and Winkler score. Each score is capped at 0.25, with a gating mechanism that suppresses sharpness credit when

calibration falls below an acceptable threshold, preventing a model from scoring well by producing confidently wrong intervals. This interval component is blended with a point accuracy score derived from RMSE, MAE, directional bias, and MAPE relative to the data range, weighted at 75% interval quality and 25% point accuracy, ensuring that a forecast with a correct uncertainty envelope but a systematically wrong level cannot dominate the ranking. The Evaluator then reasons about what the current parameters may be failing to capture, and communicates structured JSON feedback to the Forecaster agent, which re-runs its modelling process accordingly. This autonomous optimisation cycle repeats for up to three iterations per hypothesis, balancing the goal of parameter refinement against latency and API cost constraints.

4.1.5 Aggregation Agent

When the iterative process is finished, the Aggregator agent reviews the multi-dimensional confidence scores produced by the Evaluator for each hypothesis and selects the single best-performing hypothesis to carry forward as the statistical baseline forecast. Hypotheses that fail the Evaluator's validity check or score below a minimum confidence threshold are pruned. The selected hypothesis and its associated forecast are then passed to the Context Router and, where applicable, the Scenario Generator.

4.1.6 Context Router Agent

The Context Router Agent functions as a strategic gatekeeper within the pipeline, tasked with the taxonomic classification of qualitative inputs to determine the specific stage of numerical adjustment. By distinguishing

between historical structural shifts requiring pre model training modifications and upcoming exogenous events necessitating post forecast refinements, the agent ensures that contextual information is applied with mathematical precision. This routing logic facilitates a selective execution strategy where computational resources are only deployed when the provided context contains actionable signals. Consequently, the agent optimizes the workflow by bypassing subsequent context processing phases when the input is deemed irrelevant or not existent, thereby preserving the statistical rigor of the baseline models while allowing for targeted human intervention.

4.1.7 Scenario Generator Agent

When the system is provided with a future event impacting the forecast horizon, the Scenario Generator agent translates this qualitative input into a structured probabilistic impact estimate. Rather than producing a single deterministic adjustment, the agent generates three impact specifications corresponding to expected, optimistic, and conservative scenarios, each with associated phase-by-phase magnitude estimates that are subsequently applied as multiplicative adjustments to the statistical baseline forecast. The mathematical construction of those magnitude estimates is described in detail in Section 4.1.9.

4.1.7.1 Inputs

The agent operates on three complementary evidence sources together with the statistical baseline. The first is the own-history knowledge, consisting of dated historical events in the target series' own history together with their measured quantitative imprints extracted from the

historical realisations of that series. The second is the external precedent knowledge, a set of structured precedent cases drawn from analogous events in related but distinct series, each annotated with phase-by-phase percentage ranges. The third source consists of domain insights produced by the Domain Expert agent, which contextualise the future event within the operational characteristics of the target series.

4.1.7.2 Two-stage retrieval mechanism

The agent matches the future event against both evidence sources through a two-stage retrieval mechanism. In the first stage, a deterministic scoring algorithm computes a weighted overlap between the canonicalised tokens and event categories of the future event and each candidate in the two sources. Event texts are first normalised through regex-based phrase aliasing and morphological canonicalisation. Similarity is then scored as a weighted sum of shared categories and shared canonical tokens, augmented with discriminator bonuses for high-information tokens and exclusivity penalties for category mismatches. This stage produces a pre-ranked shortlist of up to four own-history analogues with similarity score above a minimum threshold, together with up to three external precedent cases ranked by token overlap.

In the second stage, the language model receives this shortlist as a structured evidence packet and performs a semantic relevance assessment, selecting the analogues whose event type most closely matches the future event rather than relying on raw textual similarity alone. This allows the system to distinguish, for example, that a single release should anchor on other single releases rather than album releases, even when their textual similarity scores are comparable. The selected analogues and precedent cases are then passed to the quantification procedure described in Section 4.1.9, which constructs the blended magnitude intervals used to bind

the three scenario specifications.

4.1.8 Separation of Quantitative and Qualitative Information

A central architectural principle of the pipeline is the explicit separation of quantitative and qualitative information sources, together with controlled points at which the two modalities interact. Language models are unreliable processors of raw numerical sequences, and the two information types operate at fundamentally different levels of abstraction: numerical signals encode local dynamics such as trend, seasonality, and autocorrelation, while text encodes causal drivers, regime context, and domain knowledge that may not be visible in the numbers alone. Combining these prematurely conflates two distinct reasoning problems and obscures where errors originate.

The pipeline therefore treats the two modalities as architecturally separate streams aligned at well-defined interfaces. The quantitative stream consists of the target time series, the statistical descriptors extracted from it by the Data Analyst agent, the measured imprints of historical events on that series, and the parametric forecasts produced by the model library; all of it is computed by deterministic procedures and presented to the language model as pre-computed input. The qualitative stream consists of the textual event descriptions in the own-history knowledge and external precedent knowledge, the user-supplied future event, and the structured domain insights produced by the Domain Expert agent.

The two streams meet at three controlled points. At the Hypothesis Generator, quantitative diagnostics and qualitative domain insights jointly inform the choice of candidate model specifications, but the model pro-

duces no numerical output. At the Scenario Generator, qualitative event descriptions are matched against historical analogues whose quantitative imprints have already been measured, and the language model assigns magnitude estimates within quantitatively derived intervals. At the Forecast Adjusters, the structured scenario specifications are translated into deterministic mathematical transformations applied to the statistical baseline. For a baseline forecast $\hat{y}_t^{\text{base}}$ produced by the aggregated statistical models, the scenario-adjusted forecast at time t takes the form

$$\hat{y}_t^{\text{scenario}} = \hat{y}_t^{\text{base}} \cdot \left(1 + \sum_k f_k(t)\right), \tag{4.1}$$

where each $f_k(t)$ is a deterministic temporal envelope corresponding to an effect phase, parameterised by a magnitude m_k and a duration τ_k derived from the quantitative evidence packet. The envelope shape is selected by the language model from a fixed library of forms, but its numerical instantiation is deterministic given the chosen parameters.

This arrangement places the language model strictly in the role of strategic orchestrator and qualitative interpreter. The qualitative-to-quantitative translation occurs at a single specifiable layer of the pipeline, the Scenario Generator, where measured historical magnitudes serve as the bridge between textual event descriptions and the numerical adjustments ultimately applied to the forecast. Figure 4.2 illustrates this information flow.

