CastFSR: A Fast–Slow–Reflect Agentic Reasoning Framework for Context-Aware Time Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Time series forecasting often requires combining historical observations with contextual information about future conditions. Numerical forecasters capture temporal patterns, while large language models (LLMs) can interpret contextual evidence and coordinate forecasting tools. Integrating these capabilities requires deciding when context should modify a forecast and checking whether the resulting revisions are consistent with temporal and domain constraints. We introduce CastFSR, an agentic framework organized as a Fast–Slow–Reflect workflow. Fast-thinking Forecasting profiles historical observations and selects a specialized forecaster to construct a numerical forecast prior. Slow Deliberative Reasoning retrieves relevant contextual evidence using context-specific look-back windows and refines the prior through contextual reasoning. Reflective Evaluation checks temporal, contextual, and domain consistency and revises the candidate forecast when violations are detected. CastFSR supports training-free orchestration with off-the-shelf LLM coordinators and a compact policy post-trained through supervised fine-tuning and multi-turn reinforcement learning. Experiments across diverse forecasting benchmarks show that CastFSR achieves strong performance against representative baselines. Module and training-stage ablations support the complementary roles of numerical priors, contextual reasoning, reflection, and post-training within the evaluated settings. [1]

1 Introduction

Time series forecasting (TSF) supports decision-making in energy markets (Lago et al., 2021), transportation systems (Guo et al., 2019; Zhang et al., 2020), and IT operations (Palaskar et al., 2024). In these domains, a time series is not merely a numeric sequence, but an interface to a system: recorded values reflect latent evolution under changing conditions (Holland, 1992; Ji et al., 2023), while future trends are shaped by historical observations and contextual features (Tao et al., 2026). These factors cannot always be inferred from historical observations alone, requiring methods to reason over temporal patterns and relevant contextual features (Ashok et al., 2024; Shao et al., 2024a).

TSF approaches span statistical models (Box et al., 2015), machine learning (Masini et al., 2023), deep neural networks (Cheng et al., 2025b; Huang et al., 2025), and foundation models (Woo et al., 2024). Despite strong temporal modeling, these approaches mainly rely on historical observations and offer limited support for evolving contexts (Eldele et al., 2021; Gao et al., 2024). Recent advances in LLMs have enabled context-aware forecasting by extending time series modeling toward semantic integration and reasoning. Existing studies represent temporal values in language-compatible forms to leverage pretrained knowledge (Jin et al., 2024). More recent work extends LLM-based forecasting toward reasoning-driven context-aware prediction with multi-step deliberation (Cheng et al., 2026; Wang et al., 2025). Coupling temporal dynamics with contextual features remains challenging (Jia et al., 2024): the forecasting workflow must select relevant evidence, determine how it should affect numerical predictions, and check the resulting revisions.

[1] Our code is available at `https://anonymous.4open.science/r/CastFSR-TS/`.

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.

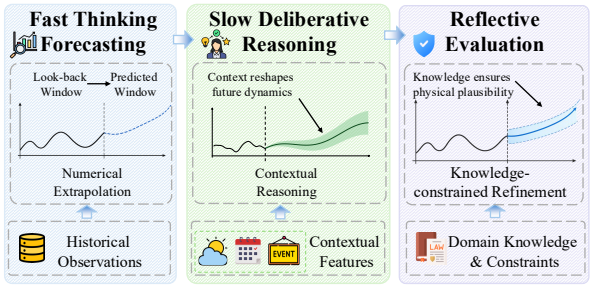

Figure 1: Illustration of CastFSR, which integrates numerical extrapolation, contextual reasoning, and knowledge-constrained reflection for TSF.

We formulate context-aware TSF as an agentic decision-making problem coordinating numerical forecasting, contextual evidence, and reflective validation. As shown in Figure 1, rather than treating LLMs as numerical predictors, we leverage their reasoning capabilities to orchestrate forecasting models, integrate contextual information, and refine forecast priors through evidence-grounded deliberation. However, realizing a Fast–Slow–Reflect forecasting agent introduces four challenges. First, the agent must construct reliable forecast priors by profiling historical observations and selecting suitable numerical models. Second, it must retrieve future-relevant evidence and reason about how contextual factors influence future dynamics, requiring adaptive retrieval and coordination among tools (Wu et al., 2026b). Third, it must validate forecasts against temporal regularities, contextual consistency, and domain constraints (Zhang et al., 2025; Cao et al., 2026). Finally, the workflow should support both off-the-shelf LLM coordinators and compact post-trained policies (Deng et al., 2022)

To address the challenges outlined above, we propose CastFSR, an agentic Fast–Slow–Reflect framework that formulates context-aware TSF as a sequential decision-making process. In fast-thinking forecasting, CastFSR profiles historical observations and autonomously selects suitable lightweight forecasters to construct a data-driven forecast prior. In slow deliberative reasoning, it retrieves future-relevant evidence from long-range contextual histories, adaptively identifies context-specific look-back windows, and reasons about how contextual factors reshape future dynamics to refine the forecast prior. In reflective evaluation, CastFSR iteratively evaluates and adjusts candidate forecasts to enforce temporal regularities, contextual consistency, and domain constraints. Importantly, CastFSR supports flexible deployment: it can be directly instantiated with off-the-shelf LLMs for training-free execution, or distilled into compact models through supervised fine-tuning and multi-turn reinforcement learning to internalize the orchestration and reasoning capabilities. Through this design, CastFSR moves beyond conventional pattern extrapolation by enabling forecasting systems to determine when historical patterns are reliable, when contextual evidence should modify predictions, and when further evaluation is required.

Our main contributions can be summarized as follows:

- We introduce a Fast–Slow–Reflect formulation that reframes context-aware time series forecasting as an agentic process integrating numerical prediction, contextual reasoning, and knowledge-constrained refinement.
- We propose CastFSR, which constructs forecast priors via autonomous forecaster selection, refines them through context retrieval and influence reasoning, and validates predictions with temporal, contextual, and domain consistency checks.
- We evaluate training-free LLM coordinators and a compact post-trained policy across diverse forecasting benchmarks, with ablations examining the workflow modules and the two training stages.

2 Related Work

This section reviews two complementary lines of related work: advances in time series forecasting and developments in agentic decision-making systems.

2.1 Time Series Forecasting

Time series forecasting has evolved from statistical models to machine learning, deep learning, and foundation models (Zhang et al., 2024). Early approaches, including exponential smoothing (Winters, 1960) and automatic statistical forecasting (Hyndman & Khandakar, 2008), provide

interpretable formulations for extrapolating historical patterns. Learning-based methods, such as recurrent networks (Wang et al., 2019), temporal convolutions (Cheng et al., 2025a), and Transformers (Shi et al., 2025), improve the modeling of complex temporal dependencies. More recently, foundation models learn transferable patterns from large time series corpora (Ansari et al., 2025). Despite these advances, most methods still infer future values through numerical pattern extrapolation. LLM-based approaches also explore prompting and modality alignment for forecasting (Xue & Salim, 2023; Jin et al., 2024). Multivariate methods model dependencies among correlated series, while context-aware approaches introduce contextual features (Liu et al., 2025a; Wang et al., 2026; Wu et al., 2026a). However, such information is supplied as predefined inputs within a fixed, single-pass inference process (Wu et al., 2021). Consequently, existing methods remain limited in retrieving context, assessing whether historical patterns remain applicable, and adaptively revising forecasts when contextual factors reshape future dynamics.

2.2 LLM-DRIVEN AGENTS

Agentic systems coordinate sequential decisions, tool use, and feedback (Varambally et al., 2026; Gwiazda et al., 2026). LLM-driven agents combine multi-step reasoning with external tools (Wei et al., 2022; Yao et al., 2022), while recent work explores reinforcement learning for reasoning-based forecasting (Luo et al., 2025). Related work also studies adaptive model selection and automated machine learning (Baratchi et al., 2024; Zhang et al., 2026; Kayaalp et al., 2026). TimeART (Wu et al., 2026b) studies tool-augmented time series question answering; TimeSeriesScientist (Zhao et al., 2025) automates forecasting through diagnostics, model selection, and validation. AlphaCast (Zhang et al., 2025) integrates contextual evidence with interactive reasoning and reflection, while TimeReasoner (Cheng et al., 2026) investigates inference-time temporal reasoning. Relative to these approaches, CastFSR organizes forecasting around a numerical prior generated by a specialized forecaster, selective context retrieval and reasoning, and explicit checks of temporal, contextual, and domain consistency. This separation assigns numerical extrapolation to forecasting tools and evidence-guided revision to the LLM. The same staged workflow supports both training-free orchestration and a compact policy post-trained with SFT and trajectory-level RL.

