LEARNING WHERE FROZEN FORECASTERS FAIL: RESIDUAL-BASED RETRIEVAL AUGMENTATION FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) supports time-series forecasting by retrieving historical trajectories and representations to inform predictions. However, the potential of historical evidence to refine a forecaster's existing predictions by targeting its remaining errors remains underexplored. Such refinement builds on the temporal patterns already captured by the forecaster, while its historical residuals provide model-specific evidence about the direction and magnitude of past errors. Using RAG to augment forecasting via residuals presents two challenges: identifying relevant historical errors when the current residual is unobserved, and determining their correction strength when historical and current prediction conditions differ. To address these challenges, we propose Residual-based Retrieval-Augmented Forecasting of Time-series (ResRAF), which organizes a frozen forecaster's historical errors into a residual-based knowledge base. Its Residual-Aware Retriever learns from historical residual supervision to predict residual pattern representations from the current input and base forecast, then combines residual proximity with forecast-shape similarity to retrieve complete historical residual sequences. State-Adaptive Retrieval Calibration adjusts the contributions of these sequences according to the prediction states of both the current and historical instances, derived from observable input and forecast characteristics. The calibrated residuals are aggregated and added to the base forecast, with the forecaster remaining frozen and all additional components fitted on training data and fixed during evaluation. Experiments across multiple datasets, forecasting architectures, and prediction horizons show that ResRAF outperforms existing methods in forecasting accuracy.

1 INTRODUCTION

Retrieval-augmented generation (RAG) supports time-series forecasting by retrieving historical trajectories and representations to inform predictions (Han et al., 2025; Zhang et al., 2025a; Ning et al., 2025). Yet using this evidence to target a forecaster's remaining errors remains underexplored. A trained forecaster already incorporates learned temporal patterns into its base forecast. Taking this forecast as the starting point allows historical evidence to focus on the errors that remain. This motivates a refinement approach that builds on the forecaster's existing predictive capabilities and uses historical evidence to guide model-specific corrections. Historical residuals provide a basis for this refinement. Each residual records the difference between an observed outcome and the prediction made by a particular forecaster, capturing the direction and magnitude of its past errors. Historical trajectories describe observed temporal patterns, while residuals relate those observations to what the model predicted. They therefore provide model-specific reference information about where and how predictions required adjustment. Residual-memory methods such as ResMem already use historical errors through nearest-neighbor regression (Yang et al., 2023). Building on this foundation, we investigate how RAG can turn a forecaster's historical residuals into correction evidence for its current predictions.

However, using RAG to augment forecasting with historical residuals presents two challenges. First, historical residuals are observable, but the current residual cannot be computed before future observations become available. Selecting relevant errors therefore requires connecting the current

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.

input and base forecast to historical residual patterns. Second, forecasting errors can vary across temporal contexts (Chen et al., 2024). Even retrieved instances with related error patterns may differ from the current instance in their input and forecast characteristics. Retrieval similarity can identify candidate corrections, but it does not directly determine how strongly each residual should influence the current forecast.

To address these challenges, we propose **Res**idual-based **R**etrieval-**A**ugmented **F**orecasting of Time-series (**ResRAF**). A *Residual-Based Knowledge Base* stores the frozen forecaster's historical errors. The *Residual-Aware Retriever* (RAR) learns to predict residual pattern representations from the current input and base forecast under historical residual supervision. It combines similarity in these representations with forecast-shape similarity to select historical instances, whose complete residual sequences supply candidate corrections. *State-Adaptive Retrieval Calibration* (SARC) derives prediction states from observable input and forecast characteristics, then adds current-state correction strengths to historical-state adjustments. The calibrated residuals are aggregated and added to the base forecast. The forecaster remains frozen, while additional components are fitted on training data and fixed during evaluation.

We validate these benefits broadly across frozen forecasters, datasets, and horizons, with full results and ablations reported in Section 4. Our contributions are as follows:

- We propose ResRAF to turn a frozen forecaster's historical errors into retrievable correction evidence, enabling their reuse to refine subsequent forecasts without updating the base model.
- We develop RAR to address residual selection when the current error is unobserved, enabling retrieval to reflect anticipated error patterns using only information available at prediction time.
- We design SARC to address uncertainty in residual transfer strength, adapting each historical residual's contribution to the prediction conditions of both the current and historical instances.

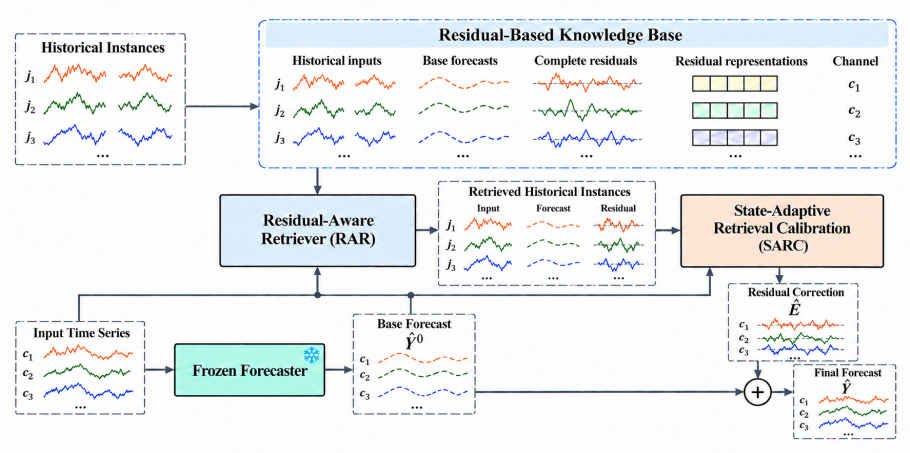


Figure 1: **Overview of ResRAF.** ResRAF uses a Residual-Aware Retriever (RAR) to retrieve relevant historical instances from a residual-based knowledge base, conditioned on the current input and base forecast. State-Adaptive Retrieval Calibration (SARC) adjusts the contributions of their complete residual sequences according to the current and historical prediction states. The calibrated residuals are then aggregated and added to the base forecast to produce the final prediction. The base forecaster remains frozen throughout training and inference.

2 Related Work

2.1 Retrieval-Augmented Time-Series Forecasting

Retrieval-augmented generation (RAG) supplements parametric knowledge with retrieved evidence and has improved performance across various tasks (Lewis et al., 2020). In time-series forecasting, kNN-MTS and RAFT use the observed continuations of retrieved historical instances to inform predictions (Zhang et al., 2025b; Han et al., 2025). PFRP combines predictions from global historical memory with those of a local forecasting model (Du et al., 2026). TimeRAF further learns retrieval

from forecasting utility (Zhang et al., 2025a), while TS-RAG adaptively fuses retrieved and model representations (Ning et al., 2025). These approaches use historical trajectories, predictions, or representations to support forecasting. However, they do not retrieve a specific forecaster's historical residuals as explicit correction evidence. Such residuals record the direction and magnitude of that forecaster's past errors. We therefore retrieve a frozen forecaster's historical residuals and calibrate their contributions before adding them to its base forecast.

2.2 Residual-aware Forecast Calibration

Residual-aware calibration uses prediction errors to adjust point forecasts. UEC-STD addresses autoregressive forecasting errors by learning separate trend and seasonal corrections without retraining the base forecaster (Nguyen et al., 2026). Historical residuals can also serve as explicit correction evidence. In macroeconomic forecasting, Guerrón-Quintana & Zhong (2023) use errors from similar historical periods to adjust ARMA/ARIMA predictions. More generally, ResMem fits a nearest-neighbor regressor to a base model's residuals and adds its output to the prediction (Yang et al., 2023). PIR instead identifies unreliable forecasting instances by estimating per-instance accuracy, then revises the base forecast via a local module conditioned on covariate and exogenous context and a retrieval-based global module for long-tail patterns (Liu et al., 2025). Context-aware methods further account for variation in forecasting conditions. SOLID selects contextually similar historical samples to adapt the prediction layer (Chen et al., 2024). TAFAS uses partially observed outcomes for gated calibration (Kim et al., 2025), while COSA updates a context-aware output adapter for frozen forecasters (Im & Kwon, 2026). Unlike PIR's contextual and covariate retrieval for long-tail coverage or the context-adaptive strategies above, ResRAF retrieves the frozen forecaster's own historical residuals as the explicit correction signal, calibrated according to the prediction states of both current and historical instances. In this study, we investigate how to implement effective residual calibration for time-series forecasting tasks using RAG.

3 Methodology

3.1 Overview

Problem Formulation. Given an input sequence $\mathbf{X} \in \mathbb{R}^{L\times C}$ with L historical time steps and C channels, and a knowledge base $\mathcal{C}_\theta$ of historical time-series instances, we aim to improve the forecasting accuracy of a trained forecaster f_θ while keeping its parameters fixed. The forecaster produces a base forecast $\widehat{\mathbf{Y}}^0 = f_\theta(\mathbf{X}) \in \mathbb{R}^{H\times C}$ for the next H time steps. We use information from instances retrieved from $\mathcal{C}_\theta$ to refine this forecast toward the actual future observations $\mathbf{Y} \in \mathbb{R}^{H\times C}$.

For a historical instance with input $\mathbf{X}_i$ and observed future $\mathbf{Y}_i$, its prediction residual is defined as $\mathbf{E}_i = \mathbf{Y}_i - f_\theta(\mathbf{X}_i)$. This residual records the direction and magnitude of the forecaster's errors across the forecast horizon. Historical residuals computed using the same forecaster thus provide model-specific information for forecast correction. Conditioned on the current input $\mathbf{X}$ and base forecast $\widehat{\mathbf{Y}}^0$, we aim to construct a correction $\widehat{\mathbf{E}} \in \mathbb{R}^{H\times C}$ based on the residuals of a set of retrieved instances. The final forecast is $\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}}^0 + \widehat{\mathbf{E}}$, with the objective of reducing its error relative to $\mathbf{Y}$.

Framework Overview. To achieve this, we propose **Res**idual-based **R**etrieval-**A**ugmented **F**orecasting of Time-series (**ResRAF**). As shown in Figure 1, ResRAF organizes historical instances and their prediction residuals in the knowledge base $\mathcal{C}_\theta$ and combines two core components: Residual-Aware Retriever (RAR) and State-Adaptive Retrieval Calibration (SARC). Given an input sequence $\mathbf{X}$, the frozen forecaster first produces the base forecast $\widehat{\mathbf{Y}}^0$. RAR uses the input and base forecast together with historical residual information to retrieve relevant instances from $\mathcal{C}_\theta$. SARC then adjusts the contributions of the retrieved prediction residuals based on the current input and base forecast, and aggregates them into the correction $\widehat{\mathbf{E}}$. Adding this correction to the base forecast yields the final prediction $\widehat{\mathbf{Y}}$. We describe knowledge base construction in Section 3.2 and the framework architecture in Section 3.3.

