BEYOND ONE-SHOT FORECASTING: PREDICT-THEN-REVEAL TIME SERIES PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term time-series forecasting typically predicts the entire future horizon in one shot from a fixed historical context. However, as the forecast unfolds, earlier observations within the prediction horizon become available and can provide valuable evidence for refining the still-unseen future. We introduce **Predict-then-Reveal Forecasting**, a forecasting paradigm in which predictions are progressively committed, revealed, and subsequently used to improve later forecasts. Under this paradigm, we propose **PRISM**, a progressive forecasting framework that maintains multiple candidate future trajectories and continuously updates them using newly revealed observations. At each stage, PRISM jointly adjusts the relative beliefs over candidate futures according to their consistency with observed feedback and revises their unrevealed trajectories based on the resulting residual patterns. To learn effective intermediate revisions, PRISM further exploits complete future outcomes as privileged supervision during training, while inference relies exclusively on observations that have already been revealed. This design enables forecasts to evolve without test-time parameter optimization or access to future targets. Extensive experiments across diverse real-world time-series benchmarks and prediction horizons demonstrate that PRISM consistently outperforms strong standard forecasting and online adaptation methods. The code is available at https://anonymous.4open.science/r/PRISM-9BC4/.

1 INTRODUCTION

Long-term time-series forecasting has witnessed substantial progress with the rapid development of deep learning, ranging from improved temporal representations to more sophisticated modeling of long-range and cross-variable dependencies (Lim & Zohren, 2021; Liang et al., 2024; Wang et al., 2024b). Despite these advances, most existing methods still follow a common **one-shot protocol, predicting the entire future horizon once from a fixed historical window.** Reliance on a fixed historical context becomes increasingly restrictive over longer horizons, where prediction errors tend to grow as forecasts extend further into the future. Meanwhile, newly observed outcomes provide valuable information about the evolving system that remains unused by one-shot forecasting.

We study **Predict then Reveal Forecasting, which reframes long horizon prediction as a progressive process of forecasting and correction**. At each stage, the model first predicts the next block and only then observes its ground truth, using the revealed outcomes to revise the remaining unseen future. This predict, reveal, and correct protocol respects the forecasting constraint, since future targets are never accessed before prediction, while allowing errors revealed at earlier stages to inform subsequent forecasts. By continuously incorporating newly revealed observations, the forecast can better adapt to the evolving trajectory throughout the prediction horizon.

This perspective is distinct from conventional online forecasting and adaptation. Existing methods primarily exploit incoming observations to update model parameters, representations, or adaptation modules for tracking evolving data distributions (Pham et al., 2023; Zhang et al., 2023; Kim et al., 2025). In many cases, adaptation is performed only after sufficient new observations have been accumulated, introducing an inherent delay between distributional changes and model updates. In contrast, our goal is to correct the *current forecast* within the same prediction horizon, without relying on repeated model adaptation. Newly revealed observations provide explicit residual information about the issued prediction, but effective correction requires more than locally compensating

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.

for the latest error: the feedback must be translated into appropriate changes to the entire remaining trajectory. The key challenge is therefore how to use progressively revealed prediction errors to produce reliable corrections for the still-unobserved future.

To this end, we propose **PRISM**, a progressive forecasting framework that maintains multiple candidate futures and continuously refines them with revealed observations. At each stage, PRISM uses the observed feedback to both *reweight* candidate trajectories according to their consistency with the revealed block and *revise* their unrevealed portions based on the resulting residuals. In this way, PRISM progressively updates both *which future to trust* and *what the remaining future should look like*. To guide these intermediate updates, PRISM further introduces training-time future supervision, which uses the complete target to provide learning signals for progressive revision. This supervision is removed at inference, where all updates rely solely on already revealed observations. Extensive experiments across diverse real-world benchmarks show that PRISM consistently improves forecasting performance over both standard and online forecasting methods, with further analyses validating the effectiveness of its progressive revision components. **Our contributions are summarized as follows.**

- ❶ We introduce **Predict-then-Reveal Forecasting**, a forecasting paradigm that progressively incorporates newly revealed observations to refine the still-unseen future.
- ❷ We propose **PRISM**, which maintains multiple candidate futures and jointly updates their relative beliefs and unrevealed trajectories as feedback becomes available.
- ❸ We develop a training-time future-guidance mechanism to supervise intermediate revisions while preserving inference based only on revealed observations.
- ❹ Extensive experiments demonstrate consistent improvements over strong standard and online forecasting baselines, with up to **87.9%** and **65.1%** reductions in MSE and MAE.

2 Related Work

Multivariate time-series forecasting. Recent forecasting architectures include Transformer-based models that capture dependencies through attention and MLP-based models that learn temporal and cross-variable mixing (Pati et al., 2026; Chen et al., 2026). Among Transformer-based approaches, iTransformer applies attention to variate tokens (Liu et al., 2024), while Timer-XL extends causal attention to long-context multivariate next-token prediction (Liu et al., 2025). More recently, PMDformer separates patch-level trends from residual shapes and introduces trend-restoration and proximal variable attention (Hu et al., 2026). MLP-based approaches pursue structured mixing with lightweight architectures. TimeMixer combines seasonal and trend components across temporal scales (Wang et al., 2024a), and WPMixer integrates wavelet decomposition, patch representations, and MLP mixing (Murad et al., 2025). XLinear further uses a global endogenous token and MLP-based interactions to incorporate exogenous information (Chen et al., 2026). Recent work further explores multimodal integration of time series and text (Jiang et al., 2025), as well as Kolmogorov–Arnold networks for frequency-specific temporal modeling (Huang et al., 2025).

Online adaptation under distribution shift. Online time-series forecasting studies how to adapt to evolving dynamics from sequential observations (Pham et al., 2023; Zhang et al., 2023; Hong et al., 2026). SOLID selects contextually similar samples to update the prediction layer (Chen et al., 2024). These approaches leverage historical knowledge and adaptation data, while multi-step forecasting faces horizon-dependent delays in complete supervision. Proceed addresses the resulting mismatch through drift-conditioned parameter adjustments (Zhao & Shen, 2025), while ADAPT-Z uses historical gradient information to adjust feature representations (Huang et al., 2026). Related test-time adaptation methods exploit newly available observations: TAFAS uses partial supervision to update calibration modules and revise unobserved predictions (Kim et al., 2025), and COSA learns context-aware output corrections from recent observations (Im & Kwon, 2026). Conventional online learning typically organizes adaptation across successive forecasting windows. PRISM operates within a single target window, combining context re-encoding and innovation-state memory to progressively revise its unissued suffix while keeping the target endpoint fixed.