3 THE PROPOSED CASTFSR

3.1 OVERVIEW

As shown in Figure 2, CastFSR formulates context-aware TSF as a Fast–Slow–Reflect sequential decision-making process. A time series is viewed as an interface to systems, where future evolution is shaped by both historical patterns and contextual conditions. Fast-thinking forecasting constructs data-driven forecast priors using specialized forecasting models that capture temporal regularities, while slow deliberative reasoning leverages LLMs to retrieve contextual evidence and reason about its influence on future dynamics. Reflective evaluation validates forecasts against temporal regularities and domain constraints. The workflow can be instantiated with multiple general-purpose LLM coordinators or internalized by a compact policy model through two-stage training.

3.2 SEQUENTIAL FORECASTING FORMULATION

We formulate TSF as a *sequential decision-making problem*, where forecasting requires a sequence of decisions, including understanding temporal dynamics, selecting numerical priors, retrieving contextual evidence, reasoning about future impacts, and validating predictions. Let $\mathcal{I} = (\mathcal{D}_{\text{task}}, \mathcal{D}_{\text{domain}}, \mathbf{X}_{1:L}, \mathcal{H}_{\text{ctx}})$ denote the input, where $\mathcal{D}_{\text{task}}$ defines the forecasting target and horizon, $\mathcal{D}_{\text{domain}}$ provides domain attributes, $\mathbf{X}_{1:L}$ represents the historical look-back window, and $\mathcal{H}_{\text{ctx}}$ denotes available contextual histories or future-known contextual resources. The specified inference interface is target-blind: forecast-horizon targets are excluded from policy inputs. Context retrieval and reflection use the available evidence, rather than future target observations. Appendix C.3 identifies the implementation checks needed to verify this separation. The objective is to predict future values $\mathbf{Y}_{1:H}$ and generate $\hat{\mathbf{Y}}_{1:H}$. Instead of treating forecasting as a direct mapping from history to future values, we represent the forecasting workflow as a policy π_θ that interacts with a forecasting environment for at most K steps. At step k, the agent observes a state:

$$s_k = \Phi(\mathcal{I}, \mathcal{M}_k, \hat{\mathbf{Y}}_k^{\text{prior}}, \mathcal{C}_k), \tag{1}$$

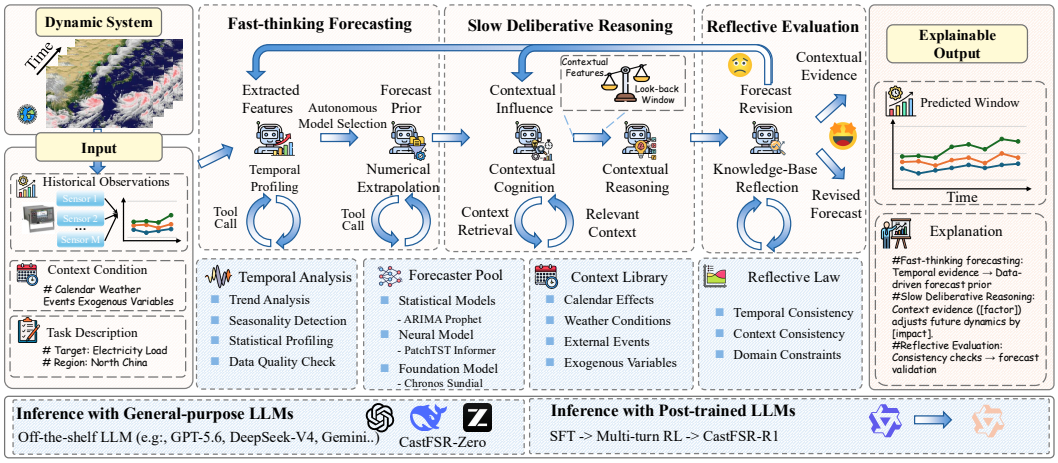

Figure 2: Overview of the CastFSR framework for agentic time series forecasting.

where $\mathcal{M}_k$ denotes the maintained memory state, $\hat{\mathbf{Y}}_k^{\text{prior}}$ is the current forecast prior, and $\mathcal{C}_k$ represents retrieved contextual evidence. The agent then selects an action:

$$a_k \sim \pi_\theta(\cdot|s_k), \qquad a_k \in \mathcal{A}, \tag{2}$$

where $\mathcal{A} = \mathcal{A}_{\text{fast}} \cup \mathcal{A}_{\text{slow}} \cup \mathcal{A}_{\text{reflect}} \cup \mathcal{A}_{\text{final}}$ includes fast-thinking forecasting, slow deliberative reasoning, reflective evaluation, and final-answer actions.

3.3 Fast-thinking Forecasting

The fast-thinking stage constructs a pattern-based forecast prior from the look-back window. It operationalizes numerical extrapolation: if past observations encode stable regularities of the underlying system, specialized forecasting models can use these regularities to generate numerical predictions. This stage combines temporal pattern analysis with adaptive numerical extrapolation, allowing CastFSR to use mature forecasters for value generation while keeping the LLM responsible for orchestration rather than unconstrained numerical generation. CastFSR is equipped with a modular toolkit:

$$\mathcal{T} = \mathcal{T}_{\text{feat}} \cup \mathcal{T}_{\text{pred}}, \tag{3}$$

where $\mathcal{T}_{\text{feat}}$ contains feature-extraction tools and $\mathcal{T}_{\text{pred}}$ contains forecasting models. In the feature-extraction phase, the agent invokes complementary tools to characterize the input series. Specifically, trend analysis extracts temporal evolution patterns, seasonality detection captures periodic behaviors, statistical profiling summarizes distributional properties, and data-quality analysis identifies irregularities. These features provide structured evidence for model selection and prevent decisions based solely on raw observations. After feature extraction, CastFSR selects a forecasting model $m \in \mathcal{P}$ through a unified prediction interface, where the implemented pool is $\mathcal{P} = \{\text{ARIMA}, \text{DLinear}, \text{PatchTST}, \text{iTransformer}, \text{Chronos-2}\}$. The selected predictor generates a forecast prior:

$$\hat{\mathbf{Y}}^{\text{prior}} = T_{\text{pred}}(\mathbf{X}_{1:L}; m), \qquad m \sim \pi_\theta(\cdot|s_k). \tag{4}$$

By treating predictors as tools rather than fixed components, CastFSR adaptively routes different inputs to models and constructs data-driven priors from historical patterns.

3.4 Slow Deliberative Reasoning

Slow deliberative reasoning examines whether the pattern-based prior remains reliable under future-relevant contextual conditions. It models context-aware temporal reasoning by recognizing that future dynamics may deviate from historical extrapolation due to external factors such as weather, calendar effects, and operational constraints. Unlike fast-thinking forecasting, slow deliberative reasoning selectively reasons about when and how contexts influence future evolution rather than replacing all forecasts. It identifies relevant factors, determines retrieval ranges, and estimates their

impacts on the forecast prior. Given the current memory, temporal features, domain description, and forecast prior, the agent performs contextual cognition, defined here as evidence retrieval and relevance assessment, to obtain horizon-aligned evidence. Instead of using a fixed look-back window, CastFSR adaptively searches long-range contextual histories and selects context-specific windows, as different factors operate at different temporal scales. The retrieved evidence is then used to reason about future changes and refine the prior:

$$\mathcal{C}_k = T_{\mathrm{ctx}}(\mathcal{D}_{\mathrm{task}}, \mathcal{D}_{\mathrm{domain}}, \mathbf{X}_{1:L}, \mathcal{H}_{\mathrm{ctx}}, \hat{\mathbf{Y}}^{\mathrm{prior}}, \mathcal{M}_k; \mathcal{W}_k), \tag{5}$$

where T_{ctx} denotes context retrieval and $\mathcal{W}_k$ contains adaptive look-back windows for different context types. The output $\mathcal{C}_k$ is stored as structured evidence.

After contextual cognition, CastFSR estimates the contextual influence by determining whether retrieved evidence is relevant to the forecast horizon, how it changes future dynamics, and which parts of the numerical prior should be revised. If the evidence is weak, redundant with historical patterns, or outside the effective context range, the agent preserves the original prior. Otherwise, it reasons about the direction, magnitude, and temporal scope of the contextual influence. The agent then performs *contextual reasoning* to transform impact assessment into a candidate forecast. Instead of generating values from scratch, the deliberation module refines the numerical prior by applying context-grounded adjustments to relevant timestamps or segments:

$$\hat{\mathbf{Y}}^{\mathrm{cand}} = G_\theta(\hat{\mathbf{Y}}^{\mathrm{prior}}, \mathcal{C}_k, \mathcal{M}_k), \tag{6}$$

where G_θ denotes the deliberation process. This design couples temporal patterns with contextual information: the prior provides a numerical starting point, while retrieved evidence guides when and where it should be revised. The resulting candidate forecast, together with contextual evidence and revision rationale, is passed to reflective evaluation.