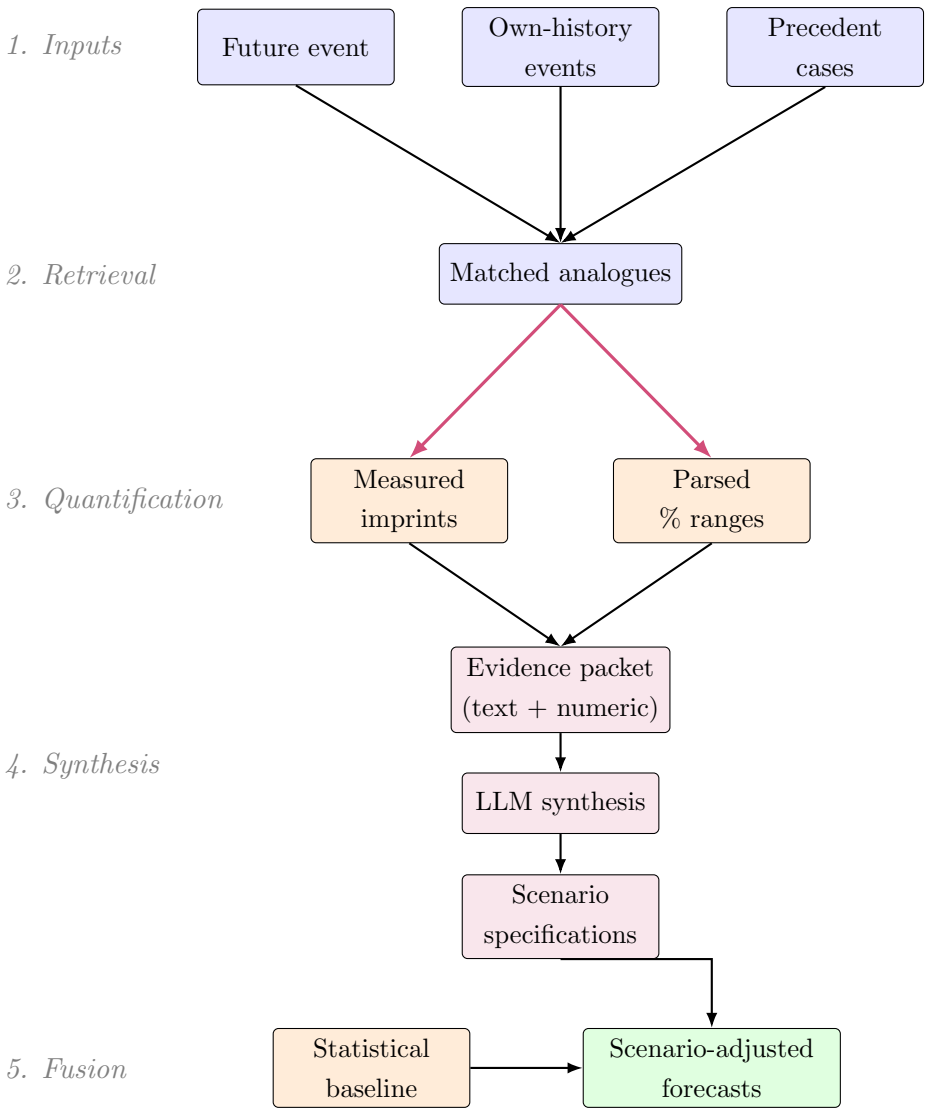

Figure 4.2: Pipeline of qualitative-to-quantitative translation. Purple nodes are textual, orange nodes are numerical, pink nodes contain both modalities, and the green node is the final numerical output. The modality transformation occurs between stages 2 and 3.

4.1.9 Quantifying the Qualitative: Construction of Magnitude Intervals

For each effect phase k of the future event, the pipeline produces a triple of magnitudes (L_k, E_k, U_k) corresponding to the lower, expected, and upper bounds of an evidence-derived interval. This subsection makes the construction of those bounds explicit, expanding on stages three through five of Figure 4.2.

4.1.9.1 Own-history interval per phase

Let $\mathcal{H} = \{h_1, \ldots, h_4\}$ denote the top four own-history analogues retained after the lexical similarity threshold ($\sigma \geq 0.35$). For each analogue h_i and effect phase k, the measured magnitude $m_{i,k}$ is extracted by the Event Impact Analyser, which detects peak, sustained, and decay imprints around the historical event date. The interval for phase k is then a triple of empirical quantiles of the absolute magnitudes:

$$\begin{aligned} L_k^{\text{hist}} &= Q_{0.25}(|m_{1,k}|, \ldots, |m_{4,k}|), \\ E_k^{\text{hist}} &= Q_{0.50}(|m_{1,k}|, \ldots, |m_{4,k}|), \\ U_k^{\text{hist}} &= Q_{0.90}(|m_{1,k}|, \ldots, |m_{4,k}|), \end{aligned} \tag{4.2}$$

where Q_p denotes the empirical p-quantile. Similarity scores are used only to select the four analogues; the quantiles themselves are uniformly weighted.

4.1.9.2 External precedent interval per phase

Let the set of external precedent cases be denoted as $\mathcal{C}$. For each matched precedent case $c \in \mathcal{C}$, the source text is parsed for annotated percentage values, classified into effect phases, and pooled into the set $\mathcal{P}_{c,k}$ for each phase k. The per-case interval is constructed as

$$\begin{aligned} L_{c,k}^{\text{prec}} &= \min(\mathcal{P}_{c,k}), \\ E_{c,k}^{\text{prec}} &= \text{median}(\mathcal{P}_{c,k}), \\ U_{c,k}^{\text{prec}} &= \max(\mathcal{P}_{c,k}). \end{aligned} \tag{4.3}$$

When multiple cases are retrieved (up to three), case-level intervals are aggregated by taking the median across cases of each bound separately:

$$\begin{aligned} L_{k}^{\text{prec}} &= \text{median}_{c \in \mathcal{C}}(L_{c,k}^{\text{prec}}), \\ E_{k}^{\text{prec}} &= \text{median}_{c \in \mathcal{C}}(E_{c,k}^{\text{prec}}), \\ U_{k}^{\text{prec}} &= \text{median}_{c \in \mathcal{C}}(U_{c,k}^{\text{prec}}). \end{aligned} \tag{4.4}$$

Annotated ranges in the source text (e.g. "+30% to +60%") contribute both endpoints to $\mathcal{P}_{c,k}$ as separate values rather than being interpreted as literal lower and upper bounds.

4.1.9.3 Blending the two evidence sources

The two interval triples are combined into a single evidence-blended triple through a support-weighted average, following the standard principle in evidence combination that sources of greater support contribute proportionally more to the aggregate estimate. Let s^{hist} and s^{prec} denote the support associated with the own-history and external precedent intervals respectively, quantifying the strength of evidence underlying each source. The own-history support accumulates rank-discounted squared similarity

scores across the retained analogues, while the precedent support scales with the number of retrieved cases. The corresponding weights are

$$w^{\text{hist}} = \frac{s^{\text{hist}}}{s^{\text{hist}} + s^{\text{prec}}}, \qquad w^{\text{prec}} = 1 - w^{\text{hist}}, \tag{4.5}$$

and the blended interval for phase k is

$$\begin{aligned} L_k &= w^{\text{hist}} L_k^{\text{hist}} + w^{\text{prec}} L_k^{\text{prec}}, \\ E_k &= w^{\text{hist}} E_k^{\text{hist}} + w^{\text{prec}} E_k^{\text{prec}}, \\ U_k &= w^{\text{hist}} U_k^{\text{hist}} + w^{\text{prec}} U_k^{\text{prec}}. \end{aligned} \tag{4.6}$$

The triple (L_k, E_k, U_k) constitutes the quantitative anchor for phase k that is passed to the language model.

4.1.9.4 Mapping the interval to the three scenarios

The language model receives the blended triple together with the textual descriptions of the matched analogues and is prompted to produce three scenario specifications, indexed by risk level r where $r \in \{\text{optimistic}, \text{expected}, \text{conservative}\}$. The prompt binds these scenarios explicitly to the bounds of the interval rather than allowing the model to generate three independent narratives: the optimistic scenario reads from the upper bound, the expected scenario from the median, and the conservative scenario from the lower bound. For an increasing event, the signed magnitude applied at phase k under risk level r is therefore $\mu_{k,r} \in \{+L_k, +E_k, +U_k\}$ for $r \in \{\text{conservative, expected, optimistic}\}$; for a decreasing event, the assignment is inverted such that the optimistic scenario corresponds to the smallest contraction.

4.1.9.5 Constructing the final forecast band

For each risk level r, the signed magnitudes $\{\mu_{k,r}\}_k$ are translated into a per-period temporal envelope $f^r(t)$ and applied to the baseline to produce three full scenario forecasts:

$$\hat{y}_t^r = \hat{y}_t^{\text{base}} \cdot (1 + f^r(t)), \quad r \in \{\text{optimistic}, \text{expected}, \text{conservative}\}. \tag{4.7}$$

The final forecast band is constructed period by period from these three curves. Let $\delta_t^r = \hat{y}_t^r - \hat{y}_t^{\text{base}}$ denote the deviation of scenario r from the baseline at period t. The band at period t is

$$\left[\hat{y}_t^{\text{base}} + \min_r \delta_t^r,\ \hat{y}_t^{\text{base}} + \max_r \delta_t^r\right], \tag{4.8}$$

clipped to a tolerance proportional to the expected deviation. Outside the event window, where the three scenarios coincide with the baseline, the band defaults to the baseline model's native 80% prediction interval. The resulting band is identical to the interval plotted in the forecast figures and evaluated by the hit rate and Winkler score in Chapter 5.