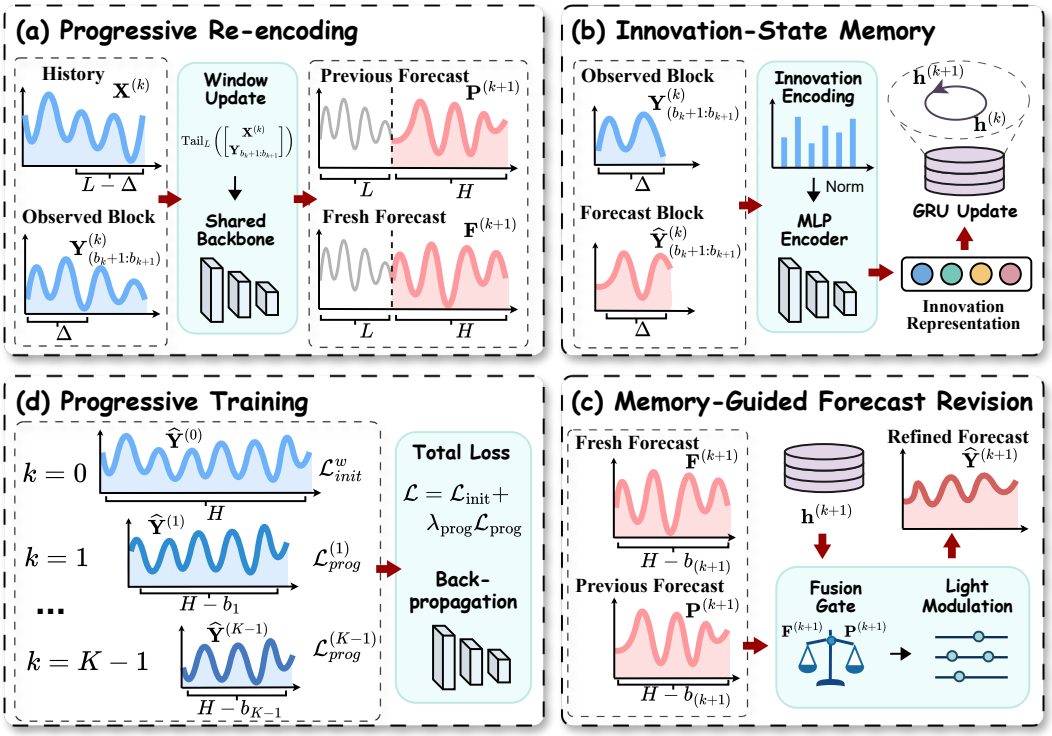


Figure 1: Overview of the proposed PRISM framework. (a) **Progressive Re-encoding** updates the context and regenerates the forecast. (b) **Innovation-State Memory** accumulates historical forecast errors. (c) **Memory-Guided Forecast Revision** fuses and refines fresh and previous forecasts. (d) **Progressive Training** jointly supervises initial and revised forecasts.

3 Problem Formulation

Time-series forecasting. Given a historical window $\mathbf{X} \in \mathbb{R}^{L\times C}$, multivariate time-series forecasting predicts the subsequent H observations $\mathbf{Y} \in \mathbb{R}^{H\times C}$ as $\widehat{\mathbf{Y}} = f_\theta(\mathbf{X})$, where L and C denote the context length and number of variables. In fixed-origin forecasting, all predictions are issued without observing any part of $\mathbf{Y}$.

Predict-then-reveal forecasting. We partition the target horizon into $K = \lceil H/B \rceil$ consecutive blocks, with boundaries $b_k = \min(kB, H)$. At stage $k \in \{0, \dots, K-1\}$, only $\mathbf{X}$ and the revealed prefix $\mathbf{Y}_{1:b_k}$ are available:

$$\widehat{\mathbf{Y}}^{(k)} = \mathcal{F}_\theta(\mathbf{X}, \mathbf{Y}_{1:b_k}) \in \mathbb{R}^{(H-b_k)\times C}, \tag{1}$$

where $\mathbf{Y}_{1:0}$ is empty. The model first issues the next $\Delta_k = b_{k+1} - b_k$ predictions, $\widehat{\mathbf{Y}}^{(k)}_{1:\Delta_k}$; only then is $\mathbf{Y}_{b_k+1:b_{k+1}}$ revealed. Feedback may revise unissued predictions, while issued predictions remain fixed. Evaluation compares the concatenation of all issued blocks with $\mathbf{Y}$. The block length B controls feedback frequency, with $B \geq H$ recovering fixed-origin forecasting.

4 Method

We introduce **PRISM** (**P**rogressive **R**e-encoding with **I**nnovation-**S**tate **M**emory) to revise forecasts using newly revealed observations and accumulated prediction errors. A shared backbone re-encodes the updated rolling context to generate fresh forecasts, while a recurrent memory integrates revealed observations and their deviations from issued predictions. A memory-conditioned gate fuses each fresh forecast with the aligned, unissued tail of the previous forecast, followed by horizon-aware affine modulation to further refine the remaining trajectory.

4.1 PROGRESSIVE INPUT RE-ENCODING

Newly revealed observations provide an updated context for forecasting the remaining trajectory. PRISM incorporates this information by advancing a fixed-length historical window and re-encoding it with a shared lightweight decoupled temporal module f_θ. Starting from $\mathbf{X}^{(0)} = \mathbf{X}$, the module produces the initial forecast $\widehat{\mathbf{Y}}^{(0)} = f_\theta(\mathbf{X}^{(0)})$. After the predictions for block k have been issued and their observations revealed, the context is updated as

$$\mathbf{X}^{(k+1)} = \mathrm{Tail}_L\left([\mathbf{X}^{(k)}; \mathbf{Y}_{b_k+1:b_{k+1}}]\right), \qquad k = 0, \ldots, K-2, \tag{2}$$

where $[\cdot;\cdot]$ denotes temporal concatenation and Tail_L retains the most recent L observations. This update advances the forecast origin while preserving the input length, allowing subsequent predictions to incorporate the observed continuation of the series.

The shared backbone combines trend and multiscale seasonal projections with periodic extrapolation and low-rank cross-variable interactions. Re-encoding the updated context recomputes these components from the newly available observations. Although the backbone returns H predictions at each call, only the prefix covering the remaining target horizon is retained. To make this fresh forecast comparable with the previous trajectory, we also remove the already issued prefix from the preceding stage's forecast:

$$\mathbf{F}^{(k+1)} = \left[f_\theta(\mathbf{X}^{(k+1)})\right]_{1:H-b_{k+1}}, \qquad \mathbf{P}^{(k+1)} = \widehat{\mathbf{Y}}^{(k)}_{\Delta_k+1:H-b_k}. \tag{3}$$

Both trajectories belong to $\mathbb{R}^{(H-b_{k+1})\times C}$ and refer to the same future timestamps. Here, $\mathbf{F}^{(k+1)}$ reflects the updated context, whereas $\mathbf{P}^{(k+1)}$ preserves the preceding stage's predictions for those timestamps. These aligned trajectories form the inputs to the memory-guided revision mechanism described below.

4.2 INNOVATION-STATE MEMORY

The updated context captures the observed evolution of the series, while discrepancies between observations and issued predictions provide complementary information about forecast errors. PRISM accumulates this information through a recurrent memory maintained for each variable. After a non-final block is revealed, the innovation for variable c is defined as $\mathbf{e}_{k,c} = \mathbf{Y}_{b_k+1:b_{k+1},c} - \widehat{\mathbf{Y}}^{(k)}_{1:\Delta_k,c}$. We normalize both the revealed observations and their innovations by the channel-wise scale $s_{k+1,c} = \max(\mathrm{Std}(\mathbf{X}^{(k+1)}_{:,c}), \epsilon)$, obtaining $\widetilde{\mathbf{y}}_{k,c}$ and $\widetilde{\mathbf{e}}_{k,c}$, respectively. A shared feedback encoder then maps these normalized sequences into a compact representation:

$$\mathbf{z}_{k,c} = \mathrm{Enc}_\phi\left(\widetilde{\mathbf{y}}_{k,c}, \widetilde{\mathbf{e}}_{k,c}\right) \in \mathbb{R}^{d_h}. \tag{4}$$

A two-layer MLP encodes clipped blockwise observations and innovation statistics, capturing both the revealed signal and its deviation from the forecast. A GRU then integrates this feedback with the previous memory state to preserve information across successive blocks:

$$\mathbf{h}_c^{(k+1)} = \mathrm{GRU}_\psi\left(\mathbf{z}_{k,c}, \mathbf{h}_c^{(k)}\right), \qquad \mathbf{h}_c^{(0)} = \mathbf{0}. \tag{5}$$

The encoder and GRU parameters are shared across variables, while each variable maintains its own state $\mathbf{h}_c^{(k)} \in \mathbb{R}^{d_h}$, initialized to zero for each forecasting episode. This recurrence allows forecast revision to depend on the accumulated error history as well as the latest discrepancy. The resulting memory conditions the fusion and modulation of the aligned trajectories introduced above.