3.2 RESIDUAL-BASED KNOWLEDGE BASE

To support forecast correction through retrieval, ResRAF associates historical time-series instances with the prediction errors of a specific trained forecaster. Given training pairs $\mathcal{D}_{\text{tr}} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{N}$ and a frozen forecaster f_θ, we construct a model-specific residual-based knowledge base $\mathcal{C}_\theta$. Each pair contains an L-step input sequence and its subsequent H observations over C channels. We compute the residual matrix $\mathbf{E}_i = \mathbf{Y}_i - f_\theta(\mathbf{X}_i)$. Each channel of each training sample corresponds to one historical instance, whose residual sequence is $\mathbf{e}_{i,c} = \mathbf{E}_i[:, c] \in \mathbb{R}^H$. We enumerate these $M = NC$ instances as knowledge-base entries indexed by j, with residual sequence $\mathbf{e}_j$ and source channel index c_j.

To represent historical instances for residual-aware retrieval, we seek to characterize their prediction error patterns. Direct comparisons between raw residual sequences can be sensitive to channel-scale differences and instance-specific fluctuations, which may obscure similarities in their error patterns. We incorporate a residual pattern transformation ϕ that adjusts for channel-scale differences and summarizes common patterns of error variation across the forecast horizon. This produces a residual pattern representation:

$$\mathbf{z}_j = \phi(\mathbf{e}_j, c_j) \in \mathbb{R}^d, \tag{1}$$

where d is the representation dimension. The transformation is fitted on training residuals and remains fixed thereafter, with implementation details provided in Appendix D.1.

Each entry stores the historical input, base forecast, complete residual sequence, residual pattern representation, and source channel index:

$$\mathcal{C}_\theta = \left\{ \left(\mathbf{x}_j, \widehat{\mathbf{y}}_j^0, \mathbf{e}_j, \mathbf{z}_j, c_j \right) \right\}_{j=1}^{M}. \tag{2}$$

Here, $\mathbf{x}_j \in \mathbb{R}^L$ and $\widehat{\mathbf{y}}_j^0 \in \mathbb{R}^H$ denote the input and base forecast for the corresponding channel of entry j. The residual pattern representation and base forecast support retrieval, while the complete residual sequence provides correction information over all H forecast steps.

3.3 FRAMEWORK ARCHITECTURE

Given an input sequence $\mathbf{X}_i$, the frozen base forecaster produces an initial forecast $\widehat{\mathbf{Y}}_i^0 = f_\theta(\mathbf{X}_i)$. ResRAF augments this forecast through two components. The Residual-Aware Retriever (RAR) selects relevant historical entries from the residual-based knowledge base and retrieves their complete residual sequences as candidate corrections. State-Adaptive Retrieval Calibration (SARC) adjusts their contributions according to the characteristics of the current and historical instances. The calibrated residual sequences are then aggregated and added to the base forecast to obtain the final prediction $\widehat{\mathbf{Y}}_i$.

Residual-Aware Retriever (RAR). The residual pattern representations stored in the knowledge base provide a basis for retrieving historical instances whose residuals may inform the current correction. However, the corresponding representation for a new input cannot be computed directly because its future observations are unavailable. We therefore introduce a residual representation predictor f^r. For channel c of the current input sequence $\mathbf{X}_i$, it produces

$$\widehat{\mathbf{z}}_{i,c} = f^r\left(\mathbf{x}_{i,c}, \widehat{\mathbf{y}}_{i,c}^0, \mathbf{h}_c\right) \in \mathbb{R}^d, \tag{3}$$

where $\mathbf{h}_c$ is a learnable channel embedding. The predictor is shared across channels and supervised by historical residual representations $\mathbf{z}_j = \phi(\mathbf{e}_j, c_j)$, aligning its outputs with the representation space used by the knowledge base. Appendix D.2 specifies the network architecture and input preprocessing.

For each channel c, RAR retrieves from knowledge-base entries with the same channel index, denoted by $\mathcal{J}_c = \{j : c_j = c\}$. For channel c of $\mathbf{X}_i$ and each historical entry $j \in \mathcal{J}_c$, we compute the joint retrieval score

$$s_{i,j}^c = s_{i,j}^{\text{pred},c} + s_{i,j}^{\text{res},c} + b_j. \tag{4}$$

The prediction similarity score $s_{i,j}^{\text{pred},c}$ measures shape similarity between the current base forecast $\widehat{\mathbf{y}}_{i,c}^0$ and the historical base forecast $\widehat{\mathbf{y}}_j^0$. The residual similarity score compares the predicted residual

representation with the stored representation of historical entry j:

$$s_{i,j}^{\mathrm{res},c} = -\frac{\|\widehat{\mathbf{z}}_{i,c} - \mathbf{z}_j\|_2^2}{\tau d}, \tag{5}$$

where $\tau > 0$ controls the scale of the residual-distance term, and b_j is a fixed recency bonus. Appendix D.3 provides the computation of the prediction similarity score and the definition of the recency bonus.

RAR then selects the K highest-scoring eligible entries:

$$\mathcal{N}_K^c(i) = \mathrm{TopK}_{j \in \mathcal{A}_c(i)}\, s_{i,j}^c, \tag{6}$$

where $\mathcal{A}_c(i) \subseteq \mathcal{J}_c$ excludes temporally overlapping entries during training, as detailed in Appendix D.4. The operator TopK returns historical entry indices. Their complete residual sequences $\{\mathbf{e}_j : j \in \mathcal{N}_K^c(i)\}$ form the candidate corrections passed to SARC.

State-Adaptive Retrieval Calibration (SARC). Prior work shows that prediction residuals can exhibit different biases across temporal contexts (Chen et al., 2024). This motivates adjusting the contribution of a historical residual according to the current instance and the historical instance from which it was obtained. Specifically, SARC summarizes their observable input and forecast characteristics as discrete prediction states. For each instance, we concatenate pooled sequence shapes, summary statistics such as the mean, standard deviation, and trend, and spectral amplitudes from its input and base forecast into a feature vector $\mathbf{r}$.

We construct a state assignment function ϕ^s by standardizing these feature vectors using training statistics and fitting R cluster centers $\{\boldsymbol{\kappa}_r\}_{r=1}^R$ with K-means. Each cluster defines a prediction state, and each instance is assigned to its nearest center:

$$\begin{aligned} g_{i,c} = \phi^s(\mathbf{x}_{i,c}, \widehat{\mathbf{y}}_{i,c}^0) &= \operatorname*{arg\,min}_{r \in \{1,\dots,R\}} \|\widetilde{\mathbf{r}}_{i,c} - \boldsymbol{\kappa}_r\|_2^2, \\ g_j = \phi^s(\mathbf{x}_j, \widehat{\mathbf{y}}_j^0) &= \operatorname*{arg\,min}_{r \in \{1,\dots,R\}} \|\widetilde{\mathbf{r}}_j - \boldsymbol{\kappa}_r\|_2^2, \end{aligned} \tag{7}$$

where $\widetilde{\mathbf{r}}$ denotes the standardized feature vector. The standardization and cluster centers remain fixed after fitting, providing a common state partition for current and historical instances. State assignment uses only the input and base forecast. Appendix D.5 specifies the feature calculations and fitting details.

For each channel c, SARC maintains two sets of coefficients fitted on training data: correction strengths $\{\alpha_r^c\}_{r=1}^R$ and historical-state adjustments $\{\delta_r^c\}_{r=1}^R$. The current instance's state $g_{i,c}$ selects $\alpha_{g_{i,c}}^c$, which is shared across the retrieved neighborhood. Each historical entry's state g_j selects $\delta_{g_j}^c$, which adjusts the contribution of its residual sequence. Their sum defines the transfer coefficient:

$$\gamma_{i,j}^c = \alpha_{g_{i,c}}^c + \delta_{g_j}^c. \tag{8}$$

To combine the candidate corrections, SARC takes a weighted average of the calibrated residual sequences. The weights are computed as:

$$w_{i,j}^c = \frac{\exp(s_{i,j}^c / T_w)}{\sum_{k \in \mathcal{N}_K^c(i)} \exp(s_{i,k}^c / T_w)}, \qquad j \in \mathcal{N}_K^c(i), \tag{9}$$

where $T_w > 0$ controls weight concentration. We set a high temperature relative to the score differences so that the selected weights are approximately uniform. Retrieval scores thus primarily determine neighborhood membership, while the transfer coefficients adjust each residual's contribution. Appendix D.4 provides the temperature setting and numerical implementation. The resulting correction is

$$\widehat{\mathbf{e}}_{i,c} = \sum_{j \in \mathcal{N}_K^c(i)} w_{i,j}^c\, \gamma_{i,j}^c\, \mathbf{e}_j. \tag{10}$$

The averaging weights sum to one, whereas the transfer coefficients can be signed and need not sum to one. Appendix D.6 details coefficient estimation, bounds, and an equivalent computation that groups residuals by historical state. Finally, we add the correction to each channel's base forecast and stack the resulting sequences as the final forecast:

$$\widehat{\mathbf{y}}_{i,c} = \widehat{\mathbf{y}}_{i,c}^0 + \widehat{\mathbf{e}}_{i,c}, \qquad \widehat{\mathbf{Y}}_i = [\widehat{\mathbf{y}}_{i,1}, \dots, \widehat{\mathbf{y}}_{i,C}]. \tag{11}$$

Learning Objectives and Training. Training proceeds in stages with the base forecaster f_θ frozen. We first fit the residual transformation ϕ and state assignment function ϕ^s on training data and construct the knowledge base $\mathcal{C}_\theta$. These components remain fixed while we train the residual representation predictor:

$$\mathcal{L}_{\text{align}} = \frac{1}{M}\sum_{j=1}^{M} \ell_{\text{align}}\big(f^r(\mathbf{x}_j, \widehat{\mathbf{y}}_j^0, \mathbf{h}_{c_j}), \mathbf{z}_j\big)\,, \tag{12}$$

where $\mathbf{z}_j = \phi(\mathbf{e}_j, c_j)$ is the historical residual representation. We optimize the shared predictor and channel embeddings using a combination of MSE and SmoothL1 loss (Appendix D.2). The predictor is selected by validation representation MSE and then frozen.