4.2 Datasets

4.2.1 Motivation for Simulated Data

A central challenge in evaluating event-aware forecasting systems is the absence of ground truth for event effects in real-world data. When a retailer observes a sales spike during Black Friday, the true counterfactual, i.e. what would have happened without the campaign, is unobservable. This makes it impossible to assess whether a forecasting system correctly attributed the effect to the right event, estimated the right magnitude,

or modeled the right temporal profile.

To address this, we construct simulated datasets with known data generating processes (DGPs). Each dataset is built from a controlled combination of trend, seasonality, autocorrelated noise, and explicitly defined event effects. Because we control the DGP, we know the exact timing, magnitude, and shape of every event injected into the series. This allows us to evaluate not only point forecast accuracy, but also whether the system's reasoning correctly identifies and quantifies the underlying causal structure.

All three datasets are designed to be structurally realistic: they exhibit day-of-week seasonality, plausible trend dynamics, and event patterns drawn from real-world domains. The trade-off is that simulated data necessarily lacks the full complexity of production time series, such as regime changes, measurement noise, and confounders that are difficult to model programmatically. We consider this an acceptable limitation for a controlled ablation study.

4.2.2 Data Generating Process

All datasets share a common multiplicative structure:

$$y(t) = \text{trend}(t) \times \text{weekday\_profile}(t) \times \big(1 + \text{event\_effects}(t)\big) \times \big(1 + \varepsilon(t)\big) \tag{4.9}$$

where the noise term $\varepsilon(t)$ follows an AR(1) process:

$$\varepsilon(t) = \varphi \cdot \varepsilon(t-1) + \eta(t), \qquad \eta(t) \sim \mathcal{N}(0, \sigma^2) \tag{4.10}$$

The multiplicative formulation means that event effects and noise scale

proportionally with the current level of the series, which is consistent with how percentage-based impacts manifest in real-world demand data.

Each dataset includes two categories of events:

- **Training-period events**: Historical events documented in the own-history knowledge and external precedent knowledge files. These provide the system with analogues for reasoning about future events.
- **Test-period events**: Events placed in the forecast window, described only in the future events file. The system must anticipate their impact using reasoning, not direct observation.

4.2.3 Dataset Descriptions

Health Center Case. This dataset simulates daily visit requests at a primary care center in Stockholm. Table 4.1 summarizes the DGP parameters.

Table 4.1: Health center case DGP parameters.

Property	Value
Observations	2,117
Date range	2019-03-01 to 2024-12-15
Seasonal pattern	Weekly (period = 7)
Trend	Logistic growth (plateau at ∼180 weekday visits)
AR(1) parameters	$\varphi = 0.35$, $\sigma = 0.04$
Random seed	42

The day-of-week profile reflects typical primary care demand: Monday 1.12, Tuesday 1.08, Wednesday 1.03, Thursday 0.98, Friday 0.90, Saturday 0.52, Sunday 0.37.

Seven training-period events are injected, including a COVID-19 health-

care avoidance drop (−45%, 8-week recovery), a nearby clinic closure causing permanent patient influx (+12%), an influenza wave (+18%, 14-day plateau with 7-day decay), and a pre-Christmas demand softening (−20% gradual ramp). Two test-period events challenge the system: a respiratory advisory surge (+15% for 12 days, 5-day decay) overlapping with a pre-Christmas demand softening (−18% gradual decline over 11 days). The primary forecasting challenge is detecting overlapping effects with different temporal profiles while maintaining the underlying weekly pattern.

Logistics Case. This dataset simulates daily parcel volumes at a logistics hub terminal. Table 4.2 summarizes the DGP parameters.

Table 4.2: Logistics case DGP parameters.

Property	Value
Observations	2,381
Date range	2019-06-01 to 2025-12-06
Seasonal pattern	Weekly (period = 7)
Trend	Linear (19,000 + 2.45 × days)
AR(1) parameters	$\varphi = 0.35$, $\sigma = 0.03$
Random seed	2026

The day-of-week profile reflects logistics operations: Monday 1.18 through Sunday 0.34, with an additional summer reduction factor (July: 0.92, August: 0.96). Training-period events include multiple years of recurring retail events: Singles Day pulses, Black Friday/Cyber Monday complex spikes (5 to 6 day buildup-peak-decay patterns), Easter dips, and Midsommar dips, totaling approximately 15 event instances across 2019 to 2024.

Four sequential test-period events are injected: a Singles Day pulse (2025-11-10, modest 2 to 4 day pulse), Black Week build-up (2025-11-24, moderate 3 to 5 day ramp), a Black Friday intake spike (2025-11-28, +90%

single-day spike), and an early-December delivery after-wave (5 to 7 days with gradual decay). The primary challenge is modeling four distinct but temporally adjacent events with different profiles, where the system must avoid conflating their individual effects.

Music Streaming Case. This dataset simulates daily global streams for a Coldplay-like artist catalogue. Table 4.3 summarizes the DGP parameters.

Table 4.3: Music streaming case DGP parameters.

Property	Value
Observations	2,000
Date range	2019-07-01 to 2024-12-26
Seasonal pattern	Weekly (period = 7) + annual sinusoidal cycle
Trend	Linear (base ∼21.5M +6,800/day)
Noise	Multiplicative Gaussian, $\sigma = 0.045$
Random seed	17

The day-of-week profile is weekend-heavy (Friday 1.18, Saturday 1.30, Sunday 1.24), opposite to the work-driven patterns in the other two cases. Training-period events follow the real-world Coldplay release calendar with stylized impact envelopes modeled as exponential decay with persistent uplift: the "Higher Power" single (+38% peak, 11-day half-life, +1.5% persistent), "My Universe" BTS collaboration (+72% peak, 13-day half-life, +3.0% persistent), "Music of the Spheres" album (+105% peak, 19-day half-life, +9.0% persistent), and a sustained world tour catalogue halo effect.

The knowledge bank contains comparable artist release benchmarks (Imagine Dragons, OneRepublic, Ed Sheeran, Taylor Swift, U2) with documented peak magnitudes, decay rates, and persistent uplift patterns. The test-period event is a single release ("feelslikeimfallinginlove", 2024-06-21: +52% peak, 14-day half-life, +2.0% persistent). The primary

challenge is estimating the spike-and-decay profile of a new release using cross-artist analogues from the knowledge bank.

4.3 Ablation Design

4.3.1 Conditions

The experiment uses an ablation design that varies three information sources: the own-history knowledge (*events.txt*, dated historical events from the target series' own history), the external precedent knowledge (*knowledge.txt*, cross-domain precedent cases from analogous series), and the future events (*suggested_events.txt*, the anticipated events whose impact the pipeline must reason about). Table 4.4 summarizes which sources are active in each condition.

Table 4.4: Ablation conditions by active information sources.

	C1	**C2**	**C3**	**C4**	**C5**	**C6**
Own-history	✓	×	✓	×	×	×
Ext. precedent	✓	✓	×	×	×	×
Future events	✓	✓	✓	✓	×	×
LLM pipeline	✓	✓	✓	✓	✓	×

The conditions are designed as follows:

- **C1** (full system): all information sources active, serving as the performance ceiling.
- **C2** (no own-history): removes own-history knowledge, isolating its marginal contribution.
- **C3** (no external precedent): removes external precedent knowledge, isolating its marginal contribution.

- **C4** (context only): removes both own-history and external precedent knowledge, leaving only statistical analysis and scenario reasoning via future events.
- **C5** (baseline): removes all augmentation; the system operates on raw time series and domain description only, with no scenario reasoning.
- **C6** (foundation model baseline): bypasses the LLM pipeline entirely and produces a zero-shot forecast directly from the concatenated train and validation series, with no access to own-history knowledge, external precedent knowledge, future events, or domain context.

This structure enables the following decompositions:

- C1 − C2: marginal value of own-history knowledge
- C1 − C3: marginal value of external precedent knowledge
- C1 − C4: combined value of both information sources
- (C1 − C2) − (C3 − C4): interaction effect between own-history and external precedent knowledge
- C4 − C5: marginal value of scenario reasoning via future events

In all conditions, the system retains access to the raw time series data, its own statistical analysis (trend, seasonality, ACF/PACF), and the domain context description.