4.3 MEMORY-GUIDED FORECAST REVISION

The fresh forecast incorporates newly revealed observations, while the aligned previous trajectory retains prior-stage predictions. PRISM reconciles the two trajectories using the innovation memory. For clarity, we omit the stage superscript below and let $R = H - b_{k+1}$ denote the remaining horizon. For each position j and variable c, we compute the normalized disagreement $d_{j,c} = \mathrm{clip}_\tau((F_{j,c} - P_{j,c})/s_c)$. We also construct a position descriptor $\mathbf{p}_j = [r_j, \sin(2\pi r_j), \cos(2\pi r_j), b_{k+1}/H]^\top$, where $r_j = (j-1)/(R-1)$ for $R > 1$ and $r_j = 0$ otherwise. This descriptor captures both

the relative position within the remaining horizon and the progress of feedback. An MLP combines these features with the memory to estimate the contribution of each trajectory:

$$\begin{aligned} G_{j,c} &= \sigma(g_\omega(\mathbf{h}_c, \mathbf{p}_j, d_{j,c}))\,, \\ Z_{j,c} &= G_{j,c}F_{j,c} + (1 - G_{j,c})P_{j,c}, \end{aligned} \tag{6}$$

where σ is the sigmoid function. The gate varies across variables and future positions, allowing the balance between fresh and previous predictions to depend on their disagreement and the accumulated error history. Beyond selecting between the two trajectories, PRISM uses the memory to adjust the fused prediction's amplitude and offset. A decoder produces two modulation coefficients, $(a_{j,c}, b_{j,c}) = D_\nu(\mathbf{h}_c, \mathbf{p}_j)$, which define a bounded affine transformation:

$$\widehat{Y}_{j,c}^{(k+1)} = (1 + \rho \tanh(a_{j,c}))\, Z_{j,c} + \rho s_c \tanh(b_{j,c}), \tag{7}$$

where ρ controls the modulation magnitude. The multiplicative term adjusts the amplitude of the fused trajectory, while the additive term provides a correction scaled by the variability of the updated context. Conditioning both coefficients on $\mathbf{p}_j$ allows the same memory state to induce different adjustments across the remaining horizon. The revised trajectory supplies the next prediction block and serves as the previous forecast at the following stage.

4.4 Training Objective

PRISM is trained to produce an accurate initial forecast and to revise the remaining trajectory as feedback becomes available. We supervise the initial forecast with a horizon-weighted absolute error and each revised trajectory with the mean squared error over its unissued suffix:

$$\mathcal{L}_{\text{init}} = \frac{1}{HC} \sum_{j=1}^{H} \sum_{c=1}^{C} j^{-\alpha} \left| \widehat{Y}_{j,c}^{(0)} - Y_{j,c} \right| \qquad \mathcal{L}_{\text{prog}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \frac{\left\| \widehat{\mathbf{Y}}^{(k)} - \mathbf{Y}_{b_k+1:H} \right\|_F^2}{(H - b_k)C}. \tag{8}$$

where $\alpha \geq 0$ controls the temporal weighting of the initial forecast. The progressive loss averages errors within each remaining horizon before averaging across revision stages, assigning equal weight to stages with different suffix lengths. We set $\mathcal{L}_{\text{prog}} = 0$ when $K = 1$. The overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{init}} + \lambda_{\text{prog}} \mathcal{L}_{\text{prog}}, \tag{9}$$

where λ_{prog} balances initial forecasting and progressive revision. The backbone, feedback encoder, recurrent memory, and revision modules are optimized jointly through the unrolled feedback process. At each stage, unrevealed targets provide supervision for the remaining trajectory but do not enter its construction, preserving the predict-then-reveal ordering.

5 Experiments

5.1 Experiment Setup

Datasets. We evaluate our method on 12 real-world multivariate time-series benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, METR-LA, AQWan, CzeLan, ZafNoo, Electricity, AQShunyi, and Exchange. These datasets span diverse application domains, including electricity, weather, transportation, air quality, ecological monitoring, and economics. Detailed dataset statistics and descriptions are provided in Appendix A.1.

Baselines. We compare PRISM with 11 representative time series forecasting methods, including six online forecasting or adaptation approaches: **Proceed** (Zhao & Shen, 2025), **SOLID** (Chen et al., 2024), **OneNet** (Zhang et al., 2023), **FSNet** (Pham et al., 2023), **COSA** (Im & Kwon, 2026), and **TAFAS** (Kim et al., 2025); and five standard forecasting methods: **PhaseFormer** (Niu et al., 2026), **MMPD** (Zhang et al., 2026), **MixLinear** (Ma et al., 2026), **TimeMixer** (Wang et al., 2024a), and **DLinear** (Zeng et al., 2023).

Table 1: Forecasting performance comparison with the historical input length set to 96. Results are averaged over five independent runs. The best results are highlighted in **red** and the second-best results in blue. In the header, denotes online forecasting models, and denotes standard forecasting models.

