NEWS-AWARE TIME SERIES FORECASTING AGENT ENHANCED BY OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate time-series forecasting in event-sensitive environments can benefit from external news, but semantically relevant news is not necessarily useful for forecasting. We study news use as a forecasting-feedback-driven selection problem. Our framework combines an LLM-based forecaster with a sequential selector that decides whether each candidate article should be kept, dropped, or summarized, conditioned on the numerical history, structured context, and information already retained. The selector is learned from fixed historical trajectories using offline reinforcement learning, with terminal reward defined by improvement in downstream forecasting loss relative to a no-news forecast. We further construct a forecasting-error reflection memory that retains counterfactual selection corrections that reduce historical forecasting loss. We evaluate the framework on electricity demand, exchange rate, and Bitcoin price forecasting. End-to-end results show that the usefulness of external news and reflection varies across tasks: electricity favors highly conservative information use, whereas selected news provides larger benefits in the financial tasks. In a controlled fixed-forecaster evaluation, changing only the supplied news context materially affects forecasting accuracy, and the best forecasting-feedback-selected context achieves lower forecasting error than both no-news and LLM-selected contexts across all three domains. These results show that forecasting utility should be learned from downstream forecasting feedback rather than inferred from semantic relevance alone.

1 INTRODUCTION

Time-series models based only on historical observations can miss abrupt changes triggered by external events that are not directly encoded in past measurements. Earlier event-driven forecasting studies have shown that textual signals and external events can provide predictive information beyond numerical history (Bollen et al., 2011; Ding et al., 2015; Li et al., 2023). Recent language-model-based forecasting methods have also improved the ability to represent and reason over numerical sequences (Gruver et al., 2023; Zhou et al., 2023; Jin et al., 2024; Sun et al., 2024), while contextual forecasting approaches increasingly incorporate textual, multimodal, or other exogenous information (Jia et al., 2024; Williams et al., 2025; Lee et al., 2025; Zhong et al., 2025; Wu et al., 2026; Wang et al., 2026). These developments make it possible to build forecasting systems that combine numerical histories with external news.

The main difficulty is not simply obtaining relevant news, but deciding which information should actually be supplied to the forecaster. Real-world news streams contain irrelevant, redundant, and potentially misleading content, and existing news-aware forecasting pipelines commonly rely on rules, semantic relevance, or LLM-based assessments to filter and summarize candidate events (Wang et al., 2024). Related retrieval-augmented language systems likewise show that indiscriminately adding retrieved context can be unhelpful (Asai et al., 2024; Zhu et al., 2025). For forecasting, semantic relevance is therefore only an indirect proxy: the usefulness of an article depends on whether including it improves the downstream forecast in the current forecasting context.

We formulate news use as a sequential selection problem. Given a time series, structured context, and an ordered sequence of candidate articles, the selector decides whether each article should be KEEP, DROP, or SUMMARIZE while conditioning on the context already retained. After all decisions are made, the selected textual context is supplied to the forecaster, and the change in forecasting loss

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.

relative to a no-news forecast provides the terminal reward. We learn this policy from fixed historical trajectories using offline reinforcement learning rather than online exploration. The formulation is not tied to a new RL optimizer: we use CQL as the primary conservative offline-RL method and compare it with DQN, BCQ, and Guarded CQL (Yan et al., 2025).

We also investigate whether verified historical selection errors can be reused. For high-loss training cases, the reflection module changes one logged selection action at a time while holding the remaining actions fixed, re-evaluates the resulting context with the same forecasting checkpoint, and stores only corrections that reduce forecasting loss. The resulting memory is constructed from training data and augments later selector states with information from forecasting-verified mistakes.

Our experiments cover electricity demand, exchange rate, and Bitcoin price forecasting. We report both end-to-end system results and a fixed-forecaster comparison in which the forecasting checkpoint is held constant and only the supplied news context changes. The results show that the amount and form of useful external information are task dependent. In the electricity setting, the strongest CQL configuration is highly conservative and retains essentially no news, while reflection does not improve the base CQL result. In the financial tasks, selected news and, in some settings, reflection provide additional gains. The fixed-forecaster comparison further shows that changing only the selected context can materially change forecasting accuracy, providing direct evidence that the selection decision itself matters in addition to forecaster adaptation.

2 Related Work

Language models and multimodal context for time-series forecasting. Recent work has adapted pretrained language models and foundation models to time-series forecasting through reprogramming, embedding alignment, prompting, autoregressive generation, and large-scale pretraining (Gruver et al., 2023; Zhou et al., 2023; Jin et al., 2024; Sun et al., 2024; Cao et al., 2024; Goswami et al., 2024; Das et al., 2024; Liu et al., 2024). Time-LLM reprograms numerical patches into the representation space of a frozen LLM (Jin et al., 2024), TEST aligns time-series embeddings with textual prototypes (Sun et al., 2024), and AutoTimes repurposes decoder-only LLMs as autoregressive forecasters with in-context forecasting capabilities (Liu et al., 2024). At the same time, recent evidence suggests that pretrained language-model components alone are not uniformly responsible for forecasting improvements, motivating closer attention to what information the language model contributes beyond numerical sequence modeling (Tan et al., 2024). A growing line of work therefore incorporates auxiliary textual, visual, or exogenous information. TimeCAP, ChatTime, CALF, and TimeCMA develop contextualization and cross-modal alignment mechanisms (Lee et al., 2025; Wang et al., 2025a; Liu et al., 2025b;a); Time-VLM combines temporal, visual, and textual representations (Zhong et al., 2025); and ExoTimer, M3Time, and VoT explicitly model exogenous or event-driven information (Wu et al., 2026; Jia et al., 2026; Wang et al., 2026). These methods establish the value of augmenting numerical histories with external context, but they do not directly optimize which individual news items should be retained according to their downstream effect on forecasting error.

News-aware forecasting and reflection. Textual events have long been used as exogenous signals for financial and event-sensitive prediction (Bollen et al., 2011; Ding et al., 2015; Li et al., 2023). The closest prior work to our setting is From News to Forecast, which uses LLM-based agents to iteratively filter irrelevant news, reason about event effects, and refine forecasting logic using prediction feedback (Wang et al., 2024). More recently, TimeXL introduces an LLM-in-the-loop multimodal prediction framework with prediction, reflection, and refinement stages: a reflection LLM compares predictions with ground truth, identifies textual inconsistencies or noise, and triggers subsequent refinement (Jiang et al., 2025). These works demonstrate that forecasting feedback can improve the use of textual context. Our reflection module differs in its role and supervision: it is attached to an RL-based news selector, tests counterfactual changes to concrete selection actions, and stores only corrections whose alternative action measurably reduces downstream forecasting loss. The resulting memories are then reused across later forecasting instances.

Reinforcement learning for forecasting and information selection. RL has been used in forecasting for adaptive model combination and, more recently, for directly improving LLM-based forecasters. RLMC learns dynamic combinations of forecasting models (Fu et al., 2022), whereas LangTime introduces TimePPO, a PPO-based fine-tuning method that optimizes an autoregressive LLM forecaster using time-series-specific rewards (Niu et al., 2025). LangTime is particularly relevant because it

establishes that reinforcement learning can improve LLM-based time-series forecasting; however, its policy optimization acts on the forecaster itself rather than on the selection of external textual information. Related work in language systems has also used RL or adaptive policies for information choice, including demonstration selection and retrieval control (Wang et al., 2025b; Asai et al., 2024; Zhu et al., 2025). Our setting is different: the decision variable is whether each candidate news item should be kept, dropped, or summarized, the reward is defined by the resulting forecasting loss, and the policy is learned from fixed historical trajectories using offline RL. Thus, the main distinction is not the use of RL or reflection in isolation, but their integration into a news-aware forecasting agent in which downstream forecasting performance supervises both news selection and reusable error memory.

Table 1: Comparison with representative recent work most closely related to our framework. ✓ denotes explicit support, × denotes that the capability is not part of the method, and △ denotes a related but different mechanism. "RL for forecasting" indicates RL used to optimize the forecasting model itself, whereas "RL for news selection" indicates RL used to decide which external news is supplied to the predictor.