4.3.2 Blinding

To prevent evaluation bias, each run is assigned a blinded identifier computed as a SHA-256 hash of the experiment ID, dataset ID, condition ID, and replicate number. The judge evaluation packet never contains the condition name or label. Judge runs are executed in a randomized order to prevent ordering effects.

4.3.3 External Baseline

To benchmark the LLM pipeline against a state of the art purely numerical foundation model, a sixth condition (C6) is added that bypasses the agent system entirely with zero shot forecast from Chronos-Bolt Small, an approximately 48-million-parameter T5-based variant of the Chronos family (Ansari et al. 2024). The model receives only the numerical training series, namely the concatenation of train and validation splits, and produces quantile forecasts for the test horizon directly. No events, knowledge bank, future events, or domain context is provided. Inference runs on CPU in bfloat16. The 80% prediction interval is constructed from the model's native q10 and q90 heads, matching the nominal coverage of the LLM scenario interval ($\alpha = 0.20$).

Unlike C1 through C5, which constitute an ablation of the LLM pipeline, C6 is an external model class included to assess whether modern numerical foundation models can match the proposed system on test windows where events occur. C6 is not evaluated by the LLM judge, since judge dimensions such as evidence grounding and causal coherence presuppose a reasoning trace that Chronos does not produce.

4.4 Forecast Pipeline

Each experimental run executes the full multi-agent pipeline in the order described in Section 4.1, from statistical decomposition through domain expert reasoning, hypothesis generation, iterative parameter refinement, aggregation, and finally scenario generation. The forecast model is GPT-5.4 (OpenAI). The forecast start date is automatically set to 7 days before the first future event, providing a baseline window before event effects begin. The evaluation window is fixed at 42 periods (6 weeks) from

the scenario anchor point, selected to ensure that all test-period events across the three datasets, including their full impact duration and decay phases, fall within the evaluation window. For the C6 condition, none of the agents are invoked and Chronos Bolt Small produces a median path and an 80% interval directly from the concatenated train and validation series.

4.5 Evaluation

4.5.1 Quantitative Metrics

Point forecast accuracy is measured over the evaluation window using the following metrics:

- **MASE** (Mean Absolute Scaled Error): scale-free metric using a seasonal naive baseline as denominator, suitable for cross-dataset comparison.
- **sMAPE** (Symmetric Mean Absolute Percentage Error): bounded and symmetric percentage error.

Interval forecast quality is assessed using:

- **Hit rate**: fraction of actuals within the forecast interval. Since the interval is constructed from the range of historically observed analogue outcomes rather than from a parametric distributional assumption, no nominal coverage level is guaranteed. The hit rate is instead interpreted as an empirical measure of how well past evidence bounds future outcomes.
- **Winkler score**: jointly penalizes under-coverage and excessive inter-

val width.

The primary evaluation window is the *event window*, defined as the period from the scenario anchor point through the inferred impact duration. This focuses evaluation on the period where event effects are present. Full-horizon metrics are reported separately.

4.5.2 Assessing Pipeline with LLM-as-a-Judge

Quantitative metrics measure what the system produces but not how it reasons. To evaluate reasoning quality, we employ an independent LLM judge (Claude Sonnet 4, Anthropic) that scores each run on seven dimensions using a 1 to 5 rubric: event identification, magnitude calibration, temporal precision, evidence grounding, causal coherence, scenario differentiation, and leakage discipline. The judge receives a blinded packet containing the system's reasoning trace, scenario specifications, and scoring contract, but not the condition label or quantitative metrics. Using a different model family (Anthropic) than the forecast model (OpenAI) reduces the risk of self-enhancement bias, one of the systematic biases documented in the LLM-as-a-judge literature.

4.6 Reproducibility

The full experiment is defined by a single configuration file specifying datasets, model identifiers, ablation conditions, and evaluation parameters. Each run produces a complete set of artifacts: reasoning trace, scenario forecasts, quantitative metrics, forecast plot, and judge evaluation. All artifacts are persisted to a timestamped results directory

with a manifest linking runs to their conditions and datasets. The code, configuration, and generated datasets are version-controlled.

Chapter 5

Results

5.1 Health Center Case

The health center case results are reported in Tables 5.1 and 5.2. The full pipeline (C1) achieves a sMAPE of 5.7±2.2%, approximately half that of C2 (11.7±3.8%) and well below the no-augmentation baseline C5 (14.0%). Removing own-history knowledge (C2) produces a larger accuracy degradation than removing external precedent knowledge (C3, sMAPE 9.4±3.7%), reflecting the richness of the health center series' own event history, which provides well-matched analogues for the test-period effects. The context-only condition C4 (11.0±2.9%) confirms that neither source alone is sufficient to recover the accuracy of the full pipeline.

Interval quality follows the same ordering. C1 achieves a Winkler score of 35.1±18.3%, markedly better than C5 (210.2%), where the absence of scenario reasoning produces intervals that are both excessively wide and poorly positioned. The high variance in C1's Winkler score across replicates reflects sensitivity to how the two overlapping test-period effects

are combined in individual runs. Figure 5.1 illustrates a representative C1 run, where the scenario-adjusted path captures both the respiratory advisory surge and the pre-Christmas softening while the unadjusted baseline systematically underestimates demand throughout.

The LLM judge scores reveal two consistent patterns. Evidence grounding (4.6±0.5 for C1) and leakage discipline (5.0 across all conditions) are the strongest dimensions, confirming that the system stays reliably within its information budget. Scenario differentiation is the weakest dimension across all conditions, with no condition exceeding 2.6, likely a consequence of the overlapping event structure leaving little room for differentiated scenario branching. Chronos (C6) recovers the weekly visit cycle accurately but systematically underestimates demand during the respiratory advisory window, producing a MASE of 0.6 against C1's 0.3, consistent with its structural inability to ingest the public health event signal.

Table 5.1: Healthcenter case: mean ± std across 5 replicates (C1 to C4), single replicate for C5 and C6. Where no ± value is reported, the standard deviation across replicates was zero. **Bold** marks the best condition per metric.

Metric	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
SMAPE (%)	$\mathbf{5.7 \pm 2.2}$	11.7 ± 3.8	9.4 ± 3.7	11.0 ± 2.9	14.0	11.7
MASE	$\mathbf{0.3 \pm 0.1}$	0.6 ± 0.1	0.4 ± 0.2	0.5 ± 0.1	0.7	0.6
Hit rate (%)	$\mathbf{48.8 \pm 8.7}$	32.0	36.0 ± 21.2	28.0	40.0	43.8
Winkler (%)	$\mathbf{35.1 \pm 18.3}$	80.5 ± 15.9	59.3 ± 35.4	55.3 ± 6.7	210.2	57.2
Judge (1–5)	**4.0**	**4.0**	3.8 ± 0.4	3.0	1.0	N/A

Table 5.2: Healthcenter case: LLM judge scores (1–5), averaged over 5 replicates. **Bold** marks the best condition per dimension.