Models		PRISM (Ours)		Proceed (2025)		SOLID (2024)		OneNet (2023)		FSNet (2023)		TAFAS (2025)		COSA (2026)		Phase* (2026)		MMPD (2026)		MixLinear (2026)		TimeMixer (2024)		DLinear (2023)	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.305**	**0.350**	0.472	0.451	0.469	0.450	0.494	0.464	0.889	0.651	0.469	0.462	0.438	0.451	0.383	0.386	0.376	0.393	0.389	0.417	0.378	0.400	0.386	0.400
	192	**0.397**	**0.405**	0.529	0.484	0.531	0.488	0.540	0.493	0.963	0.687	0.572	0.531	0.501	0.493	0.444	0.418	0.425	0.424	0.434	0.446	0.432	0.429	0.437	0.432
	336	**0.391**	**0.405**	0.575	0.511	0.584	0.516	0.589	0.523	0.982	0.697	0.671	0.584	0.613	0.558	0.474	0.436	0.467	0.448	0.489	0.464	0.481	0.451	0.481	0.459
	720	**0.391**	**0.411**	0.680	0.578	0.693	0.587	0.710	0.594	1.366	0.823	0.772	0.643	0.670	0.602	0.469	0.450	0.472	0.461	0.495	0.487	0.493	0.477	0.519	0.516
	Avg	**0.371**	**0.393**	0.564	0.506	0.569	0.510	0.583	0.519	1.050	0.715	0.621	0.555	0.556	0.526	0.442	0.422	0.435	0.431	0.452	0.454	0.445	0.439	0.456	0.452
ETTh2	96	**0.178**	**0.269**	0.235	0.318	0.235	0.317	0.238	0.319	0.450	0.428	0.242	0.326	0.258	0.341	0.293	0.337	0.322	0.364	0.337	0.393	0.284	0.331	0.333	0.387
	192	0.324	0.366	0.296	0.359	**0.293**	**0.355**	0.301	0.360	0.565	0.484	0.297	0.364	0.306	0.378	0.380	0.388	0.416	0.422	0.431	0.446	0.364	0.382	0.477	0.476
	336	**0.295**	**0.351**	0.347	0.393	0.344	0.390	0.347	0.393	0.495	0.481	0.363	0.404	0.347	0.408	0.426	0.424	0.436	0.442	0.475	0.460	0.406	0.416	0.594	0.541
	720	**0.300**	**0.364**	0.467	0.463	0.450	0.453	0.504	0.481	0.668	0.561	0.432	0.447	0.420	0.447	0.419	0.437	0.440	0.454	0.467	0.487	0.423	0.436	0.831	0.657
	Avg	**0.275**	**0.337**	0.336	0.383	0.330	0.379	0.347	0.388	0.544	0.489	0.333	0.385	0.333	0.394	0.379	0.397	0.415	0.435	0.427	0.447	0.369	0.391	0.559	0.515
ETTm1	96	**0.229**	**0.297**	0.405	0.406	0.388	0.395	0.394	0.400	0.701	0.551	0.367	0.393	0.353	0.390	0.326	0.344	0.313	0.350	0.374	0.419	0.315	0.343	0.345	0.372
	192	**0.319**	**0.347**	0.457	0.436	0.448	0.429	0.453	0.434	0.834	0.598	0.451	0.440	0.419	0.430	0.398	0.376	0.358	0.376	0.403	0.429	0.358	0.380	0.380	0.389
	336	**0.310**	**0.348**	0.509	0.464	0.508	0.461	0.514	0.468	0.999	0.664	0.537	0.488	0.488	0.472	0.425	0.401	0.400	0.404	0.470	0.443	0.385	0.400	0.413	0.413
	720	**0.307**	**0.354**	0.587	0.506	0.585	0.500	0.615	0.518	1.151	0.726	0.634	0.553	0.593	0.536	0.436	0.424	0.450	0.438	0.530	0.477	0.455	0.443	0.474	0.453
	Avg	**0.291**	**0.337**	0.490	0.453	0.482	0.446	0.494	0.455	0.921	0.635	0.497	0.469	0.463	0.457	0.382	0.388	0.380	0.392	0.444	0.442	0.378	0.392	0.403	0.407
ETTm2	96	**0.104**	**0.200**	0.155	0.253	0.161	0.259	0.156	0.252	0.253	0.320	0.168	0.271	0.174	0.271	0.179	0.258	0.175	0.260	0.207	0.326	0.178	0.260	0.193	0.292
	192	**0.182**	**0.265**	0.201	0.287	0.210	0.295	0.212	0.293	0.394	0.387	0.204	0.299	0.217	0.310	0.233	0.296	0.242	0.305	0.311	0.386	0.246	0.306	0.284	0.362
	336	**0.174**	**0.263**	0.250	0.321	0.261	0.331	0.273	0.333	0.535	0.454	0.261	0.342	0.282	0.351	0.310	0.340	0.308	0.349	0.417	0.462	0.296	0.339	0.369	0.427
	720	**0.189**	**0.279**	0.322	0.368	0.340	0.381	0.339	0.379	0.756	0.542	0.389	0.411	0.343	0.399	0.392	0.398	0.403	0.403	0.583	0.596	0.396	0.396	0.554	0.522
	Avg	**0.162**	**0.252**	0.232	0.307	0.243	0.316	0.245	0.314	0.485	0.426	0.255	0.331	0.254	0.333	0.279	0.323	0.282	0.329	0.380	0.443	0.279	0.325	0.350	0.401
Weather	96	**0.103**	**0.134**	0.174	0.216	0.173	0.214	0.167	0.209	0.337	0.307	0.200	0.255	0.165	0.209	0.172	0.204	0.183	0.222	0.213	0.286	0.163	0.210	0.196	0.255
	192	**0.178**	**0.220**	0.222	0.258	0.221	0.257	0.241	0.266	0.490	0.392	0.245	0.291	0.217	0.257	0.219	0.246	0.232	0.267	0.252	0.326	0.208	0.251	0.237	0.296
	336	**0.167**	**0.212**	0.276	0.296	0.275	0.295	0.275	0.293	0.618	0.447	0.327	0.338	0.274	0.298	0.276	0.288	0.294	0.313	0.320	0.361	0.268	0.295	0.283	0.335
	720	**0.162**	**0.211**	0.353	0.346	0.353	0.346	0.367	0.354	0.931	0.533	0.455	0.409	0.359	0.356	0.362	0.346	0.344	0.337	0.386	0.401	0.346	0.348	0.345	0.381
	Avg	**0.153**	**0.194**	0.256	0.279	0.255	0.278	0.262	0.280	0.594	0.420	0.307	0.323	0.254	0.280	0.240	0.260	0.263	0.284	0.293	0.344	0.309	0.360	0.265	0.317
METR-LA	96	**0.666**	**0.432**	9.764	0.955	1.341	0.706	2.104	0.702	1.698	0.749	2.241	0.723	1.482	0.757	1.240	0.695	1.484	0.765	1.384	0.742	1.208	0.651	1.131	0.705
	192	**1.199**	**0.665**	6.041	0.954	1.518	0.777	5.442	0.872	2.236	0.841	5.014	0.816	1.733	0.807	1.438	0.771	1.606	0.802	1.575	0.813	1.408	0.726	1.285	0.760
	336	**1.146**	**0.637**	4.707	0.950	1.824	0.804	1.563	0.742	2.261	0.898	1.684	0.799	1.721	0.798	1.443	0.771	1.501	0.814	1.600	0.810	1.448	0.748	1.318	0.775
	720	**1.104**	**0.648**	2.492	1.003	1.922	0.938	1.795	0.821	3.145	1.111	1.891	0.850	2.047	0.903	1.701	0.854	1.713	0.832	1.951	0.925	1.676	0.814	1.446	0.809
	Avg	**1.029**	**0.596**	5.751	0.966	1.652	0.807	2.726	0.784	2.335	0.900	2.708	0.797	1.746	0.816	1.456	0.773	1.313	0.699	1.628	0.822	1.435	0.735	1.295	0.762
AQWan	96	**0.568**	**0.382**	0.810	0.493	0.856	0.514	0.804	0.490	1.897	0.654	0.776	0.493	1.098	0.562	0.803	0.479	0.831	0.486	0.855	0.553	0.809	0.490	0.784	0.510
	192	**0.791**	**0.485**	0.897	0.522	0.957	0.543	0.883	0.515	1.469	0.648	0.844	0.518	0.946	0.547	0.878	0.506	0.875	0.516	0.899	0.602	0.877	0.515	0.843	0.535
	336	**0.758**	**0.475**	0.906	0.528	0.932	0.540	0.902	0.526	1.100	0.606	0.888	0.538	0.958	0.555	0.908	0.524	0.889	0.524	0.967	0.586	0.894	0.524	0.865	0.547
	720	**0.750**	**0.471**	0.975	0.552	0.989	0.561	0.960	0.543	1.308	0.628	1.059	0.586	1.070	0.581	0.951	0.546	0.957	0.539	1.065	0.592	0.951	0.547	0.932	0.563
	Avg	**0.717**	**0.453**	0.897	0.524	0.934	0.539	0.887	0.519	1.443	0.634	0.892	0.534	1.018	0.561	0.885	0.513	0.888	0.516	0.947	0.583	0.883	0.519	0.856	0.539
CzeLan	96	**0.126**	**0.194**	0.228	0.264	0.225	0.264	0.214	0.263	0.776	0.585	0.228	0.281	0.546	0.510	0.211	0.243	0.211	0.258	0.219	0.270	0.229	0.276	0.352	0.422
	192	**0.211**	**0.256**	0.259	0.290	0.258	0.290	0.244	0.282	1.385	0.811	0.264	0.305	0.442	0.440	0.237	0.258	0.252	0.289	0.257	0.304	0.262	0.297	0.471	0.475
	336	**0.198**	**0.259**	0.301	0.324	0.299	0.328	0.280	0.308	1.466	0.842	0.305	0.338	0.632	0.566	0.266	0.282	0.277	0.308	0.298	0.337	0.297	0.328	0.597	0.527
	720	**0.219**	**0.292**	0.359	0.366	0.348	0.368	0.335	0.349	0.985	0.686	0.381	0.382	0.707	0.606	0.328	0.323	0.346	0.358	0.376	0.391	0.358	0.371	0.803	0.604
	Avg	**0.188**	**0.252**	0.287	0.311	0.282	0.312	0.268	0.301	1.153	0.731	0.295	0.327	0.582	0.530	0.260	0.276	0.271	0.303	0.288	0.326	0.286	0.318	0.556	0.507
ZafNoo	96	**0.307**	**0.308**	0.477	0.419	0.478	0.421	0.462	0.406	0.941	0.678	0.481	0.429	0.568	0.481	0.465	0.385	0.476	0.419	0.492	0.482	0.484	0.425	0.454	0.433
	192	**0.462**	**0.410**	0.555	0.463	0.564	0.466	0.536	0.449	1.128	0.753	0.562	0.469	0.893	0.690	0.557	0.437	0.558	0.462	0.556	0.524	0.567	0.470	0.521	0.465
	336	**0.439**	**0.397**	0.613	0.495	0.625	0.501	0.601	0.483	1.266	0.792	0.768	0.524	0.741	0.554	0.617	0.471	0.626	0.501	0.628	0.517	0.626	0.504	0.561	0.483
	720	**0.445**	**0.407**	0.699	0.533	0.712	0.539	0.696	0.527	1.302	0.814	0.690	0.536	0.854	0.644	0.709	0.517	0.721	0.544	0.731	0.532	0.728	0.552	0.640	0.506
	Avg	**0.414**	**0.380**	0.586	0.477	0.595	0.482	0.573	0.466	1.159	0.759	0.625	0.490	0.764	0.592	0.587	0.453	0.595	0.481	0.602	0.514	0.601	0.488	0.544	0.472
Electricity	96	**0.136**	**0.222**	0.174	0.259	0.175	0.260	0.149	0.247	0.354	0.430	0.204	0.289	0.849	0.763	0.154	0.246	0.161	0.457	0.214	0.314	0.156	0.248	0.197	0.282
	192	**0.165**	**0.250**	0.182	0.268	0.182	0.267	0.175	0.261	0.367	0.440	0.214	0.297	0.376	0.437	0.167	0.263	0.180	0.277	0.210	0.313	0.170	0.259	0.196	0.285
	336	**0.149**	**0.238**	0.198	0.284	0.198	0.284	0.178	0.275	0.387	0.454	0.237	0.318	0.670	0.638	0.183	0.283	0.194	0.295	0.237	0.320	0.186	0.276	0.209	0.301
	720	**0.165**	**0.263**	0.242	0.320	0.240	0.318	0.204	0.299	0.434	0.482	0.300	0.354	0.414	0.460	0.221	0.309	0.233	0.324	0.274	0.350	0.228	0.312	0.245	0.333
	Avg	**0.154**	**0.243**	0.199	0.283	0.199	0.282	0.177	0.271	0.386	0.452	0.239	0.314	0.577	0.574	0.181	0.275	0.192	0.338	0.234	0.324	0.244	0.334	0.212	0.300
AQShunyi	96	**0.493**	**0.404**	0.727	0.508	0.765	0.523	0.718	0.502	1.379	0.674	0.702	0.508	0.783	0.537	0.724	0.505	0.835	0.565	0.766	0.533	0.712	0.493	0.696	0.529
	192	**0.697**	**0.497**	0.801	0.533	0.848	0.548	0.785	0.527	1.223	0.654	0.738	0.527	0.855	0.563	0.777	0.524	0.868	0.570	0.815	0.547	0.757	0.518	0.744	0.547
	336	**0.666**	**0.485**	0.815	0.544	0.855	0.558	0.834	0.549	1.075	0.633	0.815	0.553	0.824	0.555	0.792	0.534	0.867	0.572	0.818	0.552	0.779	0.531	0.756	0.559
	720	**0.651**	**0.482**	0.857	0.560	0.880	0.571	0.875	0.568	1.166	0.635	0.954	0.605	0.903	0.586	0.832	0.552	0.894	0.585	0.857	0.570	0.806	0.544	0.799	0.585
	Avg	**0.627**	**0.467**	0.800	0.536	0.837	0.550	0.803	0.536	1.211	0.649	0.802	0.548	0.841	0.560	0.781	0.529	0.866	0.573	0.814	0.550	0.764	0.521	0.749	0.555
Exchange	96	**0.024**	**0.104**	0.089	0.204	0.086	0.202	0.084	0.202	0.159	0.289	0.090	0.214	0.091	0.215	0.086	0.205	0.087	0.205	0.095	0.243	0.085	0.218	0.088	0.218
	192	**0.081**	**0.204**	0.186	0.304	0.186	0.306	0.189	0.306	0.278	0.385	0.176	0.305	0.200	0.324	0.196	0.312	0.199	0.301	0.189	0.347	0.175	0.315	0.176	0.315
	336	**0.081**	**0.209**	0.323	0.410	0.322	0.411	0.325	0.411	0.410	0.479	0.273	0.385	0.335	0.421	0.356	0.432	0.337	0.440	0.355	0.456	0.323	0.427	0.321	0.427
	720	**0.098**	**0.237**	0.812	0.680	0.814	0.681	0.945	0.734	1.302	0.833	0.864	0.704	0.921	0.725	0.935	0.735	0.865	0.714	0.939	0.730	0.851	0.695	0.839	0.695
	Avg	**0.071**	**0.189**	0.352	0.400	0.352	0.400	0.386	0.413	0.537	0.496	0.351	0.402	0.387	0.421	0.393	0.621	0.372	0.415	0.395	0.444	0.357	0.414	0.354	0.414

