FROM EXTERNAL INFORMATION TO FORECAST ADJUSTMENT: LLM-BASED CONSTRAINT MODELING AND REFLECTION ACCELERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In complex real-world systems, forecasts based solely on historical data often fail under non-stationary dynamics induced by external information, ranging from stochastic events to rule-based operational constraints. We propose a novel post-forecast adjustment framework that refines existing forecasts by converting unstructured external knowledge into explicit mathematical constraints using large language models (LLMs). Our approach utilizes an LLM-based agent to extract and structure external information, which is then translated into mathematical constraints by a fine-tuned LLM. Given a base forecast from any statistical or machine learning model, the adjusted forecast is derived by solving a convex optimization problem that projects the base prediction onto the feasible region defined by these constraints. We employ a reflection and memory mechanism for iterative refinement, and introduce a novel constraint selection strategy based on mutual information and intervention effect analysis to accelerate the reflection process. Empirical results show that our framework significantly improves forecasting accuracy over base methods, including those that integrate external information via LLMs, while offering superior interpretability compared to direct LLM-based approaches.

1 INTRODUCTION

Time series forecasting (Hamilton, 2020; Chatfield, 2000) plays a critical role in decision-making across economic, industrial, and social systems, supporting applications such as demand planning, resource allocation, and risk management (Hyndman & Athanasopoulos, 2018; Box et al., 2015; Arestis et al., 2010; Taylor & Buizza, 2006). A wide range of forecasting methods have been developed, including classical statistical models such as ARIMA (Shumway & Stoffer, 2017), deep learning models such as N-BEATS (Oreshkin et al., 2019) and PatchTST (Nie, 2022), and large-scale foundation models such as Chronos (Ansari et al., 2024). Despite their success, most existing approaches rely almost exclusively on historical observations and assume that future dynamics can be extrapolated from past patterns, an assumption often violated in open-world environments (Petropoulos et al., 2022).

In real-world scenarios, forecasts are frequently influenced by external information beyond numerical time series, including stochastic event-driven factors (e.g., policy changes and macroeconomic shocks) and deterministic operational knowledge (e.g., capacity limits and regulatory constraints). Such factors can induce regime shifts or impose feasibility boundaries, making them difficult to capture with purely data-driven models.

Recent studies have explored integrating external textual information into forecasting using large language models (LLMs). Methods such as Time-LLM (Jin et al., 2023) and subsequent approaches align time-series data with textual representations to extract complementary signals (Sun et al., 2023; Jin et al., 2023). More recently, (Wang et al., 2024) proposes an agent-based framework that incorporates news events via reflection mechanisms. However, these approaches typically fuse external information implicitly into predictive models, resulting in limited interpretability and an inability to enforce explicit operational constraints.

Forecasting practice has long relied on post-forecast human judgmental adjustments (Petropoulos et al., 2022). Although effective in practice, such adjustments are subjective, difficult to scale,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and prone to cognitive biases. Optimization-based forecasting methods reformulate specific models within optimization frameworks (Panagiotopoulos, 2012), but they do not systematically integrate heterogeneous external knowledge for post-hoc forecast refinement.

In this work, we propose a unified post-forecast adjustment framework that refines existing forecasts by decoupling forecasting from external knowledge integration, while maintaining interpretability and feasibility. Starting from any base forecast, external information is analyzed by an LLM-driven reasoning agent with reflection and memory mechanisms. The agent outputs structured representations, which are translated into solver-ready constraints by a fine-tuned LLM. Final forecasts are obtained by solving a convex optimization problem that minimally distorts the base prediction while enforcing these constraints. To handle conflicting constraints, we introduce slack variables and IIS-based diagnosis (Chinneck, 2008; Gurobi Optimization, LLC, 2024). We further propose a mutual information– and intervention-based constraint selection strategy to focus on high-impact external signals. Empirical results demonstrate consistent accuracy improvements over strong baselines, including LLM-based forecasting models, while providing stronger interpretability and feasibility guarantees. Moreover, the proposed constraint selection strategy substantially further accelerates the reflection process while preserving accuracy in most cases, with only occasional minor degradation.

Our contribution can be summarized as follows:

- We propose a novel post-forecast adjustment framework that refines existing forecasts by decoupling forecasting from external knowledge integration and incorporating heterogeneous external information via convex optimization.
- We introduce an LLM-driven multi-agent system with reflection and memory that transforms unstructured external information into structured solver-ready constraints with explicit interpretability.
- We propose CIMI, a mutual information and intervention effect–based strategy that ranks constraint types on validation data and restricts inference to the top-k types, enabling more efficient reflection. The ranking is held fixed between updates and can be refreshed cheaply when new outcomes arrive.

2 Related works

Time series forecasting. Time series forecasting has been studied across various paradigms, including statistical models, classical machine learning, and deep learning approaches. Early methods like ARIMA (Hyndman & Khandakar, 2008) and exponential smoothing (Kalekar et al., 2004) rely on linearity and stationarity assumptions, offering interpretability but limited performance on complex, non-stationary data. State-space models (Harvey, 1990; Zhang, 2004) and Gaussian mixture models (Lux, 2008) extend this by modeling latent dynamics, but they remain constrained in high-dimensional settings. With increasing data, machine learning techniques such as support vector regression (Sapankevych & Sankar, 2009; Pai et al., 2010), k-nearest neighbors (Martínez et al., 2019; Al-Qahtani & Crone, 2013), and tree-based models (Mayrink & Hippert, 2016; Januschowski et al., 2022) have been adopted to capture nonlinear relationships, though they struggle with long-range dependencies. Deep learning methods like RNNs (Salinas et al., 2020; Lai et al., 2018), CNNs (Luo & Wang, 2024; Liu et al., 2022; Yue et al., 2022), and Transformer-based models (Zhou et al., 2021b; Zhang & Yan, 2023; Nie, 2022) effectively address these limitations by learning representations directly from data. However, they still face challenges when incorporating external contextual information.

Large Language Models for time series forecasting. Recently, large language models (LLMs) have emerged as a promising approach to time series forecasting due to their strong representation learning capabilities (Jin et al., 2023). Early work adapts pretrained LLMs by transforming numerical sequences into tokenized or textual representations, enabling reuse of language modeling objectives for forecasting. Fine-tuning methods like AutoTimes (Liu et al., 2024b), FPT (Zhou et al., 2023), and TEMPO (Cao et al., 2023) integrate time series data with textual features. Prompt-based and zero-shot methods, such as LSTPrompt (Liu et al., 2024a) and TimeRAF (Zhang et al., 2025), explore using frozen LLMs with carefully designed prompts to perform forecasting without task-specific training. Generative agents, like those in (Wang et al., 2024), adaptively incorporate external social events into forecasting models. However, these methods face challenges such as poor

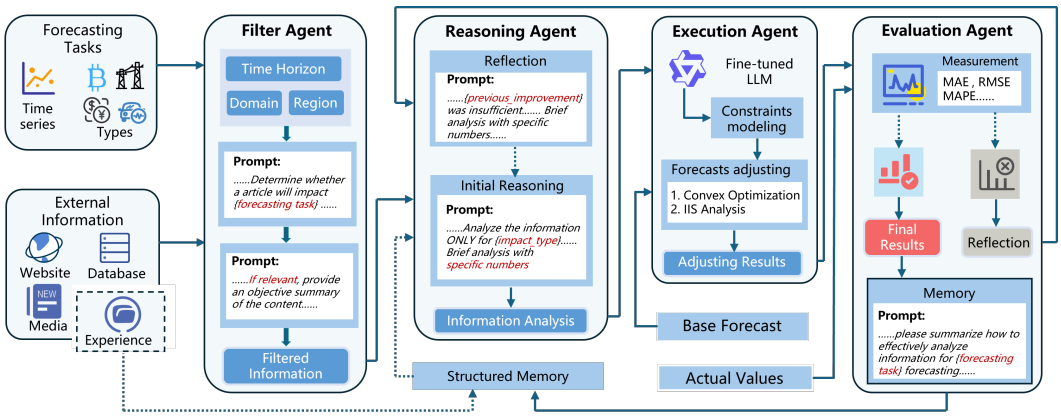


Figure 1: Given a specific prediction task, the Filter Agent retrieves and selects relevant information from external sources and produces a concise summary. The Reasoning Agent then analyzes the filtered information to identify potential factors that may influence the prediction outcomes. Based on this analysis, the Execution Agent transforms the reasoning results into structured constraints, which are used to optimize the Base Forecast. Subsequently, the Evaluation Agent assesses the optimized results. If the performance is improved, the extracted experience is stored in a memory module for future reasoning. Otherwise, the system performs reflection and re-enters the Reasoning Agent for further analysis.

interpretability, black-box forecasting, and inability to enforce explicit constraints on the resulting forecasts.

Adjustments of model-based forecasts. Judgmental adjustments to model-based forecasts, often used in practice to incorporate external information, can improve accuracy (Petropoulos et al., 2022; Fahimnia et al., 2020; Perera et al., 2019). However, they are prone to cognitive biases (Fildes et al., 2019; 2009) and limited scalability. Empirical studies show that manual interventions can introduce noise, overreact to recent events, and undermine the long-term patterns captured by statistical models (Taleb, 2010; Bolger & Harvey, 1993; Omar et al., 2023). These issues motivate interest in automated or learning-based adjustment mechanisms that systematically integrate external information, avoiding human biases and improving forecasting accuracy.

3 METHOD

In this paper, we propose a unified framework that incorporates external information into time-series forecasting by transforming it into explicit constraint conditions. While external knowledge has the potential to significantly enhance forecasting performance, effectively integrating such information poses several non-trivial challenges. First, external information is often large-scale, heterogeneous, and noisy, making its selection and preliminary analysis a non-trivial process that requires strong reasoning and filtering capabilities. Second, the extracted information is typically unstructured or semi-structured, and must be translated into structured constraints that can be directly utilized by optimization-based solvers. Third, since errors may arise during the information analysis and constraint construction stages, it is necessary to further evaluate the forecasting outcomes and reflect on potential inaccuracies, enabling iterative refinement of the overall process. To address these challenges, we design a modular framework consisting of three key components: (1) Intelligent agents for external information selection and preliminary analysis, (2) an LLM-based constraint modeling and optimization module that converts analyzed information into structured constraints and integrates them into the forecasting process, and (3) an evaluation and reflection agent that analyzes prediction results and provides feedback to improve upstream analysis and modeling. Figure 1 illustrates the interactions among these three modules. In the following sections, we describe each component in detail.

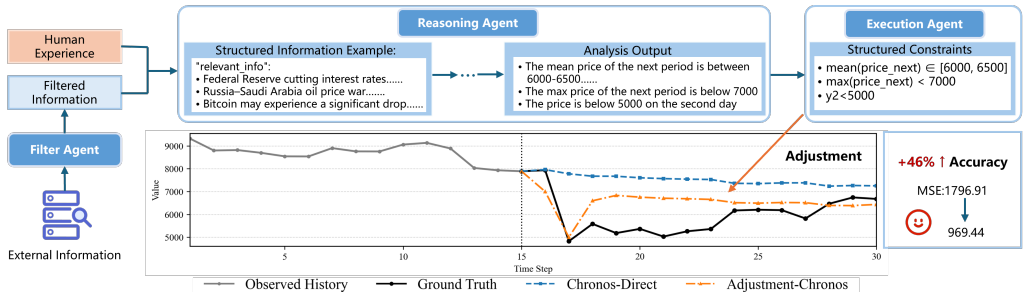


Figure 2: A concrete example of Bitcoin price forecasting optimization.

3.1 LLM-BASED FILTERING AND REASONING AGENT

External Information Filter and Analysis. The first component of our framework is LLM-based agents designed for the initial screening and analysis of external information. These two agents aim to identify, interpret, and structure potentially relevant external signals that may influence time-series dynamics. To enhance the agents' reasoning capability and robustness, we adopt a combination of Chain-of-Thought (Wei et al., 2022) reasoning and few-shot prompting (Mann et al., 2020), enabling the model to perform multi-step inference over large-scale and heterogeneous external data sources. The prompting strategy is organized into three sequential stages. In the first stage, the Filtering agent is instructed to selectively filter external information that is likely to have a meaningful impact on the forecasting target. This includes, but is not limited to, macroeconomic indicators, policy changes, regulatory announcements, and unexpected events such as natural disasters or market shocks. The goal of this stage is to reduce noise and retain only causally or temporally relevant information. In the second stage, the reasoning agent conducts a preliminary analysis of the selected information. Specifically, it performs qualitative-to-quantitative reasoning to produce coarse numerical estimations, such as the direction, magnitude, or duration of potential impacts. These estimations serve as intermediate signals for downstream constraint modeling. In the final stage, we enforce a fixed and structured output format. The reasoning agent organizes the analyzed information into a standardized JSON schema, explicitly recording information summaries, effective time spans, and preliminary quantitative assessments. This structured representation facilitates seamless integration with subsequent constraint construction and optimization modules.