Dimension	**C1**	**C2**	**C3**	**C4**	**C5**
Event identification	4.2	**5.0**	4.2	4.0	1.0
Magnitude calibration	**4.0**	**4.0**	3.2	2.8	1.0
Temporal precision	3.4	**4.0**	**4.0**	3.8	1.0
Evidence grounding	4.6	**5.0**	4.2	1.8	1.0
Causal coherence	3.6	**4.0**	3.4	3.2	1.0
Scenario differentiation	2.0	**2.6**	2.2	2.0	1.0
Leakage discipline	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Overall	**4.0**	**4.0**	3.8	3.0	1.0

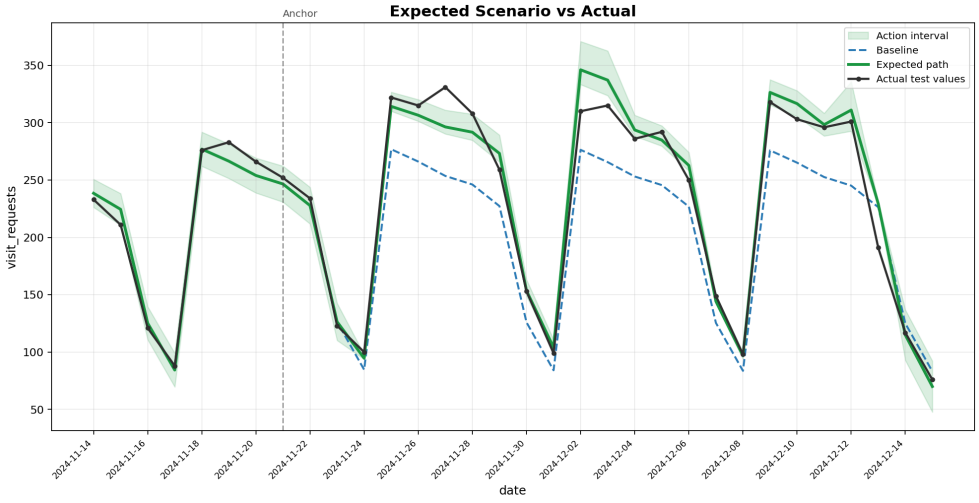

Figure 5.1: Representative C1 forecast for the health center case. The dashed vertical line marks the scenario anchor point in late November 2024. The expected scenario path (green) captures both the respiratory advisory surge and the pre-Christmas demand softening, while the unadjusted baseline (blue) systematically underestimates demand during the surge period.

5.2 Logistics case

The logistics case results are reported in Tables 5.3 and 5.4. The full pipeline (C1) achieves a sMAPE of 5.8±0.2% and a MASE of 0.2, the strongest point accuracy results observed across all three datasets. The most notable pattern relative to the health center case is the inversion in the relative importance of the two grounding sources. Here removing external precedent knowledge (C3, sMAPE 11.3±1.4%) is more damaging than removing own-history knowledge (C2, 8.4±1.0%), reflecting the availability of multiple years of well-matched Black Friday and Singles Day precedents in the knowledge bank that provide strong magnitude anchors for all four sequential test-period events. The context-only condition C4 (14.5±0.9%) and the no-augmentation baseline C5 (17.2%) confirm that both sources contribute non-redundantly when the test window contains multiple high-magnitude events.

Interval quality is the strongest observed across all three datasets, with C1 achieving a Winkler score of 22.2±0.8%. C5 produces the largest Winkler score observed across the entire experiment (327.8%), reinforcing that the absence of scenario reasoning is particularly costly when the test window is dominated by discrete, high-magnitude events such as the Black Friday spike. Figure 5.2 illustrates a representative C1 run, where the scenario path closely tracks all four sequential events while the unadjusted baseline remains near the pre-event level throughout.

Judge scores are the strongest across all datasets on event identification (5.0 for C1) and evidence grounding (5.0 for C1), reflecting the availability of well-matched precedents. Scenario differentiation remains the weakest dimension (2.8±0.4), consistent with the pattern observed in the health center case. Chronos (C6) misses all four sequential events, producing a MASE of 0.7 against C1's 0.2, but its wider baseline intervals result in a Winkler score of 85.3%, dramatically better than C5 (327.8%).

Table 5.3: Logistics case: mean ± std across 5 replicates (C1 to C4), single replicate for C5 and C6. Where no ± value is reported, the standard deviation across replicates was zero. **Bold** marks the best condition per metric.

Metric	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
SMAPE (%)	$\mathbf{5.8 \pm 0.2}$	8.4 ± 1.0	11.3 ± 1.4	14.5 ± 0.9	17.2	14.3
MASE	**0.2**	0.3 ± 0.1	0.6 ± 0.1	0.7	0.8	0.7
Hit rate (%)	$\mathbf{63.8 \pm 2.1}$	50.0 ± 4.7	47.7 ± 11.1	50.8 ± 4.2	57.7	63.6
Winkler (%)	$\mathbf{22.2 \pm 0.8}$	37.3 ± 4.2	71.3 ± 18.4	88.6 ± 6.1	327.8	85.3
Judge (1–5)	**4.0**	**4.0**	**4.0**	3.0	1.0	N/A

Table 5.4: Logistics case: LLM judge scores (1–5), averaged over 5 replicates. **Bold** marks the best condition per dimension.

Dimension	**C1**	**C2**	**C3**	**C4**	**C5**
Event identification	**5.0**	4.4	4.0	4.0	1.0
Magnitude calibration	**4.0**	3.0	3.0	2.0	1.0
Temporal precision	**4.0**	**4.0**	**4.0**	3.0	1.0
Evidence grounding	**5.0**	4.2	4.0	1.0	1.0
Causal coherence	**4.0**	3.8	3.6	3.0	1.0
Scenario differentiation	**2.8**	2.0	2.0	2.0	1.0
Leakage discipline	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Overall	**4.0**	**4.0**	**4.0**	3.0	1.0

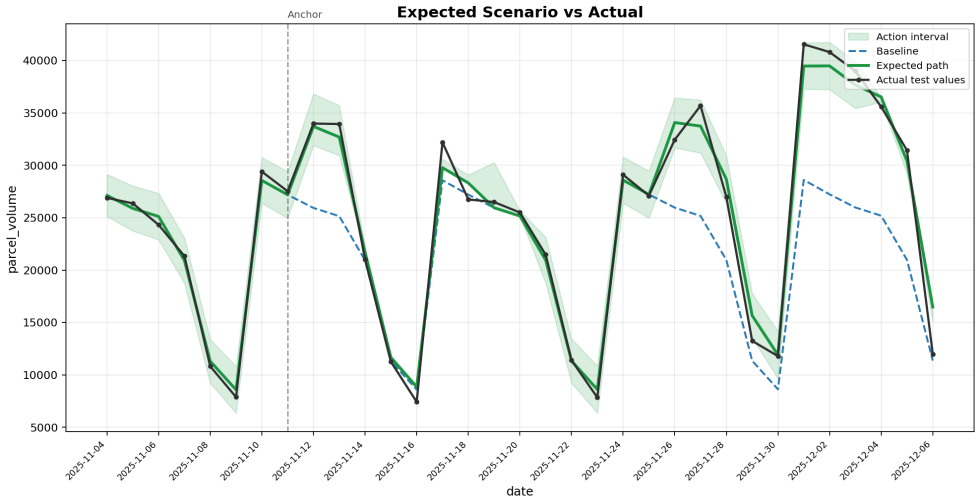

Figure 5.2: Representative C1 forecast for the logistics case. The dashed vertical line marks the scenario anchor point coinciding with the Singles Day pulse. The expected scenario path (green) accurately tracks all four sequential test-period events while the unadjusted baseline (blue) remains near the pre-event level throughout.

5.3 Music Stream Case

The Music streaming case results are reported in Tables 5.5 and 5.6. The quantitative pattern diverges substantially from the two previous cases. C1 achieves a sMAPE of 8.1±1.2%, marginally higher than both C2 (7.7%) and C3 (8.0±1.0%), indicating that neither grounding source provides a decisive additive benefit for point accuracy when the test window contains a single well-precedented spike-and-decay event. The scenario generator appears able to produce a calibrated adjustment using either source alone, consistent with the Music dataset structure where cross-artist release analogues in the knowledge bank carry sufficient magnitude information on their own.

The most instructive finding in this case is the C4 collapse. Removing both grounding sources while retaining the future event (C4) produces a sMAPE of 30.4% and a Winkler score of 325.4±0.9%, with a hit rate of 0.0%. Without any quantitative anchor for the release magnitude, the scenario generator severely underestimates the spike and produces a narrow interval that fails to capture the actual outcome entirely. Figure 5.4 illustrates this failure, and Figure 5.3 shows the corresponding C1 run where access to cross-artist precedents produces a well-calibrated forecast of the same event. This contrast is the clearest illustration in the study of what grounding contributes to LLM event reasoning.