We next fit the correction strengths and historical-state adjustments for each channel. Let $\boldsymbol{\beta}_c = [\alpha_1^c, \ldots, \alpha_R^c, \delta_1^c, \ldots, \delta_R^c]^\top \in \mathbb{R}^{2R}$ collect these coefficients. We estimate them by minimizing the corrected forecast error with a ridge penalty:

$$\widetilde{\boldsymbol{\beta}}_c = \arg\min_{\boldsymbol{\beta}_c} \left\{ \sum_{i=1}^{N} \left\| \mathbf{y}_{i,c} - \widehat{\mathbf{y}}_{i,c}^0 - \widehat{\mathbf{e}}_{i,c}(\boldsymbol{\beta}_c) \right\|_2^2 + \lambda s_c \left\| \boldsymbol{\beta}_c - \boldsymbol{\beta}_c^{(0)} \right\|_2^2 \right\}. \tag{13}$$

Here $\widehat{\mathbf{e}}_{i,c}(\boldsymbol{\beta}_c)$ is the correction in Equation 10 on training instance i. The prior $\boldsymbol{\beta}_c^{(0)}$ shrinks correction strengths toward a global channel scale and historical-state adjustments toward zero. The factor s_c scales the penalty, and λ controls its strength. After solving the unconstrained ridge problem, we clip the estimated coefficients to predefined intervals to obtain $\boldsymbol{\beta}_c^\star$. Appendix D.6 details the prior, penalty scale, estimation procedure, and clipping intervals. Training details are provided in Appendix D.7.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate on six standard long-term forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Exchange, and Weather. Following standard protocols, we use prediction horizons $H \in \{96, 192, 336, 720\}$ for all datasets. All methods use identical chronological splits, and we report mean squared error (MSE) and mean absolute error (MAE), where lower values indicate better forecasting performance. Dataset statistics and split details are provided in Appendix A.

Baselines. We evaluate five frozen backbones spanning distinct forecasting paradigms: decomposition-based DLinear (Zeng et al., 2023), inverted-Transformer iTransformer (Liu et al., 2024), orthogonal-domain OLinear (Yue et al., 2025), retrieval-augmented RAFT (Han et al., 2025), and multiscale TimeMixer (Wang et al., 2024). Each backbone is tuned on validation data and then fixed; every corrected result is paired with the exact frozen checkpoint that produces its base forecast and historical residuals. ResRAF therefore changes only the post-forecast correction, not the underlying forecaster. We additionally include the zero-shot foundation model Chronos-Bolt-Small (Ansari et al., 2024) as a sixth forecaster in the paired comparison. As post-hoc correction baselines, we compare against UEC-STD (Nguyen et al., 2026) and PIR (Liu et al., 2025), both applied to the same frozen forecasts.

4.2 Improvements over Frozen Forecasters

In both tables, "Original" denotes the frozen base forecast and "+ResRAF" its corrected forecast; red bold marks the better value within each matched setting, with ties retained. Table 1 averages the four horizons for each backbone–dataset pair, while Table 2 averages the five backbones for each dataset–horizon pair.

Broad gains across frozen backbones and datasets. Table 1 summarizes the full 6×6 backbone–dataset grid. Across the 36 matched pairs and two metrics, ResRAF improves 61 of the 72 reported MSE/MAE values. The gains are not confined to one forecasting architecture: improvements appear across linear, Transformer-based, retrieval-augmented, and multiscale backbones, as well as the zero-shot foundation model. For DLinear and iTransformer, both MSE and MAE improve on all six

Table 1: Comparing each frozen base with its ResRAF-corrected forecast.

	DLinear				iTransformer				OLinear				RAFT				TimeMixer				Chronos-Bolt			
	Original		+ResRAF		Original		+ResRAF		Original		+ResRAF		Original		+ResRAF		Original		+ResRAF		Original		+ResRAF	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.465	0.461	**0.446**	**0.450**	0.451	0.446	**0.446**	**0.442**	0.426	0.426	**0.422**	**0.423**	**0.421**	**0.436**	**0.421**	0.437	0.444	0.436	**0.437**	**0.434**	0.461	**0.424**	**0.456**	**0.424**
ETTh2	0.566	0.520	**0.554**	**0.507**	0.382	0.407	**0.374**	**0.404**	**0.367**	**0.388**	0.368	0.393	0.362	0.409	**0.351**	**0.398**	0.387	**0.408**	**0.382**	**0.408**	0.360	**0.382**	**0.354**	0.389
ETTm1	0.404	0.407	**0.381**	**0.394**	0.416	0.415	**0.407**	**0.412**	0.374	**0.377**	**0.369**	0.378	0.348	0.378	**0.345**	**0.376**	**0.382**	**0.397**	0.383	0.398	0.416	0.384	**0.376**	**0.376**
ETTm2	0.349	0.399	**0.329**	**0.374**	0.289	0.331	**0.282**	**0.330**	0.271	**0.314**	**0.268**	**0.314**	0.256	0.320	**0.255**	**0.319**	0.281	0.327	**0.277**	**0.326**	0.281	**0.316**	**0.271**	0.319
Exchange	0.346	0.414	**0.312**	**0.390**	0.368	0.409	**0.313**	**0.385**	0.353	0.397	**0.300**	**0.374**	0.441	0.439	**0.397**	**0.416**	0.405	0.425	**0.305**	**0.381**	0.334	0.390	**0.294**	**0.371**
Weather	0.265	0.317	**0.236**	**0.273**	0.261	0.281	**0.245**	**0.271**	0.238	0.260	**0.231**	**0.257**	0.240	0.286	**0.227**	**0.271**	0.244	0.274	**0.239**	**0.271**	0.252	0.269	**0.233**	**0.257**
Avg.	0.399	0.420	**0.376**	**0.398**	0.361	0.381	**0.345**	**0.374**	0.338	0.360	**0.326**	**0.357**	0.345	0.378	**0.333**	**0.369**	0.357	0.378	**0.337**	**0.370**	0.351	0.361	**0.331**	**0.356**

Table 2: Paired results averaged over the five backbones, grouped by forecast horizon ($H \in \{96, 192, 336, 720\}$).

	$H = 96$				$H = 192$				$H = 336$				$H = 720$			
	Original		+ResRAF		Original		+ResRAF		Original		+ResRAF		Original		+ResRAF	
Dataset	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.380	0.401	**0.378**	**0.399**	0.428	0.428	**0.425**	**0.427**	0.473	0.453	**0.462**	**0.448**	0.484	0.482	**0.473**	**0.476**
ETTh2	0.302	0.353	**0.295**	**0.347**	0.392	0.410	**0.384**	**0.405**	0.449	0.451	**0.442**	**0.444**	0.508	0.491	**0.501**	**0.491**
ETTm1	0.322	0.357	**0.311**	**0.349**	0.364	0.381	**0.355**	**0.376**	0.399	0.404	**0.396**	0.404	0.456	0.438	**0.446**	**0.437**
ETTm2	0.178	0.264	**0.172**	**0.257**	0.245	0.312	**0.238**	**0.305**	0.308	0.353	**0.301**	**0.346**	0.426	0.424	**0.419**	**0.422**
Exchange	0.088	0.210	**0.085**	**0.205**	0.184	0.309	**0.172**	**0.299**	0.347	0.429	**0.305**	**0.405**	0.911	0.720	**0.740**	**0.648**
Weather	0.171	0.219	**0.156**	**0.202**	0.217	0.262	**0.203**	**0.247**	0.268	0.300	**0.253**	**0.285**	0.343	0.353	**0.331**	**0.341**
Avg.	0.240	0.301	**0.233**	**0.293**	0.305	0.350	**0.296**	**0.343**	0.374	0.398	**0.360**	**0.389**	0.521	0.485	**0.485**	**0.469**

datasets; Chronos-Bolt improves in MSE on all six as well. Exchange shows particularly pronounced reductions, indicating that the benefit of historical error correction can remain substantial even when the frozen forecaster itself is already fully trained and fixed. The paired protocol is important for interpreting these results. Absolute errors differ substantially across architectures and datasets, whereas every "+ResRAF" entry differs from its control only by the correction layer applied to the same frozen predictions. The improvements therefore measure the value of reusing historical forecast errors rather than differences in backbone capacity, optimization, or checkpoint selection.

Consistent behavior across forecast horizons. Table 2 provides a complementary horizon-wise view. After averaging over the five backbones, ResRAF reduces MSE in all 24 dataset–horizon cells. For MAE, it improves 23 of the 24 cells, with the only exception (ETTm1 at H=336) within 0.3% of the averaged base. The improvement therefore persists from short to long horizons rather than being driven by a narrow prediction range. Complete per-backbone, per-horizon results in Appendix B show the same overall pattern before aggregation, supporting the use of ResRAF as a general post-hoc refinement rather than a horizon-specific adjustment.

4.3 Comparison with Post-hoc Correction Baselines

We next compare ResRAF with the UEC-STD and PIR post-hoc correction baselines on all six datasets. Because absolute forecasting errors vary substantially across datasets and frozen backbones, Table 3 reports the relative change from each matched frozen forecast; negative values denote error reduction, and red bold marks the best relative change within each backbone–dataset comparison. Across the matched backbone–dataset–metric pairs, ResRAF achieves the best relative change in most settings (49 of 60 against UEC-STD and 53 of 60 against PIR), with particularly clear margins on Exchange and Weather. In aggregate, ResRAF attains the strongest relative MSE reduction for all five backbones against both baselines. This comparison distinguishes the proposed mechanism from the generic benefit of applying a post-hoc correction: the gains persist when both methods start from the same frozen predictions, supporting the value of explicitly retrieving and transferring historical error evidence.

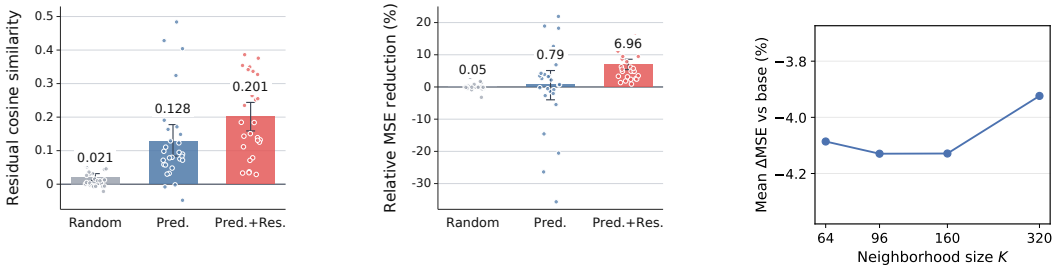
4.4 Mechanism Analysis

We examine whether retrieval selects aligned historical residuals and whether state-adaptive transfer limits harmful corrections.

Residual-guided scoring retrieves better-aligned historical residuals. For each test query, neighbors are selected before accessing its target; the true query residual is used only for post-hoc alignment evaluation. Figure 2 shows a positive association between retrieval score and residual alignment. Across 24 DLinear tasks, we compare *Random* (uniform same-channel retrieval, averaged over three

Table 3: Relative MSE change over each matched frozen backbone on all six datasets, comparing ResRAF with the UEC-STD and PIR post-hoc correction baselines (horizon-averaged; negative values indicate improvement).