Constraint Importance via Mutual Information (CIMI). Not all constraints contribute equally—some are critical while others may be redundant. We propose CIMI, a principled method to quantify constraint importance on validation data and rank constraint types before inference. The ranking is held fixed *between updates* rather than recomputed per instance: at inference time, the reasoning agent is restricted to the preselected top-k types. Because the catalog contains only eight constraint types, a CIMI update is a lightweight evaluation on validation outcomes and does not require retraining the base forecaster or fine-tuning the LLM. In deployment, the ranking can therefore be refreshed on a rolling window of newly observed outcomes, or after a detected distributional shift, at negligible cost relative to training.

The importance score for constraint type c is:

$$\text{Importance}(c) = \alpha\, \widetilde{\text{Interv}}(c) \cdot \text{Novelty}(c) + \beta\, \widetilde{\text{DMI}}(c),$$

where $\widetilde{\cdot}$ denotes normalization to $[0, 1]$ (Appendix A.7.1), and $\alpha, \beta > 0$ are balancing weights (default 0.5). Novelty multiplies only the intervention term since DMI inherently vanishes for naturally-satisfied constraints.

The intervention effect is estimated via controlled ablation: we run the full optimization pipeline on validation data with c activated and with c removed, then define:

$$\text{Intervention}(c) = \mathbb{E}_{\text{val}}[\mathcal{L} \mid \text{do}(\neg c)] - \mathbb{E}_{\text{val}}[\mathcal{L} \mid \text{do}(c)],$$

where $\mathcal{L}$ is the validation error and $\text{do}(\cdot)$ denotes constraint inclusion/exclusion within the fixed pipeline (Pearl, 1995). A positive value indicates c reduces error. The DMI term (Cover, 1999) measures distributional shift:

$$\text{DMI}(c) = \tfrac{1}{2}\, \Delta_c^2 \,/\, \text{Var}_{\text{boot}}(g_c(Y_{\text{hist}})), \quad \Delta_c = g_c(Y_{\text{true}}) - g_c(Y_{\text{hist}}),$$

Algorithm 1 CIMI-based Constraint Selection

Require: Validation observations Y_{hist}, ground-truth Y_{true}; candidate types $\mathcal{C} = \{c_1, \ldots, c_K\}$; constraint statistics $\{g_c\}$; pipeline $\mathcal{P}$; budget B (or threshold τ); hyperparameters α, β, λ
Ensure: Selected subset $\mathcal{C}^\star$ (held fixed between ranking updates)
for each $c \in \mathcal{C}$ **do**
  $\Delta_c \leftarrow g_c(Y_{\text{true}}) - g_c(Y_{\text{hist}})$ {Constraint deviation}
  $\mathrm{DMI}(c) \leftarrow \frac{1}{2} \cdot \Delta_c^2 \,/\, \mathrm{Var}_{\text{boot}}(g_c(Y_{\text{hist}}))$ {Raw information value}
  $\widetilde{\mathrm{DMI}}(c) \leftarrow 1 - \exp(-\lambda \cdot \mathrm{DMI}(c))$ {Normalize to $[0, 1]$}
  Run $\mathcal{P}$ with c active $\rightarrow \mathcal{L}^+$; with c removed $\rightarrow \mathcal{L}^-$ {Controlled ablation}
  $\mathrm{Intervention}(c) \leftarrow \mathcal{L}^- - \mathcal{L}^+$ {Raw empirical value}
end for
$\widetilde{\mathrm{Intervention}}(c) \leftarrow \mathrm{MinMaxNorm}(\{\mathrm{Intervention}(c)\}_{c \in \mathcal{C}})$ {Normalize to $[0, 1]$}
for each $c \in \mathcal{C}$ **do**
  Compute $\mathrm{Novelty}(c)$ penalizing natural satisfaction and inter-constraint correlation
  $\mathrm{Importance}(c) \leftarrow \alpha\, \widetilde{\mathrm{Intervention}}(c) \times \mathrm{Novelty}(c) + \beta\, \widetilde{\mathrm{DMI}}(c)$
end for
Rank $\mathcal{C}$ by decreasing $\mathrm{Importance}(c)$
$\mathcal{C}^\star \leftarrow$ top-K types (or $\{c : \mathrm{Importance}(c) \geq \tau\}$) {Budget or threshold}
return $\mathcal{C}^\star$ {Held fixed between ranking updates}

where $g_c : \mathbb{R}^T \to \mathbb{R}$ is the constraint-specific statistic (e.g., $g_{\text{mean}}(Y) = \frac{1}{T}\sum_t y_t$; full catalog in Appendix A.7.2), variance is bootstrap-estimated, and $\widetilde{\mathrm{DMI}}(c) = 1 - \exp(-\lambda\, \mathrm{DMI}(c))$.The novelty term downweights constraints that are naturally satisfied or explained by correlated constraints. The detailed computation of the $\mathrm{Novelty}(c)$ score is provided in Appendix A.7.3.

3.2 External Constraints and Their Integration

The reasoning outputs are converted into structured, solver-ready constraints by a fine-tuned LLM, so that the subsequent optimization can refine the base forecast while respecting domain-specific rules. Industrial sources of such constraints (event-driven, operational, temporal, regulatory, and risk/probabilistic) are described in Appendix A.4. From a modeling perspective, we classify constraints by temporal scope and mathematical structure, which determines how they enter the optimization:

- Single-point adjustment constraints, which apply to individual time steps, e.g., $a \leq y_t \leq b$;
- Multi-point joint adjustment constraints, which involve multiple time steps in a single rule, such as sum constraints $a \leq y_1 + y_2 + \cdots + y_{12} \leq b$, extremum constraints $\max(y_1, y_2, \cdots, y_H) \leq b$, and monotonicity constraints $y_1 \leq y_2 \leq \cdots \leq y_H$;
- Probabilistic (chance) constraints, which constitute a special class of multi-point joint constraints and enforce chance-based conditions. To ensure tractability, we approximate them via discrete linear inequalities by requiring that a corresponding fraction of time points satisfy the specified threshold.

The full catalog of eight constraint types is given in Appendix Table 5.

3.3 Constraint modeling and Optimization

Fine-tuning LLMs for Constraint modeling. General-purpose LLMs often produce unstable or incomplete constraint formulations. Since the goal is to learn constraint-construction patterns rather than factual knowledge, we synthesize training triples of historical series, pseudo-news, and solver-ready constraints with DeepSeek (DeepSeek-AI et al., 2025), and fine-tune the base LLM with LoRA (Hu et al., 2022). Training hyperparameters, target modules, and an example prompt–output pair are given in Appendix A.1.

Adjustment via a Two-Stage Iterative Optimization Framework. After fine-tuning, the LLM generates structured constraints that encode temporal, logical, and domain-specific requirements. These constraints are integrated into a mathematical optimization framework to refine base predictions. To handle potential inconsistencies in LLM-generated constraints due to uncertainty or

conflicting information, we propose a two-stage iterative optimization framework that combines a robust, always-feasible optimizer with conflict diagnosis and iterative repair. A example is shown in Figure 2.

Stage 1: Robust Optimization with Adaptive Soft Constraints. We begin by formulating the adjustment task as a unified convex optimization problem, incorporating slack variables s_k for each constraint to handle conflicts. These slack variables act as soft constraints, allowing for violations while ensuring the feasibility of the optimization. The objective is to minimize the distortion to the base forecast while enforcing the constraints, with each slack variable s_k quantifying the violation degree. The optimization problem is formulated as follows:

$$\min \quad \underbrace{\sum_{t=1}^{H} w_t \left(y_t - \hat{y}_t\right)^2}_{\text{minimal distortion to the base forecast}} \quad + \quad \underbrace{\tau \sum_{k=1}^{K} \alpha_k \left\|s_k\right\|_2^2}_{\text{cost of violating text constraints}} \qquad \text{s.t.} \quad \begin{cases} A_k \mathbf{y} \leq b_k + s_k, \\ \mathbf{y} \geq 0,\ \mathbf{s}_k \geq 0, \\ k = 1, \ldots, K. \end{cases}$$

Here, $\hat{y}_t$ denotes the base model prediction at time t, and y_t is the decision variable representing the adjusted forecast. The horizon length H denotes the number of future time steps produced by the base model and subject to adjustment, and K is the number of extracted LLM-based linear constraints. The term $w_t > 0$ is an optional uncertainty weight (e.g., setting $w_t = \frac{1}{H}$), α_k encodes the confidence of constraint k derived from text sources, and $\tau > 0$ is a global constraint temperature parameter. This formulation is a quadratic program (QP) (or a linear program (LP) if the ℓ_1-norm is used for both prediction errors and slack penalties). It is always feasible due to the slack variables, and as $\tau\alpha_k \to \infty$, a constraint behaves as a hard rule.

Stage 2: Conflict Diagnosis and Iterative Repair. If Stage 1 shows large constraint violations (i.e., large s_k exceeding a predefined tolerance), this indicates inconsistency in the constraints. To resolve this, we use the solver's Irreducible Infeasible Subsystem (IIS) analysis (Chinneck, 2008; Gurobi Optimization, LLC, 2024), identifying a minimal subset of incompatible constraints. This IIS report is passed to the reasoning agent to analyze the conflict and suggest repairs—such as adjusting, reweighting α_k, or removing constraints. The repaired set is then reintroduced into Stage 1, and the optimization is solved again. This process continues until a satisfactory solution is found, with key constraints satisfied or violations reduced to an acceptable level.

This two-stage design offers key benefits: the unified optimization ensures reliability by always producing an adjusted forecast, while the IIS-based feedback loop enables explainable conflict resolution and continuous improvement of the constraint set.

Optionally, a linear autoregressive shape prior can be added to the Stage 1 objective to encourage smoother, ARIMA-like trajectories while preserving convexity; the full specification is given in Appendix A.8.

3.4 Evaluation Agent

To enhance the reliability of external information analysis, we introduce an Evaluation Agent that systematically assesses optimization outcomes and iteratively refines the reasoning process. In real-world scenarios, external information—such as news articles or analyst reports—affects predictions implicitly and qualitatively. The Reasoning Agent must therefore rely on heuristic mappings to convert linguistic descriptions into numerical signals (e.g., interpreting "slight increase" as 3% gain). Such heuristic approximations are often subjective and may introduce bias or systematic errors. The Evaluation Agent addresses this limitation through outcome-driven feedback. It monitors final optimization results and evaluates whether numerical estimations derived from external information lead to consistent, stable, near-optimal performance. By comparing expected outcomes with observed results, it identifies discrepancies caused by misinterpretations, inappropriate mappings, or coarse assumptions. Based on this feedback, the Evaluation Agent adaptively adjusts quantitative parameters representing external information, establishing a closed-loop refinement process where reasoning and optimization continuously align through empirical performance signals. This progressively reduces reliance on static heuristics and improves the system's ability to incorporate external information in a calibrated, data-consistent manner. Additionally, we integrate an external memory module supporting long-term reasoning consistency and knowledge reuse. When optimization outcomes satisfy predefined criteria, the Evaluation Agent stores corresponding reasoning logic

