DMRC: RESIDUAL-MANIFOLD MEMORY ADAPTATION FOR FROZEN TIME-SERIES FORECASTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval augmentation methods have achieved notable success in time series forecasting by reusing historical input information. However, when events such as trend drifts or changes in volatility alter the prediction errors associated with similar inputs, these methods may struggle to retrieve historical correction information applicable to the current forecast. To address this limitation, we propose Dynamic Memory-enhanced Residual Correction (DMRC), which adapts frozen forecasters by dynamically adjusting the retrieval of historical residual prototypes according to their relevance to current forecasts. DMRC constructs hierarchical residual prototypes offline and retrieves complementary principal-direction and orthogonal residual components for forecast correction. During online forecasting, multi-view and temporal-overlap consistency checks provide label-free feedback to update retrieval priorities for the principal-direction component, reducing reliance on input similarity alone when historical error patterns become less applicable. This mechanism enables adaptive use of historical correction information while keeping both the forecaster and memory contents frozen. Across seven datasets, four forecast horizons, and six frozen forecasters, DMRC mostly outperforms the frozen baseline, with MSE reductions of up to 15.10%.

1 INTRODUCTION

Time series forecasting is shifting toward foundation forecasters that can be reused across domains, variables, and forecast horizons. Models such as PatchTST and iTransformer have advanced long-term forecasting (Nie et al., 2022; Liu et al., 2024b), while foundation forecasters including Chronos, TimesFM, and Moirai have improved transfer across heterogeneous tasks (Ansari et al., 2024; Das et al., 2023; Woo et al., 2024). Yet large-scale pretraining cannot eliminate deployment non-stationarity, and frozen or black-box forecasters cannot be repeatedly retrained for local shifts. Moreover, retrieval by input similarity may not identify reusable errors. We therefore hypothesize an *error manifold*: states with similar model-aware representations may exhibit more concentrated conditional residuals, while input similarity alone does not guarantee residual similarity.

Forecasting-specific test-time adaptation addresses deployment non-stationarity by updating lightweight calibration or output-space modules while keeping most of the pretrained forecaster fixed (Kim et al., 2025; Medeiros et al., 2025; Im & Kwon, 2026). Online forecasting methods further handle concept drift by estimating distribution changes and translating them into model adjustments (Zhao & Shen, 2025). Although these designs reduce adaptation cost, they still require deployment-time optimization access. Similarly, parameter-efficient time-series forecasters based on LLM reprogramming or learned prompts freeze most backbone parameters but still optimize task-specific reprogramming, projection, or prompt components (Jin et al., 2024; He et al., 2026). External retrieval avoids model updates but organizes nonparametric experience around inputs rather than frozen-forecaster errors (Khandelwal et al., 2020; Borgeaud et al., 2022; Jing et al., 2022), remaining misaligned with residual correction.

We propose Dynamic Memory-enhanced Residual Correction (DMRC), which compresses historical states and errors into residual-aware key–value prototypes. Figure 1 gives an overview. During deployment, the forecaster, projector, gate, and memory remain frozen; only the channel-wise retrieval priorities are updated. Each query identifies a coarse residual pattern, performs within-group top-K retrieval with local matching and the corresponding channel-wise priorities, and applies gated

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

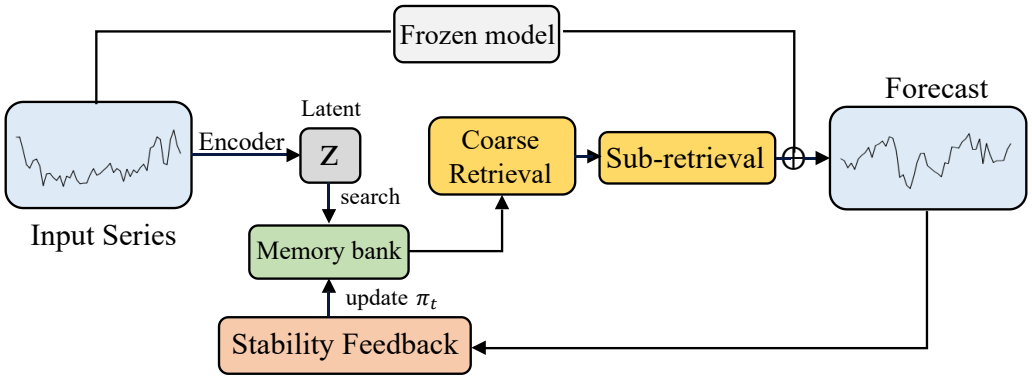


Figure 1: Overview of the DMRC framework. Offline, historical states and errors are compressed into key–value prototypes. Online, the forecaster, projector, gate, and memory remain frozen; stability feedback updates only the channel-wise retrieval distributions after each prediction.

correction. Multi-view and temporal-overlap consistency provide label-free feedback, while source-normalized responsibilities and KL-regularized mirror descent update retrieval priorities. DMRC therefore changes selected historical error evidence, not the frozen forecaster's behavior.

The practical challenge is therefore not to continually retrain the forecaster, but to determine when previously observed residual patterns remain useful for the current forecast. This requires distinguishing transferable error structures from merely similar inputs, while avoiding unnecessary changes to the deployed model. DMRC addresses this challenge by treating residual retrieval as the adaptive component and the pretrained forecaster as a fixed reference. The following sections develop this formulation and evaluate whether such residual-aware adaptation provides consistent benefits across forecasting tasks and model architectures.

Our contributions are as follows:

- We show that input similarity does not ensure residual reusability and propose the error manifold hypothesis, which organizes frozen-forecaster experience by conditional error structure.
- We propose DMRC, which freezes the forecaster, projector, gates, and memory contents, performs residual-aware coarse-to-fine top-K correction, and updates only the channel-wise retrieval distributions using label-free feedback, source-normalized responsibilities, and KL-regularized mirror descent. Coverage-aware gating suppresses unreliable corrections.
- Across seven datasets, six frozen forecasters, and four forecast horizons, DMRC mostly outperforms input retrieval and static and dynamic residual-memory baselines; ablations confirm the effectiveness of residual representation and dynamic retrieval.

2 Preliminaries

2.1 Problem Setup and Online Protocol

Let $\mathbf{x}_t \in \mathbb{R}^{L\times C}$ be the length-L history with C variables observable up to time t, and $\mathbf{y}_t \in \mathbb{R}^{H\times C_y}$ the subsequent H-step target; the frozen forecaster outputs $\hat{\mathbf{y}}_{0,t} = \mathcal{F}(\mathbf{x}_t)$. Real-world forecasting streams are often non-stationary: temporal dynamics evolve over time and the test distribution may shift continuously, so consecutive windows should not be treated as samples from a fixed i.i.d. distribution (Kim et al., 2024; 2025; Liu et al., 2024a). Current forecasts use only $\mathbf{x}_t$, frozen-model outputs, and external states formed before t; $\mathbf{y}_t$ is used neither for prediction nor online updates. Ground-truth futures in offline training or calibration data are used only to construct residual prototypes and train lightweight modules.

The online state consists of channel-wise retrieval distributions $\boldsymbol{\pi}_{t,c} \in \Delta^B$, $c = 1, \ldots, C$, over the B frozen residual prototypes for each channel:

$$\tilde{\mathbf{y}}_t = \hat{\mathbf{y}}_{0,t} + \mathcal{C}(\mathbf{x}_t, \hat{\mathbf{y}}_{0,t}; \mathcal{M}_0, \boldsymbol{\Pi}_t), \qquad \boldsymbol{\pi}_{t+1,c} = \mathcal{U}(\boldsymbol{\pi}_{t,c}, \mathbf{f}_{t,c}), \tag{1}$$

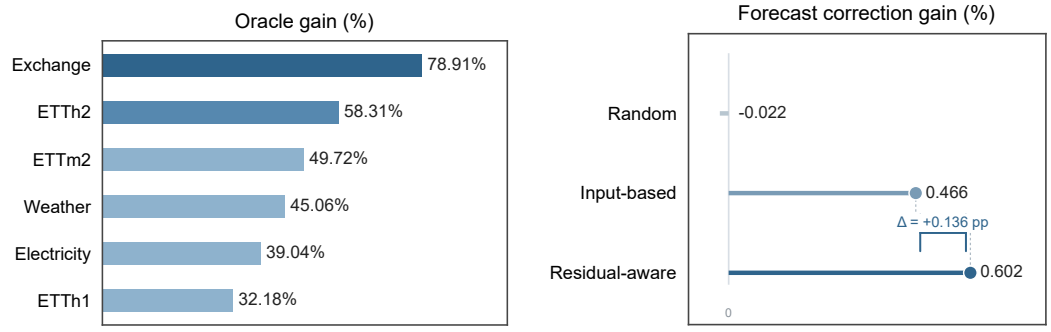
where $\boldsymbol{\Pi}_t = [\boldsymbol{\pi}_{t,1}, \ldots, \boldsymbol{\pi}_{t,C}]$ collects the channel-wise retrieval distributions, $\mathcal{C}$ is the residual correction produced by memory retrieval and channel-wise scalar gating broadcast across the forecast horizon, $\mathbf{f}_{t,c}$ is the label-free online feedback for channel c, and $\mathcal{U}$ updates that channel's retrieval probabilities. Thus, only residual priorities change; the forecaster and $\mathcal{M}_0$ remain frozen.

2.2 Residual Memory Hypothesis

DMRC rests on a limited and testable hypothesis: although the residuals of a fixed forecaster are not always predictable, systematic errors may recur in similar model-aware states. Time series representations organize general dynamics (Yue et al., 2022; Wu et al., 2023), while external retrieval reuses local nonparametric experience (Lewis et al., 2020; Borgeaud et al., 2022; Han et al., 2025; Goswami et al., 2024; Gao et al., 2024). DMRC therefore organizes historical residuals into neighborhoods defined by model states, without assuming that input similarity implies error similarity.

We examine this hypothesis through two diagnostic experiments. The first measures the correction potential contained in historical residuals. The second evaluates whether model-aware residual information improves residual retrieval.

Figure 2a reports ideal residual correction gains of 32.18% to 78.91% across six datasets, using a frozen iTransformer with $L = 512$ and $H = 96$. This analysis uses future residuals and therefore indicates potential correction capacity rather than deployable performance. Figure 2b evaluates frozen DLinear and iTransformer forecasters with $H = 96$ across seven datasets and 14 forecaster–dataset tasks. Residual-aware retrieval reduces MSE by 0.602% on average, compared with 0.466% for input-based retrieval and -0.022% for random retrieval. These results suggest that model-aware residual information can provide a more useful retrieval signal than raw input similarity in the evaluated tasks.



(a) Oracle gain

(b) Retrieval utility

Figure 2: Correction potential and retrieval value of historical residuals. (a) Relative reduction in prediction MSE of Oracle residual correction compared to Frozen across six datasets. (b) Relative reduction in prediction MSE of different retrieval methods compared to Frozen under the same historical sample pool and correction rules. Both metrics are expressed as percentages.

3 Method

3.1 Overall Framework

DMRC corrects a frozen forecaster through an external residual memory. Offline, it constructs hierarchical prototypes based on error structure predictable from the observed state and decomposes each prototype residual into a principal-subspace component and an orthogonal remainder. Online, the subspace component is retrieved using both content matching and dynamic priorities, while the remainder is retrieved using content matching alone. The two components are combined through fixed gates to correct the base forecast.

3.2 Frozen Residual Memory Construction

DMRC organizes memory using predictable error coordinates while retaining complete ground-truth residuals as correction values. The residual-aware projector takes explicit features derived from the historical window, base forecast, and local statistics as input. It predicts residuals through ridge regression and extracts a shared temporal subspace from the second moment of the predicted residuals. For a single channel, the normalized query is

$$\mathbf{q} = \frac{1}{\nu}\mathbf{U}^\top \widehat{\mathbf{r}}^{\mathrm{reg}}, \tag{2}$$

where $\widehat{\mathbf{r}}^{\mathrm{reg}}$ is the predicted residual, $\mathbf{U}$ contains the orthonormal temporal basis vectors, and ν is the root-mean-square scale of the projected training residuals. This construction places offline memory organization and online retrieval in the same coordinate system.

Memory is constructed separately for each channel. The first two query coordinates define coarse groups, within which all low-dimensional coordinates are used to form fine-grained prototypes. Each prototype stores the mean ground-truth residual $\mathbf{v}_b$ of its assigned samples, decomposed into a subspace component and an orthogonal remainder:

$$\mathbf{v}_b^{\parallel} = \mathbf{U}\mathbf{U}^\top \mathbf{v}_b, \qquad \mathbf{v}_b^{\perp} = \mathbf{v}_b - \mathbf{v}_b^{\parallel}. \tag{3}$$

The subspace component supports dynamic retrieval, while the remainder retains residual information outside the selected subspace.

Prototype keys are means of predicted or ground-truth coefficients under the same normalization, with the key type selected during validation. For each channel, DMRC constructs a separate memory containing B residual prototypes, while sharing the learned temporal subspace across channels. Feature construction, basis estimation, and normalization details are provided in Appendix C.3.