3.5 Reflective Evaluation

Reflective evaluation verifies the candidate forecast before output. It enforces the constraint-aware principle that forecasts should be not only accurate but also consistent with temporal regularities, domain knowledge, and operational constraints. The agent evaluates $\hat{\mathbf{Y}}^{\mathrm{cand}}$ through three consistency checks. Temporal consistency examines trends, seasonal patterns, turning points, and continuity with historical observations. Contextual consistency verifies whether retrieved evidence supports forecast revisions and whether their timing and direction align with the selected context windows. Domain consistency checks validity constraints, including units, non-negativity, capacity limits, timestamp correctness, and task-specific requirements. The evaluation process produces:

$$\mathcal{E}_k = E_\theta(\hat{\mathbf{Y}}^{\mathrm{cand}}, \hat{\mathbf{Y}}^{\mathrm{prior}}, \mathcal{C}_k, \mathcal{M}_k). \tag{7}$$

Based on $\mathcal{E}_k$, the agent decides whether to accept or refine the candidate forecast. When inconsistencies are localized, reflective evaluation applies targeted corrections rather than regenerating the entire horizon, preserving reliable predictions while repairing invalid segments. A forecast is accepted only after satisfying temporal and format requirements:

$$\hat{\mathbf{Y}}_{1:H} = F_\theta(\hat{\mathbf{Y}}^{\mathrm{cand}}, \mathcal{E}_k, \mathcal{M}_k). \tag{8}$$

The final output includes both the predicted window and an interpretable trajectory report describing the evidence used, constraints checked, and revision rationale.

3.6 Training-free and Post-trained Instantiation

The workflow separates forecasting logic from the policy that coordinates it. In the training-free setting, an off-the-shelf LLM follows the stage-constrained action interface and directly orchestrates feature profiling, model selection, context retrieval, and reflective evaluation. This mode requires no task-specific update to the LLM coordinator. The workflow can be instantiated with multiple general-purpose LLM coordinators, as evaluated in Appendix D.

We further investigate whether a compact coordinator can learn the workflow through two-stage post-training. First, SFT learns executable Fast–Slow–Reflect behavior from CastFSR-Zero trajectories generated for the training tasks. The validation removes trajectories with illegal tool calls,

incorrect stage order, ground-truth leakage, invalid timestamps, or incomplete forecasts. Second, multi-turn RL optimizes complete decision trajectories under delayed forecasting feedback. For a group of trajectories $\{\tau_i\}_{i=1}^{G}$, we normalize their episode rewards as:

$$A_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \tag{9}$$

where $R_i = R(\tau_i)$ is the episode reward, μ_R and σ_R are its group mean and standard deviation, and $\epsilon > 0$ stabilizes the denominator. We then optimize the clipped GRPO objective (Shao et al., 2024b). The reward evaluates the complete workflow trajectory. Output validity concerns forecast format, executable tool calls, and horizon completeness. Numerical feedback evaluates forecast error against reference targets, while structural feedback assesses agreement in trend, seasonality, and change points over the forecast horizon. These reference-dependent signals provide offline training supervision; reflective evaluation instead checks consistency using the target-blind policy state. The exact scoring and aggregation rules require specification in Appendix C.3. SFT trains the policy to follow executable workflows, while RL optimizes its decisions using trajectory-level forecasting feedback.

4 Experiments

In this section, we first introduce the experimental settings and then present experimental results, ablation studies, and case analyses to evaluate the proposed framework.

4.1 Experimental Settings

4.1.1 Datasets

Table 1 summarizes the statistics of the real-world time series datasets used in our experiments. The ETT benchmarks (Zhou et al., 2021) contain electricity-transformer measurements collected at hourly (ETTh1 and ETTh2) and 15-minute (ETTm1 and ETTm2) resolutions, providing multivariate series with long-range temporal dependencies. Wind records wind-power generation together with meteorological covariates at a 15-minute resolution. For short-term forecasting, we use five hourly electricity-price datasets from the EPF benchmark (Lago et al., 2021): Belgium (BE), Germany (DE), France (FR), the Nordic market (NP), and PJM. Together, these benchmarks span different sampling frequencies and dynamics across heterogeneous real-world forecasting scenarios. Detailed dataset descriptions are provided in Appendix A.

Table 1: Statistics of the evaluated datasets.

Setting	Dataset	Domain	Length	Variables	Frequency
Long-term	ETTh1 & ETTh2	Electricity	17,420	7	1 hour
	ETTm1 & ETTm2	Electricity	69,680	7	15 mins
	Wind	Energy	48,673	7	15 mins
Short-term	BE	Energy	14,496	3	1 hour
	DE	Energy	14,496	3	1 hour
	FR	Energy	14,496	3	1 hour
	NP	Energy	14,496	3	1 hour
	PJM	Energy	14,496	3	1 hour

Baselines. We compare CastFSR against baselines in Table 2. Statistical methods include ARIMA (Hyndman & Khandakar, 2008) and Prophet (Taylor & Letham, 2018), which capture temporal patterns through classical modeling. Deep learning approaches include DLinear (Zeng et al., 2023), ConvTimeNet (Cheng et al., 2025b), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), and TimeXer (Wang et al., 2024), covering MLP-, CNN-, and Transformer-based architectures. Foundation models include TimesFM (Das et al., 2024) and Sundial (Liu et al., 2025b). LLM-based methods include OFA (Zhou et al., 2023), Time-LLM (Jin et al., 2024), TokenCast (Tao et al., 2025), S^2IP-LLM (Pan et al., 2024), TimeReasoner (Cheng et al., 2026), and PromptCast (Xue & Salim, 2023), adapting LLMs via alignment, tokenization, prompting, and reasoning. We further compare agentic systems TimeSeriesScientist (Zhao et al., 2025) and AlphaCast (Zhang et al., 2025).

Implementation Details. For training-free inference, we instantiate CastFSR-Zero with DeepSeek-V4-Flash (Xu et al., 2026) as the reasoning engine. We further develop CastFSR-R1 by fine-tuning Qwen3-4B with SFT and RL on distilled forecasting trajectories using 16 Ascend NPUs. Deep learning baselines follow their official configurations to ensure a fair comparison. All methods use the same look-back and forecasting horizons: 96/96 for long-term tasks and 168/24 for short-term tasks. We report mean squared error (MSE) and mean absolute error (MAE) as the evaluation metrics. Additional implementation details are provided in Appendix C.

Table 2: Overall forecasting performance on benchmark datasets. Lower values indicate better performance. The best results are highlighted in bold, and the second-best are underlined.