	DLinear			iTransformer			OLinear			RAFT			TimeMixer		
Dataset	+ResRAF	+UEC	+PIR	+ResRAF	+UEC	+PIR	+ResRAF	+UEC	+PIR	+ResRAF	+UEC	+PIR	+ResRAF	+UEC	+PIR
ETTh1	**-3.83%**	+17.70%	-2.37%	**-1.09%**	+4.24%	+0.02%	**-0.79%**	+13.86%	+3.43%	-0.10%	+16.01%	**-0.51%**	**-1.52%**	+17.09%	-1.38%
ETTh2	-2.26%	-0.41%	**-3.06%**	**-2.07%**	-0.33%	-1.29%	+0.23%	**-0.22%**	+1.11%	**-3.01%**	-0.38%	+2.60%	**-1.47%**	-0.19%	-0.03%
ETTm1	-5.85%	-2.51%	**-6.78%**	-2.07%	**-3.43%**	+1.18%	-1.35%	**-1.55%**	+3.82%	**-1.12%**	+0.27%	+0.51%	+0.21%	-1.02%	**-1.51%**
ETTm2	**-6.73%**	-0.96%	+8.78%	**-2.48%**	-1.19%	+0.66%	-0.90%	**-1.25%**	+3.10%	**-0.50%**	-0.30%	+4.86%	**-1.70%**	-0.59%	+0.51%
Exchange	**-7.02%**	-0.04%	+6.52%	**-8.22%**	+0.17%	+21.66%	**-9.10%**	-0.09%	+10.31%	**-10.80%**	+0.04%	+6.09%	**-15.70%**	+0.34%	+2.24%
Weather	**-11.68%**	-0.93%	-2.16%	**-6.32%**	-0.41%	-1.76%	**-2.81%**	-0.60%	+0.53%	**-6.25%**	-0.34%	-0.40%	**-2.09%**	-0.73%	+0.05%
Avg.	**-6.23%**	+2.14%	+0.15%	**-3.71%**	-0.16%	+3.41%	**-2.45%**	+1.69%	+3.72%	**-3.63%**	+2.55%	+2.19%	**-3.71%**	+2.48%	-0.02%



(a) (b) (c)

Figure 2: Retrieval-alignment analysis on DLinear with fixed $K{=}64$. (a,b) Residual cosine similarity and relative MSE reduction across dataset–horizon tasks. *Pred.* uses s^{pred}, whereas *Pred.+Res.* uses the full RAR score. All variants use the same SARC procedure with train-fitted, variant-specific calibration. Dots denote tasks; bars and whiskers denote means and 95% task-bootstrap confidence intervals. Positive values in (b) indicate improvement. (c) Aggregate sensitivity to the retrieval neighborhood size K over the 120 tasks; more negative is better.

seeds), *Pred.* (forecast-shape similarity), and *Pred.+Res.* (adding predicted-residual similarity). Each variant uses SARC calibrated separately on training data. Compared with *Pred.*, *Pred.+Res.* increases mean residual cosine similarity from 0.128 to 0.201 and mean relative MSE reduction over the frozen base from 0.79% to 6.96%. This comparison evaluates retrieval together with its refitted calibration, rather than isolating alignment as the sole source of improvement.

Structured retrieval and conditional transfer play complementary roles. Table 4 evaluates the retrieval and transfer components over the full 120-task grid, reporting the mean and median paired relative change versus the frozen base (ΔMSE/ΔMAE; negative is better), the number of degraded tasks, the worst single-task ΔMSE, and the head-to-head win rate against full ResRAF. *Random Retrieval* replaces the retrieval score with random scores (averaged over three seeds); *w/o* s^{res} removes the residual similarity term, retaining only forecast-shape signals; and *w/o State-Adaptive Transfer* replaces SARC's calibrated coefficients with the mean retrieved residual. Randomizing retrieval nearly erases the average gain, so historical residuals help only when the relevant evidence can be identified. Removing the residual similarity term eliminates most of it (-0.5% versus -3.9% mean ΔMSE) and sharply raises the worst case: proximity in the learned residual representation space carries the bulk of the retrieval signal. This agrees with the alignment analysis in Figure 2: better neighbor selection gives better corrections.

Conditional transfer exhibits a different effect. Replacing it with direct averaging leaves the mean improvement nearly unchanged, but increases both the frequency and severity of negative-transfer cases, so average MSE alone understates its contribution. In our results, retrieval primarily determines *which* historical errors are useful, while conditional transfer controls *how safely* that evidence is applied: the former drives evidence quality, the latter robustness.

4.5 Qualitative Analysis

Figure 3 shows fixed-K examples from all six datasets. Depending on the query, ResRAF adjusts magnitude, local slope, turning points, and peak amplitudes while the frozen backbone stays fixed, moving the forecast toward the ground truth where the base systematically misses.

Table 4: Component ablation over the 120-task grid. Variants and reported statistics are defined in Section 4.4. Red bold marks the best value per column.

Variant	ΔMSE mean	ΔMSE median	ΔMAE mean	ΔMAE median	Degraded	Worst ΔMSE	vs. ResRAF
ResRAF	**-3.9%**	**-2.4%**	**-2.4%**	**-1.3%**	**8/120**	**+2.0%**	–
Random Retrieval	-0.1%	-0.0%	+0.3%	+0.0%	40/120	+3.2%	111/120
w/o s^{res}	-0.5%	-1.1%	-0.2%	-0.4%	29/120	+35.7%	76/120
w/o State-Adaptive Transfer	**-3.9%**	-2.0%	**-2.4%**	-0.8%	29/120	+12.2%	89/120

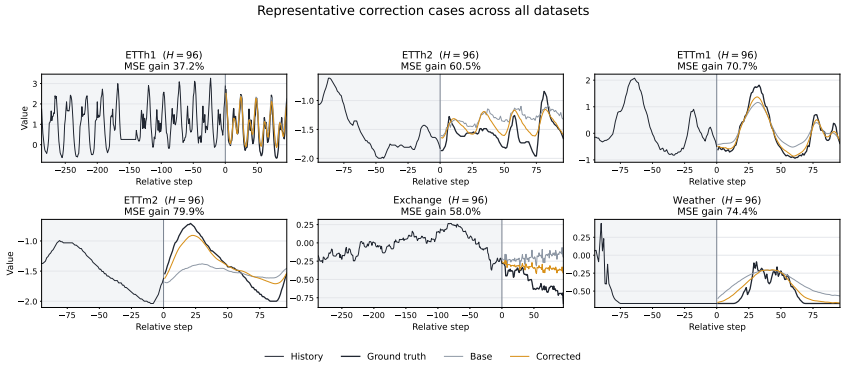


Figure 3: Illustrative fixed-K correction examples across all six datasets. Each panel shows the historical context followed by the forecast window, comparing the ground truth, frozen base forecast, and corrected forecast.

Sensitivity to the neighborhood size K. The mean gain is nearly flat for $K \in \{64, 96, 160\}$ and drops at K=320 (Figure 2c), while larger neighborhoods degrade fewer tasks (11 versus 5 of 120). The preferred K varies across backbone–dataset pairs (Table 8), so we select it per task on validation data.

Summary of empirical evidence. Reusing a frozen forecaster's historical errors helps across backbones, datasets, and horizons, and the same correction also improves a zero-shot foundation model. The gains come from retrieval rather than generic averaging: neighbors chosen by the learned residual representation match the query's unseen error better, and randomizing them erases the improvement. Calibration matters mostly for safety, cutting degraded tasks from 29 to 8 of 120. Historical errors are thus reusable evidence, provided that retrieval localizes the current query in error space and transfer is conditioned on prediction state.

5 Conclusion

In this paper, we introduced ResRAF, a residual-based retrieval-augmented framework that refines time-series forecasts using a frozen forecaster's historical errors. ResRAF organizes these errors in a model-specific knowledge base and combines residual-aware retrieval with state-adaptive calibration. To select relevant historical instances when the current residual is unavailable, the Residual-Aware Retriever predicts residual pattern representations from the current input and base forecast and combines them with forecast-shape similarity. State-Adaptive Retrieval Calibration then adjusts the contributions of the retrieved complete residual sequences according to the prediction states of both the current and historical instances. The calibrated residuals are aggregated and added to the base forecast. Experiments show consistent improvements across backbones, datasets, and horizons, while ablations confirm that residual-aware retrieval drives the gains and state-adaptive calibration improves robustness by limiting negative transfer. Together, these designs provide a way to reuse model-specific error information for forecast correction without updating the base forecaster.

The usefulness of this approach depends on whether historical error patterns remain informative for subsequent predictions. Moreover, the knowledge base is tied to a specific forecaster and requires reconstruction when that forecaster changes. Future work could investigate how to update the residual knowledge base and calibration components as new observations become available or forecasting conditions change.

AI USE STATEMENT

During the literature review and manuscript preparation, we used generative AI tools for preliminary literature search, analysis of related publications, feedback on experimental design and result presentation, drafting, and language polishing. We manually checked and verified all suggested references against primary publications to guarantee bibliographic accuracy, and reviewed all AI-assisted text and analyses to preserve our intended technical semantics. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All benchmarks are publicly available, and we follow their standard chronological splits, as detailed in Appendix A. Appendix D describes the complete procedure for knowledge-base construction, retrieval, calibration, and validation-based selection, with all structural and numerical settings specified in Table 10. Only K and λ are selected per task using validation data. Complete results for all evaluated backbone–dataset–horizon combinations are reported in Appendix B. The supplementary material includes an anonymized implementation, task configurations, and scripts for reproducing the main results.

REFERENCES

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

Mouxiang Chen, Lefei Shen, Han Fu, Zhuo Li, Jianling Sun, and Chenghao Liu. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 341–352. Association for Computing Machinery, 2024. doi: 10.1145/3637528.3671926. URL https://doi.org/10.1145/3637528.3671926.

Dazhao Du, Tao Han, and Song Guo. Predicting the future by retrieving the past. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(25):20896–20904, 2026. doi: 10.1609/aaai.v40i25.39230. URL https://ojs.aaai.org/index.php/AAAI/article/view/39230.

Pablo Guerrón-Quintana and Molin Zhong. Macroeconomic forecasting in times of crises. *Journal of Applied Econometrics*, 38(3):295–320, 2023. doi: 10.1002/jae.2951. URL https://doi.org/10.1002/jae.2951.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O. Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21774–21797. PMLR, 2025. URL https://proceedings.mlr.press/v267/han25d.html.

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://proceedings.iclr.cc/paper_files/paper/2026/hash/2a8ce71baac4c89bf9ff479d8240c7d9-Abstract-Conference.html.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17):17868–17876, 2025. doi: 10.1609/aaai.v39i17.33965. URL https://ojs.aaai.org/index.php/AAAI/article/view/33965.

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.

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

Zhiding Liu, Mingyue Cheng, Guanhao Zhao, Jiqian Yang, Qi Liu, and Enhong Chen. Improving time series forecasting via instance-aware post-hoc revision. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 35604–35629. Curran Associates, Inc., 2025. doi: 10.52202/085713-1196. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/331c41353b053683e17f7c88a797701d-Paper-Conference.pdf.