Domain	Metrics	Nbeats		PatchTST		DLinear		LightGBM		Chronos		iTransformer		LoRA		TimeLLM	
		base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C
Electricity	MAE	236.03	**229.34±4.32** (+2.83%)	247.05	**231.87±5.10** (+6.14%)	284.74	**269.52±6.18** (+5.35%)	234.85	**223.41±4.05** (+4.87%)	514.83	**458.13±11.24** (+11.01%)	259.27	**251.06±5.42** (+3.17%)	440.59	**407.25±9.37** (+7.57%)	331.97	**317.89±7.16** (+4.24%)
	RMSE	374.55	**360.42±7.85** (+3.77%)	397.11	**370.68±8.13** (+6.66%)	468.01	**442.15±9.92** (+5.53%)	383.45	**359.17±7.64** (+6.33%)	765.61	**676.29±15.83** (+11.67%)	428.85	**415.34±9.05** (+3.15%)	695.73	**650.13±14.27** (+6.55%)	505.66	**487.26±10.68** (+3.64%)
	MAPE	6.92	6.58±0.18 (+4.91%)	6.98	6.45±0.17 (+7.59%)	7.54	7.15±0.21 (+5.17%)	6.19	5.85±0.15 (+5.49%)	12.43	10.94±0.34 (+11.99%)	6.96	6.69±0.19 (+3.88%)	10.38	9.71±0.29 (+6.45%)	8.44	8.11±0.24 (+3.91%)
Bitcoin	MAE	965.11	**821.36±21.43** (+14.89%)	932.81	**816.42±20.87** (+12.48%)	1163.54	**991.28±26.15** (+14.80%)	734.86	**625.73±16.32** (+14.85%)	1421.55	**1038.51±28.74** (+26.95%)	1204.77	**982.67±25.61** (+18.44%)	1398.45	**1131.05±31.08** (+19.12%)	1299.68	**1073.94±29.45** (+17.37%)
	RMSE	1921.42	**1627.85±42.16** (+15.28%)	1495.61	**1368.29±35.28** (+8.51%)	2291.27	**1942.16±50.73** (+15.24%)	1156.65	**1011.34±26.54** (+12.56%)	2714.50	**1975.62±52.18** (+27.22%)	2309.84	**1980.43±51.62** (+14.26%)	2387.36	**1927.38±49.87** (+19.27%)	2467.26	**2068.51±54.31** (+16.16%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Traffic	MAE	16.92	**16.58±0.41** (+2.01%)	17.79	**17.21±0.44** (+3.26%)	25.02	**24.51±0.62** (+2.04%)	14.5	**14.31±0.35** (+1.31%)	20.6	**19.74±0.50** (+4.17%)	19.49	**19.12±0.48** (+1.90%)	20.14	**19.56±0.49** (+2.88%)	26.9	**26.63±0.68** (+1.00%)
	RMSE	39.04	**38.35±0.94** (+1.77%)	38.36	**37.46±0.92** (+2.35%)	51.83	**50.94±1.27** (+1.72%)	34.87	**34.49±0.85** (+1.09%)	47.47	**46.28±1.14** (+2.51%)	46.31	**45.07±1.11** (+2.68%)	50.06	**48.67±1.21** (+2.78%)	55.17	**54.62±1.37** (+1.00%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Exchange	MAE	5.50	**5.16±0.14** (+6.18%)	5.49	**5.20±0.15** (+5.28%)	5.77	**5.31±0.15** (+7.97%)	5.28	**5.02±0.13** (+4.92%)	5.35	**5.24±0.14** (+2.06%)	6.53	**5.96±0.17** (+8.73%)	6.01	**5.45±0.15** (+9.32%)	5.47	**5.13±0.14** (+6.22%)
	RMSE	7.12	**6.81±0.19** (+4.35%)	7.16	**6.90±0.20** (+3.63%)	7.47	**7.05±0.20** (+5.62%)	7.09	**6.88±0.19** (+2.96%)	7.14	**7.06±0.20** (+1.12%)	8.56	**7.91±0.23** (+7.59%)	7.89	**7.31±0.21** (+7.35%)	7.23	**6.85±0.19** (+5.26%)
	MAPE	0.77	0.72±0.02 (+6.49%)	0.76	0.74±0.02 (+2.63%)	0.80	0.73±0.02 (+8.75%)	0.74	0.70±0.02 (+5.41%)	0.74	0.73±0.02 (+1.35%)	0.91	0.83±0.03 (+8.79%)	0.83	0.76±0.02 (+8.43%)	0.76	0.72±0.02 (+5.26%)

Table 1: Performance comparison across different domains and models. CIMI-C reports mean ± std over 5 random seeds; base forecasts are deterministic, so the observed variance comes from LLM constraint extraction. The numbers in parentheses represent the percentage improvement of the CIMI-C mean over the base model. The dash (—) indicates that the value is not applicable due to inherent mathematical deficiencies of MAPE.

for future retrieval, accelerating convergence and improving stability across similar tasks. Domain experts can preload empirical knowledge or validated rules into this module, directly guiding the Reasoning Agent and constraining unreasonable assumptions. This human-in-the-loop design enhances robustness while improving interpretability and controllability of the overall system.

4 EXPERIMENTS

4.1 DATA PREPARATION

To evaluate our method on time series forecasting with external information, we prioritize datasets where exogenous factors—such as human activities and social events—significantly influence target variables, unlike physics-driven benchmarks like ETT (Zhou et al., 2021a) or Weather (Reynolds et al., 2007; Hersbach et al., 2020). We select four representative domains: electricity demand (Godahewa et al., 2021), Bitcoin price (Godahewa et al., 2021), traffic flow (Maggie et al., 2017), and foreign exchange rates (Lai et al., 2018). Electricity, traffic, and exchange rate data are sourced from (Wang et al., 2024), while Bitcoin data is collected from investing.com (Investing.com, 2026). These datasets covering 30-minute, hourly, and daily frequencies to test robustness across granularities. We use real-world news datasets from four domains (Wang et al., 2024). In addition, to enrich the experiments, we generate synthetic news using DeepSeek-V3.2 (DeepSeek-AI et al., 2025).

4.2 BASE FORECASTER

To demonstrate the effectiveness of our proposed optimization method, we conduct extensive experiments on a diverse set of mainstream time-series forecasting models. Specifically, we consider two categories of Base Forecasters: (1) history-only forecasting, including N-BEATS (Oreshkin et al., 2019), PatchTST (Nie, 2022), DLinear (Zeng et al., 2023), LightGBM (Ke et al., 2017), Chronos (Ansari et al., 2024), and iTransformer (Liu et al., 2023), which use only historical time-series inputs and remain unchanged in our framework as base forecasters, while our method is applied afterward as a post-forecast adjustment layer; and (2) a direct-prediction based on LoRA (Hu et al., 2022), which uses the same historical sequence and the same external textual information as our method, but predicts future values directly without constraint extraction or optimization, thereby isolating the effect of our constraint-based integration mechanism from the mere availability of external information. The complete list of baseline models and implementation details is provided in the Appendix A.1. Our method is not a new end-to-end predictor that replaces base models; instead, it functions as a plug-in layer that operates on top of any base forecaster.

4.3 RESULTS

Optimization to different forecasting methods. For each model, we evaluate its performance on the test set before and after applying our optimization framework. We report three widely used

evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Because the convex projection step is deterministic, we rerun the LLM-based constraint extraction over 5 random seeds and report CIMI-C as mean $\pm$ standard deviation; base forecasters remain frozen and therefore have no seed variance. As shown in Table 1, applying our optimization framework with constraints selected by the proposed CIMI strategy consistently and substantially improves forecasting performance across multiple domains, including electricity consumption, Bitcoin price prediction, and traffic flow forecasting. The standard deviations are small relative to the gains (typically within 1–3% of the mean), so the improvements are not artifacts of LLM sampling. Compared with the original base models, CIMI-guided optimization yields significant error reductions on nearly all metrics, demonstrating the strong effectiveness of selectively incorporating informative constraints into the forecast process. MAPE suffers from inherent mathematical deficiencies due to its normalization by the ground-truth value. Specifically, when the true value approaches zero, the denominator causes the error to become unbounded, leading to numerical instability and excessive amplification of small absolute errors. As a result, MAPE may provide misleading or uninformative evaluations and fail to faithfully reflect predictive performance. In the Bitcoin price and traffic flow datasets considered in this work, the observations contain values that are close to zero. Therefore, analyzing MAPE on these datasets is not meaningful. For this reason, we only report MAE and RMSE for Bitcoin price and traffic flow forecasting. Among all real-world datasets, traffic flow forecasting exhibits the smallest improvement from our optimization framework. A possible explanation is that traffic flow data is inherently governed by complex spatiotemporal dynamics, including road network topology, upstream-downstream propagation effects, and highly nonlinear congestion phenomena. These characteristics make it difficult for auxiliary textual constraints to capture the underlying mechanisms driving traffic patterns, thereby limiting the additional information gain from constraint-based optimization. Experiments on a synthetic news dataset also achieve strong optimization results, and detailed results are provided in Appendix Table 10. Notably, the performance gains on the synthetic dataset are more pronounced than those observed on most real-world datasets. A plausible explanation is that the synthetic news data is constructed with explicitly embedded information that directly governs the generation of the time series, making the underlying patterns more transparent and the relevant constraints easier to identify and exploit. In contrast, real-world datasets often contain implicit, noisy, or entangled information that is harder to disentangle and leverage through textual constraints.

Acceleration by CIMI. In addition to optimization gains, our proposed CIMI method significantly accelerates the agent's reasoning process when analyzing external information. To ensure a fair comparison, we measured the total reasoning time for analyzing 100 samples from the test set using the same agent. As illustrated in Table 11, CIMI achieves substantial efficiency improvements by prioritizing the top three or four most important constraint types, rather than exhaustively evaluating all eight (treating minimum and maximum as distinct). This selective approach drastically reduces the computational overhead associated with checking every potential constraint. All measurements were conducted using a locally deployed Qwen2.5-7B-Instruct (Qwen et al., 2025) to strictly isolate intrinsic reasoning speed from uncontrollable factors like API network latency. Consequently, the observed speedups are directly attributable to CIMI. Using the CIMI-selected subset yields accuracy comparable to the full catalog (Appendix Tables 8 and 9), indicating that moderate errors in the offline ranking do not materially degrade forecasts.

Ablation Experiment. We perform component-wise ablations on Bitcoin by separately removing the reflection loop, the memory module, and the Filter Agent. As shown in Table 2, removing any single component consistently degrades performance relative to the full framework, confirming that each module contributes positively. The effect is most pronounced when the Filter Agent is removed, where the error rises above even the base forecast across all base models. For example, on N-BEATS the MAE increases from 892.53 (no memory) and 913.46 (no reflection) to 1115.67 (no Filter), exceeding the base forecast of 965.11, with RMSE similarly jumping to 2287.94 versus the base 1921.42. This is because real-world textual inputs are heterogeneous and often noisy or contradictory; without the Filter Agent to screen them, low-quality constraints distort the projection step and make the adjustment actively harmful rather than corrective. Removing the reflection loop also raises the error (e.g., N-BEATS RMSE from 1682.35 without memory to 1743.68 without reflection), reflecting the loss of iterative refinement that corrects suboptimal constraints before projection. Removing the memory module likewise degrades performance by preventing reuse of successful adjustment experience from past instances. These trends hold across models (e.g., PatchTST no-Filter MAE 1231.31 vs. base 932.81). An overall ablation that removes all external information, as well

Model	MAE				RMSE			
	Base	No refl.	No mem.	No Filter	Base	No refl.	No mem.	No Filter
Nbeats	965.11	913.46	892.53	1115.67	1921.42	1743.68	1682.35	2287.94
PatchTST	932.81	887.31	865.58	1231.31	1495.61	1558.27	1471.53	2143.76
DLinear	1163.54	1102.38	1083.18	1238.24	2291.27	2098.44	2211.89	2512.63
LightGBM	734.86	707.03	688.87	1012.64	1156.65	1094.82	1023.57	1678.31
Chronos	1421.55	1308.27	1222.51	1436.90	2714.50	2387.16	2245.93	2891.47
iTransformer	1204.77	1143.97	1067.62	1276.82	2309.84	2278.35	2087.62	2634.18
TimeLLM	1299.68	1220.24	1178.82	1768.86	2467.26	2251.73	2378.49	3492.85

Table 2: Component-wise ablations on Bitcoin. Removing any module degrades performance relative to the full framework. Removing the Filter Agent is harmful even relative to the unadjusted base forecast.

as a comparison between CIMI-selected constraints and the full constraint set, is reported in the appendix (Table 7, Table 8, and Table 9).

Human evaluation of extracted constraints. Forecast accuracy is only an indirect signal of whether the generated constraints are faithful to the news. To evaluate extraction quality directly, we sampled 10 forecasting tasks covering 117 news articles, from which domain experts manually annotated, deduplicated, and audited a ground-truth set of 36 golden constraints. Across these tasks the framework generated 64 constraints, of which 21 matched the expert set, corresponding to a precision of 32.8% $(21/64)$ and a recall of 58.3% $(21/36)$. Manual inspection of mismatches yields three dominant error modes (Appendix A.3): over-extraction or granular redundancy ($\sim 39\%$ of generated constraints), where the LLM emits fine-grained or duplicated rules that experts omit for conciseness; magnitude estimation noise ($\sim 16\%$), from inexact qualitative-to-quantitative mapping; and temporal horizon mismatch ($\sim 9\%$). These modes are expected in news-to-constraint translation, and the architecture is designed to absorb them: Stage 1 treats constraints as soft inequalities with slack variables s_k, so small bound or horizon inaccuracies need not make the program infeasible, while Stage 2 uses IIS diagnosis and the Reflection Agent to detect, repair, or discard severely conflicting constraints before the final projection. This is consistent with the Filter Agent ablation above: unfiltered noisy constraints can make the adjustment harmful, whereas the full pipeline remains beneficial despite imperfect extraction.

5 Conclusion

We present a post-forecast adjustment framework that integrates data-driven forecasting and external knowledge through interpretable, constraint-based optimization. By decoupling base forecasting from external information, our approach accommodates various forecasting models while ensuring feasibility and transparency. Empirical results show consistent improvements in accuracy over base models, offering stronger interpretability and feasibility. This method enables more effective integration of external information, making it highly suitable for complex real-world predictions.

Limitations and Future work. While our method demonstrates effectiveness across multiple domains, it relies on the availability of meaningful external information structured into constraints, which limits its utility in fields with minimal human knowledge, such as meteorology or fundamental physics. Furthermore, although our reflection mechanism enhances robustness, it cannot fully eliminate errors arising from noisy or incomplete external data, and the overall performance remains bounded by the quality of the base predictor. Our experiments evaluate a single CIMI ranking computed on a fixed validation window; we do not empirically study rolling re-ranking under annotated regime shifts. Future work will address these challenges by optimizing the selection and analysis of external information to improve efficiency and minimize the influence of noisy signals. We also plan to expand the framework's applicability to more complex tasks, such as multivariate prediction, to handle dynamic and heterogeneous real-world data, while simultaneously focusing on improving the robustness of external information analysis for more stable forecasting outcomes.