Implementation Details. All models are implemented in PyTorch and trained with the Adam optimizer on a single NVIDIA GeForce RTX 4070Ti GPU with 12 GB of memory. The learning rate is selected from $\{0.0001, 0.0005, 0.001\}$, and the hidden dimension is chosen from $\{32, 64, 128, 256, 512\}$ based on validation performance. We apply early stopping with a patience of five epochs to prevent overfitting. The `drop_last` option is disabled during both training and evaluation so that all available samples are retained. We evaluate forecasting accuracy using MSE and MAE, and report the average results over five independent runs.

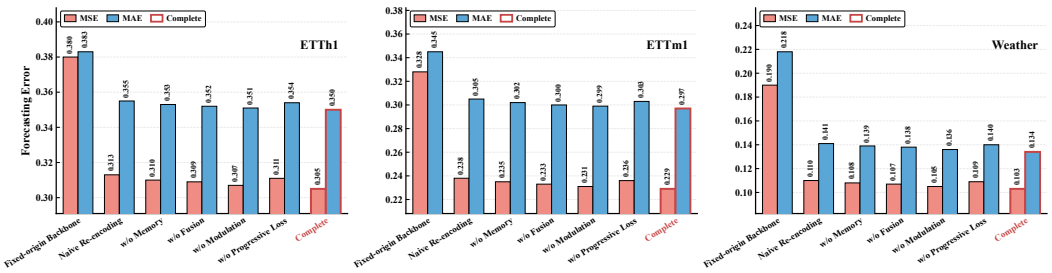


Figure 2: Ablation study.

5.2 Main Results

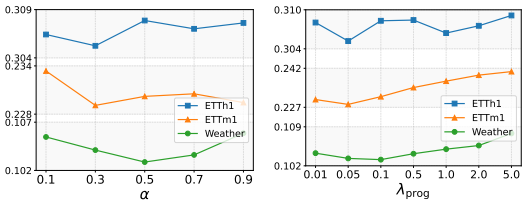
As shown in Table 1, standard forecasting methods such as Phaseformer, DLinear, TimeMixer, and MMPD remain competitive on several datasets, particularly when temporal patterns can be effectively captured from a fixed historical context. Online methods, including Proceed, SOLID, and OneNet, also perform well in some settings, indicating the benefit of adapting to newly arriving observations. However, their performance is less consistent across datasets and forecasting horizons, with FSNet exhibiting particularly noticeable degradation in several cases. A possible reason is that online methods typically rely on delayed supervision to update model parameters or adapters, and most require accumulating sufficient observations before adaptation.

In contrast, PRISM directly incorporates newly revealed observations to revise the still-unseen future, allowing prediction errors exposed at earlier stages to guide subsequent forecasting. This explicit trajectory correction leads to more consistent improvements across both short- and long-horizon settings. On METR-LA with $H = 96$, PRISM reduces MSE and MAE by **41.1%** and **33.6%**, respectively, over the strongest competing baseline. The advantage becomes even more pronounced on Exchange with $H = 720$, where the reductions reach **87.9%** in MSE and **65.1%** in MAE. Notably, PRISM shows weaker horizon-dependent degradation, with errors remaining stable across longer horizons. These results demonstrate that PRISM remains effective across substantially different forecasting horizons and is particularly beneficial when prediction errors would otherwise accumulate over long horizons.