Method	LLM-based forecasting	External text/news	Explicit news selection	RL for forecasting	RL for news selection	Offline RL	Forecast-feedback reflection/memory
Time-LLM (Jin et al., 2024)	✓	△	×	×	×	×	×
AutoTimes (Liu et al., 2024)	✓	△	×	×	×	×	×
From News to Forecast (Wang et al., 2024)	✓	✓	✓	×	×	×	△
LangTime (Niu et al., 2025)	✓	△	×	✓	×	×	×
Time-VLM (Zhong et al., 2025)	✓	✓	×	×	×	×	×
TimeXL (Jiang et al., 2025)	✓	✓	×	×	×	×	✓
Ours	✓	✓	✓	×	✓	✓	✓

3 News-Aware Time Series Forecasting Agent

Our forecasting agent integrates three complementary components: an LLM-based forecaster that predicts future values from numerical and textual context, an LLM-assisted news-processing module that represents and summarizes candidate articles, and a sequential news-selection policy trained with offline reinforcement learning. Rather than selecting news solely according to semantic relevance, the selector is optimized using downstream forecasting loss, so that external information is retained only when it is useful for the forecasting task. The agent further incorporates a forecasting-error reflection mechanism that converts historically loss-reducing selection corrections into reusable memories for subsequent decisions.

Figure 1 summarizes the complete information flow. At forecast origin T, the agent receives the historical time series $\mathbf{x}_{1:T}$, structured covariates $\mathbf{c}_T$, and an ordered candidate-news sequence $\mathcal{N}_T = (n_1, \ldots, n_M)$. The numerical history and structured context are directly available to the forecaster, while the candidate news is processed and passed to the sequential selector. For each article n_i, the selector conditions on the forecasting state and the news context already retained, and chooses whether to keep the article, discard it, or replace it with a shorter summary. After all candidates have been processed, the selected context $\mathcal{S}_T$ is combined with the numerical inputs to produce the final forecast.

The forecasting loss then closes the loop between selection and prediction. During offline policy learning, it serves as the downstream supervision signal that determines whether a sequence of news-selection decisions was beneficial for forecasting. The same feedback is also used by the reflection module: high-loss historical cases are revisited through counterfactual action tests, and only corrections that measurably reduce forecasting loss are stored as reflection memories. These memories can subsequently be retrieved and incorporated into the selector state, providing a second feedback pathway beyond the offline-RL policy itself.

3.1 Problem Formulation

Let $\mathbf{x}_{1:T}$ denote the numerical history at forecast origin T, $\mathbf{c}_T$ structured covariates, and $\mathcal{N}_T = (n_1, \ldots, n_M)$ an ordered sequence of candidate news available before the forecast origin. The target is $\mathbf{x}_{T+1:T+H}$. After news processing produces selected context $\mathcal{S}_T$, the forecasting component

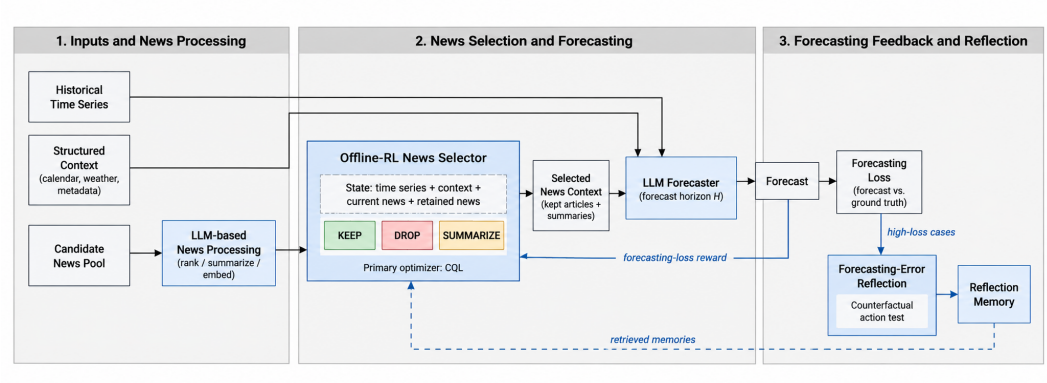


Figure 1: Overview of the proposed news-aware time-series forecasting agent. Historical observations and structured context are provided directly to the forecaster, while candidate news is first processed by an LLM-assisted news module and then examined sequentially by an offline-RL selector. At each step, the selector chooses among KEEP, DROP, and SUMMARIZE based on the current forecasting state, the candidate article, and the context retained so far. The resulting news context is supplied to the LLM forecaster. During historical training, downstream forecasting loss is computed using the historical target and provides the learning signal for the news-selection policy. Counterfactual action tests on high-error training episodes identify verified selection failures, which are stored in a training-only reflection memory and retrieved to support later selection decisions. At validation and test time, the selector and fixed memory bank operate without access to future targets or forecasting error.

predicts

$$\hat{\mathbf{x}}_{T+1:T+H} = f_\theta(\mathbf{x}_{1:T}, \mathbf{c}_T, \mathcal{S}_T), \tag{1}$$

with forecasting loss

$$\mathcal{L}(\mathcal{S}_T) = \ell(\hat{\mathbf{x}}_{T+1:T+H}, \mathbf{x}_{T+1:T+H}). \tag{2}$$

The LLM news-processing component supplies article representations and compressed summaries, while the RL policy controls which representation, if any, is passed to the forecaster. This separation allows the forecasting model and news-processing model to remain general-purpose components while the selection policy is aligned directly with forecasting feedback.

3.2 FORECASTING-AWARE NEWS SELECTION

The selector is implemented sequentially over the ordered candidate-news list, but the central objective is not the ordering mechanism itself: it is to align the information supplied to the predictor with downstream forecasting performance. At decision step i, the selector observes

$$s_i = \big(\phi_x(\mathbf{x}_{1:T}), \phi_c(\mathbf{c}_T), \phi_n(n_i), \phi_s(\mathcal{S}_{i-1})\big), \tag{3}$$

where the four blocks represent the numerical history, structured forecasting context, current news item, and previously selected context. Conditioning on $\mathcal{S}_{i-1}$ allows later decisions to account for information that has already been retained.

The action space is $\mathcal{A} = \{\text{KEEP}, \text{DROP}, \text{SUMMARIZE}\}$. Let $g(\cdot)$ denote the fixed summarization operator. Starting from $\mathcal{S}_0 = \emptyset$, the context evolves as

$$\mathcal{S}_i = \begin{cases} \mathcal{S}_{i-1} \cup \{n_i\}, & a_i = \text{KEEP}, \\ \mathcal{S}_{i-1}, & a_i = \text{DROP}, \\ \mathcal{S}_{i-1} \cup \{g(n_i)\}, & a_i = \text{SUMMARIZE}. \end{cases} \tag{4}$$

Thus, the policy can preserve a candidate in full, exclude it, or provide a compressed representation to the forecaster. Dropping every candidate is a valid trajectory.

After all M candidates are processed, the selected context is evaluated by the forecasting component. The terminal reward is the reduction in forecasting loss relative to the no-news prediction:

$$R_T = \mathcal{L}(\emptyset) - \mathcal{L}(\mathcal{S}_M). \tag{5}$$

A positive reward means that the selected context lowers forecasting loss, while a negative reward means that it degrades the forecast. Intermediate rewards are zero. The reward therefore aligns the selection policy with the downstream forecasting objective without requiring article-level relevance labels.

3.3 Offline Policy Learning

Direct online exploration would require constructing many alternative contexts and repeatedly evaluating the forecasting model. We instead learn from historical trajectories. For each training forecasting instance, a behavior selector produces actions $(a_1, \ldots, a_M)$ and the corresponding terminal forecasting-loss reward, giving a trajectory $\tau = (s_1, a_1, r_1, \ldots, s_M, a_M, r_M)$. The offline dataset is $\mathcal{D} = \{\tau^{(1)}, \ldots, \tau^{(N)}\}$. The same behavior data are used across the RL optimizers.

We use CQL as the primary optimizer (Kumar et al., 2020). For a transition (s, a, r, s'), discrete-action CQL augments Bellman regression with a conservative value penalty:

$$\mathcal{L}_{\mathrm{CQL}}(Q) = \mathbb{E}_{\mathcal{D}}\left[(Q(s,a) - y)^2\right] + \alpha\, \mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_{a'\in\mathcal{A}} e^{Q(s,a')} - \mathbb{E}_{a\sim\beta(\cdot|s)} Q(s,a)\right], \qquad (6)$$

where $y = r + \gamma \max_{a'} Q_{\mathrm{tgt}}(s', a')$ and β is the empirical behavior policy. The conservative term reduces optimistic values for actions that are weakly represented in the offline data, and the final selector is $\pi(s) = \arg\max_{a\in\mathcal{A}} Q(s,a)$. We use the same state, action, reward, and behavior trajectories for DQN, BCQ, Guarded CQL, and the other policy-learning baselines. Appendix B provides the conservative value analysis specialized to this news-selection setting.