AI USE STATEMENT

Large language models are a core component of the proposed method. We use LLM-based agents to filter and reason over external information, a fine-tuned LLM to translate unstructured text into solver-ready constraints, and an evaluation agent with reflection and memory to refine the constraint set. We generated synthetic news and constraint-modeling training data with DeepSeek-V3.2, and performed supervised LoRA fine-tuning of Qwen2.5-7B-Instruct for constraint modeling and for a direct-prediction baseline. These uses include generating synthetic datasets, implementing the proposed method, and supporting experimental analysis, and are described in Sections 3 and 4 and in the appendix. Additionally, we used generative AI tools for editing the manuscript to improve readability. We have not used generative AI tools to formulate mathematical claims or to write proofs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, crowdsourcing, or the collection of personally identifiable information. Experiments use publicly available time-series and news sources, together with synthetic news generated for controlled evaluation. The proposed method queries LLMs and solves convex programs over existing forecasts; it does not introduce a direct path to harmful applications beyond the general risks of automated forecasting in high-stakes settings, where domain experts should retain oversight. The research adheres to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the main results in the paper and appendix. Datasets, base forecasters, and evaluation metrics are described in Section 4. Architectural choices, LoRA fine-tuning settings, optimization hyperparameters, constraint types, CIMI statistics, additional tables, and full prompts are specified in the appendix. Upon acceptance, we will release source code, data, configuration files, and the scripts used to produce the reported tables.

REFERENCES

Fahad H Al-Qahtani and Sven F Crone. Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting uk electricity demand. In *The 2013 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2013.

Sassan Alizadeh, Michael W Brandt, and Francis X Diebold. Range-based estimation of stochastic volatility models. *The Journal of Finance*, 57(3):1047–1091, 2002.

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

Philip Arestis, Ambika D Luintel, and Kul B Luintel. Financial structure and economic growth: evidence from time series analyses. *Applied financial economics*, 20(19):1479–1492, 2010.

Fergus Bolger and Nigel Harvey. Context-sensitive heuristics in statistical reasoning. *The Quarterly Journal of Experimental Psychology Section A*, 46(4):779–811, 1993.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

Chris Chatfield. *Time-series forecasting*. Chapman and Hall/CRC, 2000.

John W Chinneck. *Feasibility and infeasibility in optimization: algorithms and computational methods*. Springer, 2008.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Behnam Fahimnia, Nada Sanders, and Enno Siemsen. Human judgment in supply chain forecasting. *Omega*, 94:102249, 2020. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2020.102249. URL https://www.sciencedirect.com/science/article/pii/S0305048320302954.

Robert Fildes, Paul Goodwin, Michael Lawrence, and Konstantinos Nikolopoulos. Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. *International Journal of Forecasting*, 25(1):3–23, 2009. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2008.11.010. URL https://www.sciencedirect.com/science/article/pii/S0169207008001362.

Robert Fildes, Paul Goodwin, and Dilek Önkal. Use and misuse of information in supply chain forecasting of promotion effects. *International Journal of Forecasting*, 35(1):144–156, 2019. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2017.12.006. URL https://www.sciencedirect.com/science/article/pii/S0169207018300062. Special Section: Supply Chain Forecasting.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

Gurobi Optimization, LLC. How do I use 'compute IIS', 2024. URL https://support.gurobi.com/hc/en-us/articles/15656630439441.

James D Hamilton. *Time series analysis*. Princeton university press, 2020.

Andrew C Harvey. Forecasting, structural time series models and the kalman filter. 1990.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Tao Huang, Robert Fildes, and Didier Soopramanien. Forecasting retailer product sales in the presence of structural change. *European Journal of Operational Research*, 279(2):459–470, 2019.

John Hull. *Options, futures, and other derivative securities*, volume 7. prentice hall Englewood Cliffs, NJ, 1993.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

Investing.com. Bitcoin historical data. https://ca.investing.com/crypto/bitcoin/historical-data, 2026. Accessed: 2026-01-22.

Tim Januschowski, Yuyang Wang, Kari Torkkola, Timo Erkkilä, Hilaf Hasson, and Jan Gasthaus. Forecasting with trees. *International Journal of Forecasting*, 38(4):1473–1481, 2022.

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

Prajakta S Kalekar et al. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi school of information Technology*, 4329008(13):1–13, 2004.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Cheng F Lee, Alice C Lee, and John C Lee. *Handbook of quantitative finance and risk management*, volume 1. Springer, 2010.

Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B Aditya Prakash. Lst-prompt: Large language models as zero-shot time series forecasters by long-short-term prompting. *arXiv preprint arXiv:2402.16132*, 2024a.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

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

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024b.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*, pp. 1–43, 2024.

Thomas Lux. The markov-switching multifractal model of asset returns: Gmm estimation and linear forecasting of volatility. *Journal of business & economic statistics*, 26(2):194–210, 2008.

Oren Anava Maggie, Vitaly Kuznetsov, and Will Cukierski. Web traffic time series forecasting. https://kaggle.com/competitions/web-traffic-time-series-forecasting, 2017. Kaggle competition.

Ben Mann, Nick Ryder, Melanie Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1(3):3, 2020.

Francisco Martínez, María Pilar Frías, María Dolores Pérez, and Antonio Jesús Rivera. A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, 52(3):2019–2037, 2019.

Victor Mayrink and Henrique S Hippert. A hybrid method using exponential smoothing and gradient boosting for electrical short-term load forecasting. In *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–6. IEEE, 2016.

John Mello. The impact of sales forecast game playing on supply chains. *Foresight: The International Journal of Applied Forecasting*, None(13):13–22, Spring 2009. doi: None. URL https://ideas.repec.org/a/for/ijafaa/y2009i13p13-22.html.

John Mello. Corporate culture and s&op: Why culture counts. *Foresight: The International Journal of Applied Forecasting*, (16), 2010.

Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Haytham Omar, Walid Klibi, M Zied Babai, and Yves Ducq. Basket data-driven approach for omnichannel demand forecasting. *International Journal of Production Economics*, 257:108748, 2023.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Ping-Feng Pai, Kuo-Ping Lin, Chi-Shen Lin, and Ping-Teng Chang. Time series forecasting by a seasonal support vector regression model. *Expert Systems with Applications*, 37(6):4261–4265, 2010.

Apostolos Panagiotopoulos. Optimising time series forecasts through linear programming. Technical report, University of Nottingham, 2012.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

H. Niles Perera, Jason Hurley, Behnam Fahimnia, and Mohsen Reisi. The human factor in supply chain forecasting: A systematic review. *European Journal of Operational Research*, 274(2):574–600, 2019. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2018.10.028. URL https://www.sciencedirect.com/science/article/pii/S0377221718308816.

Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J Bessa, Jakub Bijak, John E Boylan, et al. Forecasting: theory and practice. *International Journal of forecasting*, 38(3):705–871, 2022.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

R. W. Reynolds, T. M. Smith, C. Liu, D. B. Chelton, K. S. Casey, and M. G. Schlax. Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20:5473–5496, 2007. doi: 10.1175/JCLI-D-14-00293.1.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE computational intelligence magazine*, 4(2):24–38, 2009.

Robert H Shumway and David S Stoffer. Arima models. In *Time series analysis and its applications: with R examples*, pp. 75–163. Springer, 2017.

Leo WG Strijbosch and JJA Moors. The impact of unknown demand parameters on (r, s)-inventory control performance. *European Journal of Operational Research*, 162(3):805–815, 2005.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*, 2023.

Nassim Nicholas Taleb. *The Black Swan:: The Impact of the Highly Improbable: With a new section:" On Robustness and Fragility"*, volume 2. Random house trade paperbacks, 2010.

James W Taylor. Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles. *International Journal of Forecasting*, 26(4):627–646, 2010.

James W Taylor and Roberto Buizza. Density forecasting for weather derivative pricing. *International Journal of Forecasting*, 22(1):29–42, 2006.

Peter Wallström and Anders Segerstedt. Evaluation of forecasting error measurements and techniques for intermittent demand. *International Journal of Production Economics*, 128(2):625–636, 2010.

Pu Wang, Bidong Liu, and Tao Hong. Electric load forecasting with recency effect: A big data approach. *International Journal of Forecasting*, 32(3):585–597, 2016.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Ward Whitt and Xiaopei Zhang. Forecasting arrivals and occupancy levels in an emergency department. *Operations Research for Health Care*, 21:1–18, 2019.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

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

Huanyu Zhang, Chang Xu, Yi-Fan Zhang, Zhang Zhang, Liang Wang, and Jiang Bian. Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Yingjian Zhang. Prediction of financial time series with hidden markov models. 2004.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021a.

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

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Appendix

A.1 Experimental Setting / Details

Base Forecaster. To comprehensively evaluate the effectiveness of our proposed method, we compare it against a diverse set of representative baselines for time-series forecasting. These baselines cover multiple methodological paradigms, including classical statistics-based approaches, Transformer-based neural models, and recent large language model (LLM)-based methods, namely N-BEATS (Oreshkin et al., 2019), PatchTST (Nie, 2022), DLinear (Zeng et al., 2023), LightGBM (Ke et al., 2017), Chronos (Ansari et al., 2024), iTransformer (Liu et al., 2023), Time-LLM (Jin et al., 2023) and fine-tuning LLM by Lora (Wang et al., 2024). For Chronos (Ansari et al., 2024), we use "amazon/chronos-bolt-mini" model. For iTransformer (Liu et al., 2023), Time-LLM (Jin et al., 2023) and DLinear (Zeng et al., 2023), we use the code from the neuralforecast(https://github.com/Nixtla/neuralforecast). For each baseline, we strictly follow the official architectural designs and implementation details reported in the original papers or released codebases, and we carefully tune hyperparameters according to recommended settings to ensure their best possible performance.

LLM Fine-tuning. Constraint-modeling fine-tuning aims to teach the model how to emit solver-ready inequalities, not to memorize domain facts. We therefore synthesize training triples of historical time series, pseudo-generated news, and corresponding constraints with DeepSeek (DeepSeek-AI et al., 2025); an example prompt and target output are given in §A.6. For both direct prediction fine-tuning and our constraint-aware modeling fine-tuning, we perform parameter-efficient supervised fine-tuning using LoRA (Hu et al., 2022) on the Qwen2.5-7B-Instruct model. LoRA injects trainable low-rank matrices into selected layers while freezing the original weights. We apply LoRA to the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` modules, with LoRA rank set to 64 and scaling factor $\alpha = 16$. The maximum input length is 2048 tokens. We use a per-device batch size of 2 for both training and evaluation, with gradient accumulation steps of 8. The model is trained for 3 epochs using a cosine learning rate scheduler with a peak learning rate of 2×10^{-4} and a warmup ratio of 0.1. Training is performed in bfloat16 precision. All experiments are conducted on four NVIDIA RTX 5880 Ada Generation GPUs with 48GB memory each.

Hyperparameters. All hyperparameters are selected on a validation split only; no test labels are used for tuning. In our current implementation, the main tunable parameters are the AR-style shape-prior settings P and γ. For the main results, we use fixed validation-selected values: P = 3, γ = 0.1. The AR coefficients themselves are estimated separately for each instance from its historical observations (using up to the most recent 7/15/24/48 points) and do not use test labels.

Evaluation.

To evaluate the performance of the proposed model, we adopt three commonly used error metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

MAE measures the average magnitude of the absolute differences between the predicted values and the ground truth, providing a straightforward interpretation of prediction accuracy.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

RMSE computes the square root of the average squared errors, assigning a higher penalty to large deviations between predictions and true values.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

MAPE evaluates the average relative error in percentage terms, reflecting the prediction accuracy relative to the true values.

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Here, y_i and $\hat{y}_i$ denote the ground-truth value and the predicted value for the i-th sample, respectively, and N is the total number of samples.

A.2 Data Protocol and Absence of Test-Set Leakage

We clarify our data protocol and confirm that the adjustment of any test instance never uses its own ground-truth. The stages operate as follows (Table 3).

Stage	**Inputs used at inference**	**Own GT used?**	**Notes**
CIMI constraint selection	Validation split only	Validation only	Top-k types computed offline and frozen for all test instances (Alg. 1)
Hyperparameters $(\alpha, \beta, \lambda, k, P, \gamma)$	Validation split only	Validation only	Frozen before test (App. A.1)
AR shape-prior coefficients ϕ_s	Historical input window preceding the horizon	No	Estimated per instance from past observations only
Reflection loop and Evaluation Agent	Historical values, base forecast, adjusted forecast, retrieved memory	No	Operate purely on quantities available before the target values are realized
Test-time inference	Historical window + external text + frozen CIMI types and retrieved memory	No	—

Table 3: Data usage across stages. The adjustment of any test instance never uses its own ground-truth.

Regarding the reflection and evaluation modules specifically: at the moment an instance is forecast and adjusted, they rely only on the historical window, the base forecast, the adjusted forecast, and the retrieved memory — the instance's own ground-truth is never accessed. The memory bank is updated in a strictly sequential, forward-only manner: after an instance's forecast horizon has fully elapsed and its outcomes are realized (i.e., the ground-truth naturally becomes observable, as it would in any real deployment), the Evaluation Agent compares the previously produced adjustment against this realized outcome. Only if the adjustment was judged beneficial is the associated reasoning analyzed and written into memory, so that it can inform subsequent instances.