3.3 Coarse-to-Fine Retrieval and Gated Correction

Given a normalized query $\mathbf{q}$, DMRC first selects the nearest coarse group using the distance between its first two coordinates and the group centers. Within the selected group, it computes content and dynamic scores:

$$e^0_{t,c,b} = -\frac{\|\mathbf{q}_{t,c} - \mathbf{k}_{c,b}\|_2^2}{\tau}, \qquad e_{t,c,b} = e^0_{t,c,b} + \rho \ell_{c,b}. \tag{4}$$

Here, $\mathbf{k}_{c,b}$ is the normalized key of prototype b for channel c, τ is a fixed temperature, $\ell_{c,b}$ is the current priority bias, and ρ controls its strength. Applying top-K selection and softmax separately to the two scores yields channel-wise content weights $\alpha_{t,c,b}$ and dynamic weights $\beta_{t,c,b}$. Unselected prototypes receive zero weight in the corresponding readout.

Dynamic weights are used only for the subspace component, whereas content weights are used for the orthogonal remainder. The final prediction is

$$\widetilde{\mathbf{y}}_{:,c} = \widehat{\mathbf{y}}_{0,:,c} + g_{\parallel,c} \sum_b \beta_{t,c,b}\mathbf{v}^{\parallel}_{c,b} + g_{\perp,c} \sum_b \alpha_{t,c,b}\mathbf{v}^{\perp}_{c,b}. \tag{5}$$

For each channel c, $g_{\parallel,c}$ and $g_{\perp,c}$ are scalar gates in $[0, 0.8]$, selected by minimizing the validation MSE of the corrected forecasts with all other components fixed. The selected gates are shared and broadcast across the H forecast steps and remain fixed during test-time evaluation.

3.4 Prediction Stability Feedback

Online state updates cannot use future targets that have not yet been observed. DMRC therefore constructs a label-free stability signal from forecasts already produced by the frozen pipeline. As shown in Figure 3, for probabilistic forecasters, $\mathcal{L}_{V,t}$ is the average width of the forecast quantile intervals; for point forecasters, V output views are constructed using semantics-preserving input transformations, and $\mathcal{L}_{V,t}$ is defined as the prediction discrepancy across these views. Each view passes through the complete frozen forecasting, retrieval, and gating pipeline to produce $\tilde{\mathbf{y}}_t^{(v)}$, which is then aligned to the same output coordinates using $\mathcal{T}_v^{-1}$. Both definitions characterize only prediction dispersion and are not interpreted as the true error.

In addition to multi-view feedback, DMRC compares overlapping intervals of forecasts made from different starting points. The stability objective is computed separately for each channel; the channel index is omitted in this subsection. The combined stability objective is given by

$$\mathcal{L}_{V,t} = (1-\delta_F)\frac{1}{VH}\sum_{v=1}^{V}\left\|\mathcal{T}_v^{-1}(\tilde{\mathbf{y}}_t^{(v)}) - \mathrm{sg}(\bar{\mathbf{y}}_t)\right\|_2^2 + \delta_F\frac{1}{H}\sum_{h=1}^{H}(Q_{t,h,0.9} - Q_{t,h,0.1}), \tag{6}$$

$$\mathcal{L}_{T,t} = \frac{1}{H-s}\left\|\bar{\mathbf{y}}_{t,1:H-s} - \mathrm{sg}\left(\bar{\mathbf{y}}_{t-s,s+1:H}\right)\right\|_2^2, \tag{7}$$

$$\mathcal{L}_{\mathrm{stab},t} = \lambda_v\mathcal{L}_{V,t} + \lambda_o\mathcal{L}_{T,t}. \tag{8}$$

Here, $\bar{\mathbf{y}}_t = V^{-1}\sum_v \mathcal{T}_v^{-1}(\tilde{\mathbf{y}}_t^{(v)})$, sg denotes the stop-gradient operation, and δ_F is a fixed indicator determined by the type of base forecaster: $\delta_F = 0$ for point forecasters and $\delta_F = 1$ for probabilistic forecasters. The former computes $\mathcal{L}_{V,t}$ from prediction discrepancies across multiple views, while the latter characterizes predictive uncertainty using the average width of the quantile intervals. The first term of $\mathcal{L}_{stab,t}$ measures cross-view consistency, the second *tests temporal overlap consistency*, and s is sampled randomly from the look-back steps available in the existing buffer that satisfy $1 \le s < H$. When no valid overlap exists, the second term is set to zero. The entire objective compares only forecasts already generated by the model and does not access ground-truth future values. Its role as a selective risk signal, rather than an estimator of true forecasting error, is examined through the post-hoc risk-stratification diagnostic in Appendix D.4.

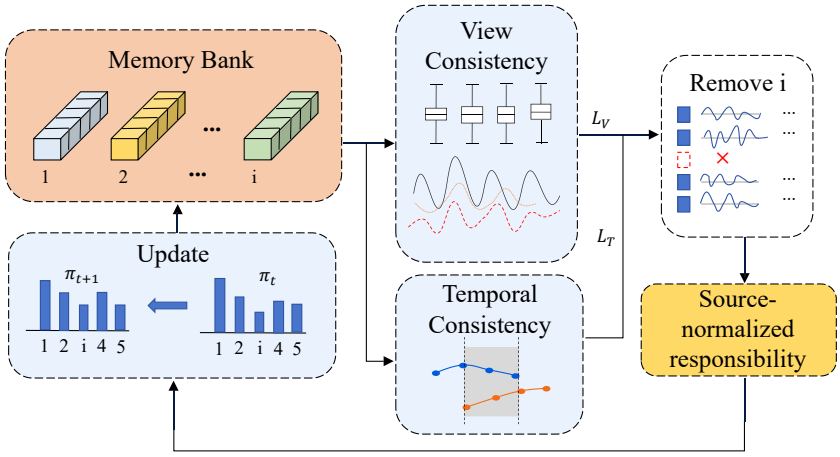


Figure 3: Evolution of the retrieval distributions in DMRC. Label-free view and temporal consistency provide stability feedback for source-normalized responsibility updates. The residual prototypes remain frozen, while only the channel-wise distributions evolve over time.

3.5 Source-Normalized Responsibilities and Dynamic Memory Weighting

For each channel c, content-based retrieval yields prototype responsibilities $\alpha_{t,c,i}^{\mathrm{content}} \approx P(z = i \mid \mathbf{q}_{t,c})$. DMRC uses the same frozen retriever to estimate the source-domain baseline $a_{\mathrm{src},c}(i)$ and aggregates across the current-window views for channel c to obtain $a_{t,c}(i)$. Responsibility estimation excludes $\boldsymbol{\pi}_{t,c}$ (i.e., $\rho = 0$), preventing the dynamic prior from reinforcing itself. The source-normalized ratio is

$$r_{t,c}(i) = \mathrm{clip}\left(\frac{a_{t,c}(i)+\varepsilon}{a_{\mathrm{src},c}(i)+\varepsilon}, R_{\max}^{-1}, R_{\max}\right), \qquad a_{\mathrm{src},c}(i) = \mathbb{E}_{\mathbf{q}_c\sim\mathcal{D}_{\mathrm{src},c}}[\alpha_{c,i}^{\mathrm{content}}(\mathbf{q}_c)]. \tag{9}$$

Normalizing and smoothing the clipped ratios gives $\tilde{r}_{t,c}(i) = B[(1-\delta)r_{t,c}(i)/\sum_j r_{t,c}(j) + \delta/B]$. Thus, $a_{t,c} = a_{\mathrm{src},c}$ when $\tilde{r}_{t,c}(i) = 1$; this removes source-domain access-frequency imbalance

without changing cluster sizes, allowing infrequent modes to receive greater attention through relative growth.

Stability determines whether these changes support the current memory. DMRC performs leave-one-out evaluation on the top-K items and adjusts their credit using source-normalized weights:

$$G_{t,c,i} = \mathcal{L}^{(-i)}_{\mathrm{stab},t,c} - \mathcal{L}_{\mathrm{stab},t,c}, \qquad u_{t,c,i} = \bar{\alpha}_{t,c,i}\ \mathrm{clip}(\tilde{r}_{t,c}(i) G_{t,c,i}, -G_{\max}, G_{\max}), \tag{10}$$

where $\bar{\alpha}_{t,c,i}$ is attention averaged across views for channel c. $G_{t,c,i} > 0$ indicates that removing the item increases prediction disagreement, while $\tilde{r}_{t,c}(i)$ directs credit toward modes that have changed relative to the source. The leave-one-out difference represents local stability credit rather than the gradient of the true forecasting error.

Let $\mathbf{u}_{t,c} = (u_{t,c,1}, \ldots, u_{t,c,B})$. Given this surrogate gain, DMRC updates memory selection while preventing a single window from overwriting retrieval behavior. It uses standard online mirror descent (Beck & Teboulle, 2003) with KL divergence on the probability simplex to constrain changes between consecutive retrieval distributions:

$$\boldsymbol{\pi}_{t+1,c} = \arg\min_{\boldsymbol{\pi}_c \in \Delta^B} \left\{ -\eta \langle \boldsymbol{\pi}_c, \mathbf{u}_{t,c} \rangle + D_{\mathrm{KL}}(\boldsymbol{\pi}_c \| \boldsymbol{\pi}_{t,c}) \right\},$$
$$\pi_{t+1,c}(i) = \frac{\pi_{t,c}(i) \exp(\eta u_{t,c,i})}{\sum_j \pi_{t,c}(j) \exp(\eta u_{t,c,j})} = \mathrm{softmax}(\log \boldsymbol{\pi}_{t,c} + \eta \mathbf{u}_{t,c})_i. \tag{11}$$

Positive gains increase subsequent retrieval probabilities for the corresponding channel, whereas the KL term limits changes caused by a single window. This update guarantees only nonnegative, normalized probabilities governed by bounded surrogate gains; it does not imply a monotonic decrease in true MSE. The closed-form solution and static-comparator bound are provided in Appendix B.

4 Experiments

4.1 Experimental Setup

Datasets and forecasting tasks. We evaluate DMRC on seven datasets: ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Weather, and Exchange (Wu et al., 2021; Liu et al., 2024b; Zhang & Yan, 2023; Han et al., 2024). We use chronological training, validation, and test splits, with $L = 512$ and $H \in \{96, 192, 336, 720\}$ in the default multivariate setting. All hyperparameters are selected on validation data.

Frozen forecasters and evaluation. The frozen forecasters are DLinear (Zeng et al., 2023), PatchTST, MICN, FreTS, iTransformer, and Chronos-Bolt-Base (Ansari et al., 2024). The first five produce point forecasts, whereas Chronos-Bolt-Base produces quantile forecasts. We follow a predict-then-update protocol and report MSE, MAE, and relative MSE changes over Frozen. KUP-BI (Chong et al., 2026) and TQNet (Lin et al., 2025) are evaluated in Appendix D.2.

Implementation details. We set $B = 192$ and $V = 3$. For each dataset, horizon, and forecaster, $K \in \{1, 4, 8, 16, 32\}$ and $\tau \in \{0.1, 0.2, 0.5\}$ are selected on validation data. The parameters $\eta = 1000$, $\lambda_o = 10^{-4}$, and $\lambda_v = 0.1$ are selected on a canonical validation suite and fixed thereafter. The leave-one-out gain is clipped to $[-1, 1]$, and all parameters remain unchanged during test-stream evaluation.

4.2 Main Results

Table 1 pairs Frozen with DMRC. Across the six frozen forecasters with completed results, DMRC obtains lower MSE in 146 of the 168 "forecaster–dataset–horizon" combinations; 12 combinations are tied, and 10 show MSE degradation. The largest reductions occur for DLinear–Exchange (H=192, 15.10%) and Chronos–Weather (H=192, 14.65%). All memory items come from the training and validation intervals, and the online memory update of DMRC never accesses the current forecasting target.

Table 1: Multivariate long-term forecasting results for different frozen forecasters and their DMRC-enhanced variants. MSE and MAE are reported for Frozen and DMRC with input length $L = 512$ and forecast horizons $H \in \{96, 192, 336, 720\}$. MSE Change (%) denotes the relative MSE change of DMRC over Frozen. The best results, determined using unrounded values, are highlighted in red.