3.4 Forecasting-Error Reflection Memory

The terminal reward evaluates the completed context but does not identify the decision responsible for a poor forecast. We therefore construct a reflection memory from historically loss-reducing fixed-suffix selection corrections. For high-loss training instances, we test three action changes:

$$\textsc{Drop} \to \textsc{Keep}, \qquad \textsc{Keep} \to \textsc{Drop}, \qquad \textsc{Summarize} \to \textsc{Keep}. \qquad (7)$$

A case is stored only when the counterfactual action reduces forecasting loss. A successful DROP→KEEP test identifies an omission, KEEP→DROP identifies a harmful retention, and SUMMARIZE→KEEP identifies over-compression. Forecasting loss determines whether a correction is valid; the LLM is used to structure the verified case into a reusable memory.

Each memory stores the forecasting context, original action, loss-reducing action correction, change in forecasting loss, and a structured description of the failure. For a later selection state, relevant verified memories are retrieved and converted into additional state features for the selector. The reflection module therefore reuses only corrections that have been validated by downstream forecasting loss, rather than relying on unverified LLM self-assessment. Implementation details, context-usage statistics, horizon analyses, and reflection behavior are reported in Appendix C.

4 Experiments

We evaluate the news-aware forecasting agent on electricity demand, exchange-rate (AUD/USD) forecasting, and Bitcoin price forecasting. The main experiment evaluates the three-part agent in a single comparison: the predictor establishes the forecasting task, the offline-RL selector determines how external news is supplied to it, and forecasting-error reflection further refines the CQL selector using verified memories of previous selection failures. The experiments ask whether forecasting-feedback-based news selection improves over no-news and LLM-based selection, whether conservative offline learning provides a strong base policy, and whether reflection can further improve or preserve the performance level of the CQL-based agent. Detailed horizon analyses, offline behavior-dataset statistics, context-usage statistics, round-wise results, and the larger electricity benchmark are reported in Appendix C.

4.1 EXPERIMENTAL SETUP

Datasets and horizons. We use three event-sensitive forecasting datasets paired with news available before the forecast origin. Because the useful temporal range of external information is task dependent, the forecasting horizon is treated as a dataset-level hyperparameter selected on a chronological validation split and then frozen before final test evaluation. The resulting configurations use a 30-minute one-step horizon for electricity demand and 7 day horizon for exchange rate and Bitcoin price. All comparisons within a dataset use the same chronological split, input window, candidate-news pool, horizon, forecasting backbone, training budget, checkpoint-selection rule, and evaluation script. Additional horizon comparisons are given in Appendix C.1.

Forecasting and news-processing components. We use Gemma-2-9B with LoRA adaptation as the forecasting backbone and Qwen2.5-72B-AWQ for LLM-based news processing. *No News* uses only numerical history and structured context. *LLM-selected News* selects candidate articles according to semantic and task relevance. The RL selectors use the same KEEP/DROP/SUMMARIZE decision space and forecasting-loss reward; the ordered article-by-article processing is an implementation mechanism within the broader agent rather than the primary source of novelty. We compare DQN, BCQ, Guarded CQL, and CQL; CQL is the primary conservative offline-RL optimizer.

Reflection. The reflection module is built on top of the CQL selector rather than treated as a separate selection algorithm. It revisits high-loss historical cases, tests counterfactual changes to previous KEEP, DROP, and SUMMARIZE decisions, and stores only corrections that reduce downstream forecasting loss. Retrieved memories are then added to later selection states. In the main table we report a single *CQL + Reflection* result for the final reflection-enhanced configuration used in each dataset.

Metrics. We report mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R^2. Lower MSE, RMSE, MAE, and MAPE are better, while higher R^2 is better.

4.2 MAIN RESULTS: SELECTION AND REFLECTION

Table 2 reports the base selectors together with the reflection-enhanced CQL agent. CQL provides the strongest base result on electricity demand and remains a strong base policy across the two financial tasks. In the electricity setting, the selected CQL policy retains essentially no news on average, indicating that highly conservative news use is preferable in this particular task. Reflection improves the reported CQL result on both financial forecasting tasks, but it does not improve the electricity result; the validation-selected electricity reflection configuration remains close to the base CQL forecasting level.

Base selector. CQL provides the strongest base result on electricity demand, with RMSE 54.7, MAE 36.6, and MAPE 1.41%. The selected CQL policy retains essentially no news in this setting, so its deployed behavior is to suppress external textual information rather than force news into the forecast. We interpret this as a near-empty-context solution for the reported $H = 1$ setting, rather than as evidence that news is universally unhelpful for electricity forecasting. On AUD/USD, CQL also performs strongly, while Bitcoin shows closer competition among the RL optimizers. We therefore interpret CQL as a strong conservative base policy rather than claiming that it dominates every optimizer on every dataset.

Reflection on top of CQL. Reflection provides a second layer of forecasting feedback beyond the base offline policy. On AUD/USD, adding reflection reduces RMSE from 0.008509 to 0.008394 and increases R^2 from 0.830 to 0.834. On Bitcoin, reflection reduces RMSE from 5067.41 to 4792.69 and increases R^2 from 0.700 to 0.732. On electricity, the selected base CQL configuration retains essentially no external news and achieves lower RMSE than its reflection-enhanced counterpart, although the degradation is modest and the reflection-enhanced forecast remains competitive with the other baselines. This suggests that, for the reported end-to-end configuration, the near-empty context produced by base CQL is already effective, and reintroducing additional information through reflection does not provide further benefit. The appendix further shows that the selected CQL checkpoint retains almost no articles or tokens. Overall, the gains from reusable error memory are more pronounced on the financial forecasting tasks, particularly for exchange-rate forecasting, than on the short-horizon electricity setting.

Table 2: Main forecasting results across three domains. *CQL + Reflection* denotes a reflection-enhanced CQL configuration. *Bold* values denote the best result for each metric within a panel.

Panel A: Electricity Demand					
Method	MSE ($\times 10^3$) ↓	RMSE ($\times 10^1$) ↓	MAE ($\times 10^1$) ↓	MAPE (%) ↓	R^2 ↑
No News	4.422	6.65	4.51	1.66	0.999
LLM-selected News	3.725	6.10	4.14	1.51	0.999
RL selector (DQN)	3.785	6.15	4.17	1.57	0.999
RL selector (BCQ)	3.058	5.53	3.71	1.44	1.000
RL selector (Guarded CQL)	3.352	5.79	3.85	1.48	0.999
RL selector (CQL)	**2.987**	**5.47**	**3.66**	**1.41**	**1.000**
CQL + Reflection	3.178	5.64	3.87	1.43	1.000
Panel B: Exchange Rate (AUD/USD)					
Method	MSE ($\times 10^{-5}$) ↓	RMSE ($\times 10^{-3}$) ↓	MAE ($\times 10^{-3}$) ↓	MAPE (%) ↓	R^2 ↑
No News	9.604	9.800	7.72	1.029	0.774
LLM-selected News	8.852	9.409	7.80	1.036	0.792
RL selector (DQN)	7.750	8.803	7.38	0.985	0.818
RL selector (BCQ)	7.862	8.867	7.38	0.982	0.815
RL selector (Guarded CQL)	7.356	8.577	7.00	0.933	0.827
RL selector (CQL)	7.240	8.509	6.92	0.922	0.830
CQL + Reflection	**7.046**	**8.394**	**6.86**	**0.915**	**0.834**
Panel C: Bitcoin Price					
Method	MSE ($\times 10^7$) ↓	RMSE ($\times 10^3$) ↓	MAE ($\times 10^3$) ↓	MAPE (%) ↓	R^2 ↑
No News	3.538	5.948	4.960	10.673	0.587
LLM-selected News	3.748	6.122	5.049	11.167	0.562
RL selector (DQN)	2.391	4.890	**3.945**	**8.890**	0.721
RL selector (BCQ)	2.562	5.062	4.290	9.688	0.701
RL selector (Guarded CQL)	2.719	5.214	4.255	9.391	0.682
RL selector (CQL)	2.568	5.067	4.217	9.594	0.700
CQL + Reflection	**2.297**	**4.793**	3.979	8.945	**0.732**

Table 3: Fixed-forecaster comparison of alternative news contexts. For each dataset, the LLM-selection Round-0 forecasting checkpoint is held fixed while the supplied news context is changed. The CQL and Reflection conditions use Round-0 selector configurations and therefore need not coincide with the validation-selected end-to-end configurations reported in Table 2. All entries are RMSE; the scale is shown in the dataset label. Bold values denote the best result within each row.