Judge scores are largely consistent with the other two cases. Evidence grounding degrades monotonically from 5.0 for C1 to 1.6±0.5 for C4, and leakage discipline is perfect throughout. Magnitude calibration for C3 (2.8±0.4) is notably lower than for C1 and C2 (both 4.0), reflecting that the absence of own-history analogues impairs magnitude precision even when interval coverage is high. Chronos (C6) predicts a stable baseline that misses the release spike entirely, producing a MASE of 2.7 against C1's 1.1 and the lowest interval coverage observed across all C6 conditions.

Table 5.5: Music case: mean ± std across 5 replicates (C1 to C4), single replicate for C5 and C6. Where no ± value is reported, the standard deviation across replicates was zero. **Bold** marks the best condition per metric.

Metric	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
SMAPE (%)	8.1 ± 1.2	**7.7**	8.0 ± 1.0	30.4	36.1	24.2
MASE	1.1 ± 0.2	**1.0**	1.1 ± 0.2	3.3	3.9	2.7
Hit rate (%)	66.7 ± 14.7	69.5 ± 6.4	$\mathbf{95.2 \pm 6.7}$	0.0	42.9	21.4
Winkler (%)	$\mathbf{49.9 \pm 8.2}$	$\mathbf{49.9 \pm 4.0}$	54.8 ± 5.6	325.4 ± 0.9	498.5	222.4
Judge (1–5)	**4.0**	**4.0**	3.8 ± 0.4	3.0	1.0	N/A

Table 5.6: Music case: LLM judge scores (1–5), averaged over 5 replicates. **Bold** marks the best condition per dimension.

Dimension	**C1**	**C2**	**C3**	**C4**	**C5**
Event identification	**4.0**	**4.0**	**4.0**	**4.0**	1.0
Magnitude calibration	**4.0**	**4.0**	2.8	2.0	1.0
Temporal precision	**4.0**	**4.0**	3.8	3.4	1.0
Evidence grounding	**5.0**	4.4	4.0	1.6	1.0
Causal coherence	**4.0**	3.6	3.8	3.0	1.0
Scenario differentiation	2.0	**2.2**	2.0	2.0	1.0
Leakage discipline	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Overall	**4.0**	**4.0**	3.8	3.0	1.0

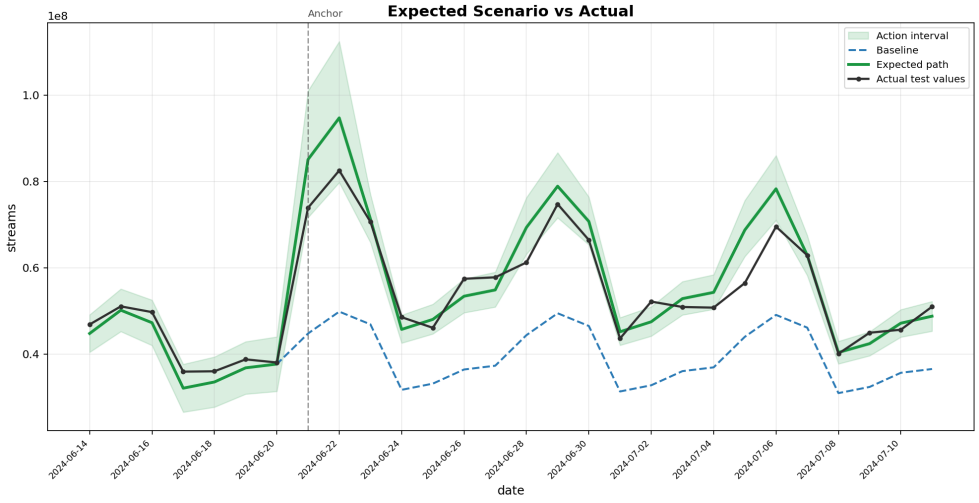

Figure 5.3: Representative C1 forecast for the Music streaming case. The expected scenario path (green) and its historically-derived interval diverge substantially from the unadjusted baseline (blue) after the anchor point, with actual test values (black) tracking the path closely through the release spike and subsequent decay.

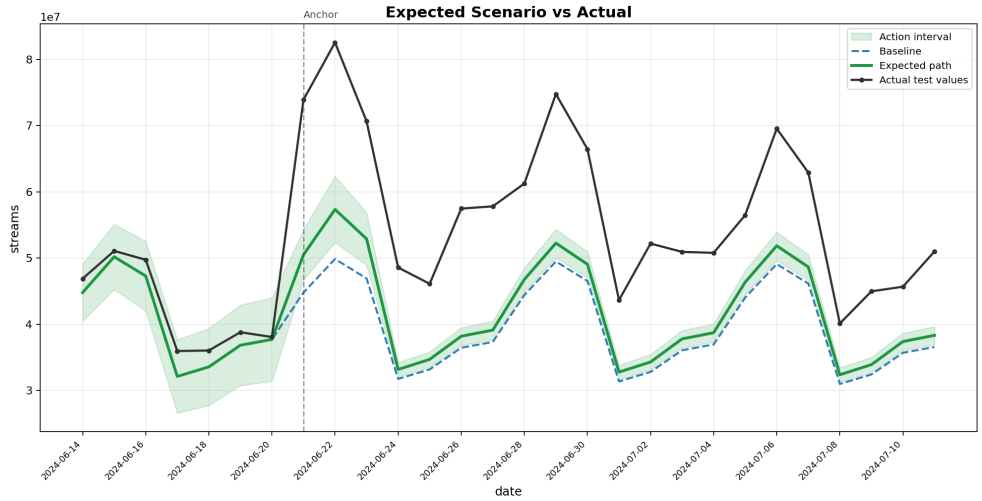

Figure 5.4: Representative C4 forecast for the Music streaming case, illustrating the interval collapse when both grounding sources are removed. Despite the future event being provided, the scenario generator lacks any quantitative anchor for the release magnitude, resulting in a severely underestimated expected path and a narrow interval that fails to capture the actual spike entirely. Compare with Figure 5.3.

5.4 Cross-Dataset Analysis

Table 5.7 synthesizes the ablation results across all three datasets and constitutes the primary empirical basis for answering the research questions posed in Section 1.1. Three cross-cutting patterns emerge from the comparison.

The first concerns the relative importance of the two grounding sources. In the health center case, removing own-history knowledge (C2) is more damaging than removing external precedent knowledge (C3), whereas the reverse holds for the logistics case. This inversion reflects the structural

match between available evidence and the test-period event type: the health center series has a rich own-history of analogous demand shocks, while the logistics case relies more heavily on cross-domain retail precedents to quantify recurring commercial events. The Music case shows negligible degradation for both C2 and C3, confirming that for a single spike-and-decay event either source alone is sufficient. The dominant grounding source is therefore determined by the evidence structure rather than by a universal hierarchy between the two sources.

The second pattern is the catastrophic failure of C4 on the Music case (+275.9%), far exceeding the degradation observed in the health center (+91.4%) or logistics (+147.2%) cases. When the test event has no structural analogue in own history and no knowledge bank to draw from, the scenario generator lacks any quantitative anchor and produces severely miscalibrated adjustments. This points to a clear boundary condition for LLM event reasoning: it requires at least one grounding source to function reliably.

The third pattern is the consistent ordering $\Delta C5 > \Delta C4$ across all three datasets, confirming that scenario reasoning via future events adds value even when both grounding sources are absent. The marginal contribution of scenario reasoning alone ranges from 53.3 percentage points in the health center case to 71.0 percentage points in the Music case, suggesting it is most valuable when the event structure is concentrated in a single well-defined effect.

Taken together, the results establish that the value of LLM event reasoning is strongly dataset-dependent and governed primarily by the match between the test-period event structure and the available grounding. The full pipeline is never the worst condition and is the best on point accuracy in two of three cases, but the Music anomaly reveals a boundary: when own-history events are structurally dissimilar to the test event, adding

them can introduce noise rather than signal.

Table 5.7: Ablation impact: relative change in SMAPE (%) when removing components, compared to the full pipeline (C1). Positive values mean the ablation *worsened* accuracy.