5.3 Ablation Study

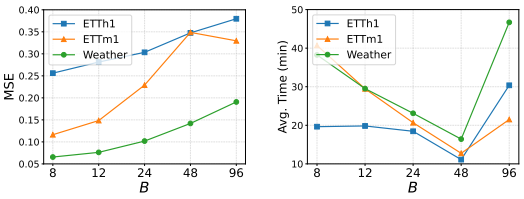
We construct six variants of **PRISM** for ablation. **'Fixed-origin Backbone'** performs one-shot forecasting without feedback. **'Naive Re-encoding'** updates the input window and re-forecasts without revision modules. **'w/o Memory'**, **'w/o Fusion'**, and **'w/o Modulation'** remove the innovation-state memory, forecast fusion, and affine modulation, respectively. **'w/o Progressive Loss'** removes stage-wise supervision. **'Complete'** denotes the full **PRISM**.

Figure 2 reports the ablation results on ETTh1, ETTm1, and Weather. The fixed-origin backbone performs substantially worse than the rolling variants, whereas Naive Re-encoding considerably reduces the forecasting errors. This comparison shows that incorporating newly revealed observations is the primary source of improvement. However, re-encoding alone does not fully exploit the available feedback. Removing the memory, fusion, or modulation component consistently degrades performance, indicating that accumulated innovations, forecast reconciliation, and horizon-aware correction provide distinct and complementary benefits. Removing the progressive loss also leads to clear performance drops, particularly on Weather, highlighting the importance of stage-wise supervision for learning effective intermediate revisions. Compared with Naive Re-encoding, the complete model further reduces MSE by 2.6%, 3.8%, and 6.4% on ETTh1, ETTm1, and Weather, respectively. These results reveal a clear performance hierarchy: updated observations provide the main improvement, while the revision modules allow the model to use the resulting feedback more effectively. Overall, the complete **PRISM** achieves the best performance across all three datasets.



(a) α. (b) λ_{prog}

Figure 3: Hyperparameter analysis.



(a) MSE. (b) Time (min).

Figure 4: Block length B analysis.

5.4 Hyperparameter Analysis

Analysis of α. Figure 3-(a) evaluates the temporal weighting exponent α, which controls how rapidly the weights assigned to prediction errors decay along the forecasting horizon. We vary α over $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The MSE fluctuates only within a narrow interval across the tested values, indicating limited sensitivity to α. Notably, the variation is more pronounced on the ETTm1 and Weather datasets than on ETTh1. The best results are obtained at $\alpha = 0.3$ on the ETTh1 and ETTm1 datasets and at $\alpha = 0.5$ on Weather. Overall, values between 0.3 and 0.5 provide a favorable balance between emphasizing reliable near-term predictions and maintaining sufficient supervision at distant horizons.

Analysis of λ_{prog}. Figure 3-(b) evaluates $\lambda_{\text{prog}} \in \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 5.0\}$, which balances progressive revision against the initial forecasting objective. The lowest MSE is obtained at $\lambda_{\text{prog}} = 0.05$ on ETTh1 and ETTm1 and at $\lambda_{\text{prog}} = 0.1$ on Weather. Beyond 0.1, the error increases noticeably on ETTm1 and Weather, while remaining relatively stable on ETTh1. An excessively large weight may overemphasize revision and weaken the initial forecast, which determines the first released block and guides subsequent forecast fusion. Overall, a modest λ_{prog} provides a better balance between initial forecasting and progressive revision.

5.5 Analysis of Block Length B

Figure 4 evaluates different block lengths, with $B \in \{8, 12, 24, 48, 96\}$ under a fixed prediction length of 96, where B controls the number of predictions issued before newly revealed observations are incorporated. Specifically, as shown in Figure 4-(a), the MSE generally increases with B because larger blocks delay feedback and leave more predictions uncorrected. Smaller blocks enable more frequent re-encoding and revision but require additional backbone evaluations. Accordingly, Figure 4-(b) shows that the training time generally decreases as B increases to 48. Notably, we find that training takes longer at $B = 96$ because the early-stopping criterion is met later, resulting in a larger number of training epochs.

5.6 Rolling Forecast Evolution Analysis

Figure 5 illustrates how the forecast evolves during rolling prediction on ETTm1. At each stage, newly revealed observations are incorporated into the rolling context to generate a fresh forecast for the remaining horizon. PRISM then reconciles this fresh prediction with the unissued portion of the previous forecast using the accumulated innovation state, while previously committed predictions remain unchanged. The visualization indicates that revision is stage-dependent: its benefit may be limited when only a small amount of feedback is available, but it can correct earlier deviations and better capture subsequent variations as more informative feedback is accumulated.

Figure 6 presents the aggregate revision gains at different rolling stages. The revised forecasts achieve positive average MSE reductions over the fresh forecasts across all stages, and the improvement becomes larger as the rolling process progresses. Although the error bars indicate variation across samples, the overall trend shows that accumulated observations and prediction errors make the innovation state increasingly informative. This result complements the trajectory-level example in Figure 5 and confirms that progressive revision provides consistent benefits at the dataset level.

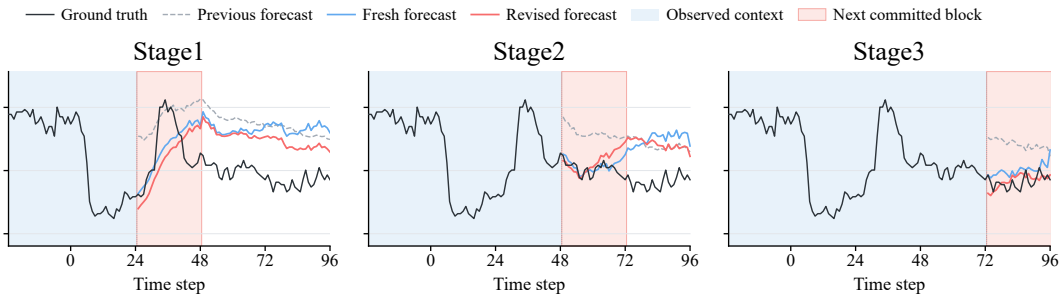


Figure 5: Rolling forecast evolution on ETTm1.

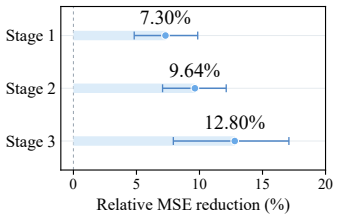


Figure 6: Stage-wise relative MSE reduction on ETTm1.

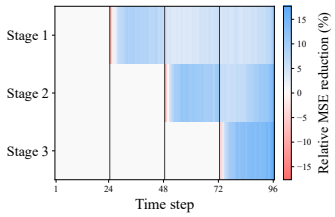


Figure 7: Progressive correction map on ETTm1.

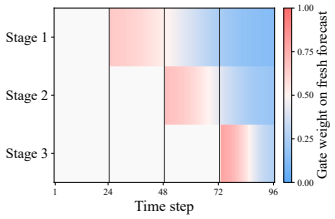


Figure 8: Fusion-gate behavior on ETTm1.

5.6.1 Progressive Correction and Fusion-Gate Analysis

To examine how the effectiveness of forecast revision varies across rolling stages and forecast positions, we compute the position-wise correction gains on ETTm1, as shown in Figure 7. Each row represents a revision stage, and each colored cell reports the relative MSE reduction of the revised forecast over the fresh forecast at a specific position in the remaining horizon. Gray regions correspond to observations that have already been revealed and are therefore no longer revised. Positive gains are mainly observed at medium- and long-range positions, whereas limited or slightly negative gains appear immediately after each reveal boundary, where the fresh forecast already captures the latest local dynamics. The correction gains also tend to become stronger at later stages, suggesting that accumulated observations and innovation information become increasingly useful for revising distant pending forecasts.