Dataset	No News	LLM-selected	CQL	Reflection 1	Reflection 2
Electricity ($\times 10^1$)	5.939	6.191	5.966	5.800	**5.640**
AUD/USD ($\times 10^{-3}$)	10.307	9.736	8.586	8.924	**8.394**
Bitcoin ($\times 10^3$)	5.693	6.169	**5.162**	5.233	5.230

Fixed-forecaster selection comparison. Table 2 reports end-to-end performance under the forecasting pipeline associated with each method, and therefore may reflect both forecaster adaptation and differences in the selected news context. To more directly examine the effect of information selection, we additionally conduct a controlled fixed-forecaster comparison. For each dataset, we hold the LLM-selection Round-0 forecasting checkpoint fixed and vary only the news context supplied at inference time: No News, the forecaster's own LLM-selected news, Round-0 CQL-selected news, and two reflection-enhanced selections. Table 3 reports the resulting RMSE. This comparison removes differences caused by using separately adapted forecasting checkpoints and therefore provides a more direct evaluation of how the selected information affects predictions for a common forecaster.

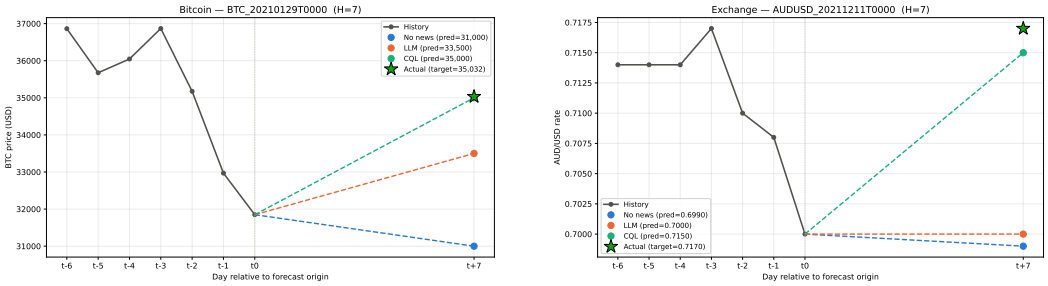
Under the shared forecaster, CQL-selected news achieves the lowest RMSE on Bitcoin, improving over both No News and the forecaster's own LLM-selected news. On AUD/USD, both CQL and the reflection-enhanced selections improve over the two baselines, with Reflection 2 achieving the lowest RMSE. On electricity, base CQL performs similarly to No News, whereas Reflection 2 achieves the strongest result in this controlled comparison. Together, these results show that forecasting performance can change substantially when the forecaster is held fixed and only the selected news

context is changed. They also show that the most effective selection behavior differs across forecasting tasks.

Cross-domain interpretation. Taken together, the end-to-end and fixed-forecaster results support a narrower conclusion than simply claiming that more news is always beneficial. Forecasting-feedback-based selection can improve the information supplied to the forecaster, but the useful amount of news differs across tasks. Electricity favors very conservative use of external text in the main CQL system, whereas the financial tasks benefit more clearly from selected news and, in the reported configurations, from reflection. The results therefore motivate forecasting-aware information selection rather than semantic relevance filtering alone, without requiring a single RL or reflection configuration to dominate every dataset.

4.3 Case Studies

Figure 2 illustrates two examples in which forecasting-loss-aligned selection yields predictions much closer to the observed target than either No News or LLM-selected News.



(a) Bitcoin: BTC_20210129T0000 (b) AUD/USD: AUDUSD_20211211T0000

Figure 2: Two representative forecasting cases. For Bitcoin, the actual value is 35,032, compared with predictions of 31,000 (No News), 33,500 (LLM), and 35,000 (CQL), giving MAEs of 4032, 1532, and 32. For AUD/USD, the actual value is 0.717, compared with 0.699 (No News), 0.700 (LLM), and 0.715 (CQL), giving MAEs of 0.0180, 0.0170, and 0.0020.

In the Bitcoin example, the LLM-selected context contains several stale or weakly directional anecdotes, while CQL retains an Ethereum all-time-high article that provides a market-wide bullish signal consistent with the subsequent rebound. In the AUD/USD example, CQL retains Australian equity and resource-sector signals that are more directly connected to the domestic currency environment, whereas the LLM selection includes an irrelevant Bitcoin article and a defensible but ultimately incorrect Fed-taper signal. These examples do not establish causality for individual articles, but they illustrate how forecasting-loss-aligned selection can prefer context that is more useful to the downstream predictor than semantic relevance alone.

4.4 Discussion and Limitations

Dataset-dependent value of news and reflection. The usefulness of external news differs substantially across forecasting tasks. Electricity demand is the clearest example: the selected CQL policy retains essentially no news in the reported configuration and achieves the strongest base result. Because the CQL pipeline and the standalone No News baseline involve separately trained forecasters, their end-to-end performance gap should be interpreted as a system-level difference. Under the fixed-forecaster comparison, base CQL performs similarly to No News, while a sparse reflection-selected context achieves the strongest result. These results suggest that the reported $H = 1$ electricity setting benefits from highly selective use of external text, whereas reflection provides clearer gains in the financial tasks, particularly for exchange-rate forecasting.

Forecast-horizon dependence. The effective horizon of textual information is task dependent. A horizon that is too short may not allow an external event to propagate into the target, whereas a horizon that is too long can dilute or even reverse its predictive relevance. Figure 3 shows that forecasting accuracy and the relative ranking of different news-selection strategies vary substantially with the prediction horizon. In particular, the benefit of incorporating news is not monotonic in H: a

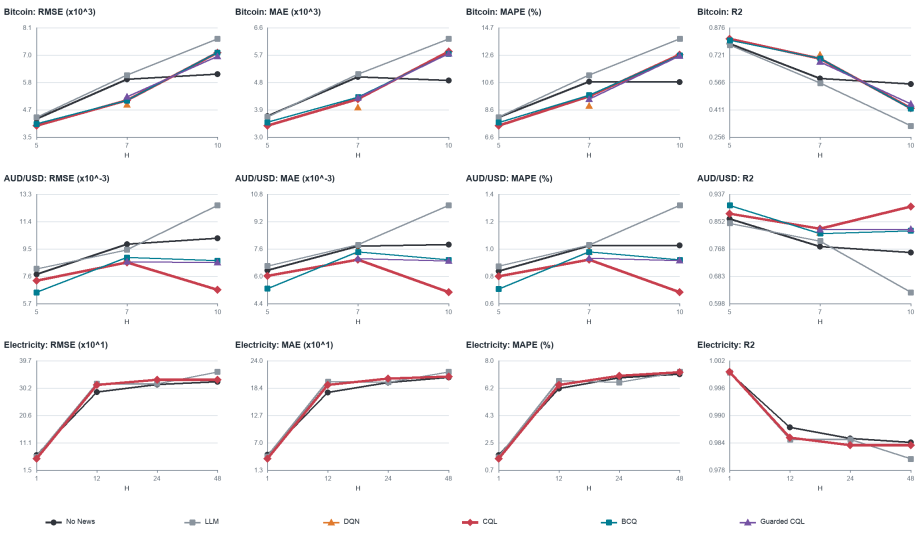


Figure 3: Forecast-horizon sensitivity across the three forecasting domains. The panels report RMSE, MAE, MAPE, and R^2 under different prediction horizons. Performance varies substantially with H, and the relative advantage of incorporating external news is not monotonic across horizons.

method that is effective at one horizon may provide little or no improvement at another. This suggests that the forecasting horizon determines not only the difficulty of the numerical prediction problem but also the temporal window over which textual events provide useful predictive information. In this work, we select the forecasting horizon separately for each dataset and use the selected setting for the main experiments. Developing a principled method for jointly determining the forecasting horizon and the effective range of external textual information remains an important direction for future work.

5 Conclusion

We studied news selection for time-series forecasting as a forecasting-feedback-driven decision problem rather than an article-level relevance filtering task. Our results support the central observation that semantically relevant news is not necessarily useful for forecasting, and that downstream forecasting loss provides a more direct learning signal for deciding how external information should be used. By formulating news selection as sequential decisions, the proposed framework learns forecasting-aware information use conditioned on both the numerical history and the context already retained.