Dataset	H	DLinear					PatchTST					iTransformer				
		Frozen		DMRC		MSE	Frozen		DMRC		MSE	Frozen		DMRC		MSE
		MSE	MAE	MSE	MAE	Change (%)	MSE	MAE	MSE	MAE	Change (%)	MSE	MAE	MSE	MAE	Change (%)
ETTh1	96	0.433	0.445	0.427	0.443	−1.26%	0.488	0.494	0.482	0.491	−1.28%	0.467	0.476	0.464	0.474	−0.82%
	192	0.479	0.480	0.471	0.476	−1.68%	0.586	0.550	0.582	0.547	−0.69%	0.516	0.510	0.508	0.506	−1.52%
	336	0.523	0.511	0.516	0.509	−1.28%	0.636	0.592	0.623	0.583	−2.12%	0.573	0.552	0.573	0.552	0.00%
	720	0.648	0.593	0.628	0.584	−3.13%	0.774	0.668	0.751	0.654	−2.92%	0.795	0.661	0.791	0.660	−0.57%
ETTh2	96	0.216	0.312	0.211	0.309	−2.09%	0.263	0.349	0.263	0.349	−0.01%	0.316	0.372	0.316	0.372	0.00%
	192	0.269	0.350	0.265	0.347	−1.65%	0.364	0.411	0.363	0.410	−0.17%	0.371	0.412	0.368	0.409	−0.81%
	336	0.315	0.381	0.303	0.372	−3.84%	0.429	0.451	0.428	0.452	−0.32%	0.499	0.476	0.499	0.476	0.00%
	720	0.397	0.431	0.371	0.414	−6.55%	0.500	0.496	0.469	0.480	−6.09%	0.632	0.553	0.632	0.553	0.00%
ETTm1	96	0.329	0.373	0.325	0.370	−1.29%	0.405	0.426	0.396	0.417	−2.36%	0.380	0.405	0.377	0.404	−0.71%
	192	0.381	0.405	0.375	0.402	−1.50%	0.476	0.462	0.472	0.456	−0.83%	0.446	0.440	0.445	0.440	−0.24%
	336	0.430	0.436	0.422	0.432	−1.90%	0.535	0.501	0.529	0.495	−1.17%	0.507	0.482	0.506	0.482	−0.10%
	720	0.484	0.474	0.474	0.469	−2.05%	0.613	0.553	0.600	0.544	−2.02%	0.553	0.521	0.546	0.516	−1.23%
ETTm2	96	0.149	0.254	0.149	0.254	0.00%	0.166	0.278	0.160	0.267	−3.57%	0.185	0.277	0.185	0.277	0.00%
	192	0.180	0.282	0.180	0.282	0.00%	0.214	0.304	0.211	0.305	−1.45%	0.229	0.310	0.229	0.310	0.00%
	336	0.217	0.306	0.217	0.306	0.00%	0.261	0.336	0.260	0.341	−0.45%	0.297	0.348	0.297	0.348	0.00%
	720	0.271	0.344	0.271	0.344	0.00%	0.338	0.396	0.333	0.395	−1.71%	0.300	0.369	0.300	0.369	0.00%
Electricity	96	0.136	0.231	0.135	0.231	−1.01%	0.134	0.236	0.134	0.231	−0.05%	0.135	0.233	0.133	0.231	−1.25%
	192	0.150	0.244	0.149	0.244	−1.12%	0.151	0.251	0.145	0.241	−4.35%	0.154	0.250	0.151	0.249	−1.96%
	336	0.166	0.260	0.162	0.259	−2.79%	0.168	0.267	0.157	0.256	−6.36%	0.168	0.266	0.163	0.262	−3.38%
	720	0.206	0.294	0.198	0.290	−3.94%	0.204	0.299	0.192	0.286	−6.09%	0.207	0.299	0.202	0.296	−2.59%
Weather	96	0.170	0.223	0.153	0.216	−10.18%	0.154	0.206	0.151	0.210	−2.33%	0.167	0.219	0.167	0.220	−0.12%
	192	0.214	0.259	0.200	0.262	−6.48%	0.215	0.262	0.208	0.267	−3.12%	0.228	0.273	0.229	0.275	+0.55%
	336	0.260	0.294	0.244	0.296	−6.18%	0.297	0.317	0.285	0.317	−3.90%	0.281	0.310	0.282	0.313	+0.49%
	720	0.325	0.340	0.298	0.338	−8.28%	0.374	0.366	0.359	0.366	−4.01%	0.383	0.373	0.382	0.374	−0.26%
Exchange	96	0.0859	0.204	0.0825	0.203	−3.94%	0.147	0.283	0.138	0.272	−6.26%	0.153	0.279	0.154	0.280	+0.16%
	192	0.181	0.300	0.154	0.278	−15.10%	0.316	0.412	0.308	0.404	−2.51%	0.228	0.346	0.236	0.352	+3.29%
	336	0.341	0.418	0.306	0.391	−10.18%	0.506	0.522	0.497	0.521	−1.74%	0.368	0.450	0.376	0.457	+2.16%
	720	0.941	0.720	0.860	0.687	−8.64%	1.42	0.903	1.39	0.896	−1.85%	1.06	0.793	1.07	0.791	+0.20%

Dataset	H	FreTS					Chronos-Bolt-Base					MICN				
		Frozen		DMRC		MSE	Frozen		DMRC		MSE	Frozen		DMRC		MSE
		MSE	MAE	MSE	MAE	Change (%)	MSE	MAE	MSE	MAE	Change (%)	MSE	MAE	MSE	MAE	Change (%)
ETTh1	96	0.455	0.472	0.452	0.470	−0.66%	0.448	0.429	0.433	0.427	−3.22%	0.584	0.532	0.581	0.531	−0.50%
	192	0.528	0.520	0.523	0.517	−1.05%	0.510	0.475	0.488	0.467	−4.39%	0.613	0.557	0.610	0.556	−0.45%
	336	0.581	0.557	0.579	0.556	−0.36%	0.560	0.516	0.544	0.514	−2.84%	0.661	0.593	0.656	0.592	−0.75%
	720	0.735	0.650	0.720	0.642	−2.01%	0.713	0.615	0.680	0.603	−4.64%	0.800	0.677	0.784	0.667	−2.08%
ETTh2	96	0.258	0.346	0.253	0.342	−1.92%	0.221	0.308	0.222	0.310	+0.38%	0.267	0.360	0.261	0.358	−2.54%
	192	0.306	0.379	0.306	0.379	−0.00%	0.273	0.348	0.272	0.347	−0.36%	0.313	0.388	0.310	0.385	−1.03%
	336	0.362	0.412	0.346	0.401	−4.27%	0.315	0.380	0.311	0.377	−1.13%	0.331	0.399	0.328	0.398	−1.06%
	720	0.436	0.454	0.410	0.437	−6.13%	0.412	0.441	0.406	0.436	−1.39%	0.451	0.470	0.442	0.467	−2.18%
ETTm1	96	0.329	0.379	0.323	0.374	−1.97%	0.371	0.363	0.336	0.357	−9.48%	0.389	0.421	0.387	0.420	−0.58%
	192	0.399	0.423	0.395	0.420	−0.84%	0.445	0.407	0.401	0.398	−9.88%	0.467	0.465	0.463	0.462	−0.86%
	336	0.445	0.452	0.439	0.449	−1.35%	0.511	0.446	0.461	0.434	−9.83%	0.537	0.509	0.530	0.505	−1.36%
	720	0.517	0.505	0.508	0.499	−1.89%	0.604	0.507	0.551	0.487	−8.70%	0.576	0.537	0.572	0.536	−0.63%
ETTm2	96	0.166	0.266	0.165	0.266	−0.07%	0.158	0.252	0.154	0.249	−2.86%	0.155	0.263	0.155	0.260	−0.10%
	192	0.200	0.297	0.199	0.294	−0.90%	0.197	0.285	0.195	0.285	−0.91%	0.208	0.300	0.207	0.303	−0.62%
	336	0.234	0.325	0.226	0.318	−3.52%	0.236	0.315	0.232	0.314	−1.63%	0.271	0.351	0.262	0.344	−3.27%
	720	0.282	0.360	0.279	0.358	−1.20%	0.310	0.363	0.300	0.358	−3.24%	0.328	0.382	0.326	0.385	−0.55%
Electricity	96	0.130	0.225	0.131	0.228	+1.35%	0.124	0.211	0.123	0.212	−1.23%	0.148	0.253	0.145	0.250	−1.80%
	192	0.148	0.242	0.145	0.239	−1.77%	0.143	0.229	0.140	0.230	−1.90%	0.166	0.269	0.163	0.267	−1.78%
	336	0.165	0.260	0.159	0.257	−3.56%	0.162	0.250	0.157	0.250	−2.99%	0.177	0.280	0.175	0.279	−0.79%
	720	0.202	0.293	0.192	0.286	−4.80%	0.210	0.293	0.194	0.287	−7.57%	0.215	0.310	0.214	0.309	−0.64%
Weather	96	0.154	0.206	0.149	0.209	−3.49%	0.177	0.210	0.151	0.201	−14.63%	0.170	0.221	0.152	0.217	−10.35%
	192	0.200	0.249	0.193	0.256	−3.49%	0.222	0.253	0.190	0.247	−14.65%	0.213	0.258	0.201	0.261	−5.90%
	336	0.251	0.288	0.242	0.293	−3.59%	0.281	0.295	0.241	0.289	−13.94%	0.259	0.292	0.243	0.293	−6.30%
	720	0.326	0.340	0.314	0.344	−3.48%	0.362	0.350	0.311	0.342	−14.09%	0.325	0.339	0.302	0.338	−7.05%
Exchange	96	0.103	0.231	0.103	0.226	−0.66%	0.0992	0.219	0.0998	0.220	+0.62%	0.0884	0.207	0.0850	0.207	−3.83%
	192	0.215	0.333	0.201	0.328	−6.37%	0.194	0.314	0.194	0.315	+0.21%	0.192	0.309	0.169	0.296	−12.01%
	336	0.346	0.432	0.325	0.418	−6.07%	0.339	0.423	0.338	0.421	−0.38%	0.360	0.428	0.326	0.402	−9.52%
	720	1.10	0.761	1.08	0.760	−1.25%	0.822	0.679	0.788	0.664	−4.18%	1.29	0.840	1.25	0.832	−2.85%

4.3 Ablation Study

This section analyzes DMRC from three perspectives: residual-aware representation, retrieval strategy, and memory updates. All comparisons use the same frozen forecaster, data split, memory capacity, gating mechanism, and evaluation protocol, changing only the component under study.

Coarse-to-fine residual retrieval. Global retrieval selects neighbors directly from the full memory bank, whereas coarse-to-fine retrieval first identifies a residual mode and then performs within-group top-K retrieval. Table 2 reports the comparison. Compared with global retrieval, coarse-to-fine retrieval reduces the average residual MSE and MAE by 2.59% and 1.35%, respectively, indicating that coarse-grained screening reduces cross-mode residual matching.

Table 2: MSE and MAE of global and coarse-to-fine residual retrieval. Results use DLinear as the frozen forecaster and are averaged over four forecast horizons.

Dataset	Global		Coarse-to-fine	
	MSE	MAE	MSE	MAE
ETTh1	0.5159	0.5050	0.5107	0.5031
ETTh2	0.2985	0.3681	0.2876	0.3605
ETTm1	0.3990	0.4183	0.3989	0.4182
ETTm2	0.2040	0.2954	0.2040	0.2954
Electricity	0.1611	0.2559	0.1606	0.2559
Weather	0.2264	0.2806	0.2236	0.2781
Exchange	0.3870	0.4119	0.3505	0.3899
Average	0.3132	0.3622	0.3051	0.3573

Memory evolution. We further compare four memory strategies. Static memory does not update weights; Absolute update updates them directly according to current access frequencies; Source-normalized update uses the source-domain access baseline to correct inherent frequency differences but does not use stability credit; Full DMRC additionally incorporates leave-one-out stability credit and updates retrieval weights under a KL constraint. Table 3 reports the comparison. Notably, on the ETTm2 dataset, all memory strategies yield identical average MSE and MAE, indicating that the gated correction falls back to the original forecast on this dataset.

Table 3: Forecasting results of different memory-weight update strategies. Results use DLinear as the frozen forecaster and are averaged over four forecast horizons.

Dataset	Static memory		Absolute update		Source-normalized update		DMRC	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.515810	0.504961	0.515932	0.505012	0.515907	0.505002	0.510734	0.503147
ETTh2	0.299134	0.368029	0.299333	0.368430	0.299215	0.367548	0.287570	0.360470
ETTm1	0.398918	0.418294	0.398938	0.418305	0.398930	0.418299	0.398898	0.418229
ETTm2	0.204020	0.295362	0.204020	0.295362	0.204020	0.295362	0.204020	0.295362
Electricity	0.161076	0.255941	0.161057	0.255936	0.161057	0.255928	0.160554	0.255912
Weather	0.226312	0.280853	0.226393	0.280841	0.226331	0.280880	0.223585	0.278052
Exchange	0.388105	0.412747	0.388436	0.413078	0.388188	0.412840	0.350515	0.389900
Average	0.313339	0.362312	0.313444	0.362423	0.313378	0.362266	0.305125	0.357296

Relative to Static memory, DMRC reduces the average MSE and MAE by 2.62% and 1.38%, respectively. The difference between Absolute update and Source-normalized update measures the effect of source-domain baseline correction, while the difference between the latter and Full DMRC measures the contribution of stability credit.

Component contributions. Finally, we remove residual-aware representation, coarse-to-fine retrieval, and memory evolution separately to assess the contribution of each core component to the full system. Coarse clustering and within-group retrieval are removed together as a unified retrieval module. Table 4 reports the results.

Compared with the corresponding ablated variants, full DMRC reduces the average MSE by 3.84%, 2.59%, and 2.62%, and the average MAE by 2.03%, 1.35%, and 1.38%, respectively. These results validate the roles of the residual matching space, hierarchical residual addressing, and online memory evolution.

