DIAGMEM: EXPERIENCE-GROUNDED DIAGNOSTIC MEMORY FOR AGENTIC TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic time series forecasting extends standalone forecasting by incorporating tool use, feedback, and historical experience into the forecasting process. However, reusing this experience to diagnose a new forecast requires relating verified error feedback from completed forecasts to a current forecast whose outcome is still unknown. We introduce DiagMem, an experience-grounded diagnostic-memory framework for agentic time series forecasting. Its key idea is to interpret disagreement among heterogeneous forecasters using verified error patterns from similar diagnostic states. DiagMem organizes completed forecasting cases into diagnostic prototypes and uses a large language model to summarize their aggregated forecast, disagreement, and error evidence as structured scores and semantic descriptions. For a new window, the historical input, base forecast, and cross-model disagreement define its diagnostic state, which guides prototype retrieval. A lightweight numerical executor combines the retrieved diagnostic knowledge with current numerical inputs to estimate trend and seasonal residual corrections while keeping the base forecaster frozen. Across seven multivariate benchmarks and four forecast horizons, DiagMem delivers broad improvements over diverse frozen base forecasters, while ablations support the contributions of cross-model disagreement, state-conditioned diagnostic memory, and the joint use of structured scores and semantic descriptions.

1 INTRODUCTION

Time series forecasting supports decisions in energy, transportation, finance, and industrial systems (Guo et al., 2026; Lim & Zohren, 2021). Models such as DLinear, PatchTST, and iTransformer improve the mapping from historical observations to future values through advances in architecture and temporal representation (Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024), while GPT4TS, Time-LLM, and Chronos extend these capabilities through pretraining and model adaptation (Zhou et al., 2023; Jin et al., 2024; Ansari et al., 2024). As forecasting tasks incorporate information acquisition, forecast diagnosis, feedback, and experience reuse, forecasting becomes an ongoing process with interconnected stages (Cheng et al., 2026; Liao et al., 2026). Agentic forecasting coordinates these stages through large language model (LLM) reasoning, tool use, and memory (Chen et al., 2026; Jalori et al., 2025; Tao et al., 2026b). This process raises a central question: *how can an agent reuse verified feedback from completed forecasts to diagnose and correct a new forecast before its future observations arrive?*

The difficulty lies in the different information available for historical and current forecasts. Observed outcomes reveal a completed forecast's error and connect it to the state in which the forecast was made. For a current window, diagnosis and correction precede this feedback. Residual correction, semantic control, and historical residual retrieval have demonstrated the value of modeling base errors and reusing past feedback (Nguyen et al., 2026; Liang et al., 2026; Kim et al., 2026; He et al., 2026). Applying this experience to a new forecast requires diagnostic clues at inference time.

Heterogeneous forecasters provide such clues through their responses to the same input. Differences in architecture, representation, and inductive bias produce forecasts whose diversity reflects complementary modeling behavior (Kang et al., 2022; Wang et al., 2023). An agent can use these forecasters as auxiliary tools and examine their forecasts relative to a designated base forecast. The resulting cross-model disagreement supplies observable evidence about the current forecasting state.

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.

Interpreting this disagreement requires the forecasting context and historical error feedback. Individual model accuracy and error dependence affect the relationship between forecast diversity and accuracy (Timmermann, 2006; Lichtendahl & Winkler, 2020); similar disagreement magnitudes may accompany different base error directions and amplitudes. As outcomes arrive, completed forecasts retain the historical input, base forecast, cross-model disagreement, and observed base error. These records capture relationships among forecasting state, model disagreement, and base error, providing experience for diagnosing subsequent forecasts.

The next question is how to turn this experience into reusable diagnostic knowledge. Retrieval methods recover relevant historical patterns and residual trajectories (Han et al., 2025; He et al., 2026). In nonstationary series, similar histories can lead to different future dynamics, making the transfer of historical evidence sensitive to changes in forecasting state (Zhou et al., 2026). Error diagnosis therefore requires identifying reusable relationships between disagreement and base error across historical states. LLM analysis of error structure and memory of forecasting experience support the abstraction and organization of such feedback (Kim et al., 2026; Tao et al., 2026a). This allows verified error feedback from completed forecasts to become reusable diagnostic knowledge for interpreting new states and guiding correction.

We propose **DiagMem**, a diagnostic-memory framework for agentic time series forecasting that integrates cross-model diagnostic evidence, historical diagnostic memory, and numerical residual correction. A base forecaster and two auxiliary forecasters generate forecasts whose trend and seasonal differences, together with historical observations and the base forecast, form an observable diagnostic state. Completed windows link these states to observed base errors and are organized into state-conditioned diagnostic prototypes. An LLM summarizes the aggregated forecast, disagreement, and error evidence within each prototype as structured diagnostic scores and semantic descriptions. For a new window, DiagMem retrieves nearby diagnostic prototypes according to its observable state and conditions a lightweight numerical executor on the retrieved diagnostic knowledge and current numerical inputs. With the base forecaster frozen, the executor estimates trend and seasonal residuals to refine its forecast.

Our contributions are:

- We formulate forecast diagnosis through an observable state that combines historical observations, the base forecast, and cross-model disagreement at trend and seasonal scales, providing inference-time evidence for diagnosing base forecast errors.
- We construct experience-grounded diagnostic memory by organizing completed forecasts into state-conditioned prototypes and using an LLM to summarize the relationship between disagreement and error as retrievable structured scores and semantic descriptions.
- We develop a memory-conditioned numerical correction mechanism that combines retrieved diagnostic knowledge with current inputs to estimate trend and seasonal residuals while keeping the base forecaster fixed.

2 Related Work

2.1 Agentic Time Series Forecasting

Agentic forecasting extends numerical forecasting toward workflows that can incorporate reasoning, tool use, feedback, and memory (Cheng et al., 2026; Chen et al., 2026). One line of work organizes forecasting through specialized roles: TimeSeriesScientist separates data curation, planning, forecasting, and reporting (Zhao et al., 2025); FLAIRR-TS iteratively refines forecasting prompts using historical outputs and retrieved analogues (Jalori et al., 2025); and Nexus integrates macro dynamics, micro dynamics, and contextual information (Das et al., 2026).

Another line develops sequential workflows around tool use and feedback. Cast-R1 maintains decision-relevant state across analysis, forecasting, and reflection (Tao et al., 2026b). CastFlow coordinates specialized roles with historical memory and diagnostic tools (Pan et al., 2026), while CastFSR combines numerical priors, contextual reasoning, and reflection (Tao et al., 2026c). Liao et al. (2026) revise base forecasts using external evidence and historical cases. These systems expand access to tools, context, and experience. Diagnosing a newly issued forecast further requires connecting this available evidence to error feedback from completed forecasts.

2.2 Forecast Error Diagnosis and Correction

Forecast post-processing treats a generated forecast as an intermediate result that can be refined. δ-Adapter learns input adjustments and output residual corrections (Liang et al., 2026); UEC-STD separately corrects trend and seasonal components (Nguyen et al., 2026); and CTRL uses LLM analysis of error structure to guide a numerical residual decoder (Kim et al., 2026). These methods demonstrate that residual errors can be modeled after the base forecast is generated. Multiple forecasters provide further information: dynamic arbitrage estimates expert losses from the current state (Cerqueira et al., 2019), and forecast diversity informs combination weights (Kang et al., 2022). However, using cross-model disagreement to diagnose a designated base forecast requires relating this observable evidence to historical base errors under comparable forecasting states.

2.3 Retrieval and Memory for Forecasting

Retrieval-augmented forecasting brings historical information beyond the input window into forecasting. RAFT retrieves histories and their future trajectories (Han et al., 2025); RAF integrates retrieval with pretrained forecasters (Tire et al., 2026); and TimeRAG and TS-RAG incorporate patterns into LLM and foundation-model forecasting (Yang et al., 2025; Ning et al., 2025). Global Temporal Retrieval expands historical coverage (Cao et al., 2026), while SARAF considers relevance, diversity, and stationarity when selecting and aggregating evidence (Zhou et al., 2026).

Memory methods further retain experience produced by forecasting itself. MemCast organizes historical patterns, reasoning experience, and temporal rules (Tao et al., 2026a); RATL retrieves and combines historical residual trajectories to correct a frozen base forecaster (He et al., 2026). Reusable information thus extends from observations and future trajectories to forecasting experience and model errors. For the setting considered here, the remaining question is how to organize such feedback around forecasting state, cross-model disagreement, and verified base errors so that historical experience can support the diagnosis of new forecasts.

3 Methodology

3.1 Problem Setup and Overview

Given a historical window $X_i \in \mathbb{R}^{L\times D}$, we forecast $Y_i \in \mathbb{R}^{H\times D}$, where L, H, and D denote the lookback length, forecast horizon, and number of variables. DiagMem uses a base forecaster f_A and two heterogeneous auxiliary forecasters f_B, f_C, independently trained and then frozen:

$$A_i = f_A(X_i), \qquad B_i = f_B(X_i), \qquad C_i = f_C(X_i), \tag{1}$$

where $A_i, B_i, C_i \in \mathbb{R}^{H\times D}$. The base forecast A_i is the object of diagnosis and correction; B_i, C_i provide complementary responses. These responses and X_i define an observable diagnostic state. Completed forecasts associate this state with observed errors, forming historical cases that are grouped into prototypes. An LLM summarizes their evidence as diagnostic knowledge, which is retrieved for each new window to condition numerical residual correction. Figure 1 presents the overall framework.