Crucially, this means the ground-truth of instance t can only influence the processing of later instances $t' > t$, and never the adjustment of instance t itself. There is therefore no leakage of test-set outcomes into the prediction of the corresponding instance — the memory update uses only past, already-realized values, exactly as would be available in a real forecasting deployment.

A.3 Human Evaluation of Constraint Extraction

Downstream forecast metrics do not by themselves establish that the extracted constraints match what a domain expert would write from the same news. We therefore constructed a small expert-annotated benchmark and compared the pipeline output against it.

We sampled 10 forecasting tasks covering 117 news articles. Domain experts read the associated news, wrote candidate constraints in the same catalog used by the model (Table 5), then deduplicated and audited the annotations into a golden set of 36 constraints. On the same tasks, the full extraction pipeline (Filter, Reasoning, and constraint-modeling LLMs) produced 64 constraints. A generated constraint is counted as a match if experts judged it to encode the same type, direction, and effective horizon as a golden constraint after this audit, *and* the associated numerical bounds or thresholds do not differ substantially from the golden values. Larger magnitude discrepancies are therefore recorded as extraction errors rather than matches. This yields 21 matches, i.e., precision 32.8% $(21/64)$ and recall 58.3% $(21/36)$, summarized in Table 4.

Quantity	**Value**
Forecasting tasks / news articles	10 / 117
Expert golden constraints	36
Generated constraints	64
Matches	21
Precision $(21/64)$	32.8%
Recall $(21/36)$	58.3%
Over-extraction or granular redundancy	$\sim 39\%$ of generated
Magnitude estimation noise	$\sim 16\%$ of generated
Temporal horizon mismatch	$\sim 9\%$ of generated

Table 4: Human evaluation of LLM-generated constraints against an expert-audited golden set. A generated constraint is counted as a match if experts judged it to encode the same type, direction, and effective horizon as a golden constraint after deduplication, and the numerical bounds or thresholds do not differ substantially from the golden values.

The precision is deliberately not high: the dominant mismatch is over-extraction rather than hallucinated opposite-direction rules. Experts typically keep a concise set of high-level constraints, whereas the LLM often emits several finer-grained or redundant inequalities that cover overlapping aspects of the same news item. Magnitude noise arises from mapping qualitative language (e.g., "slight increase") to numerical bounds, and horizon mismatch from small disagreements about the effective window. These three categories account for most non-matching generated constraints ($\sim 39\%$, $\sim 16\%$, and $\sim 9\%$, respectively).

This evaluation is a targeted audit rather than a substitute for the main forecast tables: the sample is small by design because constructing a golden constraint set requires expert reading of the full news bundle for each task. It is nonetheless useful because it explains *why* soft constraints, IIS repair, and the Filter Agent are necessary. Small bound or horizon errors can be absorbed by the slack variables s_k in Stage 1 without rendering the quadratic program infeasible; severe conflicts are diagnosed via IIS and handed to the Reflection Agent in Stage 2; and the Filter Agent ablation in Table 2 shows that skipping this screening can make the projection worse than the unadjusted base forecast.

A.4 Sources of External Information

External knowledge can originate from real-time news, business rules, regulatory requirements, historical events, and market analysis. These sources induce the following industrial classes of constraints, which the modeling-oriented taxonomy in the main text then maps onto solver-ready inequalities (Table 5):

- Event-driven constraints, which arise from sudden events such as policy changes, market shocks, or unexpected disruptions and require immediate forecast adjustments (Huang et al., 2019; Panagiotopoulos, 2012);
- Operational constraints, which represent capacity limits, inventory bounds, production schedules, and other process-driven rules that maintain feasibility in real-world operations (Mello, 2010; 2009; Panagiotopoulos, 2012);
- Temporal constraints, which ensure forecasts follow consistent trends or patterns, such as seasonality, cyclic behavior, or monotonic trends (Wang et al., 2016; Whitt & Zhang, 2019; Taylor, 2010);
- Regulatory constraints, which capture legal or compliance requirements, such as safety thresholds, permitted production levels, or contractual obligations (Strijbosch & Moors, 2005; Wallström & Segerstedt, 2010);
- Risk and probabilistic constraints, which specify that certain outcomes occur with a given probability (e.g., requiring that a price exceeds a threshold within a horizon with a specified probability). These constraints are typically derived from risk-management policies and can be approximated by discrete linear formulations (Alizadeh et al., 2002; Hull, 1993; Lee et al., 2010).

A.5 Constraint Types

See Table 5

A.6 Example of Input for Constraint Modeling Fine-tuning LLM

1.instruction: "The historical power load data is: .....6685.9,6592.7,6518.1,6479.1,6551.4,6752.8,6896.6....."

2.input: "National Grid forecasts a peak demand not exceeding 9547 MW for the upcoming 24-hour period, according to a new system operations report, The upcoming 48 periods power consumption is projected to settle between 7835 MW and 8320 MW as detailed in the latest market analysis, PowerCo analysts anticipate the load at the 25th hour will be higher than at the 47th hour, AEMO's short-term outlook indicates the 14th period's demand will be lower than the 15th period's, Gridwatch data suggests the 11th interval will see higher electrical load than the 10th interval. Task Description: You are an assistant that converts natural-language descriptions of power system load constraints into precise mathematical inequalities.Please read the above description and output all constraints on the next 48-period load sequence.Notation rules:1. Use the name load_next for the next 48-period load sequence.2. Use min(load_next), max(load_next), mean(load_next) for the minimum, maximum and mean of the sequence.3. Use y1, y2, ..., y48 to denote the load in period 1 to 48 (1-based index).4. Output ONLY the mathematical constraints, one constraint per line, with no extra explanation. Examples of format: min(load_next) $\geq$ 6000, max(load_next) $\geq$ 12000, mean(load_next) $\in$ [8000, 9000], y42 $>$ 16100, y48 $>$ y9. "

3.output_constraint: "max(load_next) $\leq$ 9547.57, mean(load_next) $\in$ [7835.83, 8320.42], y25 $>$ y47, y14 $<$ y15, y11 $>$ y10"

A.7 CIMI: Supplementary Details

A.7.1 Normalization and Dimensional Consistency

Both component scores are mapped to $[0, 1]$ before combination. **Intervention:** min-max normalized across all candidates on validation data: $\widetilde{\mathrm{Interv}}(c) = (\mathrm{Interv}(c) - \min_{c'} \mathrm{Interv}(c')) \,/\, (\max_{c'} \mathrm{Interv}(c') - \min_{c'} \mathrm{Interv}(c'))$. **DMI:** mapped via a saturating transform $\widetilde{\mathrm{DMI}}(c) = 1 - \exp(-\lambda\, \mathrm{DMI}(c))$ with $\lambda = 1$ by default, ensuring the score lies in $[0, 1)$ regardless of the raw scale of g_c. Since both terms are dimensionless and bounded in $[0, 1]$, the weighted sum $\alpha\, \widetilde{\mathrm{Interv}} \cdot \mathrm{Novelty} + \beta\, \widetilde{\mathrm{DMI}}$ is well-defined for any $\alpha, \beta > 0$.

Type	Mathematical Form	Description	Example
Min Boundary	$\min(y_{\text{next}}) \geq L$	Constrains the minimum value of the forecast sequence to be above a lower bound	$\min(y_{\text{next}}) \geq 45.2$
Max Boundary	$\max(y_{\text{next}}) \leq U$	Constrains the maximum value of the forecast sequence to be below an upper bound	$\max(y_{\text{next}}) \leq 52.8$
Mean Interval	$\text{mean}(y_{\text{next}}) \in [L, U]$	Constrains the mean of the forecast sequence to lie within a specified interval. *Equivalent to Sum constraint via* $\sum y_{\text{next}} \in [L \cdot n, U \cdot n]$	$\text{mean}(y_{\text{next}}) \in [48.5, 50.1]$
Probability	$P(y_{\text{next}} > \tau) \leq p$	Constrains the probability that forecast values exceed a threshold	$P(y_{\text{next}} > 49.0) \leq 0.65$
Pointwise	$y_i \circ \tau$	Constrains individual forecast points using relational operators $(<, \leq, >, \geq)$	$y_3 \geq 48.5$, $y_7 < 51.2$
Relative	$y_i \circ y_j$	Constrains relative ordering between different forecast time points	$y_2 > y_5$, $y_4 < y_8$
Smoothness Bound	$\lvert y_{i+1} - y_i \rvert \leq R$	Constrains the smoothness by limiting maximum changes between consecutive time points	$\lvert y_2 - y_1 \rvert \leq 2.5$
Window Sum	$\sum_{k=i}^{i+W-1} y_k \leq B$	Constrains the sum over sliding windows of width W to be below a bound. *Equivalent to Window Mean constraint via* $\frac{1}{W}\sum_{k=i}^{i+W-1} y_k \leq \frac{B}{W}$	$y_1 + y_2 + y_3 \leq 150.0$

Table 5: Constraint Types for Time Series Forecasting

A.7.2 Definition of Constraint-Specific Statistics $g_c(\cdot)$

Each constraint type c is associated with a sufficient statistic $g_c : \mathbb{R}^T \to \mathbb{R}$ that extracts the scalar quantity the constraint governs. Table 6 provides formal definitions for the constraint classes used in our experiments.

Constraint Type	**Name**	$g_c(Y)$	**Description**
Location	`mean`	$\frac{1}{T}\sum_{t=1}^{T} y_t$	Arithmetic mean
Bound	`max`	$\max_t y_t$	Maximum value
	`min`	$\min_t y_t$	Minimum value
Scale	`std`	$\left(\frac{1}{T}\sum_t (y_t - \bar{y})^2\right)^{1/2}$	Standard deviation
Derivative	`ramp`	$\max_t \lvert y_{t+1} - y_t\rvert$	Max absolute step change
	`smoothness`	$\mathrm{Std}(\{y_{t+1} - y_t\}_{t=1}^{T-1})$	Variability of first differences
Temporal	`autocorr`	$\frac{\sum_{t=1}^{T-1}(y_t-\bar{y})(y_{t+1}-\bar{y})}{\sum_t (y_t-\bar{y})^2}$	Lag-1 autocorrelation
Probability	$\texttt{prob}_\theta$	$\frac{1}{T}\sum_t \mathbf{1}[y_t > \theta]$	Exceedance probability
Pointwise	$\texttt{point}_\tau$	y_τ	Value at index τ
Pairwise	$\texttt{pair}_{(s,e)}$	$y_e - y_s$	Difference at indices s, e
Aggregate	$\texttt{window}_w$	$\max_t \sum_{i=t}^{t+w-1} y_i$	Max sliding-window sum

Table 6: Formal definitions of constraint-specific statistics $g_c(\cdot)$ for a sequence $Y = (y_1, \ldots, y_T) \in \mathbb{R}^T$.

A.7.3 Novelty score calculation.

Let $\mathbf{x} = (x_1, \ldots, x_n)^\top$ denote the historical series and $\mathbf{y} = (y_1, \ldots, y_H)^\top$ the true future values. Novelty is defined as:

$$\text{Novelty}(c) = 1 - \text{Redundancy}(c), \quad \text{Redundancy}(c) = 0.6\, S(c) + 0.4\, R(c) \tag{1}$$

where $S(c)$ is the natural satisfaction score:

$$S(c) = \begin{cases} \exp\Big(-\dfrac{|g_c(\mathbf{y}) - g_c(\mathbf{x})|}{\sigma(\mathbf{x}) + \epsilon}\Big) & \text{location} \\ \exp\Big(-5 \cdot \dfrac{|g_c(\mathbf{y}) - g_c(\mathbf{x})|}{|g_c(\mathbf{x})| + \epsilon}\Big) & \text{derivative} \\ 1 - \dfrac{1}{H}\sum_{i=1}^{H} \mathbf{1}[y_i \notin \mathcal{B}_{\text{hist}}] & \text{bound} \\ 0.3 & \text{otherwise} \end{cases} \tag{2}$$

Here $g_c(\cdot)$ is the statistic extractor for constraint c, $\sigma(\mathbf{x})$ is the standard deviation of $\mathbf{x}$, $\mathcal{B}_{\text{hist}} = [\min(\mathbf{x}), \max(\mathbf{x})]$ is the historical range, and $\epsilon = 10^{-10}$.