Model	Long-term forecasting										Short-term forecasting									
	ETTh1		ETTh2		ETTm1		ETTm2		Wind		BE		DE		FR		NP		PJM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ARIMA	0.110	0.243	0.443	0.505	**0.055**	0.169	0.376	0.429	3.682	1.361	1.978	0.814	3.269	1.076	6.410	1.189	1.798	0.801	0.907	0.710
Prophet	0.630	0.546	0.942	0.710	0.091	0.216	0.216	0.313	7.874	2.094	2.321	0.918	2.103	0.986	6.831	1.280	0.707	0.583	0.410	0.500
PatchTST	0.114	0.247	0.251	0.375	0.061	0.184	0.133	0.245	2.556	1.212	1.905	0.647	0.983	0.666	5.021	0.740	0.454	0.441	0.190	0.323
iTransformer	0.102	0.243	0.257	0.390	0.058	0.180	0.138	0.256	2.657	1.196	1.684	0.569	1.298	0.697	6.066	0.869	0.373	0.404	0.238	0.344
ConvTimeNet	0.097	0.236	0.252	0.380	0.056	0.174	0.134	0.247	2.544	1.206	1.482	0.581	0.964	0.665	5.376	0.800	0.393	0.411	0.226	0.342
TimeXer	0.108	0.248	0.255	0.386	0.058	0.185	0.136	0.249	2.479	1.214	1.437	0.502	1.009	0.708	5.578	0.827	0.412	0.416	0.197	0.318
DLinear	0.103	0.249	0.252	0.386	0.061	0.184	0.138	0.253	2.441	1.149	1.697	0.677	1.000	0.699	6.369	1.000	0.431	0.446	0.286	0.387
TimesFM	0.104	0.237	0.266	0.388	0.059	0.176	0.353	0.412	3.171	1.273	1.349	0.510	0.967	0.660	4.580	0.651	0.317	0.361	0.207	0.324
Sundial	0.106	0.247	0.273	0.391	0.058	0.172	0.213	0.331	2.780	1.244	1.478	0.513	1.155	0.700	6.105	0.682	0.362	0.378	0.201	0.324
OFA	0.100	0.242	0.247	0.378	0.058	0.179	**0.127**	0.242	2.708	1.250	1.522	0.610	1.065	0.699	5.659	0.868	0.383	0.420	0.252	0.369
Time-LLM	0.097	0.240	0.250	0.378	0.059	0.183	0.138	0.254	2.648	1.230	1.758	0.701	1.025	0.707	6.450	0.971	0.422	0.447	0.253	0.368
PromptCast	0.123	0.322	0.299	0.372	0.085	0.241	0.149	0.296	4.404	1.200	1.870	0.637	1.229	0.708	7.302	1.173	0.565	0.449	0.316	0.454
TokenCast	0.121	0.277	0.385	0.472	**0.055**	0.177	0.137	0.259	2.926	1.265	1.485	0.547	0.973	0.702	6.363	0.961	0.412	0.411	0.215	0.319
S^2IP-LLM	0.119	0.277	0.290	0.424	0.056	0.176	0.129	0.249	2.638	1.184	1.591	0.567	1.315	0.722	5.765	0.888	0.389	0.391	0.237	0.339
TimeReasoner	0.126	0.283	0.375	0.425	0.072	0.205	0.173	0.295	3.553	1.390	1.299	0.466	1.082	0.760	4.644	0.650	0.368	0.386	0.207	0.328
TimeSeriesScientist	0.213	0.290	0.498	0.505	0.092	0.205	0.150	0.300	5.907	1.680	2.057	0.712	1.028	0.673	5.977	1.033	0.606	0.476	0.302	0.439
AlphaCast	0.123	0.225	0.279	0.372	0.075	0.185	0.131	0.271	2.375	**0.856**	1.484	0.510	1.043	0.662	5.318	0.740	0.316	0.365	0.176	0.292
CastFSR-Zero	0.081	0.212	0.268	0.361	**0.055**	**0.168**	**0.127**	0.238	**1.596**	0.886	0.960	0.386	0.393	0.408	**3.939**	**0.501**	0.221	0.277	0.140	0.268
CastFSR-R1	**0.077**	**0.210**	**0.242**	**0.351**	**0.055**	**0.168**	**0.127**	**0.237**	1.757	0.918	**0.927**	**0.375**	**0.386**	**0.405**	4.032	0.502	**0.172**	**0.247**	**0.133**	**0.260**

Table 3: Ablation study of Fast–Slow–Reflect modules. We compare CastFSR-Zero against variants removing fast-thinking forecasting, slow deliberative reasoning, or reflective Evaluation.

Variant	ETTh1		ETTh2		ETTm1		ETTm2		Wind		BE		DE		FR		NP		PJM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o Fast-thinking	0.089	0.224	0.339	0.403	0.061	0.184	0.176	0.283	2.403	1.111	1.225	0.455	0.481	0.451	4.251	0.534	0.334	0.336	0.168	0.300
w/o Slow Deliberative Reasoning	0.086	0.216	0.298	0.373	**0.055**	0.169	0.157	0.265	2.310	1.054	1.147	0.431	0.438	0.426	**3.936**	0.504	0.292	0.301	0.149	0.273
w/o Reflective Evaluation	0.086	0.216	0.320	0.388	**0.055**	0.169	0.139	0.255	2.032	0.997	1.011	0.398	0.421	0.416	3.952	0.505	0.284	0.306	0.150	0.274
CastFSR-Zero	**0.081**	**0.212**	**0.268**	**0.361**	**0.055**	**0.168**	**0.127**	**0.238**	**1.596**	**0.886**	**0.960**	**0.386**	**0.393**	**0.408**	3.939	**0.501**	**0.221**	**0.277**	**0.140**	**0.268**

4.2 Main Results

As shown in Table 2, CastFSR achieves strong performance across both long-term and short-term forecasting benchmarks, obtaining the best or second-best results on most reported metrics. Its consistent gains across diverse forecasting settings suggest that combining numerical forecast priors with contextual reasoning and reflective validation can effectively improve forecasting performance beyond conventional numerical and LLM-based baselines. Comparing the two variants, CastFSR-R1 generally outperforms CastFSR-Zero, indicating that supervised fine-tuning and reinforcement learning can effectively internalize the multi-stage forecasting workflow into a compact policy. Meanwhile, CastFSR-Zero remains highly competitive and performs better in several cases, demonstrating that training-free orchestration can also coordinate numerical forecasting, contextual evidence, and reflective refinement effectively. Overall, these results support both deployment modes and highlight their complementary strengths across different forecasting scenarios.

4.3 Ablation of Workflow Modules

Table 3 evaluates each component by removing fast-thinking forecasting, slow deliberative reasoning, or reflective evaluation from CastFSR-Zero. Removing a component generally degrades performance, with ties on ETTm1 MSE and a small improvement on FR MSE without slow deliberative reasoning. Removing fast-thinking forecasting causes the largest degradation among the ablated variants on most metrics, showing that data-driven forecast priors provide the foundation for subsequent reasoning. Without this prior, the agent relies more on LLM generation and produces less accurate forecasts. Removing slow deliberative reasoning also reduces performance on most metrics, highlighting the importance of contextual evidence retrieval and reasoning. Removing reflective evaluation generally increases error, supporting the use of temporal consistency and domain constraint checks.

4.4 Analysis of Model Selection

Figure 3 shows that model-selection frequencies vary across diagnostic features and datasets in the Fast stage, suggesting that the selector does not rely on a single forecaster uniformly across instances. Instead, CastFSR conditions model selection on the observed temporal profile. As shown in

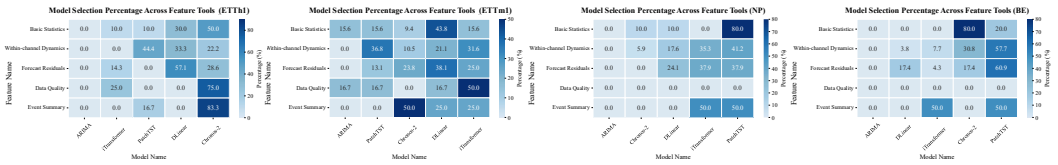

Figure 3: Feature-conditioned model selection patterns. Model-selection frequencies (%) conditioned on diagnostic features. Rows denote feature tools and columns denote candidate models; darker cells indicate routing frequency.

Table 4: Intermediate model-selection-stage results comparing the best fixed forecasting baseline with feature-conditioned selection under the same stage-specific protocol.

Strategy	ETTh1		ETTm1		BE		NP	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Best Fixed Baselines	0.916	**0.213**	0.063	0.305	1.164	0.496	0.386	0.046
CastFSR Selection	**0.890**	0.225	**0.057**	**0.176**	**1.099**	**0.448**	**0.338**	**0.043**
Named fixed model	Chronos-2		iTransformer		PatchTST		PatchTST	

Table 4, the resulting selection strategy achieves lower error than the best fixed baseline on seven of the eight reported metrics, with ETTh1 MAE as the only exception. These results provide evidence that feature-conditioned selection can yield stronger numerical forecast priors under the evaluated settings, which are then passed to the subsequent Slow reasoning and Reflective Evaluation stages.

4.5 Analyzing Adaptive Slow Deliberative Reasoning

Figure 4 illustrates how CastFSR refines forecast priors using contextual evidence retrieved from heterogeneous external sources and future conditions that are potentially relevant to the target horizon. In these representative examples, it selectively retains the prior where historical patterns remain sufficiently informative and revises only those segments where contextual signals suggest meaningful deviations, while avoiding unnecessary changes to stable regions. The examples clearly illustrate selective, evidence-guided contextual reasoning, with the scope and magnitude of revisions varying adaptively across different parts of the forecast horizon.

4.6 Effect of Training Strategies

Figure 5 compares CastFSR-R1 with variants trained without SFT or without RL across representative forecasting settings. The full model consistently achieves the best or tied-best results on the plotted metrics, indicating the overall effectiveness of the complete post-training pipeline. The two variants tie the full model on ETTm1 MSE, and the variant without SFT also ties on ETTm1 MAE despite these isolated cases overall. The remaining differences support complementary benefits from supervised workflow imitation and trajectory-level reinforcement learning.