Minh Hoang Nguyen, Dai Do, Huu Hiep Nguyen, Dung Nguyen, Kien Do, and Hung Le. Reviving error correction in modern deep time-series forecasting. arXiv preprint arXiv:2605.21088, 2026. URL https://arxiv.org/abs/2605.21088.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. TS-RAG: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5448. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/eed25c037bc08afcbefab6f7a6b700e0-Abstract-Conference.html.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Xiaohao Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Zitong Yang, Michal Lukasik, Vaishnavh Nagarajan, Zonglin Li, Ankit Rawat, Manzil Zaheer, Aditya K. Menon, and Sanjiv Kumar. ResMem: Learn what you can and memorize the rest. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2654. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/bf0857cb9a41c73639f028a80301cdf0-Abstract.html.

Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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.

Huanyu Zhang, Chang Xu, Yi-Fan Zhang, Zhang Zhang, Liang Wang, and Jiang Bian. TimeRAF: Retrieval-augmented foundation model for zero-shot time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5654–5665, 2025a. doi: 10.1109/TKDE.2025.3579137. URL https://doi.org/10.1109/TKDE.2025.3579137.

Huiliang Zhang, Ping Nie, Lijun Sun, and Benoit Boulet. Nearest neighbor multivariate time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):12606–12618, 2025b. doi: 10.1109/TNNLS.2024.3490603. URL https://doi.org/10.1109/TNNLS.2024.3490603.

A Datasets

Table 5 summarizes the six benchmarks used in this work, including the number of variates, total timesteps, and temporal granularity. Each dataset is evaluated at four forecast horizons ($H \in \{96, 192, 336, 720\}$) with six frozen forecasters (five trained backbones and the zero-shot Chronos-Bolt-Small). All datasets use standard chronological train/validation/test splits, and preprocessing follows the protocol of the original backbone implementations.

Table 5: Dataset statistics.

Dataset	Variates	Timesteps	Granularity
ETTh1	7	17,420	1 hour
ETTh2	7	17,420	1 hour
ETTm1	7	69,680	15 min
ETTm2	7	69,680	15 min
Exchange	8	7,588	1 day
Weather	21	52,696	10 min

Table 6: Full per-horizon paired MSE results (144 tasks). Bold marks corrected values that improve over the matched frozen base (per-horizon rows) or the best value within each backbone group (average rows). MAE results are provided in Table 7.

		DLinear				iTransformer				OLinear				RAFT				TimeMixer				Chronos-Bolt	
Dataset	H	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF
ETTh1	96	0.396	**0.388**	**0.392**	**0.374**	0.392	**0.388**	**0.388**	0.399	0.362	0.363	**0.359**	0.374	0.372	**0.371**	0.382	**0.365**	0.379	**0.379**	**0.379**	**0.370**	0.393	**0.389**
	192	0.446	**0.436**	0.459	**0.430**	0.440	**0.436**	**0.434**	**0.427**	0.416	**0.415**	**0.415**	0.422	0.403	0.404	0.419	**0.402**	0.435	**0.434**	0.492	**0.423**	0.452	**0.445**
	336	0.496	**0.472**	0.522	**0.489**	0.487	**0.481**	0.497	**0.462**	0.459	**0.453**	0.468	0.468	0.436	0.437	0.609	**0.434**	0.487	**0.468**	0.508	**0.468**	0.495	**0.488**
	720	0.521	**0.488**	0.852	0.526	0.486	**0.481**	0.570	0.516	0.468	**0.458**	0.722	0.500	0.473	**0.472**	0.558	0.476	0.475	**0.466**	0.717	0.492	0.503	**0.501**
	Avg.	0.465	**0.446**	0.556	0.455	0.451	**0.446**	0.472	0.451	0.426	**0.422**	0.491	0.441	0.421	0.421	0.492	**0.419**	0.444	**0.437**	0.524	0.438	0.461	**0.456**
ETTh2	96	0.344	**0.332**	**0.343**	0.366	0.301	**0.292**	**0.301**	**0.296**	0.286	**0.285**	0.287	0.291	0.285	**0.278**	**0.285**	**0.279**	0.293	**0.285**	**0.292**	**0.290**	0.282	**0.279**
	192	0.482	**0.470**	**0.481**	**0.476**	0.379	**0.370**	**0.375**	**0.374**	0.366	**0.364**	**0.364**	0.371	0.356	**0.343**	0.356	0.377	0.377	**0.375**	0.378	0.377	0.351	**0.346**
	336	0.596	**0.592**	0.605	**0.554**	0.420	**0.412**	0.421	**0.417**	0.404	**0.403**	0.407	0.409	0.379	**0.373**	**0.378**	0.388	0.445	**0.431**	0.447	**0.444**	0.393	**0.386**
	720	0.842	**0.820**	**0.819**	**0.754**	0.427	**0.423**	**0.425**	**0.420**	0.411	0.419	**0.405**	0.411	0.428	**0.409**	**0.423**	0.446	0.435	0.436	**0.430**	0.439	0.413	**0.405**
	Avg.	0.566	0.554	0.562	**0.538**	0.382	**0.374**	0.380	0.377	0.367	0.368	**0.366**	0.371	0.362	**0.351**	0.360	0.372	0.387	**0.382**	0.387	0.388	0.360	**0.354**
ETTm1	96	0.345	**0.316**	**0.332**	**0.313**	0.334	**0.326**	**0.327**	0.336	0.302	**0.299**	**0.301**	0.317	0.304	**0.292**	**0.301**	**0.294**	0.322	**0.321**	**0.319**	**0.315**	0.328	**0.303**
	192	0.382	**0.358**	**0.372**	**0.355**	0.386	**0.376**	**0.376**	0.406	0.356	**0.351**	**0.351**	0.362	0.327	**0.326**	**0.326**	**0.325**	0.368	**0.366**	**0.364**	**0.359**	0.383	**0.345**
	336	0.413	**0.399**	**0.409**	**0.388**	0.448	**0.445**	**0.433**	**0.440**	0.389	**0.385**	**0.381**	0.403	0.355	**0.355**	**0.354**	0.366	0.390	0.394	**0.386**	**0.387**	0.430	**0.393**
	720	0.478	**0.453**	**0.465**	**0.455**	0.496	**0.483**	**0.468**	0.499	0.451	**0.440**	**0.439**	0.473	0.406	**0.405**	0.417	0.418	0.449	0.451	**0.445**	**0.446**	0.523	**0.465**
	Avg.	0.404	0.381	0.394	**0.378**	0.416	0.407	**0.401**	0.420	0.374	0.369	**0.368**	0.389	0.348	**0.345**	0.350	0.351	0.382	0.383	0.378	**0.377**	0.416	**0.376**
ETTm2	96	0.194	**0.176**	**0.192**	**0.193**	0.183	**0.179**	**0.183**	0.183	0.170	**0.168**	**0.169**	0.173	0.164	**0.163**	**0.164**	0.166	0.176	**0.172**	**0.176**	**0.175**	0.174	**0.171**
	192	0.288	**0.261**	**0.287**	0.301	0.247	**0.240**	**0.242**	0.248	0.233	**0.232**	**0.229**	0.239	0.220	**0.219**	**0.219**	0.224	0.240	**0.236**	**0.240**	**0.237**	0.241	**0.233**
	336	0.358	**0.341**	**0.357**	0.445	0.311	**0.301**	**0.307**	0.314	0.291	**0.290**	**0.285**	0.294	0.275	**0.274**	0.275	0.298	0.307	**0.299**	**0.304**	0.307	0.303	**0.292**
	720	0.556	**0.539**	**0.546**	0.595	0.416	**0.409**	**0.409**	0.419	0.389	**0.383**	**0.386**	0.415	0.366	**0.362**	**0.363**	0.394	0.401	**0.400**	**0.397**	0.417	0.407	**0.388**
	Avg.	0.349	**0.329**	0.345	0.383	0.289	**0.282**	0.285	0.291	0.271	0.268	**0.267**	0.280	0.256	**0.255**	**0.255**	0.271	0.281	**0.277**	0.279	0.284	0.281	**0.271**
Exchange	96	0.094	**0.088**	0.094	**0.093**	0.086	**0.084**	0.086	0.117	0.082	0.082	0.082	0.089	0.090	**0.086**	0.090	0.100	0.089	**0.085**	0.090	**0.088**	0.087	**0.079**
	192	0.186	**0.183**	**0.185**	0.208	0.174	**0.168**	**0.173**	0.207	0.171	**0.164**	**0.171**	0.201	0.201	**0.163**	**0.201**	0.207	0.190	**0.180**	0.190	0.212	0.173	**0.152**
	336	0.333	**0.314**	0.334	0.379	0.321	**0.302**	0.321	0.383	0.323	**0.280**	**0.321**	0.349	0.384	**0.337**	**0.383**	0.445	0.375	**0.293**	0.377	**0.366**	0.309	**0.270**
	720	0.773	**0.662**	**0.771**	0.781	0.892	**0.699**	0.893	0.997	0.836	**0.676**	0.838	0.891	1.088	**1.002**	1.091	**1.032**	0.964	**0.663**	0.965	0.978	0.766	**0.676**
	Avg.	0.346	**0.312**	0.346	0.365	0.368	**0.313**	0.368	0.426	0.353	**0.300**	0.353	0.383	0.441	**0.397**	0.441	0.446	0.405	**0.305**	0.406	0.411	0.334	**0.294**
Weather	96	0.195	**0.164**	**0.194**	**0.186**	0.178	**0.162**	**0.178**	**0.172**	0.152	**0.149**	0.152	0.152	0.167	**0.148**	**0.167**	**0.166**	0.162	**0.159**	**0.162**	**0.160**	0.172	**0.153**
	192	0.238	**0.204**	**0.234**	**0.227**	0.225	**0.212**	**0.225**	**0.220**	0.202	**0.196**	**0.201**	0.202	0.211	**0.197**	**0.210**	**0.210**	0.209	**0.205**	**0.207**	0.210	0.216	**0.199**
	336	0.281	**0.250**	**0.280**	**0.279**	0.280	**0.263**	**0.278**	**0.277**	0.258	**0.251**	**0.257**	0.259	0.258	**0.245**	**0.257**	**0.256**	0.261	**0.255**	**0.261**	0.262	0.272	**0.254**
	720	0.347	**0.327**	**0.344**	0.352	0.359	**0.345**	**0.358**	**0.358**	0.340	**0.328**	**0.335**	0.346	0.325	**0.320**	**0.324**	0.326	0.345	**0.337**	**0.340**	0.348	0.349	**0.327**
	Avg.	0.265	**0.236**	0.263	0.261	0.261	**0.245**	0.260	0.257	0.238	**0.231**	0.236	0.240	0.240	**0.227**	0.240	0.240	0.244	**0.239**	0.242	0.245	0.252	**0.233**

B Full Per-Horizon Results

Tables 6 (MSE) and 7 (MAE) report the complete per-horizon paired results on the $5 \times 6 \times 4$ backbone–dataset–horizon grid plus the zero-shot Chronos-Bolt-Small forecaster (144 tasks in total; $H \in \{96, 192, 336, 720\}$). For every frozen backbone we report the original base forecast (Original), the full ResRAF correction (+ResRAF), UEC-STD (+UEC), and PIR (+PIR); the final column group reports Chronos-Bolt-Small and its ResRAF correction. The neighborhood size K is selected per task on validation data; in per-horizon rows, bold marks corrected values that improve over the matched frozen base, and in average rows, bold marks the best value within each backbone group.