3.2 Observable Diagnostic State

Trend and seasonal disagreement. Heterogeneous forecasts can differ in long-term evolution and local fluctuations. We separate these responses through the decomposition

$$\mathcal{T}(V) = \mathrm{MA}_k(V), \qquad \mathcal{S}(V) = V - \mathcal{T}(V), \tag{2}$$

where $V \in \mathbb{R}^{H\times D}$ and MA_k is a moving average of width k. The trend $\mathcal{T}(V)$ captures smooth variation, and the seasonal component $\mathcal{S}(V)$ captures deviations from it. Relative to A_i, the auxiliary disagreements are

$$\begin{aligned} d^t_{B,i} &= \mathcal{T}(B_i) - \mathcal{T}(A_i), \quad d^t_{C,i} = \mathcal{T}(C_i) - \mathcal{T}(A_i), \\ d^s_{B,i} &= \mathcal{S}(B_i) - \mathcal{S}(A_i), \quad d^s_{C,i} = \mathcal{S}(C_i) - \mathcal{S}(A_i). \end{aligned} \tag{3}$$

To relate disagreement to error, let $\Phi_t = \mathcal{T}$ and $\Phi_s = \mathcal{S}$. The component residual of forecaster M is

$$E^c_{M,i} = \Phi_c(Y_i) - \Phi_c(M_i), \qquad M \in \{A, B, C\}, \quad c \in \{t, s\}. \tag{4}$$

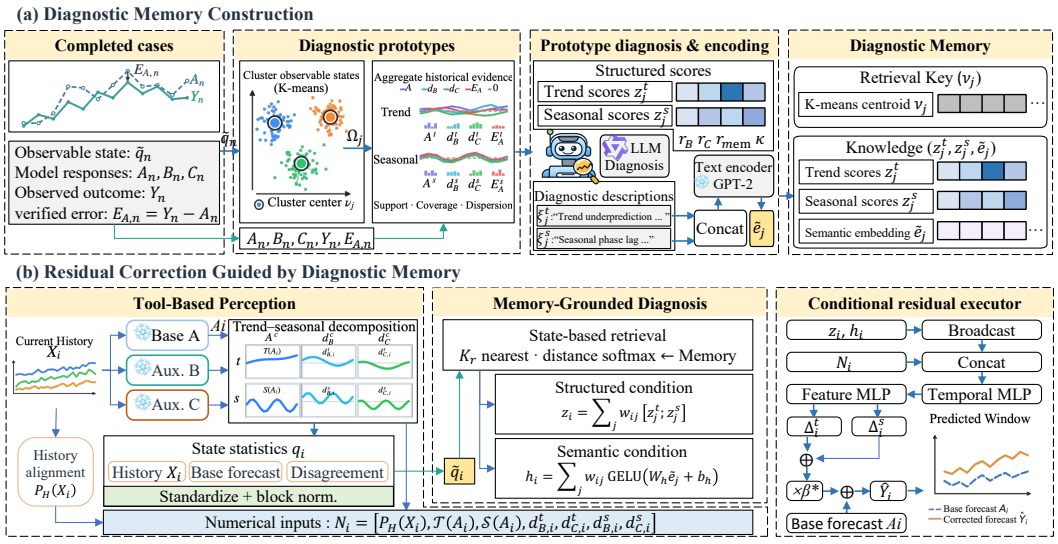

Figure 1: Overview of DiagMem. (a) Completed cases are clustered by normalized observable states, and aggregated prototype evidence is summarized by an LLM into structured scores and semantic descriptions, which are encoded and stored with prototype centers as diagnostic memory. (b) For a new window, the observable state retrieves nearby prototypes, and the resulting diagnostic conditions guide a numerical executor to estimate trend and seasonal residual corrections for the frozen base forecast.

Since $E^c_{B,i} = E^c_{A,i} - d^c_{B,i}$, the difference in squared error satisfies

$$\Delta \ell^c_{B,i} = \|E^c_{B,i}\|_F^2 - \|E^c_{A,i}\|_F^2 = \|d^c_{B,i}\|_F^2 - 2\langle E^c_{A,i}, d^c_{B,i}\rangle_F, \tag{5}$$

where $\|\cdot\|_F$ and $\langle\cdot,\cdot\rangle_F$ denote the Frobenius norm and inner product. The identity also holds for C. Disagreement's diagnostic value thus depends on its magnitude and alignment with the base residual. DiagMem uses historical feedback to learn these state-dependent relationships and organize them as diagnostic experience.

State representation. The diagnostic state combines historical dynamics, base forecast structure, and cross-model disagreement. For each variable, we summarize X_i by its mean, standard deviation, linear slope, lag-one autocorrelation, and endpoint change, giving $q_i^X \in \mathbb{R}^{5D}$. We summarize A_i by trend mean and slope, and seasonal standard deviation, mean absolute value, and lag-one autocorrelation, giving $q_i^A \in \mathbb{R}^{5D}$. The mean, mean absolute value, standard deviation, and slope of each disagreement yield $q_i^{\text{dis}} \in \mathbb{R}^{16D}$. We concatenate these blocks:

$$q_i = [q_i^X; q_i^A; q_i^{\text{dis}}] \in \mathbb{R}^{26D}. \tag{6}$$

Using the historical memory set $\mathcal{D}_{\text{mem}}$, we standardize each coordinate and divide each block by the square root of its dimension:

$$\widetilde{q}_i = \left[\frac{u_i^X}{\sqrt{5D}}; \frac{u_i^A}{\sqrt{5D}}; \frac{u_i^{\text{dis}}}{\sqrt{16D}}\right], \tag{7}$$

where $u_i^X, u_i^A, u_i^{\text{dis}}$ are the standardized blocks. Coordinate standardization aligns statistical scales, and dimension-aware block scaling balances their contributions to distance. The resulting observable state $\widetilde{q}_i$ supports both prototype construction and retrieval.

3.3 Historical Diagnostic Memory

Historical cases and prototypes. Completed forecasts connect observable states with verified error feedback. For $i \in \mathcal{D}_{\text{mem}}$, the historical diagnostic instance is

$$\mathcal{I}_i = \left(\widetilde{q}_i, \{A_i^c, d^c_{B,i}, d^c_{C,i}, E^c_{A,i}\}_{c\in\{t,s\}}\right), \qquad A_i^c = \Phi_c(A_i). \tag{8}$$

K-means clusters the observable states $\{\widetilde{q}_i\}$ into prototypes with centers ν_j and member sets Ω_j. This provides a compact prototype bank in the same Euclidean state space used for retrieval, allowing cases with similar observable states to share aggregated diagnostic evidence while avoiding instance-level memory and diagnosis. To preserve variation across the horizon, we partition it into B_p contiguous bins $\{I_b\}_{b=1}^{B_p}$ and aggregate model responses and observed errors. For $V \in \{A^c, d_B^c, d_C^c, E_A^c\}$, the signed and magnitude profiles are

$$\begin{aligned} P_{j,b}^{\mathrm{sgn}}(V) &= \underset{i\in\Omega_j,\, m=1,\dots,D}{\mathrm{median}} \left(\frac{1}{|I_b|} \sum_{u\in I_b} V_{i,u,m} \right), \\ P_{j,b}^{\mathrm{abs}}(V) &= \underset{i\in\Omega_j,\, m=1,\dots,D}{\mathrm{median}} \left(\frac{1}{|I_b|} \sum_{u\in I_b} |V_{i,u,m}| \right), \end{aligned} \tag{9}$$

where u and m index forecast positions and variables. These profiles summarize direction and magnitude within each horizon bin. Together with prototype support, temporal coverage, state dispersion, and the componentwise signed bias and mean absolute error of all three forecasters, they form the evidence $\mathcal{P}_j^c$.

Prototype diagnostic knowledge. During memory construction, the LLM diagnostic module analyzes base forecast structure, cross-model disagreement, historical errors, and prototype statistics to produce structured scores and a semantic description for each component:

$$(z_j^c, \xi_j^c) = \mathcal{G}_c(\mathcal{P}_j^c), \qquad z_j^c = [r_{B,j}^c, r_{C,j}^c, r_{\mathrm{mem},j}^c, \kappa_j^c], \tag{10}$$

where $r_{B,j}^c, r_{C,j}^c$ score the evidence supporting each auxiliary disagreement as a base residual direction, $r_{\mathrm{mem},j}^c$ scores the reusability of the historical base error pattern, and κ_j^c scores evidence consistency. Each uses the scale $\{0, 0.25, 0.5, 0.75, 1\}$, with larger values indicating stronger support. We concatenate ξ_j^t and ξ_j^s, encode them with frozen GPT-2 and masked mean pooling, then center each feature across prototypes and apply per-prototype ℓ_2 normalization. The resulting embedding is $\widetilde{e}_j \in \mathbb{R}^{d_e}$. Each memory item stores $(\nu_j, z_j^t, z_j^s, \widetilde{e}_j)$.

Retrieval for a new window. For the current state, we compute distances and retrieval weights:

$$d_{ij} = \|\widetilde{q}_i - \nu_j\|_2^2, \qquad w_{ij} = \frac{\exp(-d_{ij}/\tau)}{\sum_{k\in\mathcal{N}_{K_r}(i)} \exp(-d_{ik}/\tau)}, \quad j \in \mathcal{N}_{K_r}(i), \tag{11}$$

where $\mathcal{N}_{K_r}(i)$ contains the K_r nearest prototypes and $\tau > 0$ is the temperature. The retrieved diagnostic conditions are

$$z_i = \sum_{j\in\mathcal{N}_{K_r}(i)} w_{ij}[z_j^t; z_j^s], \qquad h_i = \sum_{j\in\mathcal{N}_{K_r}(i)} w_{ij}\, \mathrm{GELU}(W_h \widetilde{e}_j + b_h), \tag{12}$$

where $z_i \in \mathbb{R}^8$ combines the component scores, $h_i \in \mathbb{R}^{d_h}$ summarizes semantic knowledge, and $W_h \in \mathbb{R}^{d_h \times d_e}, b_h \in \mathbb{R}^{d_h}$ are learned projection parameters. Each window receives diagnostic conditions through its own prototype neighborhood and retrieval weights.

3.4 Residual Correction Guided by Diagnostic Memory

Conditional residual executor. Let $P_H : \mathbb{R}^{L\times D} \to \mathbb{R}^{H\times D}$ be a learned linear map along time, shared across variables, with the identity used when $L = H$. The executor combines aligned history, base components, auxiliary disagreements, and retrieved conditions:

$$\begin{aligned} F_i = \mathrm{Concat}\big(&P_H(X_i), \mathcal{T}(A_i), \mathcal{S}(A_i), d_{B,i}^t, d_{C,i}^t, d_{B,i}^s, d_{C,i}^s, \\ &\mathbf{1}_H z_i^\top, \mathbf{1}_H h_i^\top\big), \qquad (\Delta_i^t, \Delta_i^s) = \mathcal{R}_\theta(F_i), \end{aligned} \tag{13}$$

where concatenation is along features, $F_i \in \mathbb{R}^{H\times(7D+8+d_h)}$, and $\Delta_i^t, \Delta_i^s \in \mathbb{R}^{H\times D}$. The vector $\mathbf{1}_H$ broadcasts the diagnostic conditions across the horizon. The residual executor $\mathcal{R}_\theta$ uses a two-stage MLP. The first stage models temporal dynamics independently for each input feature using shared parameters. The second stage aggregates features at each forecast position and jointly outputs the trend residual Δ_i^t and seasonal residual Δ_i^s.