The correlation redundancy $R(c)$ is defined via the normalized shift $\delta(c) = (g_c(\mathbf{y}) - g_c(\mathbf{x})) \,/\, \max(|g_c(\mathbf{x})|, |g_c(\mathbf{y})|, \epsilon)$:

$$R(c) = \max_{c' \in \mathcal{C} \setminus \{c\}} \max\big(0,\ 1 - |\delta(c) - \delta(c')|\big) \tag{3}$$

A.8 Optional Shape Prior

To encourage smoother trajectories and capture ARIMA-like temporal dynamics while preserving solver tractability, we optionally introduce a light linear autoregressive (AR) regularizer to the Stage 1 objective:

$$\min \sum_{t=1}^{H} w_t (y_t - \hat{y}_t)^2 + \gamma \sum_{t=1}^{H} \Big(y_t - \sum_{s=1}^{P} \phi_s y_{t-s}\Big)^2 + \tau \sum_{k=1}^{K} \alpha_k \|s_k\|^2$$

where ϕ_s are estimated from a linear autoregressive model fitted to the historical observations preceding the forecast horizon and are held fixed during optimization. The order P is a hyperparameter with $P < H$. For indices $t - s \leq 0$, we initialize y_{t-s} using the most recent historical observations prior to the forecasting horizon (e.g., y_0 corresponds to the last observed value, y_{-1} to the second last, etc.), or alternatively treat them as predefined constants. Setting $\gamma = 0$ recovers the default *Minimal-Distortion Projection*.

This linear AR regularizer provides a convex, tractable surrogate for ARIMA-like dynamics. The quadratic AR term and convexity of other components ensure that the resulting objective is a convex quadratic program with linear constraints, efficiently solvable using standard solvers like Gurobi. This formulation guarantees temporal coherence, noise robustness, and stability under missing or irregular observations, maintaining the reliable solvability of the optimization problem (Panagiotopoulos, 2012).

A.9 More Results

Overall ablation of external information. We used Bitcoin data to conduct a coarse-grained ablation that removes the entire external-information module while retaining the downstream reflection, modeling, optimization, and memory pipeline. As shown in Table 7, performance degrades substantially under this configuration. This result confirms that the reflection, modeling, and optimization modules, in isolation, do not yield meaningful performance improvements without access to informative external signals. In other words, the primary source of gain in our framework is the projection of a base forecast onto a feasible region defined by LLM-generated constraints, rather than the iterative refinement pipeline itself. The downstream modules serve to enhance and stabilize this projection process, but cannot substitute for the rich structural knowledge distilled from external sources. Component-wise ablations of the Filter Agent, reflection loop, and memory module are reported in the main text (Table 2).

Model	**MAE (base / no_ext, $\Delta\%$)**	**RMSE (base / no_ext, $\Delta\%$)**
Nbeats	965.11 / 1541.04, -59.68%	1921.42 / 3731.60, -94.21%
PatchTST	932.81 / 1714.01, -83.75%	1495.61 / 6383.44, -326.81%
DLinear	1163.54 / 1470.46, -26.38%	2291.27 / 2397.37, -4.63%
LightGBM	734.86 / 1059.19, -44.13%	1156.65 / 1327.50, -14.77%
Chronos	1421.55 / 1367.36, $+3.81\%$	2714.50 / 3025.55, -11.46%
iTransformer	1204.77 / 1330.90, -10.47%	2309.84 / 2875.23, -24.48%
TimeLLM	1299.68 / 2288.64, -76.09%	2467.26 / 2632.00, -6.68%

Table 7: Overall ablation on Bitcoin: comparison of models with and without external information, while retaining the reflection, modeling, optimization, and memory pipeline.

Comparison across all Constraint types and Constraint types selected by CIMI. We next bypass the CIMI constraint selection mechanism and instead feed all generated constraints directly into the optimization module. Table 8 and Table 9 compare the optimization results obtained using all constraint types with those obtained using the constraint types selected by our CIMI method on the synthetic news dataset and the true news dataset. Overall, CIMI achieves performance comparable to using all constraints, with only slight accuracy degradation observed in a few cases. The performance gap is therefore marginal, indicating that CIMI does not sacrifice optimization quality. Rather, its value lies in substantially reducing inference overhead by filtering out redundant or uninformative constraints, thereby enabling more efficient reasoning without compromising forecast accuracy.

Experiments on synthetic news dataset. Notably, the performance gains on the synthetic dataset in Table 10 are more pronounced than those observed on most real-world datasets in Table 1. A plausible explanation is that the synthetic news data is constructed with explicitly embedded information that directly governs the generation of the time series, making the underlying patterns more transparent and the relevant constraints easier to identify and exploit. In contrast, real-world datasets often contain implicit, noisy, or entangled information that is harder to disentangle and leverage through textual constraints.

Domain	Metrics	Nbeats		PatchTST		DLinear-Direct		LightGBM		Chronos		iTransformer		LoRA		TimeLLM	
		ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C
Electricity	MAE	193.47	**196.51** (–1.57%)	193.17	**197.77** (–2.38%)	216.71	**221.77** (–2.33%)	193.83	**197.27** (–1.77%)	418.76	**425.58** (–1.63%)	202.5	**207.09** (–2.27%)	291.87	**296.38** (–1.55%)	243.9	**247.33** (–1.41%)
	RMSE	281.64	**286.68** (–1.79%)	285.89	**293.32** (–2.60%)	322.95	**330.06** (–2.20%)	294.26	**299.29** (–1.71%)	601.52	**610.14** (–1.43%)	305.05	**311.79** (–2.21%)	441.26	**446.30** (–1.14%)	349.70	**354.73** (–1.44%)
	MAPE	5.61	5.69 (–1.43%)	5.43	5.56 (–2.39%)	5.80	5.96 (–2.76%)	5.16	5.25 (–1.74%)	9.94	10.10 (–1.61%)	5.49	5.62 (–2.37%)	7.26	7.37 (–1.52%)	6.39	6.49 (–1.56%)
Bitcoin	MAE	619.56	**617.87** (+0.27%)	556.8	**558.65** (–0.33%)	626.04	**629.75** (–0.59%)	615.63	**618.44** (–0.46%)	679.40	**694.52** (–2.23%)	649.55	**652.81** (–0.50%)	699.42	**694.82** (+0.66%)	645.57	**651.82** (–0.97%)
	RMSE	1493.36	**1513.83** (–1.37%)	1275.89	**1266.02** (+0.77%)	1534.26	**1544.97** (–0.70%)	1488.18	**1468.22** (+1.34%)	1607.67	**1621.52** (–0.86%)	1690.25	**1679.81** (+0.62%)	1762.57	**1744.30** (+1.04%)	1659.06	**1656.85** (+0.13%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Traffic	MAE	16.27	**15.93** (+2.09%)	16.7	**16.56** (+0.84%)	24.8	**23.61** (+4.80%)	14.13	**14.03** (+0.71%)	21.52	**20.22** (+6.04%)	20.56	**19.13** (+6.96%)	18.65	**18.01** (+3.43%)	27.18	**25.94** (+4.56%)
	RMSE	35.23	**35.22** (+0.03%)	34.51	**34.96** (–1.30%)	48.15	**46.89** (+2.62%)	31.56	**32.24** (–2.15%)	43.41	**42.53** (+2.03%)	42.34	**41.47** (+2.05%)	41.12	**41.29** (–0.41%)	51.23	**50.78** (+0.88%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Exchange	MAE	6.83	**5.32** (+22.11%)	6.91	**5.4** (+21.85%)	7.19	**5.66** (+21.28%)	6.74	**5.10** (+24.33%)	6.86	**5.27** (+23.18%)	7.35	**6.08** (+17.28%)	7.26	**5.71** (+21.35%)	6.77	**5.34** (+21.12%)
	RMSE	9.39	**6.89** (+26.62%)	9.47	**7.06** (+25.45%)	9.74	**7.36** (+24.44%)	9.4	**6.79** (+27.77%)	9.48	**7.00** (+26.16%)	9.93	**7.94** (+20.04%)	9.84	**7.40** (+24.80%)	9.47	**7.07** (+25.34%)
	MAPE	0.95	0.74 (+22.11%)	0.96	0.75 (+21.88%)	1.00	0.79 (+21.00%)	0.94	0.71 (+24.47%)	0.95	0.73 (+23.16%)	1.02	0.85 (+16.67%)	1.01	0.79 (+21.78%)	0.94	0.74 (+21.28%)

Table 8: Performance comparison across all constraint types and constraint types selected by CIMI on synthetic news dataset.

Domain	Metrics	Nbeats		PatchTST		DLinear		LightGBM		Chronos		iTransformer		LoRA		TimeLLM	
		ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C	ALL-C	CIMI-C
Electricity	MAE	229.68	**228.71** (+0.42%)	233.92	**232.55** (+0.59%)	273.94	**270.17** (+1.38%)	227.59	**222.78** (+2.11%)	462.08	**459.6** (+0.54%)	249.09	**250.38** (-0.52%)	405.96	**406.08** (-0.03%)	322.67	**318.46** (+1.30%)
	RMSE	358.75	**359.09** (-0.09%)	375.11	**372.01** (+0.83%)	446.01	**440.68** (+1.20%)	365.72	**360.25** (+1.50%)	672.67	**674.74** (-0.31%)	412.00	**416.78** (-1.16%)	651.73	**648.84** (+0.44%)	494.18	**488.5** (+1.15%)
	MAPE	6.58	6.54 (+0.61%)	6.49	6.48 (+0.15%)	7.26	7.12 (+1.93%)	6.00	5.88 (+2.00%)	10.97	10.89 (+0.73%)	6.74	6.72 (+0.30%)	9.75	9.75 (0.00%)	8.17	8.07 (+1.22%)
Bitcoin	MAE	813.20	**818.95** (-0.71%)	820.94	**818.98** (+0.24%)	972.64	**988.74** (-1.66%)	630.09	**627.19** (+0.46%)	1127.75	**1035.97** (+8.14%)	980.26	**985.33** (-0.52%)	1137.65	**1128.23** (+0.83%)	1049.62	**1076.72** (-2.58%)
	RMSE	1602.57	**1623.31** (-1.29%)	1378.92	**1371.67** (+0.53%)	1928.04	**1938.70** (-0.55%)	1013.25	**1008.91** (+0.43%)	1959.14	**1979.93** (-1.06%)	1978.45	**1976.74** (+0.09%)	1924.58	**1930.90** (-0.33%)	2034.65	**2064.97** (-1.49%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Traffic	MAE	16.82	**16.52** (+1.78%)	17.54	**17.27** (+1.54%)	24.98	**24.44** (+2.16%)	14.73	**14.35** (+2.58%)	20.53	**19.8** (+3.56%)	19.74	**19.06** (+3.44%)	20.13	**19.61** (+2.58%)	27.01	**26.57** (+1.63%)
	RMSE	37.67	**38.2** (-1.41%)	36.85	**37.58** (-1.98%)	51.37	**50.81** (+1.09%)	34.25	**34.57** (-0.93%)	46.99	**46.17** (+1.74%)	45.60	**45.18** (+0.92%)	48.11	**48.54** (-0.89%)	54.80	**54.78** (+0.04%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Exchange	MAE	5.13	**5.13** (0.00%)	5.18	**5.23** (-0.97%)	5.65	**5.27** (+6.73%)	5.02	**5.05** (-0.60%)	5.12	**5.2** (-1.56%)	5.85	**6.00** (-2.56%)	5.41	**5.42** (-0.18%)	5.08	**5.1** (-0.39%)
	RMSE	6.77	**6.78** (-0.15%)	6.89	**6.94** (-0.73%)	7.37	**7.01** (+4.88%)	6.87	**6.91** (-0.58%)	6.95	**7.02** (-1.01%)	7.84	**7.96** (-1.53%)	7.24	**7.27** (-0.41%)	6.85	**6.88** (-0.44%)
	MAPE	0.72	0.71 (+1.39%)	0.72	0.73 (-1.39%)	0.79	0.74 (+6.33%)	0.70	0.71 (-1.43%)	0.71	0.72 (-1.41%)	0.81	0.84 (-3.70%)	0.75	0.75 (0.00%)	0.71	0.71 (0.00%)

Table 9: Performance comparison across all constraint types and constraint types selected by CIMI on true news dataset.

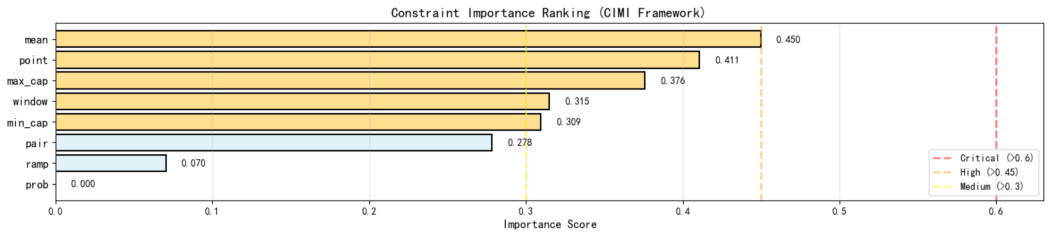
Acceleration by CIMI. As illustrated in Figure 11, CIMI achieves substantial efficiency improvements by prioritizing the top three or four most important constraint types, rather than exhaustively evaluating all eight (treating minimum and maximum as distinct). This selective approach drastically reduces the computational overhead associated with checking every potential constraint.

Constraint type selection results by CIMI. Figures 3 present the importance of different constraint types under the CIMI method on the electricity, Bitcoin, traffic flow, and exchange rate datasets, respectively. The results clearly indicate that the relative importance of constraints varies substantially across datasets, reflecting the heterogeneous characteristics of different domains. No single constraint consistently dominates in all scenarios, highlighting the necessity of dataset-specific constraint selection. From a practical perspective, these findings suggest that practitioners can effectively reduce reasoning time by selecting only the top three or four most important constraints for each dataset.