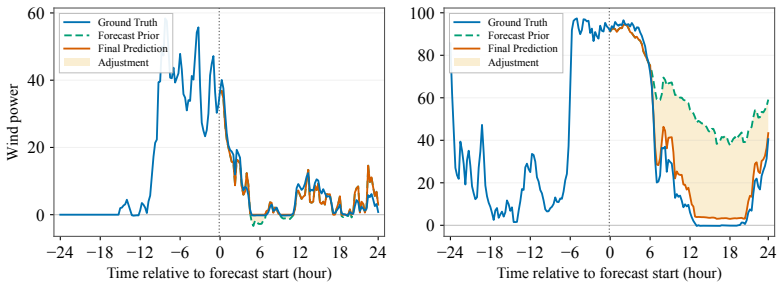

Figure 4: Examples of forecast prior refinement through slow deliberative reasoning.

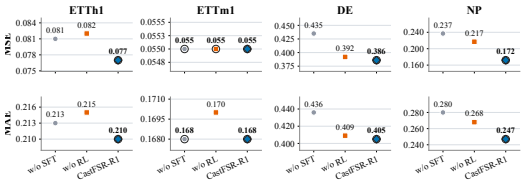

Figure 5: Effect of removing SFT or RL from CastFSR-R1 during training.

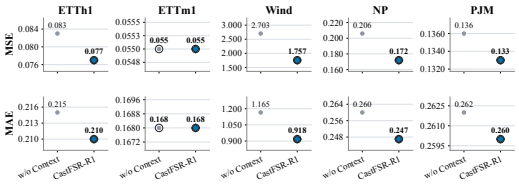

Figure 6: Comparison of w/o Context and CastFSR across datasets.

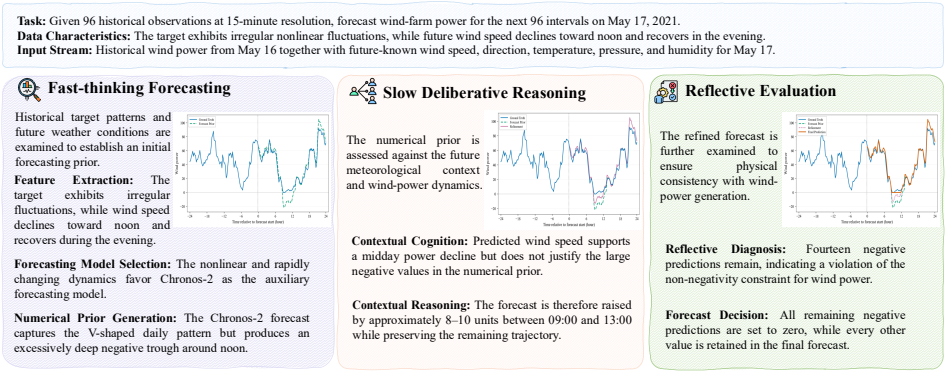

Figure 7: Supplementary case study of CastFSR for context-aware TSF.

4.7 Analysis of Context Utilization

Figure 6 compares CastFSR with its w/o Context variant across five datasets. Incorporating contextual information improves both MSE and MAE on ETTh1, Wind, NP, and PJM, while ETTm1 remains unchanged at the reported precision. The largest improvement is observed on Wind, with consistent gains also appearing on NP and ETTh1 and smaller improvements on PJM. Overall, these results suggest that contextual reasoning provides additional forecasting benefits beyond the numerical prior under the evaluated settings.

4.8 Case Study Analysis

Figure 7 presents a wind power example. Fast-thinking Forecasting selects Chronos-2 to construct a forecast prior that captures the daily pattern but contains an excessively negative midday trough. Slow Deliberative Reasoning finds that the meteorological context supports a midday decline, but not the depth of the negative trough in the numerical prior. It raises the affected segment while retaining the remaining trajectory. Reflective Evaluation then enforces the non-negativity constraint by setting the remaining negative predictions to zero. The example illustrates distinct roles for context-guided revision and domain-constraint checks.

5 Conclusion

In this work, we proposed CastFSR, an agentic TSF framework that reformulates context-aware forecasting as a Fast–Slow–Reflect sequential decision-making process. By coordinating Fast-thinking Forecasting, Slow Deliberative Reasoning, and Reflective Evaluation, CastFSR offers a principled alternative to single-pass numerical extrapolation. The framework supports both training-free execution with off-the-shelf LLMs and compact deployment through supervised fine-tuning and multi-turn reinforcement learning. Experiments on ten energy forecasting benchmarks show competitive performance across different temporal resolutions and forecasting settings. Ablations support the complementary roles of forecast prior construction, contextual reasoning, and constraint checks within this evaluated scope.

AI USE DISCLOSURE

Generative AI tools were used to assist with language editing, improving readability, literature search and organization, and limited code debugging and formatting. They were not used to generate experimental data, determine the core research methodology, formulate mathematical claims, or independently draw scientific conclusions. All AI-assisted text, code, references, and suggestions were manually reviewed and verified by the authors. The authors take full responsibility for the final content, experimental results, and claims presented in this work.

REFERENCES

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Arjun Ashok, Andrew Robert Williams, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. Context is key: a benchmark for forecasting with essential textual information. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

Mitra Baratchi, Can Wang, Steffen Limmer, Jan N Van Rijn, Holger Hoos, Thomas Bäck, and Markus Olhofer. Automated machine learning: past, present and future. *Artificial intelligence review*, 57(5):122, 2024.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition, 2015.

Defu Cao, Wen Ye, Yizhou Zhang, Sam Griesemer, and Yan Liu. Pinfdit: Energy-based physics-informed diffusion transformers for general-purpose time series tasks. In *International Conference on Learning Representations*, 2026.

Mingyue Cheng, Zhiding Liu, Xiaoyu Tao, Qi Liu, Jintao Zhang, Tingyue Pan, Shilong Zhang, Panjing He, Xiaohan Zhang, Daoyu Wang, et al. A comprehensive survey of time series forecasting: Concepts, challenges, and future directions. *Authorea Preprints*, 2025a.

Mingyue Cheng, Jiqian Yang, Tingyue Pan, Qi Liu, Zhi Li, and Shijin Wang. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 171–180, 2025b.

Mingyue Cheng, Jiahao Wang, Daoyu Wang, Xiaoyu Tao, Qi Liu, and Enhong Chen. Can slow-thinking llms reason over time? empirical studies in time series forecasting. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 99–110, 2026.

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

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7665–7680, 2022.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. In *IJCAI*, pp. 2352–2359, 2021.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24, 2024.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Malgorzata Gwiazda, Yifu Cai, Mononito Goswami, Arjun Choudhry, and Artur Dubrawski. Timeseriesexamagent: Creating time series reasoning benchmarks at scale. In *International Conference on Learning Representations*, 2026.

John H Holland. Complex adaptive systems. *Daedalus*, 121(1):17–30, 1992.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. In *International Conference on Learning Representations*, 2025.

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

Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4356–4364, 2023.

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

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. In *The Twelfth International Conference on Learning Representations*, 2024.

Mert Kayaalp, Caner Turkmen, Oleksandr Shchur, Pedro Mercado, Abdul Fatir Ansari, Michael Bohlke-Schneider, and Yuyang Wang. Test-time efficient pretrained model portfolios for time series forecasting. In *International Conference on Learning Representations*, 2026.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025a.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *International Conference on Machine Learning*, pp. 39295–39317. PMLR, 2025b.

Yucong Luo, Yitong Zhou, Mingyue Cheng, Jiahao Wang, Daoyu Wang, Tingyue Pan, and Jintao Zhang. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630*, 2025.

Ricardo P Masini, Marcelo C Medeiros, and Eduardo F Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Santosh Palaskar, Vijay Ekambaram, Arindam Jati, Neelamadhav Gantayat, Avirup Saha, Seema Nagar, Nam H Nguyen, Pankaj Dayama, Renuka Sindhgatta, Prateeti Mohapatra, et al. Automixer for improved multivariate time-series forecasting on business and it observability data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 22962–22968, 2024.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. S^2ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17 (9):2363–2377, 2024. doi: 10.14778/3665844.3665863.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiaoyu Tao, Shilong Zhang, Mingyue Cheng, Daoyu Wang, Tingyue Pan, Bokai Pan, Changqing Zhang, and Shijin Wang. From values to tokens: An llm-driven framework for context-aware time series forecasting via symbolic discretization. *arXiv preprint arXiv:2508.09191*, 2025.

Xiaoyu Tao, Mingyue Cheng, Ze Guo, Shuo Yu, Yaguo Liu, Qi Liu, and Shijin Wang. Memcast: Memory-driven time series forecasting with experience-conditioned reasoning. *arXiv preprint arXiv:2602.03164*, 2026.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Sumanth Varambally, Marshall Fisher, Jas Thakker, Yiwei Chen, Zhirui Xia, Yasaman Jafari, Ruijia Niu, Manas Jain, Veeramakali Vignesh Manivannan, Zachary Novack, Luyu Han, Srikar Eranky, Salva Rühling Cachay, Taylor Berg-Kirkpatrick, Duncan Watson-Parris, Yi-An Ma, and Rose Yu. Zephyrus: An agentic framework for weather science. In *International Conference on Learning Representations*, 2026.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, 16(6):1–87, 2025.