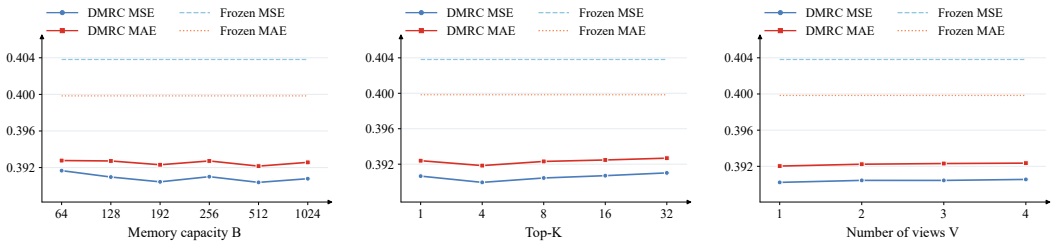
Table 4: Ablation results for the core components of DMRC. Results use DLinear as the frozen forecaster and are averaged over four forecast horizons.

Dataset	w/o Residual-aware		w/o Coarse-to-fine Retrieval		w/o Memory Evolution		DMRC	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.5169	0.5055	0.5159	0.5050	0.5158	0.5050	0.5107	0.5031
ETTh2	0.2993	0.3684	0.2985	0.3681	0.2991	0.3680	0.2876	0.3605
ETTm1	0.4019	0.4209	0.3990	0.4183	0.3989	0.4183	0.3989	0.4182
ETTm2	0.2040	0.2954	0.2040	0.2954	0.2040	0.2954	0.2040	0.2954
Electricity	0.1618	0.2560	0.1611	0.2559	0.1611	0.2559	0.1606	0.2559
Weather	0.2335	0.2829	0.2264	0.2806	0.2263	0.2809	0.2236	0.2781
Exchange	0.4035	0.4240	0.3870	0.4119	0.3881	0.4127	0.3505	0.3899
Average	0.3173	0.3647	0.3132	0.3622	0.3133	0.3623	0.3051	0.3573

Note: In the Electricity row, the results of "w/o coarse-to-fine retrieval" and "w/o memory evolution" appear identical after rounding to four decimal places (MSE: 0.1611 vs. 0.1611; MAE: 0.2559 vs. 0.2559). Their unrounded values are $(0.1610626,\ 0.255924)$ and $(0.161076,\ 0.255941)$, respectively.

4.4 Sensitivity Analysis

This section investigates the effects of memory capacity, retrieval granularity, the number of views, and the weights of the stability losses on prediction error. By performing grid searches over the target parameters while keeping all other configurations fixed, and selecting the optimal configurations based on validation performance, we find that DMRC exhibits strong overall robustness to hyperparameter choices within the considered ranges. Figure 4 reports the sensitivity to memory capacity B, the number of retrieved prototypes K, and the number of views V. For the two stability loss terms, we set their weights according to their respective numerical scales to balance their contributions to model updates, and ultimately use $\lambda_v = 0.1$ and $\lambda_o = 10^{-4}$. Detailed experimental results are provided in Appendix D.5.



(a) Memory capacity B (b) top-K (c) Number of views V

Figure 4: Parameter sensitivity: (a) Memory capacity B; (b) top-K; (c) Number of views V.

5 Conclusion and Future Work

We propose DMRC, which enhances frozen forecasters through residual-aware memory, coarse-to-fine retrieval, and constrained online weight updates. DMRC keeps both forecaster parameters and memory contents fixed, adapting only retrieval priorities and using a gate to fall back to the original forecast when memory support is insufficient. Experiments across multiple datasets, horizons, and frozen forecasters show that DMRC generally improves MSE and MAE, demonstrating the value of historical residual structures for deployment adaptation.

DMRC is limited by offline memory coverage and the accuracy of its stability proxy. Future work will explore online residual-mode augmentation and more reliable stability signals. Since the stability assessment requires no future labels, the framework can potentially generalize to other sequence forecasting domains such as energy, transportation, finance, healthcare, and industrial monitoring.

AI USE STATEMENT

This paper used generative AI tools to assist with drafting, sentence polishing, grammar checking, and improving clarity of expression. AI tools were not used to formulate the research question, design DMRC, generate experimental data, run experiments, modify experimental results, or determine the paper's conclusions. The formulas, algorithms, data processing, experimental settings, numerical results, and references in this paper were provided and verified by the authors, who take responsibility for them; all AI-assisted text was reviewed and revised paragraph by paragraph by the authors.

REFERENCES

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

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(Oct):1705–1749, 2005.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Liu Chong, Yingjie Zhou, Hao Li, Pengyang Wang, Qingsong Wen, and Ce Zhu. Beyond extrapolation: Knowledge utilization paradigm with bidirectional inspiration for time series forecasting. *arXiv preprint arXiv:2605.19249*, 2026.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175, 2024.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2505.04163*, 2025.

Hui He, Kun Yi, Yuanchi Ma, Qi Zhang, Zhendong Niu, and Guansong Pang. Sempo: Lightweight foundation models for time series forecasting. *Advances in Neural Information Processing Systems*, 38:161846–161879, 2026.

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. URL https://openreview.net/forum?id=L7Z5wBMPrW.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. Retrieval based time series forecasting. *arXiv preprint arXiv:2209.13525*, 2022.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020.

Dongmin Kim, Sunghyun Park, and Jaegul Choo. When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 13113–13121, 2024.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. *arXiv preprint arXiv:2505.12917*, 2025.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. *arXiv preprint arXiv:2410.04442*, 2024a.

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*, volume 2024, pp. 11116–11140, 2024b.

Heitor R Medeiros, Hossein Sharifi-Noghabi, Gabriel L Oliveira, and Saghar Irandoust. Accurate parameter-efficient test-time adaptation for time series forecasting. *arXiv preprint arXiv:2506.23424*, 2025.

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.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining v. 1*, pp. 1185–1196, 2025.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2025.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in neural information processing systems*, 36:76656–76679, 2023.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

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

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

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 V. 1*, pp. 2020–2031, 2025.

A Module Details

This section provides additional implementation details of the dynamic residual memory described in the main text. During deployment, DMRC keeps the base forecaster, residual-aware projector, prototype keys and values, and dual-branch gating coefficients fixed, updating only C independent channel-wise retrieval distributions $\boldsymbol{\pi}_{t,c} \in \Delta^B$. The online priorities affect only the readout of the principal-direction component, while the orthogonal residual component always uses content-based retrieval.

A.1 Online State and Causal Execution Order

Test windows arrive in their original temporal order. Retrieval and correction are performed separately for each channel c. Given a window $\mathbf{x}_t$, the system first computes the frozen forecast $\hat{\mathbf{y}}_{0,t} = \mathcal{F}(\mathbf{x}_t)$. Ridge regression then predicts the residual from explicit state features, which is projected onto a shared temporal subspace and normalized by scale to obtain the query $\mathbf{q}_t$.

Coarse retrieval selects the nearest coarse group using the first two dimensions of the query. Within this group, fine-grained retrieval performs top-K selection using the content score and the dynamic score augmented with the current priorities, yielding the content weights $\alpha_{t,c,b}$ and dynamic weights $\beta_{t,c,b}$. These weights retrieve the orthogonal residual and principal-direction components, respectively, producing the current forecast for channel c:

$$\tilde{\mathbf{y}}_{t,:,c} = \hat{\mathbf{y}}_{0,t,:,c} + g_{\parallel,c} \sum_b \beta_{t,c,b} \mathbf{v}^{\parallel}_{c,b} + g_{\perp,c} \sum_b \alpha_{t,c,b} \mathbf{v}^{\perp}_{c,b}. \tag{12}$$

Here, $g_{\parallel,c}$ and $g_{\perp,c}$ are channel-wise scalar gates in $[0, 0.8]$. They are shared and broadcast across the H forecast steps and remain fixed during deployment.

Only after the current forecast has been generated does the system construct stability feedback, perform leave-one-out evaluation for the dynamic principal-direction branch, and update $\boldsymbol{\pi}_{t+1,c}$. Feedback is computed using the same pre-update state, while the orthogonal-component readout and gating coefficients remain unchanged. Consequently, the update affects only principal-direction retrieval for subsequent windows. The current target $\mathbf{y}_t$ is used neither for forecasting nor for state updates, but only for ex-post computation of MSE, MAE, and diagnostic quantities.

A.2 PREDICTION STABILITY FEEDBACK

Stability feedback is computed separately for each channel; the channel index is omitted in this subsection.

Because the future target of the current window is unavailable online, DMRC constructs a stability signal from outputs already generated by the forecaster. This signal measures whether equivalent views or forecasts from different origins agree on the same future interval. It serves as a label-free surrogate for retrieval credit and is not interpreted as the true error or a guarantee of prediction-interval coverage.

Point forecasters. For DLinear, PatchTST, MICN, FreTS, and iTransformer, we first construct V equivalent views from the same historical window by applying controlled input perturbations $\{\mathcal{T}_v(\mathbf{x}_t)\}_{v=1}^{V}$. These views involve only magnitude-controlled observation perturbations, masking, or invertible scale changes, without changing the temporal direction of the forecasting task. Each view passes through the same frozen forecaster, retriever, and gates in sequence; if a transformation changes the numerical coordinates, $\mathcal{T}_v^{-1}$ restores the forecast to a common coordinate system before comparison. Denote the aligned outputs by

$$\mathbf{y}_t^{(v)} = \mathcal{T}_v^{-1}\left(\tilde{\mathbf{y}}_t^{(v)}\right), \qquad \bar{\mathbf{y}}_t = \frac{1}{V}\sum_{v=1}^{V}\mathbf{y}_t^{(v)}. \tag{13}$$

For point forecasters the cross-view loss is defined as

$$\mathcal{L}_{V,t} = \frac{1}{VH}\sum_{v=1}^{V}\left\|\mathbf{y}_t^{(v)} - \mathrm{sg}(\bar{\mathbf{y}}_t)\right\|_2^2, \tag{14}$$

where sg denotes stop-gradient. A smaller $\mathcal{L}_{V,t}$ indicates that the current retrieval-based correction remains stable under mild input perturbations, but does not imply that it is necessarily closer to the true target.

Probabilistic forecasters. Chronos-Bolt-Base directly outputs the quantiles $Q_{0.1}$, $Q_{0.5}$, and $Q_{0.9}$. We use the median $Q_{0.5}$ as the point forecast for computing MSE and MAE. For stability feedback, we use the average width of the prediction interval rather than perturbation-based views:

$$\mathcal{L}_{V,t} = \frac{1}{H}\sum_{h=1}^{H}\left(Q_{t,h,0.9} - Q_{t,h,0.1}\right). \tag{15}$$

The interval-based stability signal is used with the same residual retrieval, fixed channel-wise gating, and leave-one-out memory-update interface as point forecasters. It provides a relative uncertainty cue only and does not imply a frequentist coverage guarantee.

Temporal overlap consistency. In sequential forecasting, forecasts originating at time $t-s$ and time t overlap at $H-s$ future positions. For $1 \le s < H$, the overlap loss is

$$\mathcal{L}_{T,t} = \frac{1}{H-s}\left\|\bar{\mathbf{y}}_{t,1:H-s} - \mathrm{sg}(\bar{\mathbf{y}}_{t-s,s+1:H})\right\|_2^2. \tag{16}$$

When the cache contains no valid overlapping forecast, this term is set to zero without changing the other update steps. The final stability objective is

$$\mathcal{L}_{\mathrm{stab},t} = \lambda_v\mathcal{L}_{V,t} + \lambda_o\mathcal{L}_{T,t}. \tag{17}$$

The cross-view term reflects local sensitivity around the current window, while the overlap term reflects temporal consistency between consecutive forecasts over their shared future interval. Because the two terms capture different sources of information, their weights are selected separately on the validation stream and kept fixed during testing.

A.3 SOURCE-NORMALIZED RESPONSIBILITIES

Updating memory directly from current prototype access frequencies conflates two factors: some residual modes are inherently more common in the source domain, whereas others become relatively more frequent in the target stream. Relying only on absolute access counts would continually assign larger weights to frequent source-domain prototypes, making rare but rapidly growing modes difficult to identify. We therefore compare current responsibilities against the long-term source-domain baseline rather than comparing their absolute magnitudes directly.

To prevent self-reinforcement of the dynamic prior, responsibilities are computed using content-based retrieval without the $\log \pi_{t,c}(i)$ term. Let $a_{t,c}(i)$ denote the average content responsibility of prototype i across views for channel c in the current window. The source-domain baseline is defined as

$$a_{\mathrm{src},c}(i) = \mathbb{E}_{\mathbf{q}_c \sim \mathcal{D}_{\mathrm{src},c}} \left[\alpha_{c,i}^{\mathrm{content}}(\mathbf{q}_c) \right]. \tag{18}$$

where $\mathcal{D}_{\mathrm{src},c}$ contains only queries for channel c from the training and validation intervals. The baseline is computed and frozen before deployment; the test stream does not retrospectively modify this distribution.