Dataset	**Δ C2**	**Δ C3**	**Δ C4**	**Δ C5**
Healthcenter	+104.8	+64.1	+91.4	+144.7
Logistics	+44.4	+92.9	+147.2	+194.3
Music	-5.0	-0.6	+275.9	+346.9

5.5 Latency and Cost

Table 5.8 reports the mean runtime per condition across datasets. C5 is consistently the fastest condition, averaging 29.3±3.4 seconds compared to 43.6±5.6 seconds for the full pipeline (C1), representing an overhead factor of approximately 1.5 for the complete agentic architecture. This overhead is modest relative to the accuracy gains observed in the health center and logistics cases, where C1 reduces sMAPE by 59% and 66% respectively compared to C5. Conditions C1 through C4 show comparable latency across all datasets, as all four conditions share the same pipeline architecture. The elevated mean for C4 in the health center case (54.6±8.8 seconds) is a consequence of two outlier replicates pulling the mean upward; the remaining three replicates fall within the normal range.

In terms of token usage, the forecast pipeline consumed an estimated ∼1.21M input tokens and ∼324K output tokens across the 75 runs, averaging ∼16,100 input and ∼4,300 output tokens per run. The input-to-output ratio of approximately 3.7:1 reflects the grounding-heavy nature of the pipeline, where evidence packets, historical event logs, and knowl-

edge bank entries constitute the bulk of each prompt. The total API cost across the 75-run experiment was an estimated $5.58, yielding an average of approximately $0.07 per forecast run, covering the forecasting pipeline (GPT-5.4) only.

The external baseline (C6) bypasses the LLM API entirely and runs locally on CPU, with a cold-start latency of approximately 12 seconds per dataset and zero marginal API cost after a one-time model download. This represents a runtime improvement of roughly 3.5× relative to C1 for single dataset deployment, at an accuracy cost of roughly 2.5× higher MASE on event-driven test windows.

Table 5.8: Mean runtime in seconds per condition, averaged over 5 replicates (C1 to C4) and a single replicate for C5 and C6. C6 measurements reflect a single batch process: the first dataset (Healthcenter) pays the full Chronos model load cost, while subsequent datasets reuse the cached model in memory. The cold start cost of 12.4 seconds is the relevant figure for single dataset deployment where each forecast runs as an independent process. **Bold** marks the fastest condition per dataset.

Dataset	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
Healthcenter	41.7 ± 4.9	40.5 ± 5.2	41.2 ± 5.3	54.6 ± 8.8	30.9 ± 3.6	**12.4**
Logistics	46.0 ± 8.2	42.1 ± 3.9	44.5 ± 2.3	43.0 ± 2.6	30.9 ± 2.0	**0.4**
Music	43.1 ± 2.8	42.9 ± 3.8	44.6 ± 1.2	42.2 ± 2.2	26.1 ± 2.0	**0.4**
Average	43.6 ± 5.6	41.9 ± 4.1	43.4 ± 3.6	46.6 ± 7.7	29.3 ± 3.4	**4.4**

Chapter 6

Discussion

6.1 Does LLM Event Reasoning Produce Measurable Signal?

A central question motivating this thesis is whether LLMs can meaningfully reason over natural language event context to produce calibrated adjustments to time series forecasts. The results suggest that they can, under the right grounding conditions. The full pipeline reduces sMAPE by approximately 59% in the health center case and 66% in the logistics case relative to the no-augmentation baseline, and this gap is not attributable to the numerical machinery of the pipeline, which is identical across conditions, but to the LLM's ability to translate natural language event descriptions into calibrated multiplicative adjustments anchored in historical evidence.

The Music streaming case qualifies this conclusion. When the test window contains a single well-precedented event, either grounding source

alone carries sufficient evidence for the reasoning layer to produce a calibrated adjustment, and the marginal contribution of the full pipeline over its better-grounded ablations is small. The measurable signal produced by LLM event reasoning therefore scales with the structural heterogeneity of the events to be modelled and the richness of the available grounding, not with architectural complexity per se.

6.2 What Chronos Reveals About the LLM's Contribution

The inclusion of Chronos Bolt as an external baseline is not primarily a competitive comparison but a controlled contrast that isolates what the LLM contributes to the pipeline. Any performance difference between C1 and C6 on event-driven test windows reflects the value of the LLM's linguistic reasoning capability specifically, since the numerical foundations of the two systems are otherwise comparable.

The contrast is unambiguous. Across all three datasets, C6 produces MASE values 2.0–2.5 times those of C1, and its prediction intervals systematically miss the event-driven peaks in the health center surge and the music release. This is not a failure of the numerical model but a structural consequence of its design: Chronos recovers seasonal and trend structure accurately but cannot anticipate level shifts driven by events it has never observed. Critically, C6 outperforms C5 across all three datasets, confirming that the competitive ordering is grounded event reasoning versus ungrounded prediction rather than LLM versus foundation model. The LLM's value lies in its reasoning over linguistic event context, not in its numerical computation.

6.3 Numerical Accuracy and Reasoning Quality Are Distinct

The LLM-as-a-judge results expose a consistent dissociation between numerical accuracy and reasoning quality. The clearest illustration is C2 in the health center case, which scores comparably to C1 across most judge dimensions despite producing more than double the sMAPE. Coherent and well-grounded reasoning does not guarantee well-calibrated forecasts, and the inverse also holds: accurate forecasts can be produced through reasoning that would not survive operational scrutiny. Neither metric class is therefore sufficient on its own, and the dual evaluation framework employed here is a methodological necessity rather than a redundancy for any LLM-based forecasting system intended to support human decision-making.

6.4 Grounding Discipline and the Limits of Reasoning

A key finding across all datasets and conditions is that the system behaves transparently with respect to its evidence base. Leakage discipline is perfect throughout, and evidence grounding degrades monotonically from C1 to C4, confirming that the LLM does not generate false confidence when grounding sources are removed. The Music case C4 collapse is the most instructive illustration: rather than fabricating wider intervals to achieve nominal coverage, the system correctly produces a narrow interval consistent with the limited evidence available. The cost is that the actual outcome falls entirely outside the bound, but this is the correct behaviour. A system that invents confidence it does not have is far

more dangerous in an operational setting than one that signals its own uncertainty honestly.

The weakest reasoning dimension across all conditions is scenario differentiation, with no condition exceeding 2.8 on the 1–5 scale. The current prompting strategy anchors all three scenario branches in the same evidence packet, leaving little room for genuinely distinct optimistic and conservative narratives. This is a targeted limitation of the scenario generation layer rather than a fundamental property of LLM event reasoning, and is addressed in the future work section.

6.5 Limitations

The most significant limitation of this study is that the evaluation is conducted entirely on simulated datasets with known data generating processes. While this design choice is methodologically justified, since it enables ground truth for event effects and allows the ablation to isolate reasoning contributions cleanly, it also means that the datasets are structurally aligned with the system's design in ways that real-world data would not be. The event effects injected into the DGPs are multiplicative, temporally bounded, and recoverable from historical analogues, precisely the conditions under which the scenario generation layer is designed to perform well. Real-world time series may exhibit event effects that are non-stationary in magnitude, confounded by simultaneous drivers, or structurally dissimilar to any available analogue, all of which would challenge the retrieval and grounding mechanisms in ways the current evaluation cannot capture. The performance gaps reported here should therefore be interpreted as an upper bound on what the system can achieve under favourable conditions rather than as an estimate of expected real-world performance. Validating the findings on production

time series is the most important direction for future work.

The experimental design uses five replicates for the LLM-driven conditions and a single replicate for C5 and C6. This is sufficient to expose the directional effects emphasised in this discussion, but finer comparisons between similarly performing conditions, such as the Music case ranking of C1, C2, and C3, would benefit from a larger replicate count to distinguish genuine differences from sampling variation in the LLM's stochastic outputs.

The LLM-as-a-judge methodology is subject to known biases including position bias, verbosity bias, and self-enhancement bias (Zheng et al. 2023). The use of a different model family for judging mitigates the last of these, but absolute judge scores should be interpreted as ordinal rankings rather than cardinal measurements. Finally, all forecasts use a single LLM and a single judge model, meaning the findings cannot cleanly separate architectural effects from model-specific capabilities.