Siyuan Wang, Peng Chen, Yihang Wang, Wanghui Qiu, Chenjuan Guo, Bin Yang, and Yang Shu. Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. In *International Conference on Learning Representations*, 2026.

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

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pp. 6607–6617. PMLR, 2019.

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.

Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

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

Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Chenjuan Guo. Aurora: Towards universal generative multimodal time series forecasting. In *International Conference on Learning Representations*, 2026a.

Xingjian Wu, Junkai Lu, Zhengyu Li, Xiangfei Qiu, Jilin Hu, Chenjuan Guo, Christian S Jensen, and Bin Yang. Timeart: Towards agentic time series reasoning via tool-augmentation. *arXiv preprint arXiv:2601.13653*, 2026b.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 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.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE transactions on pattern analysis and machine intelligence*, 46(10):6775–6794, 2024.

Tengxue Zhang, Biao Ouyang, Yang Shu, Xinyang Chen, Chenjuan Guo, and Bin Yang. Swiftts: A swift selection framework for time series pre-trained models via multi-task meta-learning. In *International Conference on Learning Representations*, 2026.

Weijia Zhang, Hao Liu, Yanchi Liu, Jingbo Zhou, and Hui Xiong. Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1186–1193, 2020.

Xiaohan Zhang, Tian Gao, Mingyue Cheng, Bokai Pan, Ze Guo, Yaguo Liu, and Xiaoyu Tao. Alphacast: A human wisdom-llm intelligence co-reasoning framework for interactive time series forecasting. *arXiv preprint arXiv:2511.08947*, 2025.

Haokun Zhao, Xiang Zhang, Jiaqi Wei, Yiwei Xu, Yuting He, Siqi Sun, and Chenyu You. Timeseriesscientist: A general-purpose ai agent for time series analysis. *arXiv preprint arXiv:2510.01538*, 2025.

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, 2021.

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 Detailed Dataset Descriptions

We evaluate CastFSR on ten real-world energy forecasting datasets, including long-term time series forecasting (LTSF) benchmarks and short-term electricity price forecasting (EPF) benchmarks.

A.1 Long-term Time Series Forecasting

For long-horizon forecasting tasks, we consider benchmarks with complex temporal dependencies and diverse temporal patterns under challenging forecasting conditions.

- **ETT (Electricity Transformer Temperature):** The ETT benchmark (Zhou et al., 2021) provides two years of electricity transformer measurements, with Oil Temperature (OT) as the forecasting target. As a key indicator of transformer operating conditions, OT exhibits complex temporal dependencies and long-term variations.
  - **ETTh1:** Sampled at a 1-hour frequency. It contains the target OT and six distinct power load features (HUFL, HULL, MUFL, MULL, LUFL, LULL).
  - **ETTh2:** Sampled at a 1-hour frequency from a second transformer, with the same target and load feature structure settings.
  - **ETTm1:** Sampled at a 15-minute frequency, containing the same set of load features but with higher temporal resolution, capturing more granular fluctuations.
  - **ETTm2:** Sampled at a 15-minute frequency from a second transformer, matching the ETTm1 feature structure configuration.
- **Wind:** We use the Wind dataset distributed with the TFB forecasting benchmark (Qiu et al., 2024). The forecasting target is wind-power generation, and the six contextual variables are predicted humidity, pressure, temperature, wind direction, wind speed, and observed wind speed. The dataset contains 48,673 observations sampled at 15-minute intervals.

A.2 Short-term Electricity Price Forecasting

We utilize five widely adopted datasets from the electricity price forecasting benchmark established by Lago et al. (Lago et al., 2021). These datasets represent different regional day-ahead markets, each with a sampling frequency of 1 hour. The task is to forecast day-ahead prices (24 steps) using historical observations and exogenous variables.

- **BE (Belgium):** Represents the Belgian electricity market, including hourly electricity prices supplemented with national load forecasts and generation forecasts from the neighboring French grid.
- **DE (Germany):** Represents the German electricity market. The forecasting target is Germany's hourly electricity price, and the two exogenous variables are wind power and the Amprion zonal load, giving three variables in total.
- **FR (France):** Represents the French electricity market, comprising hourly electricity prices, grid load forecasts, and domestic generation forecasts.
- **NP (Nord Pool):** Represents the Nordic electricity market, containing hourly electricity prices along with exogenous forecasts of grid load and wind power generation.
- **PJM (Pennsylvania-New Jersey-Maryland):** Sourced from the PJM Interconnection in the United States, this dataset contains zonal electricity prices for the Commonwealth Edison (COMED) region, together with system-wide load forecasts and zonal load forecasts.

B Detailed Baseline Descriptions

We evaluate CastFSR against a diverse set of representative baselines, ranging from classical statistical methods to state-of-the-art foundation models.

Table 5: Summary of real-world time series datasets used in the experiments. Variable counts include the forecasting target.

Setting	Dataset	Domain	Length	Variables	Frequency	Description
Long-term	**ETTh1**	Electricity	17,420	7	1 hour	Transformer oil temperature and 6 power load features.
Long-term	**ETTh2**	Electricity	17,420	7	1 hour	Transformer oil temperature and 6 power load features.
Long-term	**ETTm1**	Electricity	69,680	7	15 mins	Transformer oil temperature and 6 power load features.
Long-term	**ETTm2**	Electricity	69,680	7	15 mins	Transformer oil temperature and 6 power load features.
Long-term	**Wind**	Energy	48,673	7	15 mins	Target with meteorological forecasts and observed wind speed.
Short-term	**BE**	Energy	14,496	3	1 hour	Belgian market prices with load and generation forecasts.
Short-term	**DE**	Energy	14,496	3	1 hour	German market prices with wind-power and Amprion zonal-load covariates.
Short-term	**FR**	Energy	14,496	3	1 hour	French market prices with load and generation forecasts.
Short-term	**NP**	Energy	14,496	3	1 hour	Nord Pool market prices with load and wind forecasts.
Short-term	**PJM**	Energy	14,496	3	1 hour	PJM (ComEd) prices with system and zonal load forecasts.

Statistical Methods.

- **ARIMA** (Hyndman & Khandakar, 2008): A classic statistical method that models temporal patterns using autoregression, differencing, and moving averages to capture linear dependencies.
- **Prophet** (Taylor & Letham, 2018): An additive regression model designed for business time series, which effectively decomposes data into trends, seasonality, and holiday effects.

Deep Learning Baselines.

- **DLinear** (Zeng et al., 2023): A simple yet effective MLP-based model that utilizes a decomposition layer to handle trend and seasonal components separately.
- **ConvTimeNet** (Cheng et al., 2025b): A deep hierarchical fully convolutional network that captures multi-scale temporal patterns through adaptive segmentation and deformable patching.
- **PatchTST** (Nie et al., 2023): A Transformer-based model that introduces channel independence and patch-based tokenization to capture local semantic information and reduce computational complexity.
- **iTransformer** (Liu et al., 2024): An inverted Transformer architecture that embeds the whole time series of each variate as a token and applies attention mechanisms across multivariate channels.
- **TimeXer** (Wang et al., 2024): An advanced Transformer framework designed to empower time series forecasting by incorporating and aligning exogenous variables.

Foundation Models.

- **TimesFM** (Das et al., 2024): A decoder-only foundation model developed by Google, pretrained on a massive corpus of over 100 billion real-world and synthetic time points. It utilizes a patch-based architecture to capture long-range temporal dependencies and supports zero-shot forecasting across multiple benchmarks.
- **Sundial** (Liu et al., 2025b): A family of pretrained time series foundation models designed to provide strong zero-shot and transfer forecasting performance.

LLM-based Methods.

- **OFA (One Fits All)** (Zhou et al., 2023): A generalized framework that leverages frozen pre-trained language models (e.g., GPT-2) for time series analysis. It adapts the LLM to forecasting tasks by fine-tuning only specific layers (such as positional embeddings and normalization layers) while keeping the self-attention and feedforward networks frozen.
- **Time-LLM** (Jin et al., 2024): A comprehensive framework that aligns time series modalities with the text space of LLMs using reprogramming techniques and prompt-as-prefix strategies.
- **TokenCast** (Tao et al., 2025): A value-tokenization method that converts numerical time series into language-compatible token sequences for LLM-based forecasting.