Across electricity demand, exchange rate, and Bitcoin forecasting, offline RL provides an effective mechanism for learning selection policies, although the relative performance of individual RL algorithms varies across tasks. CQL performs strongly on electricity and exchange rate and remains competitive on Bitcoin. The forecasting-verified reflection memory reuses historically verified selection corrections across forecasting instances. In the reported configurations, reflection improves the CQL results on the two financial tasks, with the clearest gain on exchange-rate forecasting, but does not provide an additional gain on electricity.

Several limitations remain. The reported end-to-end comparisons include adaptation of separately trained forecasters to their corresponding information-selection regimes, so improvements should be interpreted at the forecasting-system level rather than solely as isolated inference-time selector effects. In addition, the usefulness of external information depends strongly on the forecasting horizon, which is selected using held-out validation data in the present study. Future work could study joint adaptation of the forecasting horizon and information window, as well as continual forecasting settings in which selection policies and verified memories evolve as new events and forecasting feedback become available.

AI USE STATEMENT

In this work, we used generative AI tools to assist with methodological discussion, provide feedback on experimental design and result interpretation, and support the drafting and polishing of portions of the manuscript. We did not use generative AI tools to generate synthetic datasets or to autonomously produce experimental results. We also did not rely on generative AI tools as the sole basis for mathematical claims. All AI-assisted methodological suggestions, manuscript text, and interpretations were reviewed by the authors. Experimental implementations and reported results were produced and verified by the authors, and citations and scientific claims were checked before inclusion in the manuscript. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011. doi: 10.1016/j.jocs.2010.12.007.

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. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5132940b1bced8a7b28e9695d49d435a-Abstract-Conference.html.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2327–2333, 2015.

Yuwei Fu, Di Wu, and Benoit Boulet. Reinforcement learning based dynamic model combination for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6639–6647, 2022. doi: 10.1609/aaai.v36i6.20618. URL https://ojs.aaai.org/index.php/AAAI/article/view/20618.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3eb7ca52e8207697361b2c0fb3926511-Abstract-Conference.html.

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. doi: 10.1609/aaai.v38i21.30383. URL https://ojs.aaai.org/index.php/AAAI/article/view/30383.

Shuning Jia, Baijun Song, Canming Ye, and Chun Yuan. M3Time: LLM-enhanced multi-modal, multi-scale, and multi-frequency multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22265–22273, 2026. doi: 10.1609/aaai.v40i27.39383. URL https://ojs.aaai.org/index.php/AAAI/article/view/39383.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Timexl: Explainable multi-modal time series prediction with llm-in-the-loop. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4520.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/680b2a8135b9c71278a09cafb605869e-Abstract-Conference.html.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Geon Lee, Wenchao Yu, Kijung Shin, Wei Cheng, and Haifeng Chen. TimeCAP: Learning to contextualize, augment, and predict time series events with large language model agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18082–18090, 2025. doi: 10.1609/aaai.v39i17.33989. URL https://ojs.aaai.org/index.php/AAAI/article/view/33989.

Shuqi Li, Weiheng Liao, Yuhan Chen, and Rui Yan. PEN: Prediction-explanation network to forecast stock price movement with better explainability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5187–5194, 2023. doi: 10.1609/aaai.v37i4.25648. URL https://ojs.aaai.org/index.php/AAAI/article/view/25648.

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. doi: 10.1609/aaai.v39i18.34067. URL https://ojs.aaai.org/index.php/AAAI/article/view/34067.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. CALF: Aligning LLMs for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025b. doi: 10.1609/aaai.v39i18.34082. URL https://ojs.aaai.org/index.php/AAAI/article/view/34082.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3882.

Wenzhe Niu, Zongxia Xie, Yanru Sun, Wei He, Man Xu, and Chao Hao. LangTime: A language-guided unified model for time series forecasting with proximal policy optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46712–46734, 2025. URL https://proceedings.mlr.press/v267/niu25e.html.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. TEST: Text prototype aligned embedding to activate LLM's ability for time series. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a4352e2c9d93582898a2a20e1f514e8f-Abstract-Conference.html.

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

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. ChatTime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025a. doi: 10.1609/aaai.v39i12.33384. URL https://ojs.aaai.org/index.php/AAAI/article/view/33384.

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. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/34a1fc7890141f1ada3d8bc6199cce07-Abstract-Conference.html.

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.

Xubin Wang, Jianfei Wu, Yuan Yichen, Deyu Cai, Mingzhe Li, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64937–64954, 2025b. URL https://proceedings.mlr.press/v267/wang25dp.html.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66887–66944, 2025. URL https://proceedings.mlr.press/v267/williams25a.html.

Lan Wu, Xuebin Wang, Chenglong Ge, Ruijuan Chu, and Linyu Wang. ExoTimer: Leveraging large language models for time series forecasting with exogenous variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26956–26964, 2026. doi: 10.1609/aaai.v40i32.39908. URL https://ojs.aaai.org/index.php/AAAI/article/view/39908.

Runze Yan, Xun Shen, Akifumi Wachi, Sebastien Gros, Anni Zhao, and Xiao Hu. Offline guarded safe reinforcement learning for medical treatment optimization strategies. In *Advances in Neural Information Processing Systems*, volume 38, pp. 108662–108703, 2025. doi: 10.52202/085713-3275.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78478–78497, 2025. URL https://proceedings.mlr.press/v267/zhong25a.html.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One Fits All: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html.

Yun Zhu, Jia-Chen Gu, Caitlin Sikora, Ho Ko, Yinxiao Liu, Chu-Cheng Lin, Lei Shu, Liangchen Luo, Lei Meng, Bang Liu, and Jindong Chen. Accelerating inference of retrieval-augmented generation via sparse context selection. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/411fa9d368b5485be4c6bb62615b365e-Abstract-Conference.html.

A Full Agent and Training Details

The main paper presents the news-aware forecasting agent and its selection policy in compact form. This appendix gives additional details on forecasting inputs, state construction, actions, reward, offline trajectory construction, policy learning, and reflection-memory features.

Forecasting task and candidate construction. At forecast origin T, the forecaster observes numerical history $\mathbf{x}_{1:T}$, structured context $\mathbf{c}_T$, and candidate news $\mathcal{N}_T = (n_1, \ldots, n_M)$, and predicts $\mathbf{x}_{T+1:T+H}$. Each dataset uses a fixed numerical input window and final prediction horizon, which are shared by all compared methods within that domain and summarized in Appendix C. Candidate pools contain only articles available before the forecast origin. The electricity and exchange-rate tasks use Australian news, while the Bitcoin task uses cryptocurrency-specific news. Each candidate is processed once, and all policy-learning methods use the same candidate order within an experiment.

State representation. At step i, the state is

$$s_i = \big(\phi_x(\mathbf{x}_{1:T}), \phi_c(\mathbf{c}_T), \phi_n(n_i), \phi_s(\mathcal{S}_{i-1})\big). \tag{8}$$

The numerical block ϕ_x represents recent temporal behavior, including trend, seasonal information, volatility, and numerical-history features. The structured-context block ϕ_c represents calendar and available exogenous variables. The candidate block ϕ_n contains the current article representation and metadata. The selected-context block ϕ_s summarizes previously retained information and remaining context capacity.

Action space and transition. The action space is $\mathcal{A} = \{\textsc{Keep}, \textsc{Drop}, \textsc{Summarize}\}$. With fixed summarization operator $g(\cdot)$ and $\mathcal{S}_0 = \emptyset$,

$$\mathcal{S}_i = \begin{cases} \mathcal{S}_{i-1} \cup \{n_i\}, & a_i = \textsc{Keep}, \\ \mathcal{S}_{i-1}, & a_i = \textsc{Drop}, \\ \mathcal{S}_{i-1} \cup \{g(n_i)\}, & a_i = \textsc{Summarize}. \end{cases} \tag{9}$$

Keep retains the article, Drop contributes no textual context, and Summarize inserts a compressed representation using a fixed summary-length rule. Dropping all candidates is a valid trajectory.

Forecasting-loss reward. After the final selection decision, the forecaster is evaluated with $\mathcal{S}_M$. We use the unnormalized change in forecasting loss relative to the no-news prediction:

$$R_T = \mathcal{L}(\emptyset) - \mathcal{L}(\mathcal{S}_M). \tag{10}$$

Thus, $R_T > 0$ indicates lower forecasting loss with the selected context, $R_T < 0$ indicates higher loss, and $R_T = 0$ indicates no change. Intermediate rewards are zero. All selectors use the same maximum context budget.