Explicit residual learning. For correction training, define $E_i = Y_i - A_i$ and $E_i^c = E_{A,i}^c$. The additive decomposition gives $E_i = E_i^t + E_i^s$. We jointly supervise the residual and its components:

$$\mathcal{L}_i = \ell_{\text{res}}(\Delta_i^t + \Delta_i^s, E_i) + \lambda_{ts} \sum_{c \in \{t,s\}} \lambda_c \ell_{ts}(\Delta_i^c, E_i^c), \tag{14}$$

where ℓ_{res} is mean squared error and ℓ_{ts} is SmoothL1 loss. The nonnegative weights $\lambda_{ts}, \lambda_t, \lambda_s$ control overall component supervision and its trend and seasonal contributions. The first term fits the total residual; the second supervises its component estimates.

Training and inference. The three forecasters are frozen after training and model selection. A historical calibration pool held out from their training and selection is divided chronologically into $\mathcal{D}_{\text{mem}}, \mathcal{D}_{\text{corr}}, \mathcal{D}_{\text{val}}$, with boundary gaps allowing earlier targets to be observed before the next stage. On $\mathcal{D}_{\text{mem}}$, we estimate standardization parameters, construct prototypes, and generate diagnostic knowledge. With this memory fixed, $\mathcal{D}_{\text{corr}}$ trains the executor, history alignment, and semantic projection; $\mathcal{D}_{\text{val}}$ selects the correction model and strength. LLM diagnosis and text encoding occur only during memory construction; individual test windows make no LLM calls. For a new window, DiagMem obtains the tool forecasts, constructs the diagnostic state, retrieves (z_i, h_i), and estimates the residual components. The final forecast is

$$\widehat{Y}_i = A_i + \beta^*(\Delta_i^t + \Delta_i^s), \tag{15}$$

where $\beta^* \in [0, 1]$ is selected on $\mathcal{D}_{\text{val}}$ and fixed during inference. This workflow connects forecasting tool responses, the observable diagnostic state, historical memory retrieval, and continuous residual correction.

4 Experiments

We evaluate DiagMem on seven multivariate forecasting benchmarks and four forecast horizons. The experiments assess complete-system forecasting accuracy, comparisons with uncorrected base forecasters, the contributions of auxiliary information and diagnostic memory, and representative correction behavior.

4.1 Experimental Setup

Datasets and evaluation. We use ETTh1, ETTh2, ETTm1, and ETTm2 (Zhou et al., 2021), GEFCom2012Wind (Hong et al., 2014), Electricity (ECL), and Solar (Lai et al., 2018). The four ETT datasets and the wind dataset each contain seven variables; ECL and Solar contain 321 and 137 variables, respectively. All tasks use a lookback length of 96 and forecast all variables over horizons $H \in \{96, 192, 336, 720\}$. We report mean squared error (MSE) and mean absolute error (MAE) in standardized units, averaged over test windows, forecast positions, and variables. Dataset-level summaries average the four horizons equally.

Forecasting tools and baselines. The forecasting tool pool comprises PatchTST (Nie et al., 2023), DLinear (Zeng et al., 2023), and iTransformer (Liu et al., 2024). Each serves in turn as the base forecaster, with the other two providing auxiliary forecasts; the resulting DiagMem configurations are denoted F-PT, F-DL, and F-iT. We compare with these three forecasters, FEDformer (Zhou et al., 2022), TimesNet (Wu et al., 2022), GPT4TS (Zhou et al., 2023), Time-LLM (Jin et al., 2024), TimesFM (Das et al., 2024), and Chronos-2 (Ansari et al., 2025). GPT4TS and Time-LLM are trained on each task, while TimesFM and Chronos-2 are evaluated in the zero-shot setting.

Data protocol. The ETT datasets use chronological train/calibration/test splits of 12/4/4 months, and the remaining datasets use a 70%/10%/20% split. Standardization statistics are fitted on the training split. For DiagMem, the training windows are partitioned chronologically for forecaster fitting and model selection, after which all three forecasting tools remain frozen. The calibration split is further divided into $\mathcal{D}_{\text{mem}}$, $\mathcal{D}_{\text{corr}}$, and $\mathcal{D}_{\text{val}}$ for memory construction, executor training, and correction-model selection, respectively. Horizon-sized purge intervals separate adjacent stages. Test forecasts are generated from the historical input, forecasting tool responses, and the fixed diagnostic memory. The standalone supervised baselines in Table 1 fit on the full training split and select

Table 1: Performance averaged over horizons 96, 192, 336, and 720. F-PT, F-DL, and F-iT use PatchTST, DLinear, and iTransformer as the base forecaster. Wind denotes GEFCom2012Wind. Best values are bold and second-best values are underlined, using unrounded metrics.

Methods	F-PT		F-DL		F-iT		DLinear		PatchTST		iTransformer		FEDformer		TimesNet		GPT4TS		Time-LLM		TimesFM		Chronos-2	
Metric	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	0.490	0.476	0.462	0.459	0.488	0.476	0.465	0.462	0.461	0.454	0.462	0.454	0.466	0.478	0.475	0.464	**0.458**	0.439	0.481	0.464	0.484	**0.435**	0.492	0.439
ETTh2	**0.376**	0.406	0.408	0.437	0.382	0.410	0.457	0.460	0.392	0.414	0.390	0.411	0.438	0.454	0.421	0.429	0.389	0.415	0.394	0.415	0.413	0.406	0.406	**0.405**
ETTm1	**0.387**	0.408	0.393	0.411	0.398	0.414	0.409	0.414	0.404	0.408	0.414	0.414	0.439	0.459	0.416	0.418	0.395	**0.397**	0.402	0.406	0.875	0.584	0.616	0.477
ETTm2	**0.267**	**0.331**	0.272	0.336	0.273	0.333	0.321	0.373	0.289	0.334	0.291	0.334	0.305	0.351	0.302	0.334	0.288	0.334	0.296	0.338	0.372	0.378	0.318	0.338
ECL	0.191	0.282	0.202	0.294	**0.179**	**0.271**	0.210	0.296	0.191	0.281	0.181	0.272	0.212	0.326	0.199	0.295	0.214	0.296	0.250	0.339	0.239	0.308	0.234	0.294
Solar	0.236	0.277	0.242	0.301	**0.235**	**0.269**	0.327	0.398	0.249	0.282	0.246	0.274	0.297	0.380	0.283	0.302	0.270	0.306	0.320	0.324	0.913	0.520	0.528	0.327
Wind	1.227	0.934	1.168	0.907	1.267	0.939	**1.168**	**0.895**	1.194	0.927	1.276	0.930	1.300	0.945	1.319	0.943	1.210	0.909	1.220	0.922	1.609	0.986	1.485	0.967

checkpoints on the calibration split. In correction comparisons, Backbone denotes an uncorrected base forecaster trained under DiagMem's internal fitting and model-selection protocol.

Implementation details. We construct 16 diagnostic prototypes and retrieve the three nearest prototypes with temperature $\tau = 1$. Trend–seasonal decomposition uses a moving-average kernel of 25, and prototype evidence uses eight horizon bins. Qwen (`Qwen3.8-27B`) generates prototype diagnoses with decoding temperature zero; frozen GPT-2 encodes their semantic descriptions, which are projected to 32 dimensions. The executor uses hidden dimension 32, dropout 0.1, batch size 64, and learning rate 10^{-3}. Training runs for at most 100 epochs with early-stopping patience 10. We set $\lambda_{ts} = 0.1$ and $\lambda_t = \lambda_s = 0.5$. Checkpoints and correction strength β^* are selected by forecast MSE on $\mathcal{D}_{\text{val}}$. Appendix A provides exact data partitions, training configurations, diagnostic prompts, and results for individual horizons.

4.2 Overall Forecasting Performance

We first compare the complete DiagMem configurations with standalone forecasting methods, then examine their performance relative to uncorrected Backbone references trained under the DiagMem protocol.

Overall results. Table 1 reports performance averaged over the four horizons. DiagMem achieves the lowest MSE on five of the seven datasets: F-PT leads on ETTh2, ETTm1, and ETTm2, while F-iT leads on ECL and Solar. On ETTm2, F-PT reaches an MSE of 0.267, a 7.1% reduction relative to the strongest standalone result, GPT4TS. On Solar, F-iT reaches an MSE of 0.235, a 4.7% reduction relative to standalone iTransformer. Across individual horizons, F-PT, F-DL, and F-iT attain the lowest MSE in 8, 7, and 6 settings, respectively, covering 21 of the 28 dataset–horizon settings.

The ordering also varies with the evaluation metric. DiagMem achieves the lowest average MAE on ETTm2, ECL, and Solar, and the lowest per-horizon MAE in 10 settings across its three configurations. GPT4TS attains the lowest average MSE on ETTh1 and MAE on ETTm1; TimesFM and Chronos-2 attain the lowest average MAE on ETTh1 and ETTh2, respectively. DLinear attains both minima on the wind dataset. These results identify the dataset and metric regimes in which each configuration is most effective.

Comparison with uncorrected base forecasters. This comparison uses the uncorrected Backbone references trained under the DiagMem protocol. F-PT, F-DL, and F-iT achieve lower MSE in 27, 25, and 26 of the 28 dataset–horizon settings, with mean relative reductions of 5.80%, 13.50%, and 3.60%, respectively. The corresponding MAE improvements occur in 24 settings for each configuration, with mean reductions of 2.01%, 9.18%, and 0.97%. Relative reductions are computed within each dataset–horizon setting before averaging. The comparisons favor DiagMem across the three base architectures, with more frequent improvements under MSE, the objective used to train and select the correction module.

Table 2: Four-horizon average ablation results with PatchTST as the base forecaster. RO: ResidualOnly; GM: GlobalMemory; NC: NumericCondition; PID: PrototypeID. Best values are bold and second-best values are underlined, using unrounded metrics.