The source-normalized ratio is

$$r_{t,c}(i) = \mathrm{clip}\left(\frac{a_{t,c}(i) + \varepsilon}{a_{\mathrm{src},c}(i) + \varepsilon}, R_{\max}^{-1}, R_{\max} \right). \tag{19}$$

Ratio clipping limits the small-denominator effect for prototypes that are rare in the source domain. Normalization and uniform smoothing are then applied:

$$\widetilde{r}_{t,c}(i) = B \left[(1-\delta) \frac{r_{t,c}(i)}{\sum_{j=1}^{B} r_{t,c}(j)} + \frac{\delta}{B} \right]. \tag{20}$$

When current responsibilities agree with the source-domain baseline, all $\widetilde{r}_{t,c}(i)$ are close to 1, so source normalization itself does not favor any prototype in channel c. When the relative access frequency of a mode increases, its responsibility exceeds 1, but whether its memory priority actually increases is still determined by the subsequent stability credit.

Source-normalized responsibilities do not change the coarse grouping, memory keys, memory values, or content similarity, nor do they directly control the final correction magnitude. They affect only credit allocation for memory updates; channel-wise scalar gates, broadcast across the forecast horizon, independently determine whether the retrieved residual enters the final output. This distinction avoids equating more frequent access in the current stream with a need for stronger correction in the current forecast.

A.4 TOP-K LEAVE-ONE-OUT EVALUATION

The stability loss evaluates the current retrieval result as a whole, but does not identify which prototype in the top-K set contributes to or undermines that result. DMRC performs leave-one-out evaluation on the current candidate set for each channel. For prototype $i \in \mathcal{N}_{K,c}(t)$, we temporarily remove the item, renormalize the remaining attention weights, and recompute the stability loss for channel c using the same frozen gates: $\mathcal{L}_{\mathrm{stab},t,c}^{(-i)}$. The leave-one-out credit is defined as

$$G_{t,c,i} = \mathcal{L}_{\mathrm{stab},t,c}^{(-i)} - \mathcal{L}_{\mathrm{stab},t,c}. \tag{21}$$

If $G_{t,c,i} > 0$, removing prototype i increases forecast disagreement, indicating a positive contribution to current stability. If $G_{t,c,i} < 0$, forecasts become more consistent after removal, and the item receives negative credit. Prototypes outside the top-K set receive zero credit for the current window, so each step evaluates only the limited set of candidates that actually participate in the decision.

Leave-one-out evaluation neither reruns the base forecaster nor changes the gating parameters. In implementation, candidate indices, attention weights, residual values, and gating inputs can be cached for each view, and K counterfactual sets can be constructed through batched masking. The remaining attention weights must be renormalized after an item is removed; otherwise, the comparison also changes the overall residual scale, preventing the difference from being attributed to candidate composition.

$G_{t,c,i}$ is a discrete finite difference based on the stability loss for channel c, not a gradient of the true forecasting MSE. It may be affected by top-K boundaries, gating nonlinearities, and surrogate-objective mismatch. Accordingly, the theoretical results in the main text and appendix treat it only as a bounded surrogate gain.

A.5 Credit Combination and Weight Updates

Let $\bar{\alpha}_{t,c,i}$ denote the average retrieval attention of prototype i across views for channel c. DMRC combines content participation, source-normalized responsibility, and leave-one-out stability credit as

$$u_{t,c,i} = \bar{\alpha}_{t,c,i}\ \mathrm{clip}(\widetilde{r}_{t,c}(i)G_{t,c,i}, -G_{\max}, G_{\max})\,. \tag{22}$$

The attention term concentrates credit on prototypes that actually participate in the current forecast, the source-normalization term emphasizes residual modes that have changed relative to the source domain, and gain clipping limits the maximum influence of a single window. These three components respectively capture participation, relative change, and contribution to stability.

Given $\mathbf{u}_{t,c} = (u_{t,c,1}, \ldots, u_{t,c,B})$, the retrieval distribution for each channel at the next time step is updated independently through KL-regularized mirror descent (Shalev-Shwartz, 2025):

$$\boldsymbol{\pi}_{t+1,c} = \arg\min_{\boldsymbol{\pi}_c \in \Delta^B} \{-\eta\langle \boldsymbol{\pi}_c, \mathbf{u}_{t,c}\rangle + D_{\mathrm{KL}}(\boldsymbol{\pi}_c \| \boldsymbol{\pi}_{t,c})\}\,. \tag{23}$$

The closed-form solution is

$$\pi_{t+1,c}(i) = \frac{\pi_{t,c}(i)\exp(\eta u_{t,c,i})}{\sum_{j=1}^{B}\pi_{t,c}(j)\exp(\eta u_{t,c,j})}. \tag{24}$$

In practice, normalization uses the log-sum-exp form of $\log \boldsymbol{\pi}_{t,c} + \eta\mathbf{u}_{t,c}$, with a fixed probability floor applied before taking logarithms. The floor only prevents $\log 0$ under finite precision; after normalization, all components remain on the probability simplex. If a window has no valid candidates or all stability credits are zero, then $\boldsymbol{\pi}_{t+1,c} = \boldsymbol{\pi}_{t,c}$.

The complete online sequence is to finish the main forecast, generate stability views, compute source-normalized responsibilities and leave-one-out credits, and finally update each channel's memory distribution for the next time step. Gating determines the current correction magnitude, while $\boldsymbol{\pi}_{t,c}$ determines future retrieval priorities for channel c; they share no parameters that are trainable online. This design prevents a window from changing its own retrieval result through the same update and keeps test targets out of the online state in any form.

B Theoretical Foundations

This section derives the closed-form solution of Eq. 11 in the main text and establishes its basic properties. The derivation applies independently to each channel, with the channel index omitted. Mirror descent uses a Bregman divergence to control changes between successive decisions; when the regularizer is the KL divergence induced by negative entropy, the update is equivalent to exponential weighting (Banerjee et al., 2005). These results characterize memory-distribution updates under bounded surrogate gains and do not equate the stability surrogate with the true forecasting error.

B.1 Closed-Form Solution of Eq. 11

Consider the single-step optimization problem

$$\min_{\boldsymbol{\pi}\in\Delta^B}\left\{-\eta\sum_{i=1}^{B}\pi(i)u_{t,i} + \sum_{i=1}^{B}\pi(i)\log\frac{\pi(i)}{\pi_t(i)}\right\}, \tag{25}$$

where $\Delta^B = \{\boldsymbol{\pi} \geq 0 : \sum_i \pi(i) = 1\}$. When $\pi_t(i) > 0$ for all i, the optimal solution lies in the interior of the simplex. Introducing a Lagrange multiplier λ for the equality constraint gives

$$\mathcal{J}(\boldsymbol{\pi}, \lambda) = -\eta\sum_{i=1}^{B}\pi(i)u_{t,i} + \sum_{i=1}^{B}\pi(i)\log\frac{\pi(i)}{\pi_t(i)} + \lambda\left(\sum_{i=1}^{B}\pi(i) - 1\right). \tag{26}$$

Taking the partial derivative with respect to each $\pi(i)$ and setting it to zero yields

$$-\eta u_{t,i} + \log \frac{\pi(i)}{\pi_t(i)} + 1 + \lambda = 0. \tag{27}$$

Rearranging gives

$$\pi(i) = \pi_t(i) \exp(\eta u_{t,i} - 1 - \lambda). \tag{28}$$

Substituting this expression into $\sum_i \pi(i) = 1$, the common factor satisfies

$$\exp(-1 - \lambda) = \left[\sum_{j=1}^{B} \pi_t(j) \exp(\eta u_{t,j}) \right]^{-1}. \tag{29}$$

Therefore,

$$\pi_{t+1}(i) = \frac{\pi_t(i) \exp(\eta u_{t,i})}{\sum_{j=1}^{B} \pi_t(j) \exp(\eta u_{t,j})}, \tag{30}$$

which is Eq. 11 in the main text. The KL term in the objective is strictly convex in the interior of the simplex, and the linear gain term does not alter convexity. The stationary point above is therefore the unique global optimum.

B.2 Simplex Preservation and Relative Weight Changes

In Eq. 30, both the numerator and denominator are positive, so $\pi_{t+1}(i) > 0$. Summing over all i immediately gives $\sum_i \pi_{t+1}(i) = 1$. As long as the initial distribution has strictly positive components, all subsequent states remain in the interior of the probability simplex under ideal numerical precision.

The relative weights of any two prototypes satisfy

$$\frac{\pi_{t+1}(i)}{\pi_{t+1}(j)} = \frac{\pi_t(i)}{\pi_t(j)} \exp(\eta[u_{t,i} - u_{t,j}]). \tag{31}$$

Thus, the update depends only on gain differences between prototypes. If all prototypes receive the same gain, the normalization constant cancels this common shift and the memory distribution remains unchanged. If $u_{t,i} > u_{t,j}$, the relative advantage of prototype i over j increases multiplicatively (Littlestone & Warmuth, 1994).

For a fixed learning rate, expanding the recurrence gives

$$\pi_{T+1}(i) = \frac{\pi_1(i) \exp\left(\eta \sum_{t=1}^{T} u_{t,i}\right)}{\sum_{j=1}^{B} \pi_1(j) \exp\left(\eta \sum_{t=1}^{T} u_{t,j}\right)}. \tag{32}$$

This expression shows that the online state accumulates historical surrogate gains rather than making a hard selection based only on the most recent window. KL regularization retains the previous distribution as a multiplicative reference, avoiding a fresh selection of memory items from the uniform distribution at every step.

If $|u_{t,i}| \le G_{\max}$, the normalization constant $Z_t = \sum_i \pi_t(i) \exp(\eta u_{t,i})$ satisfies $e^{-\eta G_{\max}} \le Z_t \le e^{\eta G_{\max}}$, and hence

$$\left| \log \frac{\pi_{t+1}(i)}{\pi_t(i)} \right| = |\eta u_{t,i} - \log Z_t| \le 2\eta G_{\max}. \tag{33}$$

Gain clipping and the learning rate jointly bound the single-step change in log weights. This bound does not guarantee a small probability change for every prototype, but rules out unbounded single-step exponential growth under bounded gains.

B.3 Static Comparator Bound

Let

$$Z_t = \sum_{i=1}^{B} \pi_t(i) \exp(\eta u_{t,i}). \tag{34}$$

For any fixed comparator distribution $\boldsymbol{\pi}^\star \in \Delta^B$, the closed-form update gives

$$\begin{aligned} &D_{\mathrm{KL}}(\boldsymbol{\pi}^\star \| \boldsymbol{\pi}_{t+1}) - D_{\mathrm{KL}}(\boldsymbol{\pi}^\star \| \boldsymbol{\pi}_t) \\ &= \log Z_t - \eta \langle \boldsymbol{\pi}^\star, \mathbf{u}_t \rangle. \end{aligned} \tag{35}$$

Since $u_{t,i} \in [-G_{\max}, G_{\max}]$, Hoeffding's lemma gives

$$\log Z_t \le \eta \langle \boldsymbol{\pi}_t, \mathbf{u}_t \rangle + \frac{\eta^2 G_{\max}^2}{2}. \tag{36}$$

Substituting into Eq. 35 and rearranging yields

$$\begin{aligned} \langle \boldsymbol{\pi}^\star - \boldsymbol{\pi}_t, \mathbf{u}_t \rangle \le \frac{D_{\mathrm{KL}}(\boldsymbol{\pi}^\star \| \boldsymbol{\pi}_t) - D_{\mathrm{KL}}(\boldsymbol{\pi}^\star \| \boldsymbol{\pi}_{t+1})}{\eta} \\ + \frac{\eta G_{\max}^2}{2}. \end{aligned} \tag{37}$$

Summing over $t = 1, \dots, T$ makes the KL terms telescope. If the initial distribution is uniform, then $D_{\mathrm{KL}}(\boldsymbol{\pi}^\star \| \boldsymbol{\pi}_1) = \log B - \mathcal{H}(\boldsymbol{\pi}^\star) \le \log B$, yielding

$$\sum_{t=1}^{T} \langle \boldsymbol{\pi}^\star - \boldsymbol{\pi}_t, \mathbf{u}_t \rangle \le \frac{\log B}{\eta} + \frac{\eta T G_{\max}^2}{2}. \tag{38}$$

Setting $\eta = \sqrt{2 \log B / (T G_{\max}^2)}$ reduces the right-hand side to $G_{\max} \sqrt{2T \log B}$. Thus, relative to the best fixed memory distribution in hindsight, the average surrogate-gain gap decays at a rate of $\mathcal{O}(\sqrt{\log B / T})$. This is the standard static comparator result for exponential weighting and online mirror descent.