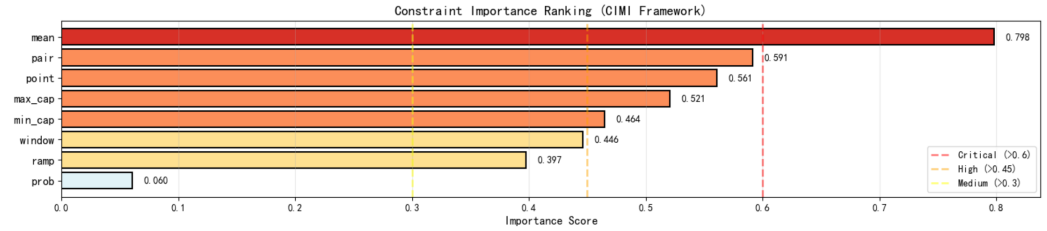
B Full Prompt Design

B.1 Prompt for constructing external information

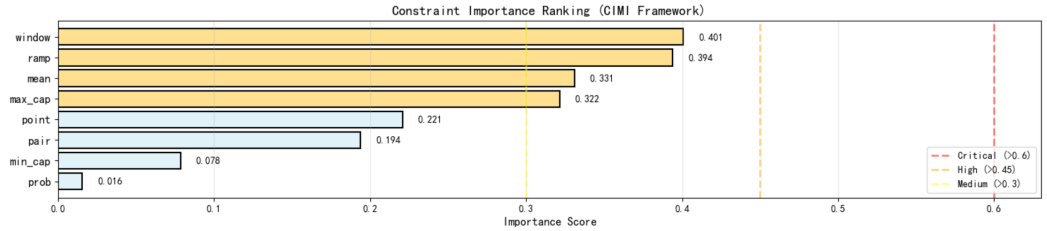
This section uses traffic flow data as an example to detail a complete prompt template used in constructing external information. The system employs role-based prompt engineering with eight distinct professional personas to generate diverse natural language descriptions.



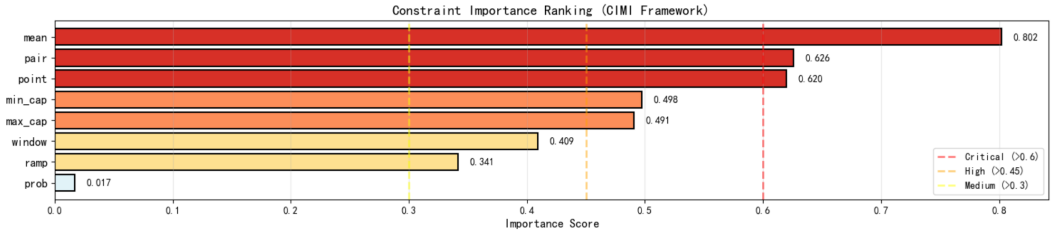
(a) CIMI-AULF



(b) CIMI-Bitcoin



(c) CIMI-Traffic



(d) CIMI-Exchange

Figure 3: Constraint type selection results by CIMI

Domain	Metrics	Nbeats		PatchTST		DLinear		LightGBM		Chronos		iTransformer		LoRA		TimeLLM	
		base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C	base	CIMI-C
Electricity	MAE	236.03	**196.51** (+16.74%)	247.05	**197.77** (+19.95%)	284.74	**221.77** (+22.11%)	234.85	**197.27** (+16.00%)	514.83	**425.58** (+17.34%)	259.27	**207.09** (+20.13%)	440.59	**296.38** (+32.73%)	331.97	**247.33** (+25.50%)
	RMSE	374.55	**286.68** (+23.46%)	397.11	**293.32** (+26.14%)	468.01	**330.06** (+29.48%)	383.45	**299.29** (+21.95%)	765.61	**610.14** (+20.31%)	428.85	**311.79** (+27.30%)	695.73	**446.30** (+35.85%)	505.66	**354.73** (+29.85%)
	MAPE	6.92	5.69 (+17.77%)	6.98	5.56 (+20.34%)	7.54	5.96 (+20.95%)	6.19	5.25 (+15.19%)	12.43	10.10 (+18.74%)	6.96	5.62 (+19.25%)	10.38	7.37 (+29.00%)	8.44	6.49 (+23.10%)
Bitcoin	MAE	965.11	**617.87** (+35.98%)	932.81	**558.65** (+40.11%)	1163.54	**629.75** (+45.88%)	734.86	**618.44** (+15.84%)	1421.55	**694.52** (+51.14%)	1204.77	**652.81** (+45.82%)	1398.45	**694.82** (+50.32%)	1299.68	**651.82** (+49.85%)
	RMSE	1921.42	**1513.83** (+21.21%)	1495.61	**1266.02** (+15.35%)	2291.27	**1544.97** (+32.57%)	1156.65	**1068.22** (+7.65%)	2714.50	**1621.52** (+40.27%)	2309.84	**1679.81** (+27.28%)	2387.36	**1744.30** (+26.94%)	2467.26	**1656.85** (+32.85%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Traffic	MAE	16.92	**15.93** (+5.85%)	17.79	**16.56** (+6.91%)	25.02	**23.61** (+5.64%)	14.5	**14.03** (+3.24%)	20.6	**20.22** (+1.84%)	19.49	**19.13** (+1.85%)	20.14	**18.01** (+10.58%)	26.9	**25.94** (+3.57%)
	RMSE	39.04	**35.22** (+9.78%)	38.36	**34.96** (+8.86%)	51.83	**46.89** (+9.53%)	34.87	**32.24** (+7.54%)	47.47	**42.53** (+10.41%)	46.31	**41.47** (+10.45%)	50.06	**41.29** (+17.52%)	55.17	**50.78** (+7.96%)
	MAPE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Exchange	MAE	5.50	**5.32** (+3.27%)	5.49	**5.4** (+1.64%)	5.77	**5.66** (+1.91%)	5.28	**5.10** (+3.41%)	5.35	**5.27** (+1.50%)	6.53	**6.08** (+6.89%)	6.01	**5.71** (+4.99%)	5.47	**5.34** (+2.38%)
	RMSE	7.12	**6.89** (+3.23%)	7.16	**7.06** (+1.40%)	7.47	**7.36** (+1.47%)	7.09	**6.79** (+4.23%)	7.14	**7.00** (+1.96%)	8.56	**7.94** (+7.24%)	7.89	**7.40** (+6.21%)	7.23	**7.07** (+2.21%)
	MAPE	0.77	0.74 (+3.90%)	0.76	0.75 (+1.32%)	0.80	0.79 (+1.25%)	0.74	0.71 (+4.05%)	0.74	0.73 (+1.35%)	0.91	0.85 (+6.59%)	0.83	0.79 (+4.82%)	0.76	0.74 (+2.63%)

Table 10: Performance comparison across different domains and models on synthetic news dataset. The numbers in parentheses represent the percentage improvement effect of our method. The dash (—) indicates that the value is not applicable due to inherent mathematical deficiencies of MAPE.

Domain	Time		Improvement
	All-C	CIMI-C	
Electricity	1172.52	442.37	62.27%
Bitcoin	932.11	388.13	58.36%
Traffic	1362.50	709.10	47.96%
Exchange	1012.55	457.53	54.81%

Table 11: Comparison of time consumption for constraint type selection

B.1.1 Role Configurations

We configured eight distinct professional roles to diversify the linguistic patterns (see Table 12).

Role	**Source**	**Expertise**
Senior Traffic Flow Analyst	Municipal Traffic Management Center	Real-time sensor networks and predictive modeling
Transportation Correspondent	Urban Mobility Daily	Commuter behavior patterns and infrastructure analysis
AI Traffic Prediction Engineer	Intelligent Transportation Systems Lab	Machine learning forecasting and probabilistic modeling
Urban Infrastructure Analyst	City Planning & Transportation Dept.	Road capacity planning and demand management
Traffic Safety Coordinator	Road Safety Monitoring Bureau	Congestion risk assessment and incident prevention
Smart City IoT Specialist	Connected Vehicle Platform	Real-time sensor fusion and edge computing analytics
Transportation Economist	Urban Economics Research Institute	Traffic efficiency and economic impact analysis
Public Transit Coordinator	Metropolitan Transit Authority	Multi-modal integration and schedule optimization

Table 12: Professional roles and their associated expertise domains

B.1.2 System Prompt Template

For each role, we construct a system prompt that establishes professional identity, domain expertise, and encoding philosophy:

System Prompt Template

`You are a {role} working with {source}, specializing in {expertise}.`

PROFESSIONAL BACKGROUND:

- 15+ years experience in traffic flow forecasting and transportation systems analysis
- Expert in next-day 24-hour traffic prediction using historical sensor data
- Published contributor to major urban mobility research publications
- Specialist in constraint-aware predictive modeling

YOUR WRITING STYLE: {*style*} **YOUR COMMUNICATION TONE:** {*tone*}

PREFERRED TECHNICAL VOCABULARY: {*comma-separated vocabulary list*}

CORE COMPETENCY: You excel at transforming mathematical traffic flow constraints into sophisticated natural language forecasts where: (1) Exact numerical values are NEVER stated directly; (2) Sufficient quantitative anchors are embedded so trained readers can reconstruct values within ±5%; (3) Domain-specific terminology creates technical credibility; (4) Hour-specific references ground predictions in real operational contexts.

FORECAST CONTEXT:

- Prediction horizon: Next-day 24-hour period
- Temporal resolution: Hourly intervals (y1 = 00:00–01:00, ..., y24 = 23:00–24:00)
- Key periods: Morning peak (y8–y9: 07:00–09:00), Evening peak (y17–y18: 16:00–18:00)
- Flow metric: Vehicles per hour (or normalized flow index)
- Data source: Previous day's complete 24-hour observation cycle

ENCODING PHILOSOPHY: You transform constraints using contextual engineering anchors such as capacity utilization percentages with explicit baseline values, comparative temporal references with magnitude deltas, probabilistic bounds with explicit thresholds, operating bands with defined upper/lower limits, cumulative throughput windows with hour ranges and totals, and inter-hour fluctuation caps with specific time transitions. Your sentences read like professional transportation analysis reports: technically precise, quantitatively grounded, but naturally flowing without raw mathematical notation.

B.1.3 User Prompt Template

The user prompt provides the mathematical constraint and encoding strategies:

User Prompt Template

ASSIGNMENT: Next-Day Hourly Traffic Flow Forecast Generation

TECHNICAL CONSTRAINT TO ENCODE: {`ground_truth_constraint`}

OPERATIONAL CONTEXT:

- Forecast period: Tomorrow, 00:00–24:00 (24 hourly intervals)
- Hour mapping: y1→00:00–01:00, y2→01:00–02:00, ..., y24→23:00–24:00
- Traffic metric: Hourly vehicle flow volumes (vehicles/hour or standardized units)
- Morning peak window: 07:00–09:00 (y8, y9)
- Evening peak window: 16:00–18:00 (y17, y18)
- Data basis: Previous day's complete 24-hour traffic observation cycle

ENCODING STRATEGY LIBRARY

STRATEGY 1: NORMALIZED FLOW UNITS
Purpose: Direct magnitude indication with approximate quantifiers
Examples: "hovering just above the 60-vehicle-per-hour mark"; "operating in the low-to-mid sixties on a per-hour basis"; "stabilizing near the 65-vehicle hourly threshold"
MANDATORY: Include explicit flow unit (vehicles per hour)

STRATEGY 2: CAPACITY UTILIZATION RATIOS
Purpose: Percentage relative to explicit baseline or capacity
Examples: "running at roughly 85% of the nominal 75-vehicle-per-hour lane capacity"; "approximately 120% of the late-night baseline near 50 vehicles per hour"; "tracking at about 90% of the modeled 80-vehicle saturation threshold"
MANDATORY: Include BOTH percentage AND baseline value

STRATEGY 3: EXPLICIT FLOW CORRIDORS
Purpose: Define bounded operating ranges
Examples: "confined to a narrow 60-to-65 vehicles-per-hour operating band"; "within a tightly controlled ±3% corridor centered on the low-60s range"; "restricted to a 58-67 vehicles-per-hour envelope"
MANDATORY: Specify both bounds or center with margin

STRATEGY 4: TEMPORAL BENCHMARKING
Purpose: Comparison to other time periods with quantified differences
Examples: "roughly 15% below the evening-peak volumes observed near 75 vehicles per hour"; "slightly above the stabilized overnight baseline by approximately 8 vehicles per hour"; "10% lighter than typical mid-morning patterns near 70 vehicles per hour"
MANDATORY: Include percentage or absolute delta with reference point

STRATEGY 5: INTER-HOUR RELATIONSHIPS
Purpose: Relative comparisons between specific hours
Examples: "early-evening (17:00–18:00) exceeding early-morning (07:00–08:00) by roughly 12 vehicles per hour"; "mid-morning flow (09:00–10:00) remaining 20% higher than pre-dawn conditions (04:00–05:00)"; "afternoon throughput (15:00–16:00) surpassing late-night by approximately 15 vehicles per hour"
MANDATORY: Specify BOTH time periods with explicit magnitude