Variant	Backbone		RO		NoAux		GM		NC		PID		Full	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.498	**0.476**	0.497	0.477	0.592	0.543	0.636	0.568	0.600	0.545	0.525	0.498	**0.490**	0.476
ETTh2	0.397	0.413	0.388	0.411	0.414	0.434	0.429	0.447	0.417	0.434	0.407	0.427	**0.376**	**0.406**
ETTm1	0.403	0.410	0.394	0.410	0.440	0.455	0.435	0.451	0.436	0.453	0.401	0.427	**0.387**	**0.408**
ETTm2	0.301	0.340	0.281	0.339	0.345	0.384	0.352	0.390	0.313	0.362	0.289	0.346	**0.267**	**0.331**
ECL	0.195	0.287	0.193	0.285	0.205	0.296	0.199	0.292	0.201	0.293	0.194	0.286	**0.191**	**0.282**
Solar	0.270	0.308	0.249	0.292	0.261	0.302	0.260	0.298	0.258	0.297	0.249	0.290	**0.236**	**0.277**
Wind	1.279	**0.909**	1.231	0.934	1.254	0.953	1.256	0.954	1.263	0.954	1.246	0.946	**1.227**	0.934

4.3 Ablation Study

Configurations. We compare Full with the uncorrected Backbone and five correction variants. ResidualOnly uses historical observations and the base forecast for numerical residual learning. NoAux removes auxiliary forecasts from diagnostic construction and numerical correction, while retaining the executor's feature width through zero-filled disagreement slots. GlobalMemory summarizes historical cases into one global prototype. NumericCondition learns structured conditions from numerical prototype evidence and represents prototypes with learned embeddings. PrototypeID retains the LLM-generated structured scores and replaces the semantic text representation with learned prototype embeddings. These comparisons examine numerical correction, auxiliary information, memory granularity, and diagnostic representations.

Ablation results. Table 2 presents the PatchTST results. Full attains lower MSE and MAE than each correction variant in all 28 dataset–horizon settings. Relative to ResidualOnly, its mean reductions are 2.47% in MSE and 1.53% in MAE. On ETTm2, the four-horizon average MSE changes from 0.301 for Backbone to 0.281 for ResidualOnly and 0.267 for Full, showing the progression from the base forecast through numerical correction to the complete diagnostic workflow.

Auxiliary information and state-conditioned memory produce larger separations in the ablation comparisons. Full achieves mean MSE reductions of 11.42% relative to NoAux and 12.08% relative to GlobalMemory; MAE reductions are 8.34% and 8.90%. On ETTh1, Full reaches an average MSE of 0.490, compared with 0.592 for NoAux and 0.636 for GlobalMemory. These comparisons favor the joint use of heterogeneous forecasting responses and state-conditioned historical context.

The diagnostic representation distinguishes the configurations. Full achieves mean MSE reductions of 9.72% relative to NumericCondition and 4.62% relative to PrototypeID, together with MAE reductions of 7.10% and 3.52%. NumericCondition changes both the structured and semantic branches, whereas PrototypeID preserves the structured diagnosis. These results support the effectiveness of the full representation, which combines structured scores with diagnostic descriptions.

Cross-backbone analysis. Tables 9 and 10 extend the comparisons to DLinear and iTransformer. Across the three base forecasters, Full attains lower MSE than every correction variant in all 84 evaluated settings. It also attains lower MAE than NoAux, GlobalMemory, and NumericCondition in all 84 settings, and than ResidualOnly and PrototypeID in 84 and 83 settings, respectively. The mean MSE reductions relative to ResidualOnly range from 1.26% for iTransformer to 7.10% for DLinear. The consistent MSE ordering across distinct forecasters connects these ablation results to the overall performance of the diagnostic workflow. Detailed DLinear and iTransformer tables appear in Appendix A.6.

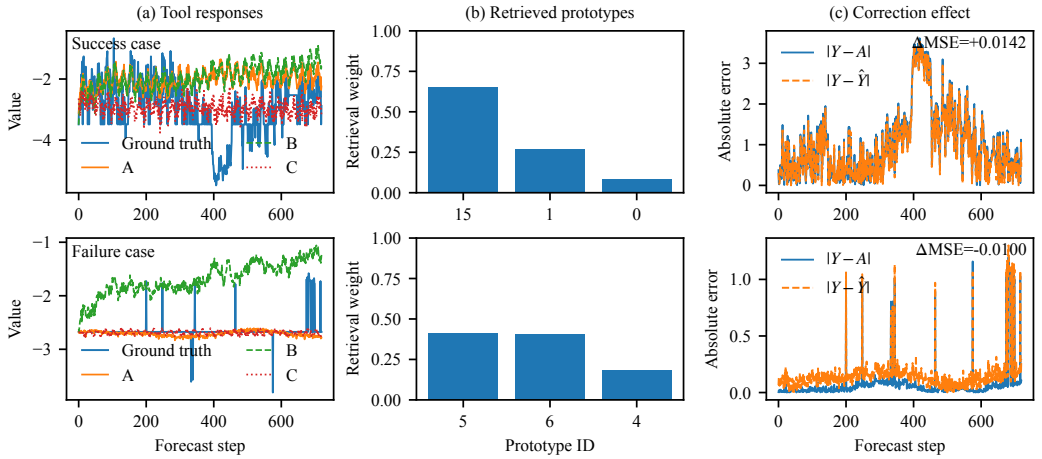

Figure 2: Representative success and failure cases of DiagMem on ETTh2 with $H = 720$. (a) Forecasts from the base forecaster A, auxiliary forecasters B and C, and the target. (b) Retrieved diagnostic prototypes and their weights. (c) Pointwise absolute errors before and after correction. Panels (a) and (c) show one representative variable, while ΔMSE is computed over all variables and forecast positions; positive values indicate improvement.

4.4 Qualitative Analysis

Diagnostic memory case study. We use a deterministic median-based protocol to select representative cases. Among the 28 F-PT dataset–horizon settings, the setting whose mean run-level relative MSE reduction is closest to the median is ETTh2 at $H = 720$, with a mean run-level reduction of 1.57%. Within this setting, we select the run closest to the median run-level relative MSE reduction, and then choose the windows closest to the median positive and negative correction gains as the success and failure cases, respectively. This post-hoc selection is used only for visualization and does not affect model selection or reported quantitative results.

Figure 2 shows the resulting representative cases. In the successful case, DiagMem reduces the window-level MSE by 0.0142 and retrieves prototypes 15, 1, and 0 with weights 0.650, 0.267, and 0.083, respectively. The correction reduces the absolute error over multiple forecast positions. In the failure case, the window-level correction gain is $\Delta\text{MSE} = -0.0100$, while prototypes 5, 6, and 4 are retrieved with weights 0.412, 0.405, and 0.183. Here the correction increases the error over several positions. Together, the two cases illustrate how state-dependent retrieval leads to different correction behaviors across forecasting windows.

5 Conclusion

We presented DiagMem, a diagnostic-memory framework for agentic time series forecasting that corrects forecasts from a frozen base forecaster. Completed forecasting cases connect cross-model disagreement with verified error feedback, while state-conditioned prototypes organize this experience and an LLM summarizes the aggregated evidence as structured scores and semantic descriptions. Retrieved diagnostic knowledge conditions a numerical executor that estimates trend and seasonal residual corrections. Across seven benchmarks and four forecast horizons, DiagMem configurations attain the lowest horizon-averaged MSE on five datasets. Relative to uncorrected Backbone references trained under the DiagMem protocol, they achieve lower MSE in 78 of 84 evaluated settings across base forecasters, datasets, and horizons. Ablation results support the contributions of auxiliary disagreement, state-conditioned diagnostic memory, and the joint use of structured scores and semantic descriptions. Future work will investigate jointly evolving diagnostic memory and tool-use policies under nonstationary dynamics and open-ended forecaster pools.

AI USE STATEMENT

In this work, generative AI tools were used to provide feedback on research methodology and experimental design, assist with method implementation and result interpretation, and aid manuscript editing and language polishing. All AI-assisted content, code, and analyses were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and analyses produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, forecasting and data protocols, baseline configurations, evaluation metrics, and key implementation settings in the main text. Additional hyperparameters, data-processing procedures, model configurations, ablation settings, and experimental details are provided in the Appendix to improve experimental transparency and support reproducibility.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026.

Vitor Cerqueira, Luís Torgo, Fábio Pinto, and Carlos Soares. Arbitrage of forecasting experts. *Machine Learning*, 108(6):913–944, June 2019. ISSN 1573-0565. doi: 10.1007/s10994-018-05774-y.

Yilong Chen, Xiao Qin, Chenghao Liu, Liang Wu, Noelle I. Samia, and Kaize Ding. LLM agents for time-series: A survey, 2026.

Mingyue Cheng, Xiaoyu Tao, Qi Liu, Ze Guo, and Enhong Chen. Position: Beyond model-centric prediction – agentic time series forecasting, 2026.

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*, ICML'24, Vienna, Austria, 2024. JMLR.org.

Sarkar Snigdha Sarathi Das, Palash Goyal, Mihir Parmar, Nanyun Peng, Vishy Tirumalashetty, Chun-Liang Li, Rui Zhang, Jinsung Yoon, and Tomas Pfister. Nexus : An agentic framework for time series forecasting, 2026.

Qi Guo, Baolin Zhao, Mingchen Song, and Guoqiang Zhong. A survey of deep learning for time series forecasting: Taxonomy, analysis and future directions. *IEEE Transactions on Knowledge and Data Engineering*, 38(5):2541–2560, 2026. doi: 10.1109/TKDE.2026.3671378.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21774–21797. PMLR, July 2025.

Yuchen He, Yueyang Cang, Zhiyuan Ning, Ningyu Wang, and Li Shi. RATL: Learning from Retrieved Residuals for Robust Multivariate Time-Series Forecasting, 2026.

Tao Hong, Pierre Pinson, and Shu Fan. Global energy forecasting competition 2012. *International Journal of Forecasting*, 30(2):357–363, 2014. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2013.07.001.

Gunjan Jalori, Preetika Verma, and Sercan O Arik. FLAIRR-TS - forecasting LLM-agents with iterative refinement and retrieval for time series. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15427–15437, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.834.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James 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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23857–23880, 2024.

Yanfei Kang, Wei Cao, Fotios Petropoulos, and Feng Li. Forecast with forecasts: Diversity matters. *European Journal of Operational Research*, 301(1):180–190, 2022. ISSN 0377-2217. doi: 10.1016/j.ejor.2021.10.024.

Minkyoung Kim, Daeun Ji, Yohan Lee, Beomsoo Kim, and Beakcheol Jang. CTRL: Control-based time series forecasting with LLM-guided residual learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21952–21968, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1104.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Sigir '18, pp. 95–104, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210006.