B.4 Gated Fallback and Theoretical Scope

Define the retrieved components before gating as

$$\mathbf{c}_t^{\parallel} = \sum_b \beta_{t,b} \mathbf{v}_b^{\parallel}, \qquad \mathbf{c}_t^{\perp} = \sum_b \alpha_{t,b} \mathbf{v}_b^{\perp}, \qquad \mathbf{r}_t^{\mathrm{ret}} = \mathbf{c}_t^{\parallel} + \mathbf{c}_t^{\perp}. \tag{39}$$

Because $\mathbf{c}_t^{\parallel}$ lies in the selected temporal subspace and $\mathbf{c}_t^{\perp}$ lies in its orthogonal complement, we have $\langle \mathbf{c}_t^{\parallel}, \mathbf{c}_t^{\perp} \rangle = 0$. Using Eq. 12 and $0 \le g_{\parallel}, g_{\perp} \le 0.8$, the correction magnitude satisfies

$$\begin{aligned} \|\tilde{\mathbf{y}}_t - \hat{\mathbf{y}}_{0,t}\|_2^2 &= g_{\parallel}^2 \left\| \mathbf{c}_t^{\parallel} \right\|_2^2 + g_{\perp}^2 \left\| \mathbf{c}_t^{\perp} \right\|_2^2 \\ &\le 0.64 \left(\left\| \mathbf{c}_t^{\parallel} \right\|_2^2 + \left\| \mathbf{c}_t^{\perp} \right\|_2^2 \right) \\ &= 0.64 \left\| \mathbf{r}_t^{\mathrm{ret}} \right\|_2^2. \end{aligned} \tag{40}$$

When $g_{\parallel} = g_{\perp} = 0$, the output falls back exactly to $\hat{\mathbf{y}}_{0,t}$. This bound limits the correction magnitude relative to the frozen forecast, but it does not compare the corrected output with the true target. Therefore, it provides no guarantee of monotonically improved MSE for every window.

Likewise, the comparator in Eq. 38 is a fixed $\boldsymbol{\pi}^\star$, and the gains are label-free stability surrogates. Persistent distribution shift more generally calls for time-varying comparator sequences and dynamic regret (Besbes et al., 2015). We make no assumption on the variation budget of the optimal memory distribution and therefore claim no dynamic regret bound. Whether stability credit aligns with true error reduction, and whether gating reduces worst-case risk under low coverage, remain empirical questions.

C Reproducibility Details

This section describes data processing, frozen forecasters, offline memory construction, hyperparameter selection, and online evaluation. Training, validation, and testing remain strictly separated; test results are not used for training, memory construction, or hyperparameter selection.

Table 5: Datasets and forecasting tasks. All data are split chronologically; the ETT datasets use 6:2:2 and the remaining datasets use 7:1:2 training, validation, and test ratios.

Dataset	Domain	Variables	Temporal resolution	Split ratio
ETTh1 / ETTh2	Electricity transformers	7	1 hour	6:2:2
ETTm1 / ETTm2	Electricity transformers	7	15 minutes	6:2:2
Electricity	Electricity load	321	1 hour	7:1:2
Weather	Meteorological observations	21	10 minutes	7:1:2
Exchange	Exchange rates	8	1 day	7:1:2

C.1 Datasets and Temporal Splits

The main experiments use seven public long-term forecasting datasets (Qiu et al., 2024). ETTh1, ETTh2, ETTm1, and ETTm2 contain electricity transformer temperature records; Electricity contains electricity loads from multiple users; Weather contains multivariate meteorological observations; and Exchange contains daily exchange rates from multiple countries or regions. Dataset properties are summarized in Table 5.

Splits are determined before constructing sliding windows. Validation and test windows may use earlier historical context, but their targets must lie entirely within the corresponding interval. This preserves a complete context of length L without allowing targets to cross split boundaries. Splits are fixed by explicit boundary indices that are shared across all backbones, so every method compared in the tables reuses exactly the same window origins. Window origins follow the chronological order of each dataset and are never shuffled, and the number of evaluation windows depends only on the dataset length and the forecast horizon.

Each variable is z-score normalized using the mean and standard deviation computed from the training interval. The same training statistics are used during training, validation, and testing; scaling parameters are not re-estimated on the validation or test set. MSE and MAE are computed in the standardized space used by public long-term forecasting benchmarks (Murad et al., 2025; Qiu et al., 2025). If results in the original units are reported, predictions and targets are first inverse-transformed using the same training statistics and reported separately from results in the standardized space.

C.2 Forecasting Tasks and Frozen Models

The historical window length is fixed at $L = 512$, and forecast horizons are $H \in \{96, 192, 336, 720\}$. All datasets use multivariate forecasting. The supervised forecasters include DLinear, PatchTST, MICN, FreTS, and iTransformer; the probabilistic foundation model is Chronos-Bolt-Base. These models cover linear decomposition, channel-independent patches, convolution, frequency-domain MLPs, cross-variable attention, and pretrained probabilistic forecasting. Table 6 summarizes their main characteristics.

The five supervised models restore their architectures and parameters from their respective checkpoints and are switched to inference mode before residual memory construction. Chronos-Bolt-Base uses publicly available pretrained weights without fine-tuning on the target dataset. Freezing is verified by disabling gradients for all parameters, disabling training mode, checking that parameters remain unchanged before and after memory construction, and making no optimizer calls during testing. DMRC is always compared with the Frozen output of the same checkpoint on the same test windows.

Deterministic models directly output point forecasts. Chronos-Bolt-Base outputs the $Q_{0.1}$, $Q_{0.5}$, and $Q_{0.9}$ quantiles, with $Q_{0.5}$ used for the common MSE and MAE comparison and the outer quantiles used only to construct the probabilistic stability signal. This clearly separates point-forecast performance from probabilistic forecast dispersion, avoiding the inclusion of interval width in point-error metrics. Each (backbone, dataset, horizon) triple keeps its own offline memory, since the residual target dimension depends on the forecast horizon. Forecast caches are organized per backbone, dataset, lookback, and horizon, and are released once all seeds and methods of that group have finished, so reported results can be regenerated from a clean cache.

Table 6: Baseline forecasters used in the forecasting tasks and their main characteristics. All supervised models restore parameters from their checkpoints, switch to inference mode before residual memory construction, and remain frozen with gradients disabled and no optimizer calls; Chronos-Bolt-Base uses publicly available pretrained weights without fine-tuning on the target dataset.

Model	**Main characteristics**
DLinear (Zeng et al., 2023)	Linear decomposition separates trend and seasonal components and forecasts each through a linear mapping; the architecture is simple, with few parameters and low computational overhead.
PatchTST (Nie et al., 2022)	Divides the sequence into local patches and models each channel independently, using self-attention to capture long-range temporal dependencies.
MICN (Wang et al., 2023)	Uses multi-scale convolutions to extract and combine local and global temporal patterns.
FreTS (Yi et al., 2023)	Maps the sequence to the Fourier frequency domain and models frequency-domain features with an MLP to capture periodic structure.
iTransformer (Liu et al., 2024b)	Uses variables rather than time steps as tokens and applies attention to model cross-variable correlations.
Chronos-Bolt-Base (Ansari et al., 2024)	A probabilistic time-series foundation model based on a pretrained Transformer; large-scale pretraining provides temporal representations and forecasting capabilities across datasets, while quantile outputs represent the predictive distribution, enabling both point and probabilistic forecasts.

C.3 Offline Training and Memory Construction

The frozen forecaster is run on the training interval to train the residual-aware projector and construct the residual memory. For each channel, the two scalar gates are selected on the validation interval by minimizing validation MSE, and are then frozen before test-time evaluation. All online updates affect only the channel-wise retrieval distributions.

The base residual is $\mathbf{r}_i = \mathbf{y}_i - \mathcal{F}(\mathbf{x}_i)$. The residual-aware projector jointly encodes the historical window, base forecast, and local statistics for error matching. Samples are grouped into coarse residual modes and then aggregated into fine-grained key–value prototypes. Each key is the center of its query group, and each value is the mean residual. Memory capacity is B, with $\pi_{1,c}(i) = 1/B$ for every channel.

Training fits the projector, gates, and memory prototypes; validation selects memory capacity, top-K, temperature, correction strength, and update parameters. Configurations are fixed before testing. Separate horizons use separate memories to match target dimensions, and forecasters do not share residual values because their errors differ.

C.4 Hyperparameter Selection

The sensitivity experiments cover $B \in \{64, 128, 192, 256, 512, 1024\}$, $K \in \{1, 4, 8, 16, 32\}$, and $V \in \{1, 2, 3, 4\}$. For online adaptation, DMRC performs a per-window KL-regularized mirror-descent update after each forecast window. We evaluate $\eta \in \{5000, 1000\}$ together with $\lambda_o \in \{10^{-4}, 0\}$ on the canonical validation suite. The selected configuration is $\eta = 1000$ and $\lambda_o = 10^{-4}$. The view-consistency weight is fixed at $\lambda_v = 0.1$. The selected dynamic configuration is shared across all datasets, forecasting horizons, backbone forecasters, and memory seeds, and remains fixed during test-stream evaluation. No block-level update scheme is used in the reported experiments. For each remaining sensitivity analysis, one target hyperparameter is varied while all other settings are fixed to the selected configuration.

Because per-window and block-level updates have different frequencies, their learning rates are not directly comparable. Block-level updates aggregate credit over fixed consecutive windows, whereas

per-window updates occur after each forecast. The final configuration is selected by validation-stream average MSE and fixed during testing. If probabilistic and point-forecast stability scales differ, their learning rates are selected separately on validation data.

A fixed ε is added to all ratio denominators before division. The source-normalized ratio is restricted to $[R_{\max}^{-1}, R_{\max}]$, and the leave-one-out gain is restricted to $[-G_{\max}, G_{\max}]$. These numerical constraints only stabilize online updates; they do not change the memory keys and values or test targets.

C.5 Software and Hardware

Hardware. All experiments run on three single-GPU Linux workstations, equipped with an NVIDIA RTX 3090, an RTX 4080 SUPER, and an RTX 5060. Each (backbone, dataset, horizon, seed) combination is executed serially on a single GPU; no multi-GPU or distributed training is involved.

Software. The operating system is Ubuntu. We use Python 3.10.21 and PyTorch 2.1.0 with the CUDA 12.1 runtime and driver version 580.178.04. The main dependencies are numpy 1.23.5, pandas 1.5.3, scipy 1.10.1, scikit-learn 1.2.2, transformers 4.44.2, and accelerate 1.14.0. Chronos inference is built on chronos-forecasting 2.3.1 with the publicly available pretrained weights `amazon/chronos-bolt-base`. All experiments share a single code snapshot.

D Supplementary Experimental Results

This section provides a complete comparison among frozen forecasters, external retrieval-augmented methods, and DMRC, and specifies how results are aggregated. Whereas the main text highlights only paired Frozen–DMRC results, the appendix also includes KUP-BI and TQNet to assess the gains of residual memory over input- or continuation-pattern retrieval.

D.1 Comparison Methods and Fair Evaluation Protocol

Frozen directly uses the original frozen forecaster and serves as the deployment baseline without external augmentation. KUP-BI extracts continuation transformations from training-time history–target–subsequent continuation chains and retrieves and aggregates them based on the current history. TQNet retrieves historical segments similar to the current input and uses their associated future segments to assist forecasting. DMRC retrieves historical residual prototypes of the frozen forecaster and updates only their priorities on the test stream.

The four methods share the same data splits, input length, forecast horizon, frozen checkpoint, and test windows. Candidate information for KUP-BI, TQNet, and DMRC is restricted to the training and validation intervals; current test targets never enter the retrieval database. Each external augmentation method selects its own hyperparameters independently on the validation stream, and test results are used only for final reporting. All comparisons are grouped by frozen forecaster; absolute errors across different backbones cannot replace paired conclusions within the same backbone. Each (backbone, dataset, horizon) configuration is repeated over three memory seeds; reported values are arithmetic means over seeds.

D.2 Complete Model Comparisons

Tables 7–12 report complete results on seven datasets and four forecast horizons. Each cell gives MSE followed by MAE; within each frozen forecaster, the lowest error is highlighted in red and the second-lowest error in blue. Equal values share the same rank color. The Frozen column is repeated in every table so that each augmentation method is compared against the same deployment baseline.