C Hyperparameter Sensitivity

Neighborhood size K. We evaluate the sensitivity of ResRAF to the retrieval neighborhood size across all five frozen backbones (Figure 2c). The preferred neighborhood size varies across backbone architectures, and no single value of K dominates all settings. DLinear favors a moderate neighborhood, TimeMixer performs best with a smaller one, whereas OLinear continues to benefit from larger neighborhoods; iTransformer and RAFT are comparatively stable across intermediate and larger values. At the aggregate level, the mean error is nearly flat for $K \in \{64, 96, 160\}$ and rises at K=320, while larger neighborhoods yield fewer degraded tasks (11/120 degraded at K=64 versus 5/120 at K=320). This heterogeneity motivates selecting K on validation data rather than fixing a single neighborhood size across all tasks. Table 8 breaks this down by backbone and dataset: larger neighborhoods win more rows (13 of 30 backbone–dataset pairs for K=320 versus 6 for K=64), confirming that the preferred K is genuinely task-dependent.

Number of prediction states R. On DLinear with $K = 64$, aggregate forecasting performance changes only marginally over a broad range of state counts. This suggests that ResRAF is not sensitive to the exact choice of R, supporting the use of a shared value $R = 24$ across tasks.

Table 7: Full per-horizon paired MAE results (144 tasks). Bold marks corrected values that improve over the matched frozen base (per-horizon rows) or the best value within each backbone group (average rows). MSE results are provided in Table 6.

		DLinear				iTransformer				OLinear				RAFT				TimeMixer				Chronos-Bolt	
Dataset	H	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF	+UEC	+PIR	Original	+ResRAF
ETTh1	96	0.411	**0.408**	**0.409**	**0.401**	0.409	**0.406**	**0.409**	0.417	0.383	0.384	0.383	0.402	0.402	**0.400**	0.407	**0.397**	0.398	**0.398**	0.399	**0.397**	0.382	0.385
	192	0.441	**0.437**	0.447	**0.436**	0.435	**0.431**	**0.434**	**0.432**	0.415	**0.415**	0.421	0.433	0.419	0.420	0.430	0.419	0.431	**0.430**	0.440	**0.429**	0.414	**0.412**
	336	0.474	**0.462**	0.479	**0.471**	0.459	**0.454**	0.464	**0.449**	0.439	**0.434**	0.443	0.459	0.443	0.445	0.484	0.445	0.449	**0.443**	0.461	0.452	0.438	**0.435**
	720	0.517	**0.494**	0.602	0.524	0.479	**0.477**	0.502	0.499	0.466	**0.460**	0.527	0.489	0.480	0.482	0.489	0.486	0.467	**0.467**	0.524	0.478	0.462	0.465
	Avg.	0.461	**0.450**	0.484	0.458	0.446	**0.442**	0.452	0.450	0.426	**0.423**	0.443	0.446	**0.436**	0.437	0.453	0.437	0.436	**0.434**	0.456	0.439	**0.424**	**0.424**
ETTh2	96	0.398	**0.385**	**0.397**	0.407	0.351	**0.344**	**0.351**	**0.346**	0.330	0.330	0.330	0.340	0.344	**0.339**	0.344	0.346	0.343	**0.340**	**0.343**	0.343	0.324	0.326
	192	0.479	**0.465**	**0.479**	**0.471**	0.398	**0.394**	**0.398**	**0.397**	0.381	0.383	0.381	0.393	0.397	**0.386**	0.397	0.411	0.396	0.397	0.397	**0.396**	0.370	0.376
	336	0.542	**0.532**	0.543	**0.522**	0.432	**0.426**	0.432	**0.430**	0.413	0.414	0.413	0.421	0.425	**0.411**	**0.425**	0.433	0.443	**0.438**	0.443	**0.439**	0.404	0.413
	720	0.662	**0.645**	**0.657**	**0.622**	0.446	0.451	**0.445**	**0.441**	0.430	0.445	**0.429**	0.435	0.470	**0.455**	**0.467**	0.477	0.448	0.458	**0.448**	0.450	0.430	0.443
	Avg.	0.520	0.507	0.519	**0.506**	0.407	**0.404**	0.406	**0.404**	**0.388**	0.393	**0.388**	0.397	0.409	**0.398**	0.408	0.417	0.408	0.408	0.408	**0.407**	**0.382**	0.389
ETTm1	96	0.372	**0.348**	**0.365**	**0.355**	0.368	**0.363**	**0.364**	0.371	0.334	**0.332**	**0.334**	0.356	0.351	**0.344**	**0.351**	**0.348**	0.361	**0.359**	**0.358**	**0.355**	0.332	**0.326**
	192	0.390	**0.376**	**0.390**	**0.377**	0.397	**0.391**	**0.395**	0.406	0.363	0.364	**0.363**	0.382	0.365	0.365	**0.364**	0.366	0.387	**0.384**	**0.386**	**0.381**	0.365	**0.360**
	336	0.412	**0.406**	0.412	**0.400**	0.431	0.434	**0.429**	**0.428**	0.387	0.391	**0.387**	0.411	0.383	0.384	**0.382**	0.395	0.404	0.408	**0.404**	**0.403**	0.395	**0.387**
	720	0.454	**0.446**	**0.445**	**0.443**	0.461	**0.460**	**0.453**	0.463	0.425	0.426	**0.423**	0.454	0.412	**0.411**	0.415	0.427	0.437	0.442	0.438	0.438	0.444	**0.432**
	Avg.	0.407	**0.394**	0.403	**0.394**	0.415	0.412	**0.410**	0.417	**0.377**	0.378	**0.377**	0.400	0.378	**0.376**	0.378	0.384	0.397	0.398	0.396	**0.394**	0.384	**0.376**
ETTm2	96	0.293	**0.265**	**0.288**	**0.287**	0.264	**0.262**	**0.264**	0.266	0.249	**0.247**	**0.249**	0.257	0.256	**0.255**	**0.256**	0.258	0.259	**0.255**	0.259	**0.258**	0.244	0.244
	192	0.364	**0.334**	**0.364**	0.369	0.306	**0.305**	**0.306**	0.309	0.290	**0.290**	**0.290**	0.300	0.296	**0.295**	**0.296**	0.299	0.301	0.301	**0.301**	**0.299**	0.290	0.293
	336	0.411	**0.388**	**0.411**	0.460	0.347	**0.344**	**0.347**	0.353	0.328	**0.328**	**0.328**	0.338	0.335	**0.333**	0.335	0.349	0.344	**0.339**	**0.344**	0.344	0.333	0.336
	720	0.526	**0.509**	**0.525**	0.551	0.407	0.411	**0.407**	0.413	0.387	0.390	**0.387**	0.415	0.395	**0.391**	**0.395**	0.408	0.403	0.410	**0.403**	0.413	0.397	0.402
	Avg.	0.399	**0.374**	0.397	0.417	0.331	**0.330**	0.331	0.335	**0.314**	**0.314**	**0.314**	0.327	0.320	**0.319**	0.320	0.329	0.327	**0.326**	0.327	0.328	**0.316**	0.319
Exchange	96	0.227	**0.216**	0.227	**0.222**	0.206	**0.203**	0.207	0.239	0.199	**0.199**	0.200	0.210	0.208	**0.204**	0.208	0.221	0.209	**0.204**	0.209	**0.205**	0.207	**0.198**
	192	0.324	**0.319**	0.324	0.339	0.299	**0.294**	0.299	0.325	0.293	**0.289**	0.293	0.316	0.317	**0.291**	**0.317**	0.329	0.310	**0.303**	0.311	0.324	0.295	**0.281**
	336	0.435	**0.420**	0.437	0.463	0.413	**0.403**	0.416	0.448	0.408	**0.384**	**0.408**	0.430	0.447	**0.425**	**0.447**	0.488	0.441	**0.394**	0.446	**0.436**	0.403	**0.383**
	720	0.670	**0.607**	**0.670**	0.673	0.718	**0.641**	0.719	0.751	0.688	**0.625**	0.689	0.715	0.784	**0.746**	0.786	**0.764**	0.739	**0.622**	0.739	0.745	0.655	**0.623**
	Avg.	0.414	**0.390**	0.414	0.424	0.409	**0.385**	0.410	0.441	0.397	**0.374**	0.397	0.418	0.439	**0.416**	0.439	0.450	0.425	**0.381**	0.426	0.428	0.390	**0.371**
Weather	96	0.256	**0.212**	**0.251**	**0.248**	0.218	**0.204**	**0.218**	**0.213**	0.188	**0.187**	0.188	0.196	0.225	**0.200**	**0.224**	0.225	0.209	**0.206**	**0.209**	**0.207**	0.206	**0.189**
	192	0.296	**0.251**	**0.281**	**0.286**	0.257	**0.249**	**0.257**	**0.253**	0.238	**0.234**	**0.237**	0.244	0.264	**0.250**	0.264	**0.262**	0.252	**0.250**	**0.251**	**0.252**	0.247	**0.234**
	336	0.331	**0.287**	**0.329**	0.333	0.298	**0.288**	**0.296**	**0.297**	0.280	**0.277**	**0.279**	0.287	0.301	**0.287**	0.303	**0.300**	0.291	**0.288**	**0.290**	**0.291**	0.287	**0.276**
	720	0.384	**0.344**	**0.382**	0.391	0.351	**0.344**	**0.349**	0.354	0.334	**0.330**	**0.330**	0.345	0.353	**0.347**	**0.352**	0.355	0.345	**0.340**	**0.341**	0.349	0.337	**0.327**
	Avg.	0.317	**0.273**	0.311	0.315	0.281	**0.271**	0.280	0.279	0.260	**0.257**	0.259	0.268	0.286	**0.271**	0.286	0.286	0.274	**0.271**	0.273	0.275	0.269	**0.257**

Table 8: Sensitivity to the Top-K neighborhood size by backbone and dataset. Entries are horizon-averaged MSE; bold marks the best K per row.