Chapter 7

Conclusion and Future Work

This thesis investigated whether Large Language Models can meaningfully reason over natural language event context to produce calibrated adjustments to time series forecasts, and under what conditions that reasoning produces reliable signal. The core design principle of delegating all numerical computation to validated statistical implementations and isolating the LLM's contribution to the reasoning layer allows forecast quality differences between conditions to be attributed directly to event reasoning rather than to numerical machinery.

The results support a qualified affirmative answer. LLM event reasoning produces substantial accuracy gains in two of three cases studied, with the full pipeline reducing sMAPE by approximately 59% and 66% in the health center and logistics cases respectively relative to the no-augmentation baseline. The Music streaming case qualifies this: when a single well-precedented event dominates the test window, either grounding source alone is sufficient and the marginal contribution of the full architecture is small. The value of LLM event reasoning therefore scales with the structural heterogeneity of the events to be modelled and the

richness of the available grounding, not with architectural complexity per se.

The comparison with Chronos Bolt and the ablation results together establish that the competitive ordering is grounded event reasoning versus ungrounded prediction, not LLM versus foundation model. The dual evaluation framework further reveals that numerical accuracy and reasoning quality are partially independent, underscoring that neither metric class alone is sufficient to characterise LLM reasoning capability. The system's consistently strong grounding behaviour, transparent degradation as evidence is removed, and correct interval collapse under insufficient grounding, indicates that the reasoning layer behaves in a trustworthy and auditable way.

Taken together, the results position agentic LLM orchestration as a principled instrument for translating natural language event knowledge into calibrated numerical adjustments, rather than a general purpose forecasting improvement. Its practical value is conditional on the availability of grounding sources and the presence of event-driven deviations from baseline patterns.

Future Work

Several directions follow naturally from the findings and limitations of this study. The most immediate is extending the evaluation to real-world time series, where lower signal-to-noise ratios and unobserved confounders may compress the performance gaps observed here. A covariate-aware foundation model supplied with the same event indicators the LLM pipeline consumes as text would also provide a stronger isolation of the reasoning contribution, clarifying whether the pipeline's advan-

tage stems from event identification or from the reasoning process itself. Within the current architecture, the consistently weak scenario differentiation across all conditions motivates refinement of the scenario generation layer, specifically constructing optimistic and conservative branches from distinct evidence subsets rather than anchoring all three branches in the same evidence packet. Finally, the current evaluation focuses exclusively on test windows where events are known to occur. An important open question is whether the LLM pipeline retains its advantage on event-free windows, where the numerical foundation model has no informational handicap. This evaluation gap is particularly relevant for assessing operational reliability: a system that produces meaningful event-adjusted forecasts when events occur, but that hallucinates spurious event effects when the forecast window is genuinely uneventful, would be unsuitable for production deployment despite strong performance in the present evaluation. Characterising performance across both event and non-event windows, with particular attention to false positive event identification and to the calibration of the system's confidence when no event signal is present, would provide a more complete picture of when agentic LLM orchestration is and is not the right architectural choice.

Bibliography

Adimulam, A., R. Gupta, and S. Kumar (2026). *The Orchestration of Multi-Agent Systems: Architectures, Protocols, and Enterprise Adoption*. arXiv: 2601.13671 [cs.MA].

Ansari, A. F., L. Stella, A. C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. Pineda Arango, S. Kapoor, et al. (2024). "Chronos: Learning the Language of Time Series". In: *arXiv preprint arXiv:2403.07815*.

Chang, C., Y. Shi, D. Cao, W. Yang, J. Hwang, H. Wang, J. Pang, W. Wang, Y. Liu, W.-C. Peng, and T.-F. Chen (2025). *A Survey of Reasoning and Agentic Systems in Time Series with Large Language Models*. arXiv: 2509.11575 [cs.AI].

Chow, W., L. Gardiner, H. T. Hallgrímsson, M. A. Xu, and S. Y. Ren (2024). *Towards Time Series Reasoning with LLMs*. arXiv: 2409.11376 [cs.LG].

Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning (1990). "STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)". In: *Journal of Official Statistics* 6, pp. 3–73.

Gneiting, T. and A. E. Raftery (2007). "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477, pp. 359–378. eprint: https://doi.org/10.1198/016214506000001437.

Helske, J. (2017). "KFAS: Exponential Family State Space Models in R". In: *Journal of Statistical Software* 78.10, pp. 1–39.

Hyndman, R. J. and A. B. Koehler (2006). "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4, pp. 679–688.

Jiang, Y., Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song (2024). *Empowering Time Series Analysis with Large Language Models: A Survey*. arXiv: 2402.03182 [cs.LG].

Jin, M., S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen (2024). *Time-LLM: Time Series Forecasting by Reprogramming Large Language Models*. arXiv: 2310.01728 [cs.LG].

Kim, K., H. Tsai, R. Sen, A. Das, Z. Zhou, A. Tanpure, M. Luo, and R. Yu (2024). *Multi-Modal Forecaster: Jointly Predicting Time Series and Textual Data*. arXiv: 2411.06735 [cs.AI].

Li, H., X. Chen, Z. Xu, D. Li, N. Hu, F. Teng, Y. Li, L. Qiu, C. J. Zhang, Q. Li, and L. Chen (2025). "Exposing Numeracy Gaps: A Benchmark to Evaluate Fundamental Numerical Abilities in Large Language Models". In: *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20004–20026.

Makridakis, S. (1993). "Accuracy measures: theoretical and practical concerns". In: *International Journal of Forecasting* 9.4, pp. 527–529.

Mathew, J. G. and J. Rossi (2025). “Large Language Model Agents”. In: *Engineering Information Systems with Large Language Models*. Springer, pp. 173–205.

Panjala, M., N. C. Bhatracharyulu, V. Alugolu, A. K. Singh, G. K. Gupta, and R. Kishor (2025). “Statistical Analysis of Time Series Data”. In: *Signal Processing, Telecommunication and Embedded Systems: Automation and Sustainability Applications*. Ed. by V. Bhateja, W. Flores-Fuentes, A. Bhattacharya, and P. S. R. Chowdary. Singapore: Springer Nature Singapore, pp. 255–262.

Shen, X., Y. Liu, Y. Dai, Y. Wang, R. Miao, Y. Tan, S. Pan, and X. Wang (2025). *Understanding the Information Propagation Effects of Communication Topologies in LLM-based Multi-Agent Systems*. arXiv: 2505.23352 [cs.MA].

Wang, S., P. Chen, Y. Wang, W. Qiu, C. Guo, B. Yang, and Y. Shu (2026). “Unlocking the Value of Text: Event-Driven Reasoning and Multi-Level Alignment for Time Series Forecasting”. In: *International Conference on Learning Representations (ICLR)*.

Wang, X., M. Feng, J. Qiu, J. Gu, and J. Zhao (2024). *From News to Forecast: Integrating Event Analysis in LLM-Based Time Series Forecasting with Reflection*. arXiv: 2409.17515 [cs.AI].

Winkler, R. L. (1972). “A Decision-Theoretic Approach to Interval Estimation”. In: *Journal of the American Statistical Association* 67.337, pp. 187–191.

Yao, S., D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan (2023). *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. arXiv: 2305.10601 [cs.CL].

Ye, J., Y. Wang, Y. Huang, D. Chen, Q. Zhang, N. Moniz, T. Gao, W. Geyer, C. Huang, P.-Y. Chen, N. V. Chawla, and X. Zhang (2024). *Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge*. arXiv: 2410.02736 [cs.CL].

Yeh, C.-C. M., V. Lai, U. S. Saini, X. Fan, Y. Fan, J. Wang, X. Dai, and Y. Zheng (2025). *Empowering Time Series Forecasting with LLM-Agents*. arXiv: 2508.04231 [cs.LG].

Zheng, L., W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica (2023). "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena". In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*. arXiv: 2306.05685 [cs.CL].

Zhou, Z. and R. Yu (2025). *Can LLMs Understand Time Series Anomalies?* arXiv: 2410.05440 [cs.LG].