Daojun Liang, Qi Li, Yinglong Wang, Jing Chen, Hu Zhang, Xiaoxiao Cui, Qizheng Wang, and Shuo Li. The forecast after the forecast: A post-processing shift in time series. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuhua Liao, Zetian Wang, Qiangqiang Nie, and Zhenhua Zhang. Bridging the last mile of time series forecasting with LLM agents, 2026.

Kenneth C. Lichtendahl and Robert L. Winkler. Why do some combinations perform better than others? *International Journal of Forecasting*, 36(1):142–149, 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2019.03.027.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379 (2194):20200209, February 2021. ISSN 1364-503X. 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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 11116–11140, 2024.

Minh Hoang Nguyen, Van Dai Do, Huu Hiep Nguyen, Dung Nguyen, Kien Do, and Hung Le. Reviving error correction in modern deep time-series forecasting. In *Forty-Third International Conference on Machine Learning*, 2026.

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

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. TS-RAG: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 163170–163199. Curran Associates, Inc., 2025. doi: 10.52202/085713-5448.

Bokai Pan, Mingyue Cheng, Zhiding Liu, Shuo Yu, Xiaoyu Tao, Yuchong Wu, Qi Liu, Defu Lian, and Enhong Chen. CastFlow: Learning role-specialized agentic workflows for time series forecasting, 2026.

Xiaoyu Tao, Mingyue Cheng, Ze Guo, Shuo Yu, Yaguo Liu, Qi Liu, and Shijin Wang. MemCast: Memory-driven time series forecasting with experience-conditioned reasoning. In *Forty-Third International Conference on Machine Learning*, 2026a.

Xiaoyu Tao, Mingyue Cheng, Chuang Jiang, Tian Gao, Huanjian Zhang, and Yaguo Liu. Cast-R1: Learning tool-augmented sequential decision policies for time series forecasting, 2026b.

Xiaoyu Tao, Mingyue Cheng, Bokai Pan, Chuang Jiang, Huanjian Zhang, Tian Gao, Yaguo Liu, Qi Liu, and Enhong Chen. CastFSR: A fast–slow–reflect agentic reasoning framework for context-aware time series forecasting, 2026c.

Allan Timmermann. Chapter 4 forecast combinations. volume 1 of *Handbook of Economic Forecasting*, pp. 135–196. Elsevier, 2006. doi: 10.1016/S1574-0706(05)01004-9.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4294–4302. PMLR, May 2026.

Xiaoqian Wang, Rob J. Hyndman, Feng Li, and Yanfei Kang. Forecast combinations: An over 50-year review. *International Journal of Forecasting*, 39(4):1518–1547, 2023. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2022.11.005.

Haixu Wu, Teng Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. *ArXiv*, abs/2210.02186, 2022.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. TimeRAG: Boosting LLM time series forecasting via retrieval-augmented generation. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889933.

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.

Haokun Zhao, Xiang Zhang, Jiaqi Wei, Yiwei Xu, Yuting He, Siqi Sun, and Chenyu You. TimeSeriesScientist: A general-purpose AI agent for time series analysis, 2025.

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

Shiqiao Zhou, Holger Schöner, Zipeng Wu, Edouard Fouché, IAG Wilson, and Shuo Wang. Stationarity-aware retrieval-augmented time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, Kdd '26, pp. 6875–6886, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 979-8-4007-2259-2. doi: 10.1145/3770855.3817813.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, July 2022.

Table 3: Dataset statistics and chronological segment lengths in observations. Calibration is the validation segment for standalone baselines and is subdivided for DiagMem.

Dataset	Variables	Interval	Source rows	Train	Calibration	Test
ETTh1	7	1 h	17,420	8,640	2,880	2,880
ETTh2	7	1 h	17,420	8,640	2,880	2,880
ETTm1	7	15 min	69,680	34,560	11,520	11,520
ETTm2	7	15 min	69,680	34,560	11,520	11,520
ECL	321	1 h	26,304	18,412	2,632	5,260
Solar	137	10 min	52,560	36,792	5,256	10,512
Wind	7	1 h	26,569	18,598	2,658	5,313

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

A Additional Experimental Details

A.1 Data Processing and Exact Temporal Partitions

Table 3 gives the source lengths and chronological partitions. ETT uses chronological train/calibration/test splits of 12/4/4 months, with 30 days per month. The other datasets use chronological 70%/10%/20% train/calibration/test splits; Table 3 gives the exact segment lengths. Each variable is standardized using the mean and standard deviation of the training segment. All numeric channels are forecasting targets. Wind forecasts the seven historical power series `wp1`–`wp7`. Solar uses zero-valued calendar features; the other datasets use timestamp-derived features.

Window construction. Windows advance by one observation. The input length is $L = 96$, and the entire H-step target must lie within the corresponding segment. Calibration and test windows may use the final L observations preceding their segment as history. Thus, for segment lengths $(n_{\mathrm{tr}}, n_{\mathrm{cal}}, n_{\mathrm{te}})$, the window counts are $N_{\mathrm{tr}} = n_{\mathrm{tr}} - L - H + 1$, $N_{\mathrm{cal}} = n_{\mathrm{cal}} - H + 1$, and $N_{\mathrm{te}} = n_{\mathrm{te}} - H + 1$. Table 4 lists every count used in the experiments.

Forecaster fitting and selection. Within the training-window pool, set $b = \lfloor 0.85 N_{\mathrm{tr}} \rfloor$. The forecasting tools fit on window indices $0, \ldots, b - H - 1$ and select checkpoints on $b, \ldots, N_{\mathrm{tr}} - 1$. The H intervening windows are purged. Standalone supervised baselines fit on the full training segment and select checkpoints on the calibration segment.

Memory, correction, and validation. Let $R = N_{\mathrm{cal}} - 2H$ and let $q = 4$ for ETTm1/ETTm2 and $q = 1$ otherwise. The memory and correction-training counts are $n_{\mathrm{mem}} = q\lfloor \lfloor 0.2R \rfloor / q \rfloor$ and $n_{\mathrm{corr}} = q\lfloor \lfloor 0.6R \rfloor / q \rfloor$. The remaining $n_{\mathrm{val}} = R - n_{\mathrm{mem}} - n_{\mathrm{corr}}$ windows select the correction model. Using half-open index intervals in the calibration-window pool, the three sets are

$$\begin{aligned} \mathcal{D}_{\mathrm{mem}} &= [0, n_{\mathrm{mem}}), \\ \mathcal{D}_{\mathrm{corr}} &= [n_{\mathrm{mem}} + H, n_{\mathrm{mem}} + H + n_{\mathrm{corr}}), \\ \mathcal{D}_{\mathrm{val}} &= [n_{\mathrm{mem}} + n_{\mathrm{corr}} + 2H, N_{\mathrm{cal}}). \end{aligned}$$

Each gap removes exactly H candidate windows, so earlier targets have matured before the next stage starts. The factor q rounds allocation sizes to multiples of four for the minute-level ETT datasets; window stride remains one.

Metric aggregation. MSE and MAE are computed in the standardized space across all target variables. For each dataset and horizon, the three DiagMem configurations, all ablation variants, and DLinear, PatchTST, iTransformer, FEDformer, and TimesNet are averaged over three runs. GPT4TS, Time-LLM, TimesFM, and Chronos-2 each contribute one run per setting. Four-horizon

Table 4: Exact window counts. Fit and Select are the internal forecaster partitions; Mem, Corr, and Val denote $\mathcal{D}_{\mathrm{mem}}$, $\mathcal{D}_{\mathrm{corr}}$, and $\mathcal{D}_{\mathrm{val}}$. Each of the three purge intervals contains H windows: one within Train and two within Calibration.

Dataset	H	Train	Fit	Select	Calibration	Mem	Corr	Val	Test
ETTh1	96	8449	7085	1268	2785	518	1555	520	2785
ETTh1	192	8353	6908	1253	2689	461	1383	461	2689
ETTh1	336	8209	6641	1232	2545	374	1123	376	2545
ETTh1	720	7825	5931	1174	2161	144	432	145	2161
ETTh2	96	8449	7085	1268	2785	518	1555	520	2785
ETTh2	192	8353	6908	1253	2689	461	1383	461	2689
ETTh2	336	8209	6641	1232	2545	374	1123	376	2545
ETTh2	720	7825	5931	1174	2161	144	432	145	2161
ETTm1	96	34369	29117	5156	11425	2244	6736	2253	11425
ETTm1	192	34273	28940	5141	11329	2188	6564	2193	11329
ETTm1	336	34129	28673	5120	11185	2100	6304	2109	11185
ETTm1	720	33745	27963	5062	10801	1872	5616	1873	10801
ETTm2	96	34369	29117	5156	11425	2244	6736	2253	11425
ETTm2	192	34273	28940	5141	11329	2188	6564	2193	11329
ETTm2	336	34129	28673	5120	11185	2100	6304	2109	11185
ETTm2	720	33745	27963	5062	10801	1872	5616	1873	10801
ECL	96	18221	15391	2734	2537	469	1407	469	5165
ECL	192	18125	15214	2719	2441	411	1234	412	5069
ECL	336	17981	14947	2698	2297	325	975	325	4925
ECL	720	17597	14237	2640	1913	94	283	96	4541
Solar	96	36601	31014	5491	5161	993	2981	995	10417
Solar	192	36505	30837	5476	5065	936	2808	937	10321
Solar	336	36361	30570	5455	4921	849	2549	851	10177
Solar	720	35977	29860	5397	4537	619	1858	620	9793
Wind	96	18407	15549	2762	2563	474	1422	475	5218
Wind	192	18311	15372	2747	2467	416	1249	418	5122
Wind	336	18167	15105	2726	2323	330	990	331	4978
Wind	720	17783	14395	2668	1939	99	299	101	4594

summaries take the arithmetic mean of the horizon-level values. Relative reductions are calculated from unrounded metrics as $100(m_{\mathrm{reference}} - m_{\mathrm{Full}})/m_{\mathrm{reference}}$ within each setting and then averaged with equal weight. All rankings and strict improvement counts are computed before rounding the displayed metrics.