Backbone	Dataset	Base	$K = 64$	$K = 96$	$K = 160$	$K = 320$
DLinear	ETTh1	0.465	**0.446**	**0.446**	0.447	0.449
	ETTh2	0.566	**0.554**	0.555	0.556	0.559
	ETTm1	0.404	0.382	**0.381**	0.382	0.382
	ETTm2	0.349	**0.329**	0.331	0.332	0.335
	Exchange	0.346	**0.312**	**0.312**	**0.312**	0.316
	Weather	0.265	0.236	0.236	**0.235**	0.236
iTransformer	ETTh1	0.451	0.446	0.445	0.445	**0.444**
	ETTh2	0.382	**0.374**	**0.374**	**0.374**	**0.374**
	ETTm1	0.416	**0.407**	**0.407**	**0.407**	**0.407**
	ETTm2	0.289	0.283	0.283	**0.282**	**0.282**
	Exchange	0.368	**0.307**	**0.307**	0.309	0.314
	Weather	0.261	0.246	**0.245**	**0.245**	**0.245**
OLinear	ETTh1	0.426	0.422	0.421	0.420	**0.419**
	ETTh2	0.367	**0.368**	**0.368**	**0.368**	**0.368**
	ETTm1	0.374	**0.369**	**0.369**	**0.369**	0.370
	ETTm2	0.271	0.269	0.269	**0.268**	**0.268**
	Exchange	0.353	**0.287**	0.288	0.293	0.301
	Weather	0.238	0.232	**0.231**	**0.231**	**0.231**
RAFT	ETTh1	0.421	0.423	0.423	0.422	**0.420**
	ETTh2	0.362	**0.349**	**0.349**	**0.349**	0.350
	ETTm1	0.348	**0.345**	**0.345**	**0.345**	**0.345**
	ETTm2	0.256	0.255	0.255	0.255	**0.254**
	Exchange	0.441	0.396	**0.394**	**0.394**	0.397
	Weather	0.240	0.228	**0.227**	**0.227**	**0.227**
TimeMixer	ETTh1	0.444	**0.437**	**0.437**	**0.437**	**0.437**
	ETTh2	0.387	0.384	0.383	**0.382**	**0.382**
	ETTm1	0.382	0.383	0.383	0.383	**0.382**
	ETTm2	0.281	0.278	0.278	**0.277**	**0.277**
	Exchange	0.405	**0.291**	0.293	0.292	0.305
	Weather	0.244	0.239	0.239	0.239	**0.238**

D IMPLEMENTATION DETAILS

This appendix specifies how ResRAF constructs a residual memory, retrieves complete historical residuals for a new input, and turns them into a calibrated forecast. Sections D.1–D.6 follow the information used by a single prediction. Section D.7 then gives the fixed-component inference procedure and the fitting and selection stages. The residual pattern transformation, knowledge base, residual representation predictor, state assignment function, and calibration coefficients are constructed separately for each fixed forecasting model and task. All preprocessing statistics used at validation and test time are fitted on the training split.

D.1 RESIDUAL PATTERN TRANSFORMATION AND KNOWLEDGE BASE CONSTRUCTION

Historical instances. For each training pair $(\mathbf{X}_i, \mathbf{Y}_i)$, the frozen forecaster produces $\widehat{\mathbf{Y}}_i^0 = f_\theta(\mathbf{X}_i)$ from the full multivariate input, with residual $\mathbf{E}_i = \mathbf{Y}_i - \widehat{\mathbf{Y}}_i^0$. Each channel gives one of $M = NC$ historical instances: for $j = (i-1)C + c$, $\mathbf{x}_j = \mathbf{X}_i[:, c] \in \mathbb{R}^L$, $\widehat{\mathbf{y}}_j^0 = \widehat{\mathbf{Y}}_i^0[:, c] \in \mathbb{R}^H$, $\mathbf{e}_j = \mathbf{E}_i[:, c] = \mathbf{e}_{i,c} \in \mathbb{R}^H$, and $c_j = c$. Thus, entries are indexed by channel while their base forecasts retain multivariate context.

Residual coordinates. With $e_{i,h,c} = [\mathbf{E}_i]_{h,c}$, we fit the training root-mean-square scale $\sigma_c = (\sum_{i,h} e_{i,h,c}^2/(NH))^{1/2}$ and use $\widetilde{\mathbf{e}}_j = \mathbf{e}_j/\sigma_{c_j}$ for pattern construction. The knowledge base retains the complete residual $\mathbf{e}_j$ in original target units for correction. We stack the scaled residuals as rows of $\widetilde{E} \in \mathbb{R}^{M \times H}$ and fit an uncentered randomized truncated SVD with $d = 16$ and seven iterations. Its retained right singular vectors form $V_d \in \mathbb{R}^{d \times H}$, giving projected coordinates $\mathbf{a}_j = V_d \widetilde{\mathbf{e}}_j$. Let $\bar{\mathbf{a}} = M^{-1} \sum_j \mathbf{a}_j$ and $D_s = \mathrm{diag}(s_1, \ldots, s_d)$, where s_ℓ is the training standard deviation of coordinate ℓ. The transformation in Equation 1 is

$$\phi(\mathbf{e}, c) = D_s^{-1} \left(V_d \frac{\mathbf{e}}{\sigma_c} - \bar{\mathbf{a}} \right). \tag{14}$$

Positive scaling factors are assumed. Training residuals determine all quantities in ϕ, which remain fixed for predictor training, validation, and testing.

Knowledge-base contents. Each entry is $(\mathbf{x}_j, \widehat{\mathbf{y}}_j^0, \mathbf{e}_j, \mathbf{z}_j, c_j)$, with $\mathbf{z}_j = \phi(\mathbf{e}_j, c_j)$ (Equation 2). We fit ϕ and construct the knowledge base separately for each checkpoint, dataset, and horizon. Historical states are derived from stored inputs and base forecasts (Appendix D.5); window positions are auxiliary retrieval metadata (Appendix D.4).

D.2 RESIDUAL REPRESENTATION PREDICTOR

Network architecture. For channel c of the current input $\mathbf{X}_i$, separate training-fitted transformations Z_x and Z_y standardize its input sequence $\mathbf{x}_{i,c}$ and base forecast $\widehat{\mathbf{y}}_{i,c}^0$. Each branch receives value and first-difference tokens, $\mathrm{tok}(\mathbf{a}) = [\mathbf{a}, \Delta\mathbf{a}]$. Independent history and forecast branches F_x and F_y are shared across channels. Each has three one-dimensional convolutions with widths 16/32/64, kernels 9/7/5, and stride 2, each followed by group normalization and GELU; adaptive average pooling yields eight temporal positions. A two-layer MLP with hidden width 128 maps the concatenated branch outputs and a 12-dimensional channel embedding $\mathbf{h}_c$ to $\widehat{\mathbf{z}}_{i,c} \in \mathbb{R}^d$. These components comprise $f^r(\mathbf{x}_{i,c}, \widehat{\mathbf{y}}_{i,c}^0, \mathbf{h}_c; \psi)$ in Equation 3. The channel embedding conditions the predictor but is excluded from residual-representation distances.

Learning objective. The target $\mathbf{z}_j = \phi(\mathbf{e}_j, c_j)$ is fixed while ψ is trained with coordinate-wise mean MSE plus 0.1 SmoothL1:

$$\ell_{\mathrm{align}}(\widehat{\mathbf{z}}, \mathbf{z}) = \frac{1}{d} \sum_{\ell=1}^{d} \left[(\widehat{z}_\ell - z_\ell)^2 + 0.1 \mathrm{SL1}_\beta(\widehat{z}_\ell - z_\ell) \right],$$

where $\mathrm{SL1}_\beta(u) = u^2/(2\beta)$ for $|u| < \beta$ and $|u| - \beta/2$ otherwise.

D.3 Similarity Scores and Joint Retrieval

Prediction similarity. The prediction term in Equation 4 compares channel c of the current input $\mathbf{X}_i$ with the base forecast stored for same-channel entry j. Divide the horizon into $B_H = \lceil H/96 \rceil$ contiguous segments $I_b = \{(b-1)96 + 1, \ldots, \min(96b, H)\}$. For a sequence $\mathbf{a}$, let $\mathbf{n}_b(\mathbf{a}) = (\mathbf{a}_{I_b} - \overline{a}_{I_b}\mathbf{1})/(\|\mathbf{a}_{I_b} - \overline{a}_{I_b}\mathbf{1}\|_2 + 10^{-6})$, where $\overline{a}_{I_b}$ is its segment mean. Then

$$s_{i,j}^{\mathrm{pred},c} = \frac{1}{B_H} \sum_{b=1}^{B_H} \mathbf{n}_b(\widehat{\mathbf{y}}_{i,c}^0)^\top \mathbf{n}_b(\widehat{\mathbf{y}}_j^0). \tag{15}$$

Segments contribute equally, including a shorter final segment. For example, $H = 336$ yields lengths $(96, 96, 96, 48)$; $H = 720$ yields seven length-96 segments and a final length-48 segment. Concatenating $B_H^{-1/2}[\mathbf{n}_1(\mathbf{a}); \ldots; \mathbf{n}_{B_H}(\mathbf{a})]$ gives an equivalent H-dimensional inner-product representation without learned parameters.

Residual similarity and recency. We use $d = 16$ and $\tau = 1.25$ in Equation 5. Expanding its squared distance gives $s_{i,j}^{\mathrm{res},c} = (2\widehat{\mathbf{z}}_{i,c}^\top \mathbf{z}_j - \|\mathbf{z}_j\|_2^2 - \|\widehat{\mathbf{z}}_{i,c}\|_2^2)/(\tau d)$. The query-only final term can be omitted for ranking and softmax normalization. The fixed recency bonus is $b_j = 0.01\mathbf{1}[j \in \mathcal{R}_{\mathrm{tr}}]$, where $\mathcal{R}_{\mathrm{tr}}$ contains entries from the last 30% of the training time axis, determined by window position.

D.4 Candidate Eligibility and Retrieval Weights

Candidate eligibility. Let $\mathcal{J}_c = \{j : c_j = c\}$ and let t_j be entry j's training-window position in original time steps; all channels from one window share it. The query position $t(\mathbf{X}_i)$ uses the same reference point. Retrieval uses only same-channel entries and, during training, removes overlapping input–forecast windows:

$$\mathcal{A}_c(i) = \begin{cases} \{j \in \mathcal{J}_c : |t_j - t(\mathbf{X}_i)| \geq L + H\}, & \text{training query,} \\ \mathcal{J}_c, & \text{validation or test query.} \end{cases} \tag{16}$$

The mask uses original time steps, so cached window indices must account for extraction stride. It may retain later nonoverlapping training windows: this is offline retrieval, not past-only replay, and training residuals are not out-of-sample. Validation and test targets never enter the fixed training knowledge base.