Offline behavior data. We construct the offline dataset from historical forecasting instances in the training split. A fixed behavior selector produces $(a_1, \ldots, a_M)$ for each candidate sequence, and the resulting episode is stored as $\tau = (s_1, a_1, r_1, \ldots, s_M, a_M, r_M)$. The dataset is $\mathcal{D} = \{\tau^{(1)}, \ldots, \tau^{(N)}\}$, and its empirical behavior distribution is denoted by $\hat{\beta}(a \mid s)$. The same trajectories are used for DQN, BCQ, CQL, and Guarded CQL.

Offline policy optimization. Our primary implementation uses discrete-action CQL(H) (Kumar et al., 2020):

$$\begin{aligned} \mathcal{L}_{\mathrm{CQL}}(Q) = & \frac{1}{2}\mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\big[(Q(s,a) - y)^2\big] \\ & + \alpha \mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_{a'\in\mathcal{A}} \exp Q(s,a') - \mathbb{E}_{a\sim\hat{\beta}(\cdot|s)} Q(s,a)\right], \end{aligned} \tag{11}$$

where $y = r + \gamma \max_{a'\in\mathcal{A}} Q_{\mathrm{tgt}}(s', a')$ and the bootstrap term is removed at terminal states. The first term is Bellman regression and the second is the CQL conservative penalty. The final selector is $\pi(s) = \arg\max_{a\in\mathcal{A}} Q(s,a)$. DQN uses the same MDP without the conservative regularizer; BCQ and Guarded CQL change only the policy-learning layer.

All methods within a domain use identical chronological splits, numerical input windows, final prediction horizons, candidate-news pools, forecasting backbones, and evaluation scripts. We use Gemma-2-9B with LoRA adaptation as the forecasting backbone and Qwen2.5-72B-AWQ for LLM-based news processing. Hyperparameters and checkpoints are selected using validation data only.

Counterfactual reflection memory. The reflection memory is constructed exclusively from the training split. We first identify high-error training episodes using forecasting loss, where the current implementation retains episodes above the 80th loss percentile within each region. For an episode with the logged action sequence

$$\mathbf{a} = (a_1, \ldots, a_M), \tag{12}$$

we evaluate a single-action counterfactual intervention for each candidate news item. Specifically, the counterfactual action sequence for position i is

$$\widetilde{\mathbf{a}}^{(i)} = (a_1, \ldots, a_{i-1}, g(a_i), a_{i+1}, \ldots, a_M), \tag{13}$$

where

$$g(\textsc{Drop}) = \textsc{Keep}, \qquad g(\textsc{Keep}) = \textsc{Drop}, \qquad g(\textsc{Summarize}) = \textsc{Keep}. \tag{14}$$

All other actions, including the suffix $a_{i+1}, \ldots, a_M$, remain fixed at their originally logged values. The selector is not re-executed after changing a_i. We reconstruct the final news context in the original candidate order and evaluate it using the same fixed forecasting checkpoint that scores the corresponding factual trajectory and the same evaluation procedure.

Let $\mathcal{L}$ and $\widetilde{\mathcal{L}}_i$ denote the factual and counterfactual forecasting losses, respectively. The counterfactual improvement is

$$\Delta_i = \mathcal{L} - \widetilde{\mathcal{L}}_i, \qquad \Delta_i^{\mathrm{rel}} = \frac{\mathcal{L} - \widetilde{\mathcal{L}}_i}{\mathcal{L} + \epsilon}. \tag{15}$$

where ϵ is a small constant used for numerical stability. A reflection memory is retained only when the counterfactual action strictly reduces forecasting loss. Each retained memory records the source training instance, news identifier, original action, corrected action, miss type, factual and counterfactual losses, relative improvement, region metadata, and news embedding.

Memory retrieval. At decision step i, the retrieval query is the embedding of the current candidate news item's long summary. Embeddings are produced using `Qwen/Qwen3-Embedding-0.6B` with a maximum input length of 2048 tokens, mean pooling, truncation to 64 dimensions, and L2 normalization. For query embedding $\mathbf{q}_i$ and memory embedding $\mathbf{e}_m$, relevance is defined by cosine similarity:

$$s(i, m) = \frac{\mathbf{q}_i^\top \mathbf{e}_m}{\|\mathbf{q}_i\|_2 \|\mathbf{e}_m\|_2}. \tag{16}$$

The current configuration retrieves the top $k = 3$ training memories with $s(i, m) \geq 0.8$. For the electricity dataset, retrieval is additionally restricted using the categorized region of the current news item. Region filtering is disabled for the Bitcoin and exchange-rate datasets. Memories originating from the same instance are excluded when applicable.

Test-time information and state augmentation. The reflection memory bank is fixed before evaluation and contains only memories constructed from the training split. At validation and test time, retrieval uses the current candidate news item, its semantic embedding, and the categorized region where region filtering is enabled. It does not use the current target value, the current forecasting error, validation or test labels, or memories constructed from validation or test examples. The retrieved memory texts are not directly inserted into the forecasting prompt. Instead, each memory is represented by the one-hot encoded miss type, original action, corrected action, cosine similarity, and relative loss improvement. The top-k representations, together with the memory count and availability indicator, are appended to the selector state.

B Conservative Offline Learning Analysis

This appendix restates standard CQL policy-evaluation intuition under idealized assumptions to motivate conservative learning in our setting. We analyze why conservative value learning is appropriate for the offline news-selection component of our forecasting agent. The analysis follows the conservative policy-evaluation and finite-sample arguments of CQL (Kumar et al., 2020), specialized to our discrete action space $\mathcal{A} = \{\textsc{Keep}, \textsc{Drop}, \textsc{Summarize}\}$ and forecasting-loss reward $R_T = \mathcal{L}(\emptyset) - \mathcal{L}(\mathcal{S}_M)$. Let $\beta(a \mid s)$ denote the behavior policy represented by the logged trajectories

and $\pi(a \mid s)$ a target news-selection policy. As in the standard CQL analysis, we assume $\beta(a \mid s) > 0$ whenever $\pi(a \mid s) > 0$ and bounded rewards.

Define

$$D_{\mathrm{CQL}}(\pi, \beta)(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \left(\frac{\pi(a \mid s)}{\beta(a \mid s)} - 1 \right) = \sum_{a \in \mathcal{A}} \frac{(\pi(a \mid s) - \beta(a \mid s))^2}{\beta(a \mid s)} \geq 0. \tag{17}$$

This term increases when the target policy moves away from the action distribution represented in the offline behavior data.

Canonical conservative policy evaluation. Following the policy-evaluation analysis of CQL, consider

$$\begin{aligned} \widehat{Q}_{k+1} \leftarrow \arg\min_{Q} \alpha \left(\mathbb{E}_{s \sim d^{\beta}, a \sim \pi(\cdot|s)} Q(s,a) - \mathbb{E}_{s \sim d^{\beta}, a \sim \beta(\cdot|s)} Q(s,a) \right) \\ + \frac{1}{2} \mathbb{E}_{(s,a,s') \sim \mathcal{D}} \left[\left(Q(s,a) - \widehat{\mathcal{B}}^{\pi} \widehat{Q}_k(s,a) \right)^2 \right]. \end{aligned} \tag{18}$$

The first term lowers values under the target action distribution relative to actions represented by the behavior distribution, while the second performs Bellman regression. This objective is the analytical policy-evaluation form used by CQL to expose the conservative mechanism.

Policy evaluation without sampling error. With exact tabular policy evaluation and an exact Bellman operator, the first-order condition of Equation 18 gives

$$\widehat{Q}_{k+1}(s,a) = \mathcal{B}^{\pi} \widehat{Q}_k(s,a) - \alpha \left(\frac{\pi(a \mid s)}{\beta(a \mid s)} - 1 \right). \tag{19}$$

Taking expectation over $a \sim \pi(\cdot \mid s)$ yields

$$\widehat{V}^{\pi}_{k+1} = \mathcal{B}^{\pi} \widehat{V}^{\pi}_{k} - \alpha D_{\mathrm{CQL}}(\pi, \beta). \tag{20}$$

At the fixed point,

$$\widehat{V}^{\pi} = r^{\pi} + \gamma P^{\pi} \widehat{V}^{\pi} - \alpha D_{\mathrm{CQL}}(\pi, \beta), \tag{21}$$

and therefore

$$\widehat{V}^{\pi} = V^{\pi} - \alpha (I - \gamma P^{\pi})^{-1} D_{\mathrm{CQL}}(\pi, \beta). \tag{22}$$

Since $(I - \gamma P^{\pi})^{-1}$ has non-negative entries and $D_{\mathrm{CQL}}(\pi, \beta) \geq 0$, Equation 22 implies

$$\widehat{V}^{\pi}(s) \leq V^{\pi}(s), \qquad \forall s. \tag{23}$$

Thus, when the Bellman operator is known exactly, the conservative correction directly produces a lower-bound estimate of the policy value. This is the exact-evaluation case of the CQL lower-bound analysis.