A.2 Forecasting Tools and Baseline Configurations

The three forecasting tools. PatchTST, DLinear, and iTransformer each use Adam with MSE, batch size 16 on ECL and 32 elsewhere, at most 50 epochs, and checkpoint-selection patience 3. Weight decay and gradient clipping are zero. Cosine annealing uses $T_{\max} = 50$. The learning rates are 10^{-4}, 10^{-3}, and 10^{-4} for PatchTST, DLinear, and iTransformer, respectively. All tools are frozen after the internal chronological selection described in Appendix A.1. DLinear uses a moving-average kernel of 25 and shares each component projection across channels. PatchTST uses patches of length 16, stride 8, model width 512, dropout 0.1, and GELU; its feed-forward width is 512 on Solar and 2048 elsewhere. Its encoder depth and attention-head counts appear in Table 5. iTransformer uses eight heads, dropout 0.1, and GELU. Its model and feed-forward widths are both 512 on ECL/Solar and 128 elsewhere; encoder depth is three on ECL and two elsewhere.

Table 5: PatchTST encoder depth / attention heads. These architectures apply both to the DiagMem forecasting tool and the standalone PatchTST baseline.

Dataset	$H = 96$	$H = 192$	$H = 336$	$H = 720$
ETTh1	1/2	1/8	1/8	1/16
ETTh2	3/4	3/4	3/4	3/4
ETTm1	1/2	3/2	1/4	3/4
ETTm2	3/16	3/2	1/4	3/4
ECL	2/8	2/8	2/8	2/8
Solar	2/8	2/8	2/8	2/8
Wind	2/8	2/8	2/8	2/8

Standalone supervised baselines. DLinear, PatchTST, iTransformer, FEDformer, and TimesNet use Adam with MSE, zero weight decay, a maximum of 50 epochs, early-stopping patience 3, cosine learning-rate decay, dropout 0.1, and full-precision training. Table 6 specifies the learning rate and batch size of each task. The first three models use the architectures described above. FEDformer uses model width 512, feed-forward width 2048, eight heads, two encoder layers, one decoder layer, and a moving-average kernel of 25. FEDformer uses its Fourier variant with 32 randomly selected frequency modes. TimesNet uses two layers, five selected periods, and six convolutional kernels. Its model/feed-forward widths are 16/32 on ETTh1; 32/32 on ETTh2, Solar, and Wind; 256/512 on ECL; 64/64 for ETTm1 at $H \in \{96, 192\}$ and 16/32 at $H \in \{336, 720\}$; 32/32 for ETTm2 at $H \in \{96, 192, 336\}$ and 16/32 at $H = 720$. Decoder-based models use label length 48. Checkpoint selection minimizes validation MSE.

GPT4TS. GPT4TS uses the first six blocks of pretrained GPT-2. It adapts the input/output projections and GPT-2 layer-normalization and positional parameters. Training uses Adam with MSE, zero weight decay, learning rate 10^{-4}, 10 epochs, patience 3, and no mixed precision. The model/feed-forward widths are 768, the configured head count is four, patch length is 16, and the dropout argument is 0.3. Stride is 16 on ETTm1/ETTm2 and eight elsewhere. Cosine annealing has minimum learning rate 10^{-8} and $T_{\max} = 20$ on ETTh1/ETTh2 and 10 elsewhere. Batch size is 36 on ETT and Wind, six on ECL, and one on Solar.

Time-LLM. Time-LLM uses the first six blocks of frozen GPT-2 and trains the patch embedding, reprogramming modules, and prediction head. Its model/feed-forward widths are 16/32 on ECL and 32/128 elsewhere, with eight heads, patch length 16, stride eight, and dropout 0.1. The vocabulary is mapped to 1,000 text prototypes. Its input prompt combines the dataset description with the forecast length, input length, minimum, maximum, median, trend direction, and five autocorrelation lags of each input series. Training uses Adam, MSE, zero weight decay, BF16 mixed precision, and gradient checkpointing. The effective optimizer batch size is 24, except for ETTm2 at $H = 96$, where it is 16. Training microbatches and evaluation batches contain one window on ECL/Solar and four elsewhere. Table 7 gives the executed learning rates and budgets. For type1, OneCycle initializes the optimizer and the epoch-end update sets $\eta_e = \eta_0 2^{-(e-1)}$; TST steps OneCycle at each optimizer update with a warm-up fraction of 0.2.

TimesFM and Chronos-2. Both foundation models are evaluated zero-shot on standardized test histories of length 96 at each forecast horizon. All test windows and target channels are evaluated under the same metric protocol. TimesFM uses the pretrained TimesFM 2.5 model with 200M parameters and forecast batch size 32. Its adapter forecasts each channel independently and reassembles the point forecasts. Input normalization, the continuous-quantile head, flip invariance, positivity inference, and quantile-crossing repair are enabled. Chronos-2 uses its pretrained model in float32. Each multivariate window is one task. Its quantile API requests levels 0.1, 0.5, and 0.9, with context length 96, forecast batch size 256, and cross-task learning disabled; evaluation uses the returned point forecast.

Table 6: Standalone baseline training settings: learning rate / batch size for each dataset and horizon.

Dataset	H	DLinear	PatchTST	iTransformer	FEDformer	TimesNet
ETTh1	96	$10^{-3}/64$	$10^{-4}/64$	$10^{-4}/32$	$10^{-4}/32$	$10^{-4}/64$
ETTh1	192	$10^{-3}/64$	$10^{-4}/64$	$10^{-4}/32$	$10^{-4}/64$	$10^{-4}/32$
ETTh1	336	$5\times10^{-3}/64$	$10^{-4}/64$	$10^{-4}/64$	$10^{-4}/64$	$10^{-4}/32$
ETTh1	720	$10^{-3}/64$	$10^{-3}/64$	$10^{-4}/32$	$10^{-3}/64$	$10^{-4}/64$
ETTh2	96	$10^{-3}/32$	$10^{-4}/32$	$10^{-4}/64$	$10^{-4}/64$	$10^{-4}/32$
ETTh2	192	$10^{-3}/32$	$10^{-4}/64$	$10^{-4}/32$	$10^{-4}/32$	$5\times10^{-4}/32$
ETTh2	336	$5\times10^{-3}/64$	$10^{-4}/64$	$10^{-4}/64$	$5\times10^{-4}/64$	$5\times10^{-4}/32$
ETTh2	720	$5\times10^{-3}/32$	$10^{-4}/32$	$5\times10^{-4}/64$	$10^{-4}/32$	$10^{-4}/64$
ETTm1	96	$10^{-3}/64$	$10^{-3}/32$	$10^{-3}/32$	$10^{-4}/64$	$10^{-4}/64$
ETTm1	192	$10^{-3}/64$	$5\times10^{-4}/32$	$10^{-4}/32$	$5\times10^{-4}/32$	$10^{-4}/32$
ETTm1	336	$10^{-3}/32$	$10^{-3}/64$	$10^{-3}/32$	$10^{-3}/64$	$10^{-4}/32$
ETTm1	720	$5\times10^{-3}/32$	$5\times10^{-4}/64$	$5\times10^{-4}/32$	$5\times10^{-4}/32$	$10^{-4}/64$
ETTm2	96	$10^{-3}/64$	$10^{-4}/64$	$10^{-3}/32$	$10^{-4}/64$	$10^{-4}/64$
ETTm2	192	$10^{-3}/32$	$10^{-4}/32$	$10^{-4}/64$	$10^{-4}/32$	$10^{-3}/64$
ETTm2	336	$10^{-3}/32$	$10^{-4}/64$	$10^{-4}/64$	$10^{-4}/64$	$10^{-4}/64$
ETTm2	720	$5\times10^{-3}/64$	$10^{-4}/32$	$10^{-4}/64$	$5\times10^{-4}/32$	$10^{-3}/64$
ECL	96	$10^{-4}/16$	$5\times10^{-4}/16$	$5\times10^{-4}/16$	$5\times10^{-4}/16$	$5\times10^{-4}/16$
ECL	192	$10^{-4}/16$	$10^{-4}/16$	$5\times10^{-4}/16$	$10^{-4}/16$	$5\times10^{-4}/16$
ECL	336	$10^{-4}/16$	$10^{-4}/16$	$10^{-4}/16$	$10^{-4}/16$	$10^{-4}/16$
ECL	720	$10^{-4}/16$	$5\times10^{-4}/16$	$10^{-4}/16$	$10^{-4}/16$	$10^{-3}/16$
Solar	96	$10^{-4}/32$	$5\times10^{-4}/64$	$10^{-4}/32$	$10^{-4}/32$	$10^{-4}/64$
Solar	192	$10^{-4}/32$	$5\times10^{-4}/32$	$10^{-4}/64$	$10^{-4}/32$	$10^{-4}/32$
Solar	336	$10^{-4}/32$	$5\times10^{-4}/32$	$10^{-3}/64$	$10^{-4}/64$	$10^{-4}/64$
Solar	720	$10^{-4}/32$	$5\times10^{-4}/64$	$5\times10^{-4}/32$	$10^{-4}/32$	$10^{-4}/64$
Wind	96	$10^{-4}/32$	$10^{-3}/64$	$10^{-4}/64$	$10^{-4}/32$	$10^{-4}/64$
Wind	192	$10^{-4}/32$	$5\times10^{-4}/32$	$5\times10^{-4}/64$	$10^{-4}/64$	$10^{-3}/32$
Wind	336	$10^{-4}/32$	$5\times10^{-4}/64$	$10^{-3}/32$	$10^{-4}/64$	$10^{-4}/64$
Wind	720	$10^{-4}/64$	$10^{-3}/64$	$10^{-4}/32$	$10^{-4}/32$	$10^{-3}/64$

Table 7: Time-LLM training settings recorded in the evaluated runs. Epochs are maximum budgets; patience controls validation-MSE early stopping.

Dataset	Horizons	Learning rate	Epochs	Patience	Schedule
ETTh1	96, 720	0.01	50	3	type1
ETTh1	192, 336	0.01	100	10	type1
ETTh2	All	0.01	10	10	type1
ETTm1	All	0.001	50	3	TST
ETTm2	96	0.01	10	3	type1
ETTm2	192, 336, 720	0.002	10	3	TST
ECL, Solar, Wind	All	0.01	10	3	type1

A.3 Diagnostic Memory and Executor Training

Memory construction. The observable diagnostic state has $26D$ features. State standardization is fitted on $\mathcal{D}_{\mathrm{mem}}$; K-means uses 20 initializations and 16 prototypes. Retrieval uses the three nearest prototypes with temperature one. Evidence uses eight horizon bins and a moving-average kernel of 25. Qwen diagnoses trend and seasonal components separately in batches of four prototypes, giving eight logical requests per Full memory bank. We use Qwen (`Qwen3.8-27B`) for prototype

diagnosis. Decoding uses temperature zero, top-$p = 1$, top-$k = 20$, a 4,096-token output cap, and non-thinking mode. Appendix A.4 provides the diagnostic prompt and output format.