Neighborhood weights. Selected entries supply complete residuals $\mathbf{e}_j$ and weights $w_{i,j}^c = \mathrm{softmax}_{j \in \mathcal{N}_K^c(i)}(s_{i,j}^c / T_w)$ with $T_w = 10^6$. We subtract the neighborhood's maximum score before exponentiation for numerical stability. Selected weights sum to one; other weights are zero. Large T_w makes them nearly uniform when score differences are small, but the exact weights are retained. K cannot exceed the eligible same-channel count for a query.

D.5 State Encoder

State features. Let $\mathbf{a}_y = \widehat{\mathbf{y}}_{i,c}^0$ and let $\mathbf{a}_x$ contain the most recent $\min(L, 192)$ input observations. From each sequence we extract pooled shape features P_{24}, summary statistics Q_7, and spectral features F_8:

$$\begin{aligned} \mathbf{r}_{i,c} = [&P_{24}(\mathbf{a}_y); Q_7(\mathbf{a}_y); F_8(\mathbf{a}_y); \\ &P_{24}(\mathbf{a}_x); Q_7(\mathbf{a}_x); F_8(\mathbf{a}_x)] \in \mathbb{R}^{78}. \end{aligned} \tag{17}$$

The operator P_{24} divides a sequence into 24 pooled values with within-sequence standardization. The statistical features are

$$\begin{aligned} Q_7(\mathbf{a}) = [&\,\mathrm{mean}(\mathbf{a}), \mathrm{std}(\mathbf{a}), a_1, a_n, \\ &\mathrm{trend}(\mathbf{a}), \mathrm{quantile}_{.25}(\mathbf{a}), \mathrm{quantile}_{.75}(\mathbf{a})], \end{aligned} \tag{18}$$

where n is the sequence length and trend measures its linear trend. For the spectral features, let $A_k(\mathbf{a}) = |\,\mathrm{FFT}(\mathbf{a} - \overline{a}\mathbf{1})_k|$. Denoting the normalization-frequency set by Ω and a numerical stabilizer by ϵ_F, the first eight non-DC amplitudes are transformed as

$$[F_8(\mathbf{a})]_k = \log\left(1 + 50 \frac{A_k(\mathbf{a})}{\sum_{\ell \in \Omega} A_\ell(\mathbf{a}) + \epsilon_F}\right), \qquad k = 1, \ldots, 8. \tag{19}$$

Prediction state partition. We standardize the 78-dimensional features using training statistics and fit $R = 24$ centers with K-means. Each window–channel pair is assigned to its nearest frozen center:

$$\begin{aligned} \widetilde{\mathbf{r}}_{i,c} &= Z_r(\mathbf{r}_{i,c}), \\ \{\boldsymbol{\kappa}_r\}_{r=1}^{R} &= \mathrm{KMeans}_R(\{\widetilde{\mathbf{r}}_j\}_{j\in\mathrm{Train}}), \\ \phi^s(\mathbf{x}_{i,c}, \widehat{\mathbf{y}}_{i,c}^0) &= \operatorname*{arg\,min}_{r\in\{1,\dots,R\}} \|\widetilde{\mathbf{r}}_{i,c} - \boldsymbol{\kappa}_r\|_2^2. \end{aligned} \tag{20}$$

Using the fixed centers, ϕ^s assigns states $g_{i,c}$ and g_j to current and historical instances from their respective inputs and base forecasts. These states describe observable characteristics, whereas $\mathbf{z}_j$ represents a historical residual pattern.

D.6 Calibration Computation and Estimation

State-grouped transfer. For channel c of current input $\mathbf{X}_i$, $g_{i,c}$ selects a correction strength $\alpha^c_{g_{i,c}}$ shared across the neighborhood, and each historical state g_j selects an adjustment $\delta^c_{g_j}$. Their sum $\gamma^c_{i,j} = \alpha^c_{g_{i,c}} + \delta^c_{g_j}$ scales the complete residual $\mathbf{e}_j$. Defining

$$\begin{aligned} \mathbf{m}_{i,r,c} &= \sum_{j\in\mathcal{N}_K^c(i)} w^c_{i,j}\mathbf{1}[g_j = r]\mathbf{e}_j, \qquad \mathbf{m}_{i,c} = \sum_{r=1}^{R} \mathbf{m}_{i,r,c}, \\ \widehat{\mathbf{e}}_{i,c} &= \alpha^c_{g_{i,c}}\mathbf{m}_{i,c} + \sum_{r=1}^{R} \delta^c_r \mathbf{m}_{i,r,c}, \end{aligned} \tag{21}$$

gives Equation 10. Weights remain normalized over the entire selected neighborhood, without renormalization within state groups.

Prior and ridge fit. Using temporally masked training retrieval, we obtain a channel-wise prior $\alpha_c^{(0)} = \mathrm{clip}(\sum_i \langle \mathbf{m}_{i,c}, \mathbf{e}_{i,c}\rangle / (\sum_i \|\mathbf{m}_{i,c}\|_2^2 + 10^{-8}), 0, 0.8)$. It anchors current-state strengths, while historical-state adjustments are shrunk toward zero. The unconstrained fit minimizes

$$\sum_{i=1}^{N} \left\| \mathbf{e}_{i,c} - \alpha^c_{g_{i,c}}\mathbf{m}_{i,c} - \sum_{r=1}^{R} \delta^c_r \mathbf{m}_{i,r,c} \right\|_2^2 + \lambda s_c \left[\sum_{g=1}^{R} (\alpha^c_g - \alpha_c^{(0)})^2 + \sum_{r=1}^{R} (\delta^c_r)^2 \right]. \tag{22}$$

The data-fitting term equals corrected forecast squared error in Equation 13; the training sum is not averaged.

For each training pair and lead time (i, h), let row ℓ of $D_c \in \mathbb{R}^{NH\times 2R}$ have $[D_c]_{\ell,g} = \mathbf{1}[g_{i,c} = g]m_{i,h,c}$ and $[D_c]_{\ell,R+r} = m_{i,r,h,c}$, and set $[\mathbf{b}_c]_\ell = e_{i,h,c}$. Let $s_c = \mathrm{tr}(D_c^\top D_c)/(2R)$ and $\boldsymbol{\beta}_c^{(0)} = [\alpha_c^{(0)}\mathbf{1}_R; \mathbf{0}_R]$. We solve and then clip:

$$\begin{aligned} \widetilde{\boldsymbol{\beta}}_c &= \mathrm{solve}\Big(D_c^\top D_c + \lambda s_c I, D_c^\top \mathbf{b}_c + \lambda s_c \boldsymbol{\beta}_c^{(0)} \Big), \\ \alpha_g^{c,\star} &= \mathrm{clip}(\widetilde{\beta}_{c,g}, 0, 0.8), \qquad \delta_r^{c,\star} = \mathrm{clip}(\widetilde{\beta}_{c,R+r}, -0.1, 0.1). \end{aligned} \tag{23}$$

With $R = 24$, this is a 48-dimensional system; its Gram matrix can be accumulated without storing D_c. This procedure is ridge estimation followed by coordinate-wise clipping, rather than a box-constrained least-squares solution. The resulting transfer coefficient can be negative and lies in $[-0.1, 0.9]$.

D.7 Inference, Training, and Selection

Inference. The frozen forecaster first produces $\widehat{\mathbf{Y}}_i^0 = f_\theta(\mathbf{X}_i)$. For each channel, the selected f^r predicts $\widehat{\mathbf{z}}_{i,c}$, RAR retrieves same-channel entries and their complete residuals, and ϕ^s assigns the current prediction state; historical states may be cached. Fixed SARC coefficients produce the correction in Equation 10, which is added to the base forecast. The knowledge base, transformations, predictor, state partition, coefficients, and selected K remain fixed. Inference does not use the current future observations or residuals.

Table 9: Fitting stages and data use. Validation and test targets never enter the knowledge base.

Stage	Fitted or retained objects	Data role
Knowledge construction	ϕ, ϕ^s, $\mathcal{C}_\theta$	Training residuals determine the residual pattern transformation; training inputs and base forecasts determine the state partition.
Representation learning	Parameters ψ of f^r	Training residual representations supervise learning; validation representation MSE selects the epoch.
Calibration fitting	Prior and $2R$ coefficients per channel for each (K, λ)	Temporally masked training retrieval supports fitting; validation forecast MSE selects the fitted configuration.
Inference and evaluation	All selected components fixed	Test inputs produce forecasts; test targets are used only for evaluation.

Table 10: Structural and numerical settings for residual retrieval and calibration. Representation-predictor optimization and validation selection are specified in Appendix D.7.

Component	Setting
Residual representations	$d = 16$; seven randomized-SVD iterations
Representation branches	Conv widths 16/32/64; kernels 9/7/5; stride 2; eight pooled positions
Representation projection	Channel embedding 12; MLP hidden width 128
Joint score	$\tau = 1.25$; horizon segments of at most 96 steps
Candidate scope and recency	Same-channel entries only; increment 0.01 for the final 30% of training entries
Retrieval weights	Softmax temperature 10^6 within the selected neighborhood
Observable states	78 features; history tail of at most 192 steps; $R = 24$
Calibration	Current-state strengths in $[0, 0.8]$; historical-state adjustments in $[-0.1, 0.1]$

Fitting sequence. The base forecaster remains frozen. We fit ϕ from training residuals and ϕ^s from training inputs and base forecasts, then construct $\mathcal{C}_\theta$ with $(\mathbf{x}_j, \widehat{\mathbf{y}}_j^0, \mathbf{e}_j, \mathbf{z}_j, c_j)$. Historical states are derived separately from stored inputs and forecasts. The trainable parameter set ψ comprises the two encoder branches, channel embeddings, and output MLP of f^r. Its target is the fixed historical residual representation $\mathbf{z}_j = \phi(\mathbf{e}_j, c_j)$, and validation representation MSE selects its epoch. We freeze the selected predictor before estimating calibration coefficients. For each candidate neighborhood size K, temporally masked training retrieval determines the global scaling prior and fits the $2R$ correction strengths and historical-state adjustments per channel for every candidate λ. Validation forecast MSE selects among these fitted configurations; test targets are used only to evaluate the final selected method.

Training configuration. The residual representation predictor uses AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 256, and gradient-norm clipping at 1.0. Each epoch samples at most 30,000 training entries, for at most 15 epochs. Early stopping monitors validation representation MSE, with patience 3 and a minimum improvement of 5×10^{-5}. These optimization settings apply only to ψ. The residual pattern transformation and state partition are fitted separately; calibration uses the linear system in Equation 23.

Validation selection. We search $K \in \{64, 96, 160, 320\}$, subject to eligible same-channel candidate counts, and $\lambda \in \{0.5, 1, 2, 5, 10\}$. For ridge selection, configurations within 0.3% of the best validation MSE are considered near-optimal; preference is given to regularization closer to 1 and smaller historical-state adjustments. Across neighborhood sizes, we select the smaller K when its validation MSE is within 0.1% of the best value. Table 10 summarizes the fixed structural settings.