Finite-sample Bellman error. In offline learning, $\mathcal{B}^{\pi}$ is estimated from finitely many logged transitions. Let $\widehat{\mathcal{B}}^{\pi}$ denote the empirical Bellman operator. Under the standard reward and transition concentration conditions used in CQL, the empirical backup satisfies, with probability at least $1 - \delta$,

$$\left| \widehat{\mathcal{B}}^{\pi} Q(s,a) - \mathcal{B}^{\pi} Q(s,a) \right| \leq \epsilon_{\mathcal{D}}(s,a), \qquad \epsilon_{\mathcal{D}}(s,a) = O\left(\frac{1}{\sqrt{N(s,a)}} \right), \tag{24}$$

where $N(s,a)$ is the number of logged transitions for (s,a). Hence, state–action pairs with weak behavior coverage can have larger empirical Bellman error. Define the policy-averaged sampling error

$$\bar{\epsilon}^{\pi}_{\mathcal{D}}(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[\epsilon_{\mathcal{D}}(s,a)]. \tag{25}$$

Proposition 1 (Finite-sample conservative value bound)**.** *Let $\widehat{V}^{\pi}$ be the fixed-point value obtained by conservative policy evaluation with the empirical Bellman operator. Under Equation 24, with probability at least $1 - \delta$,*

$$\widehat{V}^{\pi} \leq V^{\pi} + (I - \gamma P^{\pi})^{-1} \left(\bar{\epsilon}^{\pi}_{\mathcal{D}} - \alpha D_{\mathrm{CQL}}(\pi, \beta) \right). \tag{26}$$

Proof. Equation 24 implies

$$\widehat{\mathcal{B}}^{\pi}\widehat{Q}(s,a) \le \mathcal{B}^{\pi}\widehat{Q}(s,a) + \epsilon_{\mathcal{D}}(s,a). \tag{27}$$

The empirical counterpart of Equation 19 is

$$\widehat{Q}_{k+1}(s,a) = \widehat{\mathcal{B}}^{\pi}\widehat{Q}_k(s,a) - \alpha\left(\frac{\pi(a \mid s)}{\beta(a \mid s)} - 1\right). \tag{28}$$

Combining Equations 27 and 28, then taking expectation over $a \sim \pi(\cdot \mid s)$, gives

$$\widehat{V}^{\pi}_{k+1} \le r^{\pi} + \gamma P^{\pi}\widehat{V}^{\pi}_{k} + \bar{\epsilon}^{\pi}_{\mathcal{D}} - \alpha D_{\mathrm{CQL}}(\pi,\beta). \tag{29}$$

At the fixed point,

$$(I - \gamma P^{\pi})\widehat{V}^{\pi} \le r^{\pi} + \bar{\epsilon}^{\pi}_{\mathcal{D}} - \alpha D_{\mathrm{CQL}}(\pi,\beta). \tag{30}$$

Because $(I - \gamma P^{\pi})^{-1}$ has non-negative entries,

$$\widehat{V}^{\pi} \le (I - \gamma P^{\pi})^{-1}\left(r^{\pi} + \bar{\epsilon}^{\pi}_{\mathcal{D}} - \alpha D_{\mathrm{CQL}}(\pi,\beta)\right). \tag{31}$$

Using $V^{\pi} = (I - \gamma P^{\pi})^{-1} r^{\pi}$ yields Equation 26. □

Equation 26 makes the role of limited offline coverage explicit. The term $\bar{\epsilon}^{\pi}_{\mathcal{D}}$ captures finite-sample error in the empirical Bellman backup, while $\alpha D_{\mathrm{CQL}}(\pi,\beta)$ is the opposing conservative correction. Therefore, if

$$\alpha D_{\mathrm{CQL}}(\pi,\beta)(s) \ge \bar{\epsilon}^{\pi}_{\mathcal{D}}(s), \qquad \forall s, \tag{32}$$

then the correction term in Equation 26 is non-positive and

$$\widehat{V}^{\pi}(s) \le V^{\pi}(s), \qquad \forall s. \tag{33}$$

The exact-operator result in Equation 23 is recovered when the sampling error is zero.

Comparison with unregularized offline Q-learning. For the corresponding unregularized value update, setting $\alpha = 0$ gives

$$\widehat{V}^{\pi}_{\mathrm{unreg}} \le V^{\pi} + (I - \gamma P^{\pi})^{-1}\bar{\epsilon}^{\pi}_{\mathcal{D}}. \tag{34}$$

CQL instead replaces the finite-sample term by $\bar{\epsilon}^{\pi}_{\mathcal{D}} - \alpha D_{\mathrm{CQL}}(\pi,\beta)$. Hence, whenever the learned policy deviates from the behavior distribution, the conservative correction reduces the upper bound on optimistic value estimation. This is the mechanism absent from unregularized DQN-style offline learning, where bootstrapping and maximization can propagate erroneously high values for weakly supported actions.

Implication for the agent's news-selection policy. In our MDP, the logged frequencies of KEEP, DROP, and SUMMARIZE can be highly uneven. A weakly represented action has fewer supporting transitions and therefore a less reliable empirical Bellman estimate. An unregularized offline value learner can propagate an optimistic estimate for such an action through subsequent Q-learning updates. CQL counteracts this effect by lowering estimated values as the target policy moves away from the behavior distribution. Proposition 1 therefore specializes the CQL finite-sample argument to our news-selection setting: limited action coverage contributes an estimation-error term, while conservative regularization provides an explicit opposing correction.

The analytical objective in Equation 18 is used to expose this mechanism. Our implementation uses the discrete CQL(H) objective in Equation 11; the log-sum-exp regularizer is the practical Q-learning instantiation of the same CQL framework (Kumar et al., 2020).

C ADDITIONAL EXPERIMENTAL DETAILS

This appendix provides supporting experimental information that is intentionally separated from the main forecasting table: final task configurations, horizon sensitivity, offline behavior-dataset statistics, context usage, round-wise reflection behavior, and the larger electricity benchmark.

Datasets and final evaluation configuration. Table 4 summarizes the final task configuration. All methods within a domain use the same chronological split, input construction, candidate-news pool, forecasting horizon, forecasting backbone, and evaluation script.

Baseline details. *No News* uses only numerical history and structured forecasting context. *LLM-selected News* selects candidate news using LLM-based semantic and task relevance. DQN uses the proposed sequential MDP without conservative offline regularization. BCQ constrains policy improvement toward behavior-supported decisions. CQL uses Equation 11. Guarded CQL uses the same CQL learner with the additional guard employed by our implementation. All RL methods use the same state representation, action space, forecasting-loss reward, and offline behavior trajectories within each dataset.

Table 4: Final forecasting configuration for the three datasets.

Dataset	Frequency	Input Window	Final Horizon	Test Size	News Source
Electricity	30 min	48 steps	$H = 1$ (30 min)	250	Australian general news
AUD/USD	Daily	7 days	$H = 7$	50	Australian general news
Bitcoin	Daily	7 days	$H = 7$	50	Cryptocurrency news

C.1 Forecast-Horizon Analysis

The forecasting horizon changes both the intrinsic difficulty of the task and the interval over which external information can affect the target. Table 5 summarizes the horizon sensitivity observed during model development. Because Table 5 reports development-stage No News and LLM-news pipelines used for horizon analysis, its absolute values are not expected to coincide with the final end-to-end checkpoints reported in Table 2. The table compares the No News and LLM-news pipelines before RL selection; its purpose is to characterize how the usefulness of text changes with H, not to select a horizon from final test performance. In the final protocol, candidate horizons are compared on the chronological validation split, the selected H is frozen, and all methods are then evaluated on the same test horizon.

The effect of news is not monotonic in H. For electricity, LLM-selected news helps at the one-step horizon but becomes worse than No News at longer horizons. For AUD/USD, news is mildly beneficial at $H \in \{1, 3, 7\}$ but not at $H = 14$. Bitcoin is more unstable: the apparent RMSE advantage of news at $H = 14$ is driven by a small number of large No News errors. Removing the five largest No News outliers reverses the comparison (RMSE 4531 for No News versus 7151 for LLM news), and the No News predictor also has the lower median error and a higher per-instance win count. These observations motivate treating horizon choice as part of the forecasting task specification rather than assuming that a single horizon is optimal for textual context.