- **S^{2}IP-LLM** (Pan et al., 2024): An LLM-based forecasting method that adapts pretrained language models to time series through structured semantic and instance-level prompting.
- **TimeReasoner** (Cheng et al., 2026): A prompting-based approach that uses pretrained slow-thinking LLMs for inference-time reasoning about temporal patterns and contextual features.
- **PromptCast** (Xue & Salim, 2023): A prompt-based forecasting method that casts time series prediction into a language-model prompting task.

Agentic Forecasting Systems.

- **TimeSeriesScientist** (Zhao et al., 2025): An LLM-driven scientific agent for time series analysis and forecasting.
- **AlphaCast** (Zhang et al., 2025): A human–LLM co-reasoning framework for TSF, enabling iterative collaboration between human and LLM-based forecasting reasoning.

C Detailed Implementation Settings

In this section, we report the model, training, and evaluation settings used in our experiments.

C.1 CastFSR-Zero

To evaluate the effectiveness of the proposed Fast–Slow–Reflect workflow without task-specific optimization, we instantiate CastFSR in a training-free setting, denoted as CastFSR-Zero. Rather than directly generating numerical forecasts with the LLM, CastFSR-Zero treats the LLM as an agentic coordinator that orchestrates forecasting experts, analytical tools, contextual evidence, and reflective evaluation. Specifically, we employ DeepSeek-V4-Flash as the coordinator, which is responsible for planning, tool invocation, contextual reasoning, and forecast refinement. During inference, we set the maximum output length to 32,768 tokens. Given historical observations and task descriptions, the coordinator first performs fast-thinking forecasting by profiling temporal characteristics and selecting appropriate forecasting experts from a model pool, including ARIMA, DLinear, PatchTST, iTransformer, and Chronos-2. These experts generate data-driven forecast priors, providing a numerical basis for subsequent contextual reasoning. Training-free refers specifically to the LLM coordinator; the auxiliary forecasters follow the chronological partitions and target-blind data-access rules described below.

C.2 CastFSR-R1

To investigate whether the proposed reasoning workflow can be learned by a compact forecasting agent, we further develop CastFSR-R1 through SFT and RL. Specifically, we adopt Qwen3-4B-Instruct (Yang et al., 2025) as the backbone model and distill high-quality reasoning trajectories generated by CastFSR-Zero across diverse forecasting tasks. The SFT stage is conducted with a learning rate of 2.0×10^{-6} to teach the model the fundamental Fast–Slow–Reflect workflow, including expert selection, contextual reasoning, and reflective validation. We then apply the GRPO algorithm for reinforcement learning with a learning rate of 2.0×10^{-7} to further optimize the agent's decision-making policy. The RL stage uses a global batch size of 128, a group size of $G = 5$, and maximum prompt and response lengths of 20,480 and 6,144 tokens, respectively. The model is trained on 16 Ascend NPUs with distilled forecasting trajectories. CastFSR-R1 uses the compact model as the coordinator for numerical forecasting, contextual reasoning, and reflective evaluation.

C.3 Protocol and Reproducibility Checks

The unified data-splitting and rolling-window settings are summarized in Table 6.

Table 6: Chronological split and rolling-window evaluation protocol. Every dataset uses a 7:1:2 training/validation/test split with boundaries aligned to complete forecast windows. The test span includes the first look-back window; only forecast-horizon targets are scored.

Dataset	Split	Look-back	Horizon	Stride	Test windows	Test span (including first look-back)
ETTh1 / ETTh2	7:1:2	96	96	96	36	2018-01-28 00:00–2018-06-24 23:00
ETTm1 / ETTm2	7:1:2	96	96	96	145	2018-01-31 00:00–2018-06-25 23:45
Wind	7:1:2	96	96	96	101	2021-02-09 00:00–2021-05-21 23:45
NP / PJM	7:1:2	168	24	48	60	2018-08-21 00:00–2018-12-24 23:00
BE / FR	7:1:2	168	24	48	60	2016-08-28 00:00–2016-12-31 23:00
DE	7:1:2	168	24	48	60	2017-08-28 00:00–2017-12-31 23:00

All datasets are ordered by timestamp and divided without shuffling into training, validation, and test partitions using a 7:1:2 ratio. Split boundaries are aligned to complete forecast windows. For a unified cross-dataset comparison, this protocol is also applied to the selected 14,496-point EPF segments rather than the native two-year EPF holdout. Long-term tasks use a look-back length of 96, a forecasting horizon of 96, and a stride of 96; short-term tasks use 168, 24, and 48, respectively. To construct the first validation or test instance, the required look-back observations may come from the immediately preceding partition. These overlapping observations serve only as historical inputs: every forecast target lies entirely inside its assigned partition, and target timestamps are disjoint across the three splits.

The implementation enforces target-blind execution at every decision step. Forecast-horizon targets are excluded from the policy state and are not exposed to forecasting, retrieval, or reflection tools. SFT trajectory generation follows the same target-blind interface, while RL reference targets are drawn only from training windows and are used solely for offline feedback. Test targets are used only for final evaluation, never for trajectory selection, reward computation, policy inputs, or checkpoint selection. All compared methods receive the same historical look-back window. Future-known covariates are used only when they are available at the forecast origin; observed covariates and residual diagnostics use past data only. Reflection rules are fixed without tuning on test targets. Trajectory validation additionally rejects illegal tool calls, incorrect stage order, invalid timestamps, and incomplete forecasts.

C.4 Prompt Construction Details

As shown in Template 1, CastFSR constructs prompts according to the three-stage Fast–Slow–Reflect workflow. Each prompt contains the shared forecasting situation, including task metadata, domain priors, timestamp scope, sampling frequency, target variable, and available future-known covariates. The remaining prompt content and admissible actions are then adjusted to match the current stage.

- **Stage 1: Fast-thinking Forecasting.** The prompt provides the historical look-back window and task context, and asks the agent to profile temporal patterns, select diagnostic evidence, and route the instance to a suitable specialized forecasting model. The output of this stage is a data-driven forecast prior together with structured evidence about trend, seasonality, local dynamics, data quality, or residual behavior.
- **Stage 2: Slow Deliberative Reasoning.** The prompt includes the forecast prior from Stage 1, the retained diagnostic evidence, recent historical observations, and available contextual variables. It asks the agent to determine whether contextual signals are relevant to the forecast horizon, infer their direction and temporal scope, and refine the numerical prior without replacing it blindly.
- **Stage 3: Reflective Evaluation.** The prompt provides the candidate forecast, the forecast prior, the contextual reasoning trace, and task-specific constraints. It asks the agent to check temporal consistency, contextual support, domain validity, timestamp format, and numerical feasibility, then apply localized corrections when necessary before producing the final forecast sequence.

Table 7: Full training-free forecasting performance of CastFSR-Zero with different pretrained LLM coordinators on all datasets reported for this analysis. Lower values indicate better performance.

LLM Coordinator	ETTh1		ETTm1		Wind		DE		FR		NP	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
GPT-5.6-sol	0.081	0.213	0.055	0.168	**1.166**	**0.741**	0.400	0.414	4.056	0.505	0.218	0.266
LongCat-2.0	0.089	0.219	0.056	0.168	1.739	0.939	**0.383**	0.412	3.943	0.507	0.221	0.271
DeepSeek-V4-Pro	0.081	0.212	**0.054**	**0.165**	1.314	0.810	0.403	0.417	4.060	0.505	0.216	0.267
DeepSeek-V4-Flash	0.081	0.212	0.055	0.168	1.596	0.886	0.393	**0.408**	**3.939**	0.501	0.221	0.277
GLM-5.2	**0.080**	**0.210**	0.055	0.167	1.332	0.806	0.405	0.427	3.966	**0.498**	**0.207**	**0.260**

Table 8: Full performance comparison of CastFSR-R1 with variants removing supervised fine-tuning (w/o SFT) or reinforcement learning (w/o RL). Lower values indicate better performance.

Variant	ETTh1		ETTh2		ETTm1		ETTm2		Wind		BE		DE		FR		NP		PJM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o SFT	0.081	0.213	0.274	0.367	**0.055**	**0.168**	0.141	0.254	1.795	0.920	0.959	0.384	0.435	0.436	4.059	0.504	0.237	0.280	0.144	0.267
w/o RL	0.082	0.215	0.268	0.364	**0.055**	0.170	0.139	0.253	1.813	0.920	0.961	0.395	0.392	0.409	4.047	0.503	0.217	0.268	0.149	0.275
CastFSR-R1	**0.077**	**0.210**	**0.242**	**0.351**	**0.055**	**0.168**	**0.127**	**0.237**	**1.757**	**0.918**	**0.927**	**0.375**	**0.386**	**0.405**	**4.032**	**0.502**	**0.172**	**0.247**	**0.133**	**0.260**

D Additional Experimental Results

Table 7 reports the full training-free LLM coordinator comparison. Table 8 reports the full SFT/RL ablation results.