Semantic encoding. Frozen GPT-2 encodes the prototype descriptions. The combined trend and seasonal descriptions are tokenized with a maximum length of 256; overlength inputs are rejected. Masked mean pooling produces a 768-dimensional representation. Embeddings are centered feature-wise over the prototype bank and then normalized to unit ℓ_2 norm. A trainable linear layer and GELU project them to 32 dimensions. Both stages of the residual executor use hidden width 32 and dropout 0.1.

Optimization. The correction module uses Adam with learning rate 10^{-3}, default moment coefficients $(0.9, 0.999)$, $\epsilon = 10^{-8}$, zero weight decay, batch size 64, and gradient-norm clipping at five. Training runs for at most 100 epochs with early-stopping patience 10. ReduceLROnPlateau halves the learning rate after five epochs without improvement in validation forecast MSE. The residual output head is initialized to zero. The loss combines forecast MSE and trend and seasonal SmoothL1 supervision with $\lambda_{ts} = 0.1$ and $\lambda_t = \lambda_s = 0.5$. At each epoch, the correction strength is selected from $\{0, 0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7, 1\}$ on $\mathcal{D}_{\text{val}}$; the resulting forecast MSE selects the checkpoint. History alignment is the identity for $H = L$ and a learned temporal projection otherwise. Forecasters, prototype bank, diagnostic scores, and text encoder remain fixed during correction training.

Ablation settings. Correction variants retain the same data partitions, optimization budget, hidden width, component-loss weights, and validation criterion. NoAux uses only A in memory construction and fills the executor's auxiliary-disagreement slots with zeros. GlobalMemory replaces the 16-prototype bank with one global prototype. NumericCondition generates the eight structured scores from numerical prototype evidence using a two-layer MLP with hidden width 32, GELU, and sigmoid outputs, and uses learned prototype embeddings. PrototypeID keeps Qwen scores and substitutes learned prototype embeddings for GPT-2 representations. ResidualOnly learns from history and A without a diagnostic-memory condition. The descriptions in the main text specify the corresponding input and representation changes.

A.4 Diagnostic Prompt and Output Contract

The following system prompt diagnoses a batch of four trend prototypes. The component and requested prototype IDs vary across batches. Separate trend and seasonal calls provide the two diagnostic branches.

> You are an offline time-series trend prototype analyst. For each of the 4 multivariate historical prototypes in this batch, reason briefly from the historical error evidence and independently assign four discrete control values. Expected prototype IDs: [0, 1, 2, 3]. These values condition an additive correction of backbone A; they are not four synonyms:
>
> - b_reliability: usefulness of (B_prediction - A_prediction) as a correction direction for A. A high value requires B to have lower historical error than A and/or B-A to align consistently with A's matured error. - c_reliability: the analogous usefulness of (C_prediction - A_prediction), assessed independently from B. - memory_reliability: usefulness of the historical A-error patterns as diagnostic context. Increase it for stable signed/temporal error structure with adequate support and coverage; decrease it for dispersed or contradictory error structure. - confidence: clarity and consistency of the total evidence. It is a diagnostic input to the learned corrector, not a direct multiplier of the correction, and must not merely copy any source reliability.
>
> The four values are learned-corrector conditions, not ensemble weights or hard gates. They do not directly add a historical error vector to the forecast.
>
> Sign convention is mandatory: - A_error = actual - A_prediction. - Positive A_error means A predicted too low (under-prediction) and needs a positive residual correction. - Negative A_error means A predicted too high (over-prediction) and needs a negative residual correction. - AB_disagreement = B_prediction - A_prediction and AC_disagreement = C_prediction - A_prediction. - A disagreement with the same sign as A_error points in the needed residual direction, but an excessive magnitude can overshoot the truth. The opposite sign moves A away from truth. Assess usefulness with historical error magnitudes as well as signs. - B_error = A_error - AB_disagreement; C_error = A_error - AC_disagreement.

> Control levels: - 0.00: consistently harmful or unsupported - 0.25: usually harmful or weakly supported - 0.50: mixed or genuinely uncertain - 0.75: usually useful - 1.00: consistently useful
>
> Compare A/B/C bias and MAE, disagreement direction, eight-bin persistence, support, coverage, and dispersion before scoring. Assign every field independently. Equal values are allowed only when the evidence supports equality; never copy one score into all four fields by default.
>
> The semantic_description must contain 25 to 70 English words and state the factual relationships that justify the scores. It may describe under-prediction, over-prediction, direction, persistence, dispersion, alternation, bursts, phase, consensus, and historical deviation. It must not recommend a forecaster, prescribe a correction magnitude, or state a final forecast. Do not use reliable/unreliable, trust, prefer, better reference, worse reference, should use, or should ignore in semantic_description.
>
> Return exactly one JSON object without markdown. The top-level object must contain `component` and `prototypes`. Each item in `prototypes` must contain `prototype_id`, `semantic_description`, `b_reliability`, `c_reliability`, `memory_reliability`, and `confidence`.
>
> The prototypes array must contain exactly IDs [0, 1, 2, 3] once each, and every policy value must be one of 0.00, 0.25, 0.50, 0.75, 1.00. For each semantic_description, write about 45 to 55 English words. The hard validation range is 25 to 70 English words, but aim near 50 words to avoid boundary failures. Use two or three short factual sentences about levels, slopes, periodic patterns, dispersion, and observed historical error signs. Do not discuss actions or decisions. Never use correction, reliable, unreliable, trust, prefer, recommend, should use, should ignore, better reference, worse reference, or final forecast inside semantic_description. Keep policy judgments in the four numeric fields.

JSON output example. The complete response contains one item for each requested prototype ID. The example below illustrates the output format; the numeric values are not target scores.

```
{
  "component": "trend",
  "prototypes": [
    {
      "prototype_id": 0,
      "semantic_description":
        "Forecaster B remains above the base forecast across most horizon bins, while forecast
      "b_reliability": 0.75,
      "c_reliability": 0.50,
      "memory_reliability": 0.75,
      "confidence": 0.75
    }
  ]
}
```

User-message evidence. The user message specifies the component, horizon, prototype IDs, channel count, and eight-bin profile layout. Each prototype supplies support count, temporal-block count, temporal coverage, cluster dispersion, and the historical bias and MAE of A, B, and C. For each of A's forecast, B–A disagreement, C–A disagreement, and observed A error, it supplies eight-bin signed-mean and mean-absolute profiles. The forecast and error summaries pool window-level, per-channel statistics by their median. Support is the number of member windows. For temporal coverage, the chronologically ordered memory windows are grouped into blocks of H consecutive windows, including the final partial block; the temporal-block count is the number of blocks containing prototype members, and coverage is this count divided by the total number of blocks. State dispersion is the median Euclidean distance $\|\widetilde{q}_i - \nu_j\|_2$ from member states to their prototype center. The message states the error sign convention and notes that separately pooled medians need not preserve subtraction identities.

Parsing. Responses pass through JSON parsing and strict schema validation. The parser removes an optional reasoning prefix ending in `</think>` and surrounding Markdown code fences. Validation checks the component, exact prototype-ID set, absence of extra fields, allowed score levels, and a 25–70-word English description. The semantic descriptions exclude action recommendations and

policy-score terminology specified in the prompt. Invalid output triggers format-repair feedback; unresolved failures terminate the request.

A.5 COMPLETE FORECASTING RESULTS

Table 8 expands the main comparison into individual forecast horizons. Each dataset contains five rows: $H \in \{96, 192, 336, 720\}$ and their equally weighted average (Avg). The Avg rows reproduce Table 1, while the individual rows show how performance varies with the forecast horizon.

The horizon-level results reveal different patterns across datasets. On ETTm2, F-PT achieves the lowest MSE at all four horizons. On Solar, F-DL attains the lowest MSE at $H = 96$ and 192, whereas F-iT leads at $H = 336$ and 720. These results show how the relative performance of the three DiagMem configurations changes with the forecasting task.

Table 8: Complete forecasting results. Method abbreviations follow Table 1. Best values are bold and second-best values are underlined, with ranks computed from unrounded metrics.