Table 5: Horizon sensitivity of the No News and LLM-news pipelines. The final main experiments use $H = 1$ for electricity and $H = 7$ for AUD/USD and Bitcoin.

Dataset	H	No-news RMSE	News RMSE	No-news R^2	News R^2
Electricity	1	66.5	61.0	0.9993	0.9994
	12	287.6	319.0	0.9874	0.9845
	24	313.8	358.3	0.9850	0.9805
	48	323.3	352.6	0.9841	0.9811
AUD/USD	1	0.003516	0.003470	0.9710	0.9717
	3	0.005541	0.005475	0.9279	0.9296
	7	0.009162	0.008801	0.8028	0.8180
	14	0.009000	0.009318	0.8097	0.7960
Bitcoin	1	1664.5	1702.7	0.9676	0.9661
	3	3252.3	3224.1	0.8765	0.8786
	7	4334.4	5295.9	0.7806	0.6725
	14	10215.2	8419.1	-0.2186	0.1722

C.2 Offline Behavior-Dataset Statistics

Because the motivation for conservative offline RL depends on the support of the logged behavior data, we report the scale and composition of the fixed offline trajectories used to train the news-

selection policies. One forecasting instance corresponds to one episode, and one candidate news item corresponds to one transition. The final trajectory artifacts provide episode counts, reward-valid episode counts, transition counts, mean candidate counts, and empirical action frequencies for all three datasets; these statistics are reported directly in Tables 6 and 7.

Table 6: Scale of the offline RL datasets used in the reported configurations.

Dataset	Episodes	Reward-valid	Transitions	Mean candidates/episode
Electricity ($H = 1$)	12,000	11,999	1,972,443	164.37
AUD/USD ($H = 7$)	726	724	260,463	358.76
Bitcoin ($H = 7$)	726	726	23,989	33.04

Table 6 shows that the three domains differ substantially in offline-dataset scale and candidate-pool size. For electricity, the mean candidate count is derived directly from the reported trajectory size, $1{,}972{,}443/12{,}000 = 164.37$. The complete electricity source pool contains 43,860 episodes and 7,241,461 candidate transitions, with mean candidate count 165.1, median 163, 90th percentile 232, and maximum 318. These full-pool percentiles are reported only as source-pool context; they are not substituted for the exact distribution of the sampled 12,000-episode training trajectory.

Behavior-policy construction. The offline trajectory is generated once from the upstream LLM-selected news IDs. Articles not selected by the upstream selector are assigned DROP. Selected articles are assigned KEEP when their `long_summary` length is at most a dataset-specific threshold and SUMMARIZE otherwise. The threshold is 870 characters for electricity and AUD/USD and 656 characters for Bitcoin. This fixed logged trajectory is shared by DQN, BCQ, CQL, and Guarded CQL for all primary optimizer comparisons.

Empirical action support. Table 7 summarizes the empirical action support of the fixed offline RL trajectories. Electricity is highly imbalanced: 98.70% of transitions correspond to DROP, compared with 0.48% KEEP and 0.82% SUMMARIZE. AUD/USD is similarly drop-dominated, with 98.45% DROP, 0.65% KEEP, and 0.90% SUMMARIZE. Bitcoin is less extreme but remains imbalanced, with 82.05% DROP, 8.53% KEEP, and 9.42% SUMMARIZE. These aggregate action frequencies show that the logged behavior data are strongly imbalanced and motivate evaluating conservative offline-RL methods in this setting. They do not by themselves establish state-wise action support for the learned policy.

Table 7: Empirical action support in the offline RL trajectories.

Dataset	KEEP(%)	DROP(%)	SUMMARIZE(%)	Transitions
Electricity ($H = 1$)	0.48	98.70	0.82	1,972,443
AUD/USD ($H = 7$)	0.65	98.45	0.90	260,463
Bitcoin ($H = 7$)	8.53	82.05	9.42	23,989

C.3 POLICY BEHAVIOR AND CONTEXT USAGE

Table 8 reports the average amount of external context consumed by each method using the selected checkpoints in the standard test sets. The electricity result is especially informative: CQL selects no news on average, whereas DQN selects substantially more context. The financial tasks show a different regime in which CQL and reflection retain a nonzero amount of news.

C.4 ROUND-WISE REFLECTION BEHAVIOR

We additionally examine how the CQL and reflection policies behave across optimization rounds. Here, a "round" denotes a successive training/refinement checkpoint under the same predefined method configuration. The underlying offline behavior data used for the primary RL comparison are fixed; the round index does not denote regeneration of a new method-specific offline behavior dataset.

Table 8: Average selected news and selected-news token counts for the standard test sets.

Dataset	Method	Avg. selected news	Avg. selected tokens
Electricity	No News	0.00	0.0
	LLM-selected News	1.95	368.7
	DQN	5.06	297.9
	BCQ	0.07	3.6
	Guarded CQL	0.15	14.1
	CQL	0.00	0.0
	Reflection (excl. omission)	0.45	53.1
	Reflection (incl. omission)	0.32	37.5
AUD/USD	No News	0.00	0.0
	LLM-selected News	1.76	288.9
	DQN	9.88	914.0
	BCQ	2.78	211.5
	Guarded CQL	0.74	54.2
	CQL	2.72	189.9
	Reflection (excl. omission)	2.52	248.9
	Reflection (incl. omission)	1.90	171.2
Bitcoin	No News	0.00	0.0
	LLM-selected News	5.64	785.9
	DQN	8.72	872.4
	BCQ	3.94	345.1
	Guarded CQL	2.30	199.0
	CQL	3.70	294.3
	Reflection (excl. omission)	3.08	252.3
	Reflection (incl. omission)	2.94	206.3

For reflection, we additionally track two memory variants across rounds to illustrate the optimization behavior of the refinement stage. These round-wise results are supplementary diagnostics and are reported separately from the single *CQL + Reflection* result in the main table.

Table 9 reports test RMSE at the recorded round checkpoints. Checkpoint selection is performed using the chronological validation split and the selected checkpoint is frozen before test evaluation. Test-set values in this table are reported only to characterize round-wise behavior and are not used for checkpoint selection.

Table 9: Round-wise RMSE for the base CQL selector and two supplementary reflection-memory variants. Rounds denote successive optimization checkpoints; test values are shown only for diagnostic analysis.

Dataset	Round	CQL	Reflection Excl.	Reflection Full
Electricity	0	59.7	58.0	56.4
	1	57.7	56.3	58.1
	2	54.7	93.9	88.8
AUD/USD	0	0.008586	0.008924	0.008394
	1	0.008509	0.008615	0.008661
	2	0.008857	0.009159	0.008588
Bitcoin	0	5161.58	5233.48	5229.94
	1	5067.41	4792.69	4932.49
	2	5506.30	5508.10	5663.33

The round-wise results show that additional optimization or reflection refinement is not monotonically beneficial. For example, later rounds substantially degrade both reported reflection variants on electricity, while stronger checkpoints occur earlier for the financial tasks. We therefore use the validation split for checkpoint selection and freeze the selected checkpoint before evaluating the test set. The round-wise test values are reported only as supplementary diagnostics.

C.5 LARGER ELECTRICITY BENCHMARK

We additionally evaluate the methods on the larger electricity benchmark ($n = 4800$) reported in the experimental artifacts. This experiment provides an additional robustness check on a substantially larger test set.

Table 10 shows that CQL and BCQ achieve the strongest base-selector performance on this benchmark. Neither of the two supplementary reflection variants improves over the base CQL selector. Thus, the larger electricity benchmark provides an additional example in which verified reflection memory does not yield a forecasting improvement.

Table 10: Results on the larger electricity benchmark ($n = 4800$). The two reflection rows are supplementary variants reported to characterize reflection behavior on the larger test set.

Method	RMSE ↓	MAE ↓	MAPE ↓	R^2 ↑	Avg. News
No News	98.5	61.1	3.64%	0.999	0.00
LLM-selected News	88.2	56.1	1.95%	0.999	2.30
DQN	90.4	56.4	2.00%	0.999	3.56
BCQ	**84.5**	53.3	**1.89%**	0.999	0.06
Guarded CQL	84.8	54.1	1.93%	0.999	0.20
CQL	**84.5**	**53.2**	**1.89%**	0.999	0.01
Reflection excl. omission	85.2	54.2	1.93%	0.999	0.38
Reflection (incl. omission)	87.1	54.9	1.94%	0.999	0.42