To examine how PRISM combines fresh and previous forecasts when revising the remaining trajectory, we visualize the fusion-gate behavior on ETTm1, as shown in Figure 8. Each cell represents the mean gate value over all test samples and variables, with larger values assigning more weight to the fresh forecast and smaller values retaining more information from the previous forecast. Near each reveal boundary, the gate favors the fresh forecast, consistent with the limited additional gains observed in Figure 7. At more distant positions, the model retains more information from the previous trajectory, where larger correction gains are generally observed. Together, the two figures suggest that PRISM learns a stage- and horizon-aware revision strategy that adaptively balances newly encoded information with the existing forecast trajectory.

6 Conclusion

In this paper, we introduced Predict-then-Reveal Forecasting and proposed PRISM, a progressive framework that revises the still-unobserved future as new observations become available while keeping issued predictions unchanged. PRISM combines progressive re-encoding, innovation-state memory, and memory-guided forecast revision to accumulate revealed errors and adaptively reconcile fresh predictions with previous forecasts. Experiments on 12 multivariate benchmarks demonstrate strong improvements over standard forecasting and online adaptation methods, particularly at longer horizons. Ablation and diagnostic analyses further confirm the complementary contributions of its components and reveal a stage- and horizon-aware revision strategy.

REFERENCES

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. doi: 10.1145/3637528.3671926. URL https://arxiv.org/abs/2310.14838.

Xinyang Chen, Huidong Jin, Yu Huang, and Zaiwen Feng. XLinear: A lightweight and accurate MLP-based model for long-term time series forecasting with exogenous inputs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20325–20335, 2026. doi: 10.1609/aaai.v40i24.39121. URL https://ojs.aaai.org/index.php/AAAI/article/view/39121.

Seungha Hong, Sukang Chae, Suyeon Kim, Sanghwan Jang, and Hwanjo Yu. Dynamic multi-period experts for online time series forecasting. In *Proceedings of the ACM Web Conference 2026*, 2026. doi: 10.1145/3774904.3792731. URL https://arxiv.org/abs/2603.09062.

Ao Hu, Liangjian Wen, Jiang Duan, Yong Dai, HE YAN, Dongkai Wang, Jun Wang, Yukun Zhang, Ruoxi Jiang, and Zenglin Xu. PMDformer: Patch-mean decoupling information transformer for long-term forecasting. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/654f8bf54c8ad618dcca6cd43dd0830f-Abstract-Conference.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/46362971bfc3a97e6a271f2eb90fba17-Abstract-Conference.html.

Xiannan Huang, Shuhan Qiu, Jiayuan Du, and Chao Yang. Online time series prediction using feature adjustment. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.03810.

Jeonghwan Im and Hyuk-Yoon Kwon. Cosa: Context-aware output-space adapter for test-time adaptation in time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

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. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c5433ab4056ca58db67be4578c384cba-Abstract-Conference.html.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 2025. doi: 10.1609/aaai.v39i17.33965.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6555–6565, 2024.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021. doi: 10.1098/rsta.2020.0209.

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 *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06625.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d09820e292f915d77fa717d228c425ee-Abstract-Conference.html.

Aitian Ma, Dongsheng Luo, and Mo Sha. Mixlinear: Extreme low-resource multivariate time series forecasting with 0.1k parameters. In *International Conference on Learning Representations*, 2026.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. WPMixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19581–19588, 2025. doi: 10.1609/aaai.v39i18.34156. URL https://ojs.aaai.org/index.php/AAAI/article/view/34156.

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, 2026.

Viresh Pati, Yubin Kim, Vinh Pham, Jevon Twitty, Shihao Yang, and Jiecheng Lu. Caps: Unifying attention, recurrence, and alignment in transformer-based time series forecasting. *arXiv preprint arXiv:2602.02729*, 2026.

Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven C. H. Hoi. Learning fast and slow for online time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2202.11672.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a. URL https://arxiv.org/abs/2405.14616.

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

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. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Yifan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. OneNet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3066. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dd6a47bc0aad6f34aa5e77706d90cdc4-Abstract.html.

Yunhao Zhang, Wenyao Hu, Jiale Zheng, Lujia Pan, and Junchi Yan. MMPD: Diverse time series forecasting via multi-mode patch diffusion loss. In *International Conference on Learning Representations*, 2026.

Lifan Zhao and Yanyan Shen. Proactive model adaptation against concept drift for online time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025. doi: 10.1145/3690624.3709210. URL https://arxiv.org/abs/2412.08435.

Algorithm 1 Training pipeline for **PRISM**

Input: Historical sequence $\mathbf{X} \in \mathbb{R}^{L \times C}$, future target $\mathbf{Y} \in \mathbb{R}^{H \times C}$, block length B, Epochs (E).
Output: Trained model parameters Θ^*.
Partition the forecasting horizon into K consecutive blocks according to B.
for $e = 1, \ldots, E$ **do**
    # 1. Initialization and Initial Forecasting.
    Set the rolling context $\mathbf{X}^{(0)} = \mathbf{X}$ and initialize $\mathbf{h}^{(0)} = \mathbf{0}$.
    Generate the initial forecast $\widehat{\mathbf{Y}}^{(0)} = f_\theta(\mathbf{X}^{(0)})$.
    for $k = 0, \ldots, K-1$ **do**
        # 2. Predict-then-Reveal Forecasting.
        Issue the next prediction block from $\widehat{\mathbf{Y}}^{(k)}$.
        if $k < K-1$ **then**
            Reveal the corresponding observation block after its prediction has been issued.
            Update the rolling context $\mathbf{X}^{(k+1)}$ with the revealed observations.
            Re-encode $\mathbf{X}^{(k+1)}$ to obtain the fresh forecast $\mathbf{F}^{(k+1)}$.
            # 3. Innovation-State Memory.
            Compute the innovation $\mathbf{e}^{(k)}$ from the issued prediction and revealed observations.
            Encode the normalized observations and innovations into $\mathbf{z}^{(k)}$.
            Update the recurrent memory $\mathbf{h}^{(k+1)}$ using $\mathbf{z}^{(k)}$.
            # 4. Memory-Guided Forecast Revision.
            Align $\mathbf{F}^{(k+1)}$ with the unissued previous forecast $\mathbf{P}^{(k+1)}$.
            Compute the memory-conditioned fusion gate $\mathbf{G}^{(k+1)}$.
            Fuse and modulate the aligned forecasts to obtain $\widehat{\mathbf{Y}}^{(k+1)}$.
        end if
    end for
    # 5. Progressive Optimization.
    Compute the total objective $\mathcal{L} = \mathcal{L}_{\text{init}} + \lambda_{\text{prog}} \mathcal{L}_{\text{prog}}$.
end for
return Θ^*.

Table 2: Statistics of the datasets used in our experiments. **Used Length** denotes the number of temporal observations used under the adopted forecasting protocol, and **Dim.** denotes the number of variables.

Dataset	ETTh1	ETTh2	ETTm1	ETTm2	Weather	METR-LA	AQWan	CzeLan	ZafNoo	Electricity	AQShunyi	Exchange
Domain	Electricity	Electricity	Electricity	Electricity	Weather	Traffic	Air Quality	Nature	Nature	Electricity	Air Quality	Economics
Frequency	1 hour	1 hour	15 min	15 min	10 min	5 min	1 hour	30 min	30 min	1 hour	1 hour	1 day
Used Length	14,400	14,400	57,600	57,600	52,696	34,272	35,064	19,934	19,225	26,304	35,064	7,588
Dim.	7	7	7	7	21	207	11	11	11	321	11	8
Split	12/4/4 mo.	12/4/4 mo.	12/4/4 mo.	12/4/4 mo.	7:1:2	7:1:2	6:2:2	7:1:2	7:1:2	7:1:2	6:2:2	7:1:2

A Experiment

A.1 Dataset Details

We provide detailed statistics of the datasets used in our experiments in Table 2. The 12 datasets cover six application domains and exhibit considerable diversity in sampling frequency, sequence length, and dimensionality. All datasets are split chronologically into training, validation, and test sets following the standard protocols commonly used for each benchmark.

