LEARNING FROM FORECASTING ERRORS: SEPARATING CORRECTABLE ERRORS FROM PREDICTIVE UNCERTAINTY IN TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing time series forecasting methods typically treat forecasting errors as optimization residuals to be minimized, while rarely exploiting the information contained in forecasting failures themselves. However, forecasting errors differ substantially in correctability: some exhibit stable structures that can be further corrected, some recur under similar historical conditions and can be reused as historical evidence, whereas others lack reliable correction evidence and are better represented as predictive uncertainty. We propose EUD-RM (Error–Uncertainty Decoupling with Residual Memory), an error-centric framework that handles forecasting failures through a reduce–correct–reuse–quantify process. A Channel-Aware Mixer (CAM) first strengthens cross-variable interactions to reduce potential systematic errors. Learnability-Gated Error Correction (LGEC) then identifies prediction positions with reliable correction evidence and applies bounded parametric corrections. Residual-Aware Retrieval Memory (RARM) stores historical forecasting residuals and retrieves consistent past failures under similar input and prediction states as non-parametric correction evidence. Finally, an independent LGEC-U branch models the remaining predictive risk after deterministic corrections and produces calibrated prediction intervals without altering point forecasts. Rather than assuming a strict decomposition into deterministic error and random noise, EUD-RM exploits empirical correctability and historical reproducibility to determine how forecasting failures should be handled. Experiments on long-horizon multivariate forecasting benchmarks demonstrate consistent gains, with EUD-RM achieving leading results on most benchmark settings.

1 INTRODUCTION

Recent advances in Transformer-based forecasting (Wu et al., 2021; Nie et al., 2022), linear and MLP-based architectures (Zeng et al., 2023; Chen et al., 2023), time-frequency modeling (Zhou et al., 2022b), and multi-scale modeling (Wang et al., 2024) have substantially improved time series forecasting. Multivariate methods have further explored explicit cross-variable dependency modeling (Zhang & Yan, 2023; Liu et al., 2024), while hybrid and residual-correction systems have long exploited forecasting residuals to model structures left unexplained by a primary predictor (Zhang, 2003; Júnior et al., 2023). Nevertheless, error handling remains secondary in many modern forecasting architectures: prediction errors are primarily minimized through training objectives, although some failures exhibit persistent structure or recur under similar historical conditions, suggesting regularities left unmodeled by the predictor.

As illustrated in Figure 1, forecasting residuals differ in structure, repeatability, and cross-variable dependency. Some forecasting failures therefore contain exploitable patterns rather than serving solely as optimization residuals. Treating all errors uniformly may leave systematic biases uncorrected or encourage overfitting to unstable fluctuations. For example, in electricity demand forecasting, prediction errors under similar weather conditions or seasonal periods may repeatedly exhibit similar patterns, indicating that some failures can be corrected using historical error evidence. In contrast, sudden demand fluctuations caused by unexpected events may remain inherently difficult to predict and are more naturally represented through predictive uncertainty, as considered in probabilistic and interval forecasting methods (Salinas et al., 2020; Romano et al., 2019; Xu & Xie,

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.