Methods	H	F-PT		F-DL		F-iT		DLinear		PatchTST		iTransformer		FEDformer		TimesNet		GPT4TS		Time-LLM		TimesFM		Chronos-2	
Metric		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	0.390	0.410	**0.386**	0.399	0.391	0.411	0.393	0.409	0.400	0.411	0.392	0.409	0.392	0.431	0.408	0.424	0.387	0.397	0.425	0.427	0.429	0.402	0.424	**0.397**
	192	0.454	0.453	0.447	0.445	0.448	0.454	0.440	0.435	0.445	0.440	0.447	0.441	**0.436**	0.459	0.454	0.451	0.443	**0.428**	0.467	0.450	0.485	0.432	0.484	0.431
	336	0.521	0.493	0.503	0.483	0.506	0.488	0.512	0.490	0.475	0.458	0.495	0.467	**0.475**	0.475	0.521	0.488	0.487	**0.450**	0.496	0.465	0.526	0.450	0.536	0.456
	720	0.596	0.550	0.511	0.508	0.607	0.552	0.518	0.512	0.525	0.505	0.513	0.498	0.561	0.547	0.516	0.494	0.516	0.483	0.537	0.514	**0.498**	**0.457**	0.526	0.471
	Avg	0.490	0.476	0.462	0.459	0.488	0.476	0.465	0.462	0.461	0.454	0.462	0.454	0.466	0.478	0.475	0.464	**0.458**	0.439	0.481	0.464	0.484	**0.435**	0.492	0.439
ETTh2	96	0.297	0.355	**0.289**	0.350	0.294	0.350	0.313	0.371	0.307	0.354	0.304	0.353	0.350	0.394	0.331	0.372	0.302	0.354	0.306	0.355	0.330	0.345	0.317	**0.344**
	192	0.374	0.402	**0.369**	0.404	0.379	0.402	0.416	0.435	0.383	0.404	0.383	0.400	0.432	0.441	0.421	0.419	0.392	0.411	0.394	0.407	0.420	0.402	0.413	**0.400**
	336	**0.398**	**0.418**	0.399	0.432	0.425	0.433	0.484	0.479	0.431	0.439	0.427	0.434	0.490	0.490	0.459	0.452	0.425	0.437	0.435	0.444	0.453	0.434	0.449	0.433
	720	0.433	0.447	0.574	0.561	**0.429**	0.453	0.614	0.558	0.446	0.460	0.448	0.456	0.479	0.490	0.473	0.473	0.438	0.459	0.440	0.453	0.450	0.443	0.443	**0.442**
	Avg	**0.376**	0.406	0.408	0.437	0.382	0.410	0.457	0.460	0.392	0.414	0.390	0.411	0.438	0.454	0.421	0.429	0.389	0.415	0.394	0.415	0.413	0.406	0.406	**0.405**
ETTm1	96	**0.318**	**0.364**	0.331	0.370	0.335	0.375	0.343	0.374	0.332	0.367	0.338	0.372	0.370	0.417	0.337	0.377	0.340	0.365	0.345	0.376	0.841	0.560	0.592	0.452
	192	**0.369**	0.397	0.373	0.394	0.372	0.397	0.383	0.393	0.390	0.397	0.385	0.397	0.415	0.445	0.397	0.405	0.372	**0.382**	0.380	0.391	0.881	0.582	0.607	0.467
	336	0.404	0.418	0.405	0.417	0.411	0.421	0.418	0.423	0.407	0.411	0.432	0.425	0.464	0.477	0.419	0.424	**0.402**	**0.403**	0.410	0.409	0.872	0.588	0.618	0.482
	720	**0.457**	0.454	0.462	0.463	0.473	0.463	0.490	0.468	0.487	0.458	0.501	0.463	0.508	0.496	0.510	0.467	0.466	**0.438**	0.473	0.449	0.905	0.607	0.646	0.508
	Avg	**0.387**	0.408	0.393	0.411	0.398	0.414	0.409	0.414	0.404	0.408	0.414	0.414	0.439	0.459	0.416	0.418	0.395	**0.397**	0.402	0.406	0.875	0.584	0.616	0.477
ETTm2	96	**0.172**	**0.263**	0.177	0.269	0.175	0.267	0.192	0.288	0.181	0.266	0.185	0.267	0.200	0.289	0.189	0.268	0.182	0.267	0.188	0.273	0.257	0.312	0.210	0.273
	192	**0.232**	0.307	0.239	0.313	0.236	0.311	0.262	0.335	0.246	0.308	0.252	0.312	0.264	0.325	0.257	**0.307**	0.248	0.309	0.255	0.315	0.333	0.357	0.274	0.314
	336	**0.288**	**0.346**	0.293	0.350	0.293	0.350	0.321	0.374	0.312	0.349	0.313	0.351	0.329	0.365	0.336	0.357	0.311	0.350	0.316	0.351	0.395	0.393	0.340	0.355
	720	**0.378**	0.407	0.380	0.411	0.390	**0.404**	0.511	0.492	0.417	0.412	0.412	0.405	0.428	0.427	0.425	0.406	0.410	0.409	0.424	0.411	0.501	0.448	0.448	0.412
	Avg	**0.267**	**0.331**	0.272	0.336	0.273	0.333	0.321	0.373	0.289	0.334	0.291	0.334	0.305	0.351	0.302	0.334	0.288	0.334	0.296	0.338	0.372	0.378	0.318	0.338
ECL	96	0.161	0.254	0.174	0.268	0.150	0.243	0.195	0.277	0.163	0.254	**0.150**	**0.243**	0.201	0.316	0.171	0.273	0.194	0.278	0.231	0.325	0.223	0.293	0.214	0.274
	192	0.174	0.266	0.190	0.282	**0.167**	**0.259**	0.194	0.281	0.175	0.268	0.169	0.260	0.206	0.322	0.187	0.287	0.196	0.280	0.239	0.333	0.220	0.292	0.213	0.279
	336	0.192	0.286	0.202	0.295	**0.181**	**0.274**	0.207	0.296	0.194	0.288	0.185	0.277	0.206	0.322	0.197	0.298	0.212	0.297	0.250	0.339	0.236	0.307	0.230	0.295
	720	0.236	0.324	0.243	0.330	**0.219**	0.308	0.243	0.330	0.231	0.316	0.221	**0.308**	0.235	0.345	0.240	0.322	0.253	0.328	0.281	0.359	0.278	0.341	0.279	0.329
	Avg	0.191	0.282	0.202	0.294	**0.179**	**0.271**	0.210	0.296	0.191	0.281	0.181	0.272	0.212	0.326	0.199	0.295	0.214	0.296	0.250	0.339	0.239	0.308	0.234	0.294
Solar	96	0.197	0.248	**0.196**	0.266	0.197	**0.243**	0.287	0.374	0.213	0.258	0.214	0.254	0.286	0.382	0.240	0.282	0.239	0.291	0.277	0.300	0.875	0.503	0.639	0.343
	192	0.234	0.276	**0.233**	0.299	0.239	**0.273**	0.317	0.395	0.245	0.279	0.244	0.275	0.292	0.377	0.280	0.303	0.263	0.301	0.297	0.312	0.937	0.528	0.527	0.324
	336	0.251	0.287	0.255	0.307	**0.248**	**0.277**	0.351	0.413	0.277	0.299	0.267	0.284	0.306	0.383	0.304	0.309	0.302	0.326	0.357	0.343	0.909	0.520	0.494	0.323
	720	0.264	0.297	0.284	0.332	**0.255**	0.284	0.355	0.412	0.260	0.293	0.260	**0.282**	0.306	0.377	0.309	0.312	0.275	0.306	0.349	0.342	0.930	0.528	0.451	0.319
	Avg	0.236	0.277	0.242	0.301	**0.235**	**0.269**	0.327	0.398	0.249	0.282	0.246	0.274	0.297	0.380	0.283	0.302	0.270	0.306	0.320	0.324	0.913	0.520	0.528	0.327
Wind	96	1.144	0.889	**1.113**	0.873	1.196	0.895	1.117	**0.870**	1.143	0.905	1.224	0.900	1.219	0.902	1.247	0.907	1.149	0.879	1.173	0.888	1.533	0.948	1.400	0.922
	192	1.217	0.922	**1.158**	0.894	1.257	0.928	1.162	**0.893**	1.189	0.926	1.289	0.933	1.279	0.938	1.330	0.937	1.220	0.911	1.226	0.935	1.603	0.983	1.485	0.966
	336	1.237	0.935	1.182	0.907	1.262	0.932	**1.182**	**0.902**	1.214	0.934	1.297	0.941	1.317	0.961	1.332	0.953	1.228	0.921	1.226	0.928	1.614	0.993	1.513	0.982
	720	1.311	0.990	1.218	0.955	1.353	1.000	**1.209**	**0.915**	1.228	0.945	1.295	0.944	1.385	0.978	1.368	0.976	1.242	0.926	1.255	0.937	1.685	1.020	1.542	0.997
	Avg	1.227	0.934	1.168	0.907	1.267	0.939	**1.168**	**0.895**	1.194	0.927	1.276	0.930	1.300	0.945	1.319	0.943	1.210	0.909	1.220	0.922	1.609	0.986	1.485	0.967

A.6 Additional Cross-Backbone Ablation Results

Tables 9 and 10 report the four-horizon average ablation results with DLinear and iTransformer as the base forecasters, complementing the PatchTST results in Table 2. The corresponding cross-backbone comparisons are summarized in the main text.

Table 9: Four-horizon average ablation results with DLinear as the base forecaster. RO: ResidualOnly; GM: GlobalMemory; NC: NumericCondition; PID: PrototypeID. Best values are bold and second-best values are underlined, using unrounded metrics.

Variant	Backbone		RO		NoAux		GM		NC		PID		Full	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.458**	**0.454**	0.474	0.467	0.509	0.492	0.503	0.490	0.513	0.498	0.499	0.485	0.462	0.459
ETTh2	0.594	0.535	0.491	0.489	0.503	0.497	0.498	0.494	0.495	0.489	0.475	0.478	**0.408**	**0.437**
ETTm1	0.410	0.413	0.400	0.412	0.430	0.444	0.431	0.442	0.422	0.440	0.413	0.428	**0.393**	**0.411**
ETTm2	0.409	0.433	0.300	0.359	0.317	0.373	0.367	0.402	0.353	0.396	0.301	0.360	**0.272**	**0.336**
ECL	0.211	0.298	0.209	0.298	0.221	0.309	0.215	0.306	0.218	0.308	0.208	0.300	**0.202**	**0.294**
Solar	0.331	0.402	0.298	0.359	0.301	0.364	0.258	0.318	0.258	0.322	0.252	0.311	**0.242**	**0.301**
Wind	1.173	**0.900**	1.169	0.908	1.188	0.916	1.184	0.914	1.181	0.913	1.178	0.912	**1.168**	0.907

Table 10: Four-horizon average ablation results with iTransformer as the base forecaster. RO: ResidualOnly; GM: GlobalMemory; NC: NumericCondition; PID: PrototypeID. Best values are bold and second-best values are underlined, using unrounded metrics.

Variant	Backbone		RO		NoAux		GM		NC		PID		Full	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.495	**0.475**	0.494	0.479	0.550	0.520	0.555	0.520	0.543	0.515	0.522	0.500	**0.488**	0.476
ETTh2	0.391	0.410	0.387	0.412	0.394	0.419	0.397	0.423	0.399	0.423	0.393	0.417	**0.382**	**0.410**
ETTm1	0.418	0.417	0.405	0.417	0.432	0.444	0.428	0.440	0.420	0.436	0.407	0.423	**0.398**	**0.414**
ETTm2	0.293	0.337	0.277	0.336	0.295	0.348	0.304	0.358	0.307	0.356	0.294	0.348	**0.273**	**0.333**
ECL	0.182	0.273	0.181	0.273	0.183	0.275	0.183	0.275	0.183	0.275	0.181	0.273	**0.179**	**0.271**
Solar	0.251	0.282	0.237	0.272	0.239	0.274	0.240	0.274	0.240	0.274	0.237	0.271	**0.235**	**0.269**
Wind	1.284	**0.931**	1.277	0.941	1.283	0.949	1.297	0.955	1.303	0.952	1.282	0.946	**1.267**	0.939