ETT. The Electricity Transformer Temperature (ETT) benchmark consists of four datasets, ETTh1, ETTh2, ETTm1, and ETTm2, collected from two electricity transformers. Each dataset contains seven variables describing transformer operating conditions, including oil temperature and load-related measurements. ETTh1 and ETTh2 are recorded at an hourly resolution, whereas ETTm1 and ETTm2 provide finer-grained observations at 15-minute intervals. The ETT benchmarks therefore allow us to evaluate forecasting performance under different temporal resolutions while retaining comparable underlying variables.

Weather. Weather contains 21 meteorological variables recorded at 10-minute intervals throughout 2020. The variables characterize multiple aspects of atmospheric conditions and exhibit both regular periodic patterns and short-term fluctuations. It provides a representative environmental forecasting scenario with relatively high temporal resolution.

METR-LA. METR-LA is a large-scale traffic dataset collected from 207 loop detectors deployed across the Los Angeles County road network. Traffic speeds are aggregated at 5-minute intervals, resulting in a high-dimensional and fine-grained multivariate time series. Compared with the other benchmarks, METR-LA introduces stronger cross-variable dependencies and rapidly changing temporal dynamics, providing a challenging transportation forecasting setting.

Air-Quality Datasets. AQWan and AQShunyi are two air-quality datasets collected from monitoring stations over approximately four years. Both datasets contain 11 variables sampled hourly, including air-pollution measurements together with related environmental information. They provide non-stationary environmental time series affected by seasonal patterns and changing atmospheric conditions.

Ecological Datasets. CzeLan and ZafNoo are derived from the SAPFLUXNET project and contain sap-flow measurements together with associated environmental variables. Both datasets contain 11 variables recorded every 30 minutes. They characterize natural ecological processes whose temporal dynamics are jointly influenced by plant physiology and environmental conditions, providing forecasting scenarios that differ substantially from conventional electricity and traffic benchmarks.

Electricity. Electricity records the hourly electricity consumption of 321 clients from 2012 to 2014. With 321 variables, it is the highest-dimensional dataset in our benchmark suite. The dataset contains pronounced daily and weekly consumption patterns as well as variations across individual clients, allowing us to evaluate forecasting performance in high-dimensional multivariate settings.

Exchange. Exchange contains daily exchange rates from eight countries or regions, forming an eight-dimensional financial time series. Compared with the strongly periodic electricity and environmental datasets, Exchange exhibits relatively irregular temporal variation and weaker periodicity, providing a complementary economic forecasting scenario.

A.2 Metrics

To evaluate the forecasting performance of different methods, we employ Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics:

$$\mathrm{MSE} = \frac{1}{HC}\sum_{j=1}^{H}\sum_{c=1}^{C}\left(\hat{Y}_{j,c} - Y_{j,c}\right)^2, \qquad \mathrm{MAE} = \frac{1}{HC}\sum_{j=1}^{H}\sum_{c=1}^{C}\left|\hat{Y}_{j,c} - Y_{j,c}\right|. \tag{10}$$

Here, H and C denote the forecasting horizon and the number of variables, respectively. Lower MSE and MAE indicate better forecasting performance.

A.3 Baseline Models

We compare **PRISM** with 11 representative time-series forecasting methods, including six online forecasting or adaptation methods and five standard forecasting models.

- **Proceed** (Zhao & Shen, 2025): Proceed proactively adapts forecasting models to concept drift through drift-conditioned parameter adjustments.
- **SOLID** (Chen et al., 2024): SOLID selects contextually similar historical samples to update the prediction layer for online forecasting under distribution shifts.
- **OneNet** (Zhang et al., 2023): OneNet employs online ensembling to dynamically combine forecasting models and adapt to evolving concept drift.
- **FSNet** (Pham et al., 2023): FSNet balances fast adaptation to recent changes with the retrieval of similar previously observed patterns through an adapter and associative memory.

Table 3: Forecasting performance on Weather after integrating all baselines into the unified Predict-then-Reveal Forecasting (PTRF) paradigm with $L = 96$ and $B = H/4$.

H	Metric	Proceed	SOLID	OneNet	FSNet	TAFAS	COSA	Phase*	MMPD	MixLinear	TimeMixer	DLinear	PRISM (Ours)
96	MSE	0.1176	0.1281	0.1386	0.2794	0.1827	0.1161	0.1453	0.1330	0.1572	0.1212	0.1196	**0.1027**
	MAE	0.1640	0.1736	0.1869	0.3415	0.2294	0.1621	0.1880	0.1852	0.2019	0.1588	0.1716	**0.1339**
720	MSE	0.2414	0.2337	0.2448	0.3494	0.2708	0.2396	0.2484	0.2511	0.2479	0.2363	0.2325	**0.2078**
	MAE	0.3099	0.2986	0.3067	0.3728	0.3215	0.2941	0.2768	0.2954	0.2825	0.2677	0.2918	**0.2510**

- **COSA** (Im & Kwon, 2026): COSA introduces a context-aware output-space adapter that exploits recent observations to correct forecasts during test-time adaptation.
- **TAFAS** (Kim et al., 2025): TAFAS leverages partially revealed observations to update lightweight calibration modules and revise unobserved future predictions.
- **PhaseFormer** (Niu et al., 2026): PhaseFormer models periodicity from a phase perspective, using compact phase embeddings and a lightweight routing mechanism for efficient cross-phase interaction.
- **MMPD** (Zhang et al., 2026): MMPD introduces a multi-mode patch diffusion loss applicable to patch-based forecasting backbones, enabling the generation of multiple forecast modes with associated probabilities.
- **MixLinear** (Ma et al., 2026): MixLinear employs an extremely lightweight linear architecture for low-resource multivariate time-series forecasting.
- **TimeMixer** (Wang et al., 2024a): It decomposes time series into seasonal and trend components and performs multiscale mixing to capture patterns at different resolutions.
- **DLinear** (Zeng et al., 2023): DLinear decomposes each series into trend and seasonal components and forecasts them with separate linear layers.

A.4 Integrating Baselines into Predict-then-Reveal Forecasting

We further adapt all baselines to the Predict-then-Reveal Forecasting (PTRF) paradigm and evaluate them on the Weather dataset. At each stage, a method commits its prediction for the next block before the corresponding observations become available for subsequent forecasting. All methods use the same input length, block size, reveal schedule, forecasting origins, and evaluation interval. Standard models are re-invoked with the updated rolling context, while online methods retain their original adaptation mechanisms but operate only on the observations revealed so far. As shown in Table 3, PRISM consistently outperforms the adapted baselines under identical stage-wise information. This suggests that its advantage stems not simply from incorporating newly revealed observations, but from using them effectively through the innovation-state memory and memory-guided forecast revision.

A.5 Limitations and Future Directions

While PRISM demonstrates consistent improvements across diverse forecasting settings, several directions remain worth exploring. First, the current innovation-state memory is maintained independently for each variable. Future work could explicitly model interactions among variable-wise feedback states, potentially enabling more coordinated forecast revision in strongly coupled multivariate systems.

Second, PRISM adopts a fixed feedback block length throughout each forecasting episode. Extending the framework with adaptive feedback granularity could allow the revision frequency to vary according to temporal dynamics, prediction uncertainty, or the informativeness of newly revealed observations.

Finally, the current formulation maintains a fixed-length rolling context as forecasting progresses. Future work may explore more flexible context-retention mechanisms that selectively preserve informative historical patterns while incorporating newly revealed observations, enabling progressive forecasting over more heterogeneous temporal structures.

B AI USE STATEMENT

Generative AI tools were employed solely to assist with linguistic refinement, including improving the clarity and readability of the manuscript. The authors thoroughly reviewed all AI-assisted revisions to ensure their accuracy and appropriateness. The authors remain fully responsible for the final manuscript and its content.