Effect of Different LLM Coordinators. Table 7 evaluates the training-free CastFSR-Zero with different pretrained LLM coordinators to investigate whether the effectiveness of our agentic workflow depends on a specific foundation model. The results show that the workflow can be executed with each evaluated coordinator, with performance varying across datasets. Although different coordinators exhibit complementary strengths across datasets, no single LLM consistently dominates all forecasting scenarios. For example, GLM-5.2 achieves the best performance on ETTh1 and NP, while DeepSeek-V4-Pro performs best on ETTm1. DeepSeek-V4-Flash obtains competitive results on multiple datasets, achieving the lowest MAE on DE and the lowest MSE on FR, while GPT-5.6-sol achieves the strongest results on Wind. These variations characterize the complete workflow and do not isolate the coordinator's reasoning ability from its tool use and forecast revisions. By decoupling numerical forecasting from high-level reasoning, CastFSR-Zero enables different LLMs to leverage specialized forecasting experts and structured analytical tools. The comparison supports instantiation with multiple general-purpose LLM coordinators within the evaluated settings.

Effect of Training Strategies. Table 8 compares the full CastFSR-R1 with variants trained without SFT or without RL. The full model achieves the best or tied-best result on every reported metric. Both variants tie the full model on ETTm1 MSE, and the variant without SFT also ties on ETTm1 MAE; the remaining metrics favor the full model. This pattern supports the combination of supervised workflow imitation and reward-based trajectory optimization, while the ties show that the observed gains depend on the dataset and metric.

D.1 Visualization Analysis

Figure 8 illustrates how the selected look-back windows influence the refinement process of CastFSR. The upper examples represent successful cases where the retrieved historical contexts exhibit strong relevance to future dynamics. By identifying consistent patterns or regime transitions from the selected windows, CastFSR appropriately adjusts the forecast prior and produces predictions closer to the ground truth. In contrast, the lower examples correspond to challenging scenarios where the selected contexts provide insufficient or misleading evidence. Consequently, the contextual reasoning process may introduce inaccurate adjustments or fail to fully correct the initial forecast. These observations highlight that context-aware forecasting is not simply a matter of incorporating more historical information, but requires adaptively identifying informative temporal windows that match the underlying dynamics. These examples illustrate both the benefits of

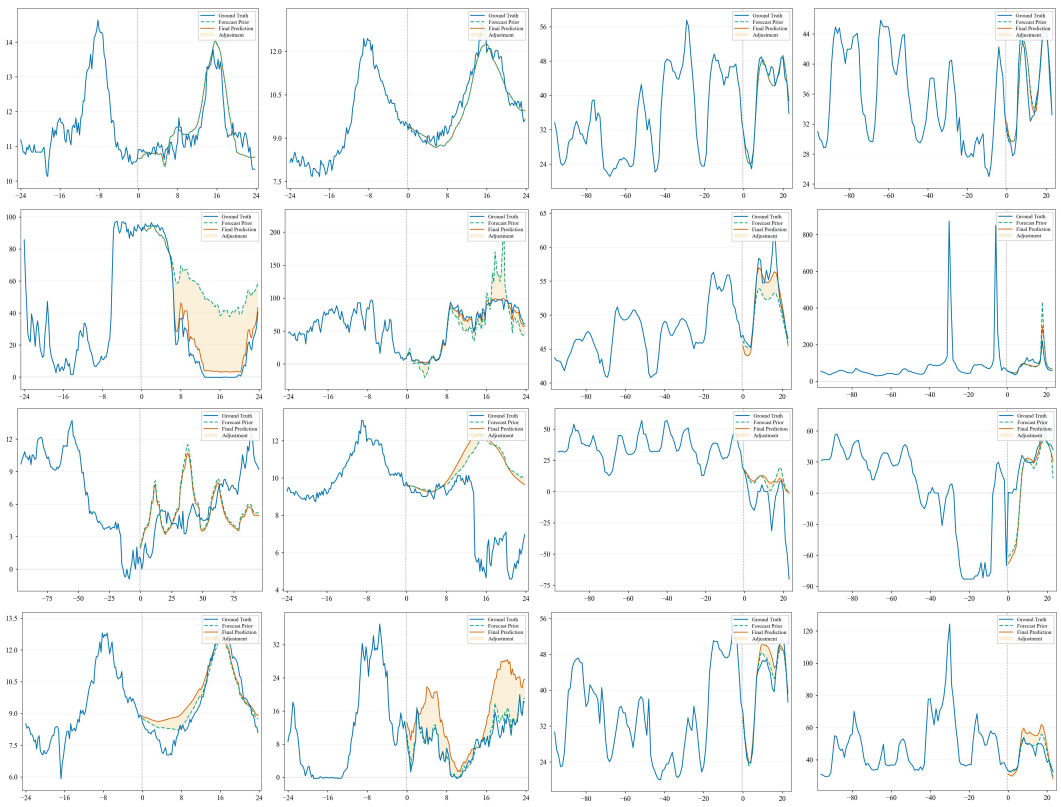

Figure 8: Case study of adaptive look-back window selection in CastFSR. The upper examples show successful cases where the retrieved contextual windows provide relevant historical evidence for future dynamics, enabling effective refinement of the forecast prior. The lower examples show challenging cases where inappropriate context windows provide limited or misleading evidence, resulting in insufficient or excessive adjustments.

relevant context and the risk of excessive or insufficient forecast revisions when retrieval is poorly aligned.

Fast–Slow–Reflect Agentic Forecasting Prompt

Role & Context: Act as the forecasting policy of a context-aware time-series agent. All decisions remain **target-blind**, grounded in historical observations, task and domain priors, temporal context, available future-known covariates, diagnostic evidence, and auxiliary forecasts.

Stage 1—Fast-thinking Forecasting: Construct a pattern-based numerical prior through evidence acquisition and context-aware predictor routing.

Evidence Acquisition. Follow a "Stop-and-Look" policy and select the diagnostic view that best resolves the dominant uncertainty:

- `extract_basic_statistics`: global structure and dependence;
- `extract_within_channel_dynamics`: local dynamics and regime shifts;
- `extract_forecast_residuals`: forecast bias and residual structure;
- `extract_data_quality`: corruption, saturation, and dropout;
- `extract_event_summary`: segment-level temporal events.

Select exactly one diagnostic action from this complementary toolkit before prediction.

Context-Aware Predictor Routing. Route the task to a single auxiliary predictor according to the situation context and acquired evidence:

- **ARIMA**: linear dynamics with stable seasonality;
- **DLinear**: decomposable trend and seasonal dynamics;
- **PatchTST**: regular dependence and multi-scale seasonality;
- **iTransformer**: pronounced cross-channel dependence;
- **Chronos-2**: irregular, nonlinear, or shifting regimes.

The preferred model acts only as a tie-breaker. Coarse target-blind calibration is reserved for evidence-supported global bias.

Stage 2—Slow Deliberative Reasoning: Treat the auxiliary trajectory as a numerical baseline rather than a final answer, preserve its temporal alignment, and reason through two coupled views:

- **Situation Synthesis:** infer correction direction, reliability, lag, and temporal scope from regime, domain, and exogenous evidence;
- **Forecasting Reasoning:** reconcile this guidance with historical features and the auxiliary level, shape, and phase.

Refinement follows natural cycles and change points. Its scope and magnitude are regularized by evidence strength, enabling conservative local adjustment under uncertainty and larger yet localized correction under coherent evidence.

Stage 3—Reflective Evaluation
Consistency Audit: Synthesize rule diagnostics with the situation context, feature evidence, auxiliary trajectory, candidate forecast, and prior reflection trace. A target-blind, dataset-specific rule gate activates reflection only for a critical violation or severe conflicts spanning multiple rule families.

Adaptive Re-entry:

- **Evidence-level inconsistency:** return to Stage 1 and reconsider feature, regime, exogenous evidence, and predictor routing;
- **Reasoning-level inconsistency:** retain valid evidence and the auxiliary trajectory, and revisit Stage 2.

The re-entry depth is matched to the diagnosed source of error, preserving valid intermediate state while revisiting only the uncertain reasoning path.

Structured Output: The rationale links situation evidence, auxiliary diagnostics, consistency validation, and uncertainty-aware decision making.

```
<think>
Part I - Situation synthesis and analysis
Part II - Situation-guided forecasting reasoning
</think>
<answer>
YYYY-MM-DD HH:MM:SS finite_decimal
...
</answer>
```