STRATEGY 6: PROBABILISTIC THRESHOLDS
Purpose: Constraints on exceedance probability
Examples: "likelihood of hourly volumes breaching the 60-vehicle threshold remains below 20%"; "only a 15% probability tail reaching beyond the 65-vehicle congestion level"; "exceedance probability for the 70-vehicle mark constrained to roughly one-fifth"
MANDATORY: Include BOTH probability percentage AND threshold value

STRATEGY 7: HOUR-TO-HOUR FLUCTUATION LIMITS
Purpose: Cap on consecutive-hour variation
Examples: "transition from 07:00–08:00 to 08:00–09:00 constrained within a 200-vehicle swing"; "hour-to-hour variation during morning ramp-up (08:00–10:00) capped at ±150 vehicles"; "inter-hour volatility between 17:00–18:00 and 18:00–19:00 bounded within 180 vehicles"
MANDATORY: Specify adjacent hours AND explicit change magnitude

STRATEGY 8: CUMULATIVE THROUGHPUT WINDOWS
Purpose: Multi-hour aggregate flow constraints
Examples: "aggregate throughput from 15:00 to 21:00 remains below a roughly 1,100-vehicle cumulative ceiling"; "the six-hour window (07:00–13:00) accumulates just under 1,200 total vehicles"; "cumulative flow spanning 16:00 to 22:00 capped near 1,050 vehicles"
MANDATORY: Specify BOTH hour range AND cumulative total

YOUR TASK:

1. Analyze the constraint structure
2. Select the MOST APPROPRIATE encoding strategy
3. Generate ONE sophisticated sentence (about 120 words) that:
   - ✓ HIDES exact numerical values from the original constraint
   - ✓ EMBEDS sufficient quantitative anchors for ±5% reconstruction
   - ✓ USES your professional vocabulary: {*vocabulary*}
   - ✓ REFERENCES specific hours when relevant
   - ✓ ATTRIBUTES findings to credible sources
   - ✓ MAINTAINS {*style*} style and {*tone*} tone
   - ✓ FLOWS naturally as a single coherent sentence

CRITICAL REQUIREMENTS:

- DO NOT state exact numbers from the constraint
- DO provide precise contextual anchors
- DO reference specific hours when constraint involves them
- DO use domain terminology consistently
- DO NOT write multiple sentences - ONLY ONE sentence
- DO NOT explain your encoding - just deliver the forecast statement

B.1.4 Generation Parameters

We used the following LLM generation parameters:

- **Model:** DeepSeek-V3.2 (via SiliconFlow API)

- **Temperature:** 0.85 (to encourage linguistic diversity)
- **Top-p:** 0.95 (nucleus sampling)
- **Max tokens:** 2400
- **Retry strategy:** Exponential backoff with up to 3 attempts

B.2 News Filtering Prompt Details

This section uses Bitcoin data as an example to detail the complete prompt template used in the external information filtering module.

B.2.1 System Prompt for News Relevance Analysis

The system prompt defines the expert role, evaluation criteria, and response format:

System Prompt: News Relevance Evaluator

```
You are an expert in Bitcoin price forecasting.  Your task is to:
1.  Determine whether a news article will impact Bitcoin price
prediction
2.  If relevant, provide an objective summary of the news content
```

EVALUATION CRITERIA:

Consider whether this news will impact Bitcoin price movements. Events that typically affect Bitcoin price include:

1. Regulatory News: Government regulations, policy changes, legal decisions, cryptocurrency bans or approvals, tax policies

2. Major Market Events: Large institutional investments, major company adoptions (e.g., Tesla, MicroStrategy), ETF approvals or rejections

3. Technical Events: Network upgrades, hard forks, halving events, security breaches, major exchange hacks

4. Macroeconomic Factors: Federal Reserve decisions, inflation data, major economic crises, currency devaluations

5. Whale Activities: Large BTC transfers, significant exchange inflows/outflows, major wallet movements

6. Industry News: Major exchange shutdowns, new cryptocurrency launches, DeFi protocol issues, stablecoin depegging

7. Market Sentiment: Celebrity endorsements, viral social media trends, major FUD (Fear, Uncertainty, Doubt) or FOMO events

8. Global Events: Geopolitical tensions, financial crises, major economic sanctions, banking system issues

9. Direct Price Predictions: Analysis or forecasts about Bitcoin price movements

RESPONSE FORMAT:
Respond in JSON format:

```
{
  "relevant": "YES" or "NO",
  "summary": "If YES: provide an objective, factual summary
             of the news. Include all key information such as:
             who, what, when, where, specific numbers, dates,
             quotes, and concrete events. Do NOT include
             subjective interpretations or predictions.
             If NO: leave empty string."
}
```

B.2.2 User Prompt Template

The user prompt provides temporal context and the news article content:

User Prompt Template: News Article Evaluation

CONTEXT:
- News Date Range: {target_date} to {end_date} (format: YYYY-MM-DD)
- These news articles will be used to predict Bitcoin price for the following 15 days

NEWS ARTICLE:
- **Title:** {title}
- **Publication Time:** {pub_time} (format: YYYY-MM-DD HH:MM:SS)
- **Full Article:** {full_article} (truncated to 8000 characters if necessary)

INSTRUCTIONS:
Analyze the above news article and determine:
1. Is this news relevant to Bitcoin price prediction?
2. If relevant, provide a factual summary with key details (who, what, when, where, specific numbers)
3. Return results in the specified JSON format

B.3 News Analysis Prompt Details

This section showcases the complete prompt template used in the External Information Analysis module of our Bitcoin price prediction system.

B.3.1 System Prompt Template

The system prompt varies based on impact type and operational mode:

System Prompt Template (Standard Mode)

```
You are a professional Bitcoin price forecasting analyst specializing
in {impact_type_name}.
```

CORE COMPETENCIES:
- Always provide **specific numerical predictions** (USD amounts, percentages, or day numbers)
- Deliver **clear causal reasoning** linking news events to predicted impacts
- Keep descriptions **concise and data-driven**
- Focus exclusively on the assigned impact type

HISTORICAL LEARNING: Learn from historical successful analysis patterns to improve forecast accuracy.

OUTPUT REQUIREMENTS:
- Each prediction MUST contain explicit numbers
- Each prediction MUST include one-sentence causal explanation
- Return results in valid JSON format only

Reflection Mode Enhancement:

When in reflection mode (after initial prediction fails), the system prompt appends:

[REFLECTION MODE ACTIVE]
You are in **REFLECTION MODE** - reconsider your previous analysis and try different approaches to improve forecast accuracy. Review past attempts, identify weaknesses, and generate alternative predictions using novel methodologies.

B.3.2 User Prompt Template: Historical Experiences Section

When historical experiences are available, they are formatted as follows:

User Prompt Component: Historical Experiences

HISTORICAL SUCCESSFUL EXPERIENCES (Learn from past optimizations):
==========

1. Experience (RMSE Improvement: 13.24):
Focus on regulatory announcements with specific policy details. Large percentage changes (>10%) correlate with institutional investment news.
Keywords: regulation, ETF approval, institutional adoption
Example: "SEC Approves First Bitcoin ETF..."
→ Identified as: Regulatory Approval (Strong Positive impact, magnitude: 15-20%)

2. Experience (RMSE Improvement: 12.87):
[Additional experience entries following same format...]

==========

APPLY THESE LEARNINGS: Use the above successful patterns to guide your analysis. Focus on the analysis methods that have proven effective in improving forecasts.

B.3.3 User Prompt Template: Reflection Feedback Section

When in reflection mode, historical attempts are summarized:

User Prompt Component: Reflection Feedback

REFLECTION MODE (Iteration {`iteration`}):
==========

PREVIOUS ATTEMPTS HISTORY ({`num_attempts`} attempts):

Attempt 1:
- RMSE Improvement: 0.0124
- Approach: Conservative regulatory impact analysis
- Constraints Sample: "Maximum price below $52,000 due to ETF approval..."

Attempt 2:
- RMSE Improvement: 0.0089
- Approach: Aggressive institutional adoption projections
- Constraints Sample: "Average price around $55,000 driven by whale accumulation..."

LEARNING FROM HISTORY - Analyze the patterns:

1. TRAJECTORY ANALYSIS:
- Are improvements increasing, decreasing, or plateauing?
- Which iteration showed the best result? What was different?
- What approaches have been tried and exhausted?

2. STRATEGY ADJUSTMENT:
- If improvements are decreasing: Try fundamentally different approach
- If improvements are increasing: Refine and enhance current direction
- If plateau: Introduce novel perspective or methodology

3. MAGNITUDE CALIBRATION:
- Review historical predictions vs actual effectiveness
- If previous attempts were too conservative: Be more aggressive
- If previous attempts were too aggressive: Be more measured
- Find the optimal balance based on trend

4. TEMPORAL PATTERN REVIEW:
- Did we correctly identify when impacts occur (early/mid/late period)?
- Should we adjust time windows based on news type?
- Consider lag effects and momentum patterns

5. CONSTRAINT QUALITY:
- Were constraints too strict (over-constraining)?
- Were constraints too loose (under-constraining)?
- Did constraints align with actual price movements?

6. NOVEL APPROACHES TO TRY:
- Different event impact categorizations
- Alternative magnitude assessment methods
- New temporal pattern recognition

- Combination effects of multiple news items

IMPORTANT INSTRUCTIONS:
- DO NOT repeat failed approaches from history
- PRIORITIZE untested strategies
- LEARN from the trajectory - what's working, what's not
- Make BOLD changes if plateau is detected
- Ensure numerical predictions are realistic and well-justified
- Focus on the MOST IMPACTFUL aspects based on historical success

==========

B.3.4 Main User Prompt Template

The complete user prompt structure:

User Prompt Template: Main Analysis Request

```
You are a Bitcoin price forecasting expert. Analyze the following news
summaries and determine their COMBINED impact on Bitcoin prices in the
next 15 days.

{combined_summaries}

[Historical Experiences Section - if available]

[Reflection Feedback Section - if in reflection mode]
```

TASK: Analyze the news ONLY for impact type {impact_type}:
{impact_type_definition}

CRITICAL REQUIREMENTS:
1. Focus ONLY on impact type {impact_type}
2. Each description MUST contain specific numerical values (USD amounts, percentages, or day numbers)
3. Reason must be one sentence explaining WHY this impact occurs based on the news
4. If the news does NOT suggest this specific type of impact, return empty array
5. **[If Reflection Mode]:** Review ALL previous attempts above and try a DIFFERENT approach that hasn't been tested yet
6. **[If Historical Experiences]:** Apply the historical successful patterns shown above to improve your analysis accuracy

Output Format (JSON):

```
{
  "impacts": [
    {
      "type": <impact_type>,
      "description": "Brief analysis with specific numbers",
      "reason": "One sentence explaining the cause
                 based on news"
    }
  ]
}
```

B.4 Memory Summarization Prompt Details

This section introduces the hint template in the memory module used to extract and store successful Bitcoin price prediction experience.

B.4.1 System Prompt Template

The system prompt defines the LLM's role and expertise:

System Prompt: Memory Extraction

```
You are an expert in Bitcoin price prediction, skilled at extracting
prediction methodologies from successful optimization cases.
```

CORE COMPETENCIES:

- Identify patterns in successful price movement forecasts
- Distill actionable prediction strategies from case studies
- Recognize feature-impact relationships that improve accuracy
- Extract transferable methodologies applicable to future predictions

OUTPUT REQUIREMENTS:

- Concise summaries (within 50 words)
- Focus on **HOW** to predict, not **WHAT** was predicted
- Emphasize generalizable patterns over case-specific details
- Provide actionable guidance for future analyses

B.4.2 User Prompt Template: Experience Extraction

The main user prompt presents the optimization case and requests methodology extraction:

User Prompt Template: Experience Summarization

You are an expert in Bitcoin price prediction. The following is a successful optimization case with an improvement of **{improvement}**.

By comparing the original features and the analysis results, please summarize **how to effectively analyze information for Bitcoin price forecasting.**

CASE INFORMATION

- Training Period: {date_range_start} to {date_range_end}
- RMSE Improvement: {improvement}

ORIGINAL FEATURES VS. ANALYSIS RESULTS

[JSON-formatted feature comparison - see structure above]

TASK: Summarize in 1–2 sentences:

1. Which features or information are most predictive for Bitcoin price movements?
2. How to determine the direction and magnitude of price impact?
3. What special patterns or indicators should be paid attention to?

EXAMPLE FORMAT:

> *"For regulatory news, focus on policy scope and implementation timeline. Major regulatory announcements typically cause 10-20% price volatility within 24-48 hours, with initial reactions often reversed after 3-5 days."*

OUTPUT REQUIREMENTS:

- Return only the summarized text (within 50 words)
- Do not include any other content or preamble
- Focus on actionable prediction guidance
- Be specific about patterns, magnitudes, and timeframes