Table 7: Complete multivariate long-term forecasting results on DLinear with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.432899	0.445155	0.433440	0.448234	0.430026	0.444467	**0.427463**	**0.443279**
	192	0.479073	0.479645	0.480181	0.483149	0.479358	0.480041	**0.471029**	**0.476228**
	336	0.523054	0.511119	0.524924	0.514819	0.523153	0.511593	**0.516335**	**0.508865**
	720	0.648415	0.593382	0.649589	0.596429	0.660851	0.600640	**0.628110**	**0.584218**
ETTh2	96	0.215853	0.311844	0.215721	0.312734	0.223660	0.317542	**0.211348**	**0.309277**
	192	0.269066	0.349638	0.269005	0.350411	0.273284	0.353330	**0.264636**	**0.346509**
	336	0.315255	0.381498	0.315604	0.382306	0.352805	0.396496	**0.303148**	**0.372434**
	720	0.397162	0.430737	0.397177	0.431297	0.397960	0.444527	**0.371148**	**0.413660**
ETTm1	96	0.328899	0.372851	0.330520	0.375769	0.328858	0.379729	**0.324642**	**0.370349**
	192	0.380648	0.404765	0.381828	0.407607	0.381991	0.405610	**0.374945**	**0.401777**
	336	0.430121	0.435648	0.430471	0.437540	0.433716	0.439639	**0.421932**	**0.431864**
	720	0.483977	0.474300	0.484811	0.476539	0.484314	0.474861	**0.474072**	**0.468925**
ETTm2	96	0.149344	0.254198	**0.147039**	**0.251884**	0.151674	0.256374	0.149344	0.254198
	192	0.180243	0.281716	**0.179297**	**0.279215**	0.186180	0.284336	0.180243	0.281716
	336	0.217156	**0.306416**	**0.216666**	0.306460	0.216765	0.306857	0.217156	**0.306416**
	720	0.270866	**0.343857**	0.270899	0.344125	**0.270747**	0.345976	0.270866	**0.343857**
Electricity	96	0.135915	0.231147	0.139859	0.240556	0.135915	0.231145	**0.134542**	**0.230788**
	192	0.150202	0.243725	0.154155	0.252834	0.150214	0.243730	**0.148515**	**0.243558**
	336	0.166162	0.260176	0.170301	0.268854	0.166166	0.260176	**0.161532**	**0.258918**
	720	0.205733	0.293516	0.210500	0.302227	0.205753	0.293518	**0.197625**	**0.290386**
Weather	96	0.170170	0.222564	0.170663	0.224033	0.153940	**0.203639**	**0.152844**	0.216386
	192	0.213552	0.258960	0.213728	0.259792	0.203391	**0.249235**	**0.199721**	0.261531
	336	0.260094	0.293949	0.260313	0.294510	0.250048	**0.285435**	**0.244011**	0.296065
	720	0.324636	0.339970	0.324652	0.340193	0.325279	0.340649	**0.297762**	**0.338227**
Exchange	96	0.085879	0.203670	0.091349	0.211049	0.087362	0.204848	**0.082496**	**0.203238**
	192	0.180900	0.300353	0.188699	0.308249	0.175820	0.295386	**0.153583**	**0.278436**
	336	0.340632	0.418391	0.349408	0.425234	0.410263	0.461886	**0.305969**	**0.391035**
	720	0.941391	0.720209	0.952746	0.724152	0.940511	0.723847	**0.860014**	**0.686890**
Average	–	0.317761	0.362979	0.319769	0.366079	0.321429	0.365554	**0.305180**	**0.357465**

Table 8: Complete multivariate long-term forecasting results on PatchTST with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.488260	0.494404	0.492480	0.497053	**0.464073**	**0.477609**	0.482010	0.491266
	192	0.586457	0.550079	0.579891	0.548233	**0.578665**	**0.543784**	0.582424	0.547349
	336	0.636474	0.592245	0.630858	0.590159	**0.611714**	**0.576246**	0.622957	0.583262
	720	0.773802	0.667501	0.761606	0.664664	0.764841	0.659524	**0.751239**	**0.654082**
ETTh2	96	0.263431	0.349214	**0.260386**	**0.344596**	0.263298	0.347315	0.263398	0.349228
	192	0.363817	0.411001	0.358708	0.408689	**0.349676**	**0.400871**	0.363183	0.410379
	336	0.429115	0.451280	**0.424259**	**0.449022**	0.428770	0.451122	0.427729	0.451918
	720	0.499798	0.496143	0.505354	0.498997	0.479846	0.485391	**0.469337**	**0.479684**
ETTm1	96	0.405398	0.426030	0.407736	0.429164	**0.389014**	**0.411655**	0.395820	0.416648
	192	0.475855	0.461549	0.477794	0.465582	**0.458830**	**0.445849**	0.471899	0.456479
	336	0.535033	0.501272	**0.527368**	0.499100	0.529236	**0.492723**	0.528786	0.494674
	720	0.612759	0.553127	0.607474	0.551630	**0.597629**	**0.542899**	0.600386	0.544098
ETTm2	96	0.166354	0.277945	0.160187	0.267402	**0.157373**	**0.261413**	0.160422	0.266868
	192	0.213688	0.304458	0.211252	0.303712	**0.205385**	**0.299114**	0.210593	0.305027
	336	0.260855	0.336283	0.257790	0.334898	**0.254030**	**0.332909**	0.259679	0.340851
	720	0.338335	0.395517	0.333923	**0.392816**	0.337705	0.395739	**0.332540**	0.394943
Electricity	96	0.133611	0.235509	0.142306	0.252605	**0.129815**	**0.224602**	0.133550	0.230883
	192	0.151296	0.251241	0.159988	0.267477	0.147421	0.241215	**0.144721**	**0.240990**
	336	0.168076	0.267476	0.176713	0.282586	0.164069	0.258747	**0.157388**	**0.256040**
	720	0.204270	0.298967	0.212692	0.313117	0.199984	0.290058	**0.191835**	**0.285747**
Weather	96	0.154454	0.205904	0.154731	0.207220	0.152295	**0.202271**	**0.150852**	0.209866
	192	0.214701	0.262095	0.214097	0.262075	0.208760	**0.256141**	**0.207997**	0.266716
	336	0.296512	0.317274	0.292622	**0.315222**	0.294355	0.315797	**0.284945**	0.316736
	720	0.374117	0.366108	0.368858	**0.363301**	0.373434	0.366163	**0.359122**	0.365541
Exchange	96	0.147485	0.282606	0.143713	0.277939	**0.130795**	**0.263734**	0.138256	0.271720
	192	0.316028	0.411785	0.309139	0.407528	**0.293010**	**0.391836**	0.308104	0.404367
	336	0.506232	0.521590	**0.496716**	0.520897	0.503007	**0.517113**	0.497408	0.521432
	720	1.418051	0.903372	**1.357584**	**0.884541**	1.377544	0.892656	1.391805	0.896066
Average	–	0.397652	0.413999	0.393794	0.414294	**0.387306**	**0.405161**	0.388871	0.409031

Table 9: Complete multivariate long-term forecasting results on iTransformer with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.467431	0.475578	**0.459910**	0.474380	0.459936	**0.470930**	0.463583	0.473951
	192	0.515527	0.510395	0.509591	0.509693	0.514210	0.510272	**0.507696**	**0.505818**
	336	0.572561	0.552282	0.571278	0.553079	**0.562811**	**0.545685**	0.572561	0.552282
	720	0.795017	0.661055	**0.780091**	**0.657102**	0.793190	0.660257	0.790524	0.659560
ETTh2	96	0.316036	0.372008	**0.308710**	0.369367	0.312875	**0.368391**	0.316036	0.372008
	192	0.371163	0.412078	**0.364155**	**0.409332**	0.370771	0.411430	0.368151	0.409469
	336	0.498916	0.476348	**0.486794**	0.472374	0.489983	**0.469576**	0.498916	0.476348
	720	0.632062	0.553026	**0.611100**	**0.545667**	0.621055	0.547007	0.632062	0.553026
ETTm1	96	0.379929	0.404893	**0.372210**	0.403052	0.373229	**0.401398**	0.377234	0.403663
	192	0.446027	0.439753	**0.436461**	**0.437806**	0.444438	0.438733	0.444945	0.439522
	336	0.506512	0.482198	**0.496251**	**0.478860**	0.501846	0.480156	0.505990	0.481978
	720	0.552665	0.520679	**0.544569**	0.518423	0.545712	**0.516090**	0.545850	0.516403
ETTm2	96	0.185213	0.277248	**0.181469**	**0.275174**	0.185463	0.277713	0.185213	0.277248
	192	0.229060	0.310464	**0.225103**	0.308536	0.226037	**0.307277**	0.229060	0.310464
	336	0.296776	0.347648	**0.290854**	**0.345797**	0.295573	0.347456	0.296776	0.347648
	720	0.299528	0.369391	0.297906	0.368706	**0.291548**	**0.364826**	0.299528	0.369391
Electricity	96	0.134750	0.233134	0.135255	0.235047	**0.132279**	**0.228649**	0.133070	0.231051
	192	0.153633	0.249985	0.157326	0.257070	0.151855	**0.248211**	**0.150622**	0.248931
	336	0.168226	0.265878	0.172400	0.273778	0.166023	0.262808	**0.162540**	**0.262300**
	720	0.207390	0.298982	0.210661	0.303997	0.207385	0.298885	**0.202015**	**0.295590**
Weather	96	0.167175	0.219122	0.168564	**0.216166**	0.170676	0.217324	**0.166980**	0.219675
	192	**0.227719**	0.273179	0.229095	**0.269165**	0.231363	0.270695	0.228975	0.274923
	336	**0.280721**	0.310368	0.282693	**0.306587**	0.285808	0.308127	0.282097	0.313142
	720	0.383321	0.373346	**0.381394**	**0.368741**	0.388341	0.372516	0.382309	0.374438
Exchange	96	0.153272	0.279423	0.146539	0.275068	**0.124124**	**0.252504**	0.153520	0.279746
	192	0.228150	0.346209	**0.226751**	**0.345128**	0.233977	0.350901	0.235657	0.352295
	336	0.368412	0.450251	**0.364840**	**0.449438**	0.436345	0.485132	0.376362	0.456624
	720	1.063831	0.792556	1.053456	0.788569	**1.032300**	**0.780905**	1.066009	0.791289
Average	–	0.378608	0.402053	**0.373765**	0.400575	0.376755	**0.399781**	0.377653	0.401742

Table 10: Complete multivariate long-term forecasting results on MICN with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.583615	0.532335	**0.570048**	**0.528624**	0.586591	0.536272	0.580688	0.530641
	192	0.613012	0.556591	**0.608007**	**0.555032**	0.615836	0.557967	0.610261	0.556112
	336	0.661335	0.593348	**0.655154**	**0.591238**	0.660915	0.592505	0.656356	0.591796
	720	0.800477	0.676788	0.789891	0.674012	0.813386	0.683483	**0.783827**	**0.667282**
ETTh2	96	0.267381	0.360262	0.265061	0.358675	0.261880	**0.356674**	**0.260590**	0.357636
	192	0.312989	0.388038	0.310661	0.386362	0.318216	0.391249	**0.309762**	**0.385384**
	336	0.331466	0.398873	0.328371	**0.396753**	0.332566	0.399763	**0.327963**	0.398088
	720	0.451497	0.470405	0.446551	0.467947	0.448973	0.468928	**0.441670**	**0.466516**
ETTm1	96	0.389062	0.420670	0.388528	0.421513	0.392734	0.423077	**0.386804**	**0.419645**
	192	0.466607	0.464547	**0.460010**	**0.461104**	0.467194	0.464272	0.462601	0.461636
	336	0.537051	0.508541	**0.526079**	**0.502990**	0.532648	0.506357	0.529721	0.504561
	720	0.575692	0.536887	**0.569707**	**0.535068**	0.574520	0.537786	0.572044	0.536202
ETTm2	96	0.155142	0.262594	**0.153532**	**0.259477**	0.154965	0.260329	0.154986	0.260068
	192	0.208463	0.299876	0.206011	**0.298356**	**0.205484**	0.299735	0.207177	0.302575
	336	0.270753	0.351431	0.266652	0.348752	0.270332	0.349691	**0.261899**	**0.344128**
	720	0.328017	0.381726	**0.324694**	**0.379892**	0.327981	0.381397	0.326220	0.384792
Electricity	96	0.148082	0.253088	0.148198	0.255414	**0.143055**	**0.246582**	0.145421	0.250157
	192	0.166097	0.269436	0.166307	0.271824	**0.162088**	**0.264188**	0.163147	0.266516
	336	0.176716	0.280167	0.177017	0.282831	**0.173572**	**0.276202**	0.175312	0.278938
	720	0.214984	0.310241	**0.208495**	**0.305897**	0.214050	0.308922	0.213600	0.309035
Weather	96	0.169883	0.221495	0.170287	0.222799	0.156124	**0.207862**	**0.152301**	0.216723
	192	0.213193	0.257679	0.213412	0.258446	**0.199559**	**0.244862**	0.200607	0.260569
	336	0.259207	0.291859	0.259429	0.292313	0.249036	**0.283316**	**0.242881**	0.293399
	720	0.324639	0.338757	0.324575	0.338691	0.325786	0.339845	**0.301759**	**0.337777**
Exchange	96	0.088390	0.206885	0.094122	0.214152	0.094872	0.213985	**0.085004**	**0.206750**
	192	0.192248	0.309110	0.200417	0.317160	0.192911	0.310949	**0.169150**	**0.295776**
	336	0.360107	0.427956	0.369776	0.435618	0.425578	0.474176	**0.325818**	**0.401564**
	720	1.288309	0.839624	1.269449	0.835428	1.309257	0.841521	**1.251537**	**0.831561**
Average	–	0.376943	0.400329	0.373944	0.399870	0.378932	0.400782	**0.367825**	**0.396994**

Table 11: Complete multivariate long-term forecasting results on FreTS with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.455126	0.472071	0.453309	0.472604	**0.441966**	**0.463396**	0.452116	0.470053
	192	0.528191	0.519759	0.523831	0.518921	**0.505258**	**0.505633**	0.522619	0.516671
	336	0.581426	0.556839	0.578906	0.556832	**0.567235**	**0.548616**	0.579310	0.555700
	720	0.735171	0.650100	0.726626	0.646974	0.730220	0.650239	**0.720408**	**0.642350**
ETTh2	96	0.258439	0.346110	0.255693	0.344918	**0.248194**	**0.339210**	0.253467	0.341727
	192	0.305617	0.379335	0.302924	0.378069	**0.301652**	**0.376757**	0.305603	0.379214
	336	0.361611	0.412246	0.358017	0.410624	0.358450	0.413555	**0.346186**	**0.401206**
	720	0.436463	0.453686	0.433051	0.452161	0.426910	0.455288	**0.409691**	**0.436722**
ETTm1	96	0.329251	0.379016	0.330208	0.382045	**0.319049**	**0.372623**	0.322749	0.374242
	192	0.398581	0.422938	0.398626	0.425616	**0.387702**	**0.416307**	0.395221	0.420252
	336	0.445164	0.452243	0.443754	0.453174	**0.433432**	**0.445650**	0.439169	0.448543
	720	0.517328	0.504727	0.514996	0.504364	0.511667	0.500577	**0.507545**	**0.499168**
ETTm2	96	0.165518	0.266424	**0.163486**	**0.265851**	0.165687	0.266785	0.165409	0.265952
	192	0.200422	0.297056	**0.197991**	0.296041	0.199775	0.296935	0.198608	**0.294056**
	336	0.234468	0.325385	0.232041	0.324179	0.235363	0.325653	**0.226212**	**0.317887**
	720	0.281968	0.359641	0.280511	0.358901	0.279572	0.358084	**0.278586**	**0.357582**
Electricity	96	**0.129737**	**0.224880**	0.133352	0.234271	0.129738	0.224885	0.131484	0.227617
	192	0.147757	0.241653	0.151298	0.250290	0.147754	0.241656	**0.145139**	**0.239426**
	336	0.164554	0.259925	0.168409	0.268055	0.164552	0.259932	**0.158697**	**0.256563**
	720	0.201736	0.292738	0.205827	0.300882	0.201747	0.292731	**0.192049**	**0.286295**
Weather	96	0.154337	0.205868	0.153594	0.206155	0.154057	**0.205608**	**0.148952**	0.209131
	192	0.199913	0.249093	0.199025	0.249089	0.199081	**0.248334**	**0.192941**	0.256477
	336	0.250749	0.288426	0.250197	0.288476	0.250366	**0.288026**	**0.241745**	0.293438
	720	0.325736	0.340446	0.325091	0.340313	0.325341	**0.339955**	**0.314411**	0.343577
Exchange	96	0.103366	0.230526	0.106552	0.231011	0.113613	0.237759	**0.102683**	**0.225866**
	192	0.214699	0.332983	0.220982	0.337922	0.201875	**0.323043**	**0.201022**	0.328381
	336	0.346160	0.431882	0.347232	0.432484	0.398847	0.468187	**0.325164**	**0.417621**
	720	1.098411	0.760644	1.087697	0.757001	**1.070011**	**0.752146**	1.084632	0.760319
Average	–	0.341853	0.380594	0.340829	0.381687	0.338183	0.379199	**0.334351**	**0.377359**

Table 12: Complete multivariate long-term forecasting results on Chronos-Bolt-Base with the frozen forecaster and external enhancement methods.

Dataset	H	Frozen		KUP-BI		TQNet		DMRC	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.447565	0.429227	0.440507	0.430855	0.440033	0.436131	**0.433136**	**0.427028**
	192	0.510203	0.474864	0.502464	0.475883	0.514415	0.487780	**0.487821**	**0.466762**
	336	0.560320	0.515706	0.553599	0.516494	0.563558	0.523970	**0.544409**	**0.513583**
	720	0.713434	0.615216	0.703829	0.613735	0.715201	0.622593	**0.680362**	**0.603068**
ETTh2	96	0.221115	**0.307992**	**0.220177**	0.308337	0.247542	0.327224	0.221954	0.309892
	192	0.272744	0.347833	**0.271561**	0.347767	0.284603	0.359753	0.271764	**0.346888**
	336	0.314501	0.379693	0.313432	0.379358	0.343550	0.401059	**0.310935**	**0.376929**
	720	0.411638	0.440962	0.409166	0.439732	**0.403070**	0.447801	0.405915	**0.435512**
ETTm1	96	0.371358	0.362956	0.364550	0.364203	0.362528	0.373632	**0.336166**	**0.357496**
	192	0.444831	0.406644	0.436145	0.407420	0.449939	0.414407	**0.400869**	**0.397918**
	336	0.510744	0.446140	0.501659	0.445383	0.512786	0.452401	**0.460540**	**0.434048**
	720	0.603531	0.506888	0.592898	0.504817	0.603180	0.509948	**0.551045**	**0.487397**
ETTm2	96	0.158410	0.252258	0.156925	0.252197	0.162320	0.257847	**0.153877**	**0.249154**
	192	0.196649	0.285007	**0.194803**	0.284571	0.201186	0.289235	0.194855	**0.284518**
	336	0.235806	0.315495	0.233670	0.314708	0.251979	0.326233	**0.231952**	**0.313979**
	720	0.309569	0.363221	0.306692	0.362155	0.301197	0.362064	**0.299533**	**0.357694**
Electricity	96	0.124023	**0.210618**	0.126087	0.219043	0.123759	0.211061	**0.122503**	0.211698
	192	0.142570	**0.229021**	0.145149	0.237334	0.142095	0.229378	**0.139860**	0.230287
	336	0.161682	0.249550	0.164926	0.257716	0.160649	**0.248987**	**0.156846**	0.250298
	720	0.210409	0.292956	0.211300	0.298958	0.208409	0.292079	**0.194490**	**0.286814**
Weather	96	0.176879	0.210039	0.174439	0.210086	0.162404	0.203537	**0.151003**	**0.200736**
	192	0.222341	0.252988	0.220079	0.253019	0.212638	0.249253	**0.189772**	**0.246521**
	336	0.280611	0.295038	0.278023	0.294861	0.269111	0.291307	**0.241495**	**0.288966**
	720	0.362142	0.350267	0.358343	0.349507	0.343944	0.342565	**0.311133**	**0.342109**
Exchange	96	**0.099183**	**0.219045**	0.101039	0.221489	0.104820	0.224391	0.099795	0.220296
	192	**0.193770**	**0.313799**	0.194991	0.314832	0.201683	0.320146	0.194177	0.315347
	336	0.338874	0.423009	0.339005	0.423163	0.396887	0.456852	**0.337586**	**0.421434**
	720	0.822011	0.678573	0.830680	0.681817	1.198558	0.754615	**0.787615**	**0.663713**
Average	–	0.336318	0.363393	0.333791	0.364623	0.352930	0.372009	**0.318265**	**0.358574**

D.3 Result Interpretation and Reporting Scope

The paired Frozen–DMRC results in the main text show that, among the frozen forecasters evaluated so far, adding DMRC reduces MSE on most tasks. Larger gains mainly occur on datasets such as Weather and Exchange, where error structures vary more noticeably over time. Representative results include a 15.10% MSE reduction for DLinear–Exchange at $H = 192$ and a 12.01% reduction for MICN–Exchange at the same horizon. Meanwhile, a small number of tasks may still exhibit increased error, while equal Frozen–DMRC entries indicate that the gated correction has fallen back to the original forecast.

Comparisons with external methods should address two distinct questions. The results of KUP-BI and TQNet relative to Frozen assess whether input- or continuation-pattern retrieval supplies useful auxiliary information; DMRC's results relative to these two methods assess whether shifting retrieval toward frozen-forecaster errors further reduces negative transfer.

Error magnitudes across different frozen forecasters are not directly comparable. DLinear, PatchTST, MICN, FreTS, and iTransformer are deterministic point forecasters, whereas Chronos-Bolt-Base uses its forecast median for point-error evaluation. The core comparison unit in the appendix tables is always the same frozen forecaster, dataset, and forecast horizon; cross-backbone averages are used only to summarize DMRC's compatibility.

D.4 Stability-based Risk Stratification

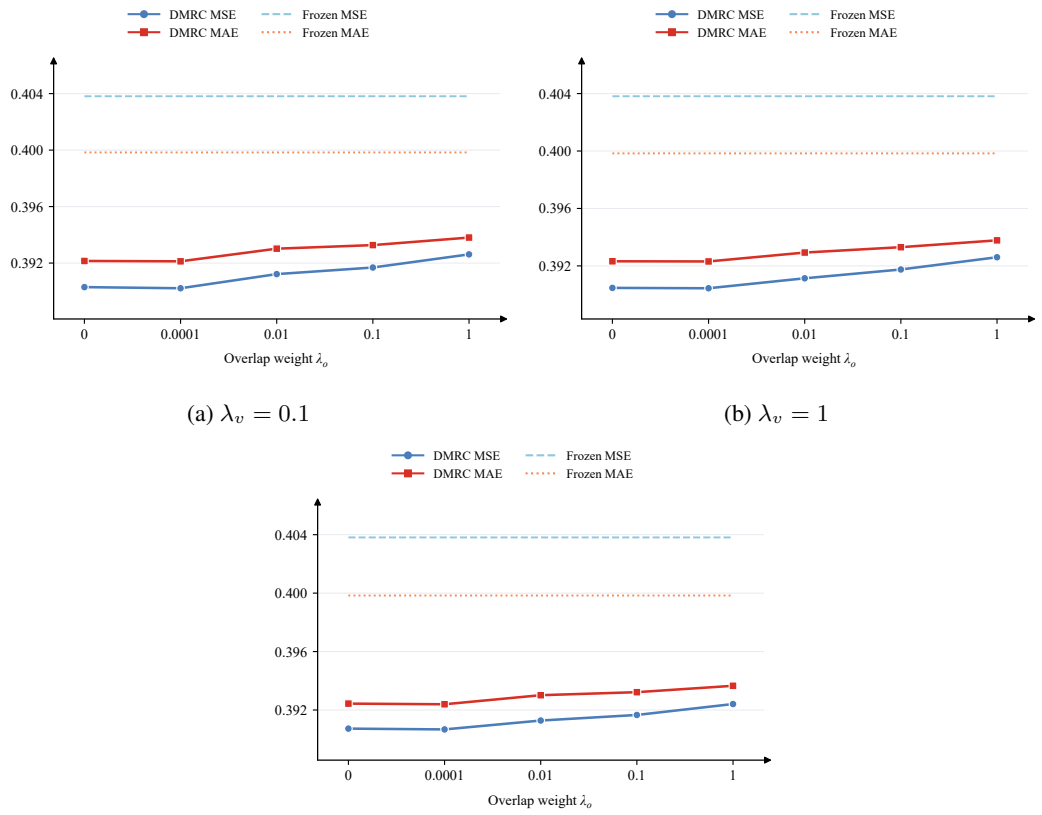
The stability proxy is not intended to provide a calibrated estimate of the exact forecasting error. We instead evaluate whether it can identify potentially unreliable windows for selective fallback. On the held-out test stream of ETTh1 with iTransformer at $H = 96$, we define the highest MSE 10% of windows as the high-risk group. Selecting the highest-proxy 10% recovers 35.7% of this group, compared with 10% under random selection. The corresponding AUROC and AUPRC are 0.709 and 0.332, respectively. The AUROC remains between 0.707 and 0.710 when the jitter scale varies from 0.01 to 0.10. These results provide evidence that the stability signal can support risk stratification in this evaluated test stream, while not implying a globally monotonic or calibrated relationship between the proxy and the true forecasting error.

D.5 Loss Weight Sensitivity Analysis

λ_v and λ_o control the contributions of the multi-view consistency term and the temporal-overlap consistency term, respectively. We fix all other experimental settings and jointly evaluate $\lambda_v \in \{0.1, 1, 10\}$ and $\lambda_o \in \{0, 10^{-4}, 10^{-2}, 10^{-1}, 1\}$. Figure 5 reports the results; from left to right, the three panels correspond to the three values of λ_v. The results are evaluated over 28 tasks on the validation streams, where the results for each task are first averaged across three memory seeds, followed by averaging across all tasks.

Overall, varying λ_v has a relatively small effect on prediction error. When $\lambda_o \leq 10^{-4}$, the model performance remains relatively stable, whereas further increasing λ_o leads to a slight increase in the average error. All parameter configurations achieve lower average MSE and MAE than the Frozen baseline, with $(\lambda_v, \lambda_o) = (0.1, 10^{-4})$ achieving the best overall performance.

These results indicate that the weights of the two consistency losses should not simply be assigned values of the same order of magnitude. Instead, they should be selected according to the numerical scales of the respective loss terms to ensure balanced contributions to model updates. In particular, a relatively small λ_o is already sufficient to provide the temporal-overlap consistency constraint, while an excessively large weight may cause this constraint to exert an overly strong influence on the optimization process.



(a) $\lambda_v = 0.1$

(b) $\lambda_v = 1$

(c) $\lambda_v = 10$

Figure 5: Sensitivity to stability-loss weights. Each panel varies the overlap weight λ_o at a fixed view-consistency weight λ_v, with all other settings held constant. MSE and MAE are averaged over 28 validation tasks, after averaging three seeds per task. Lower values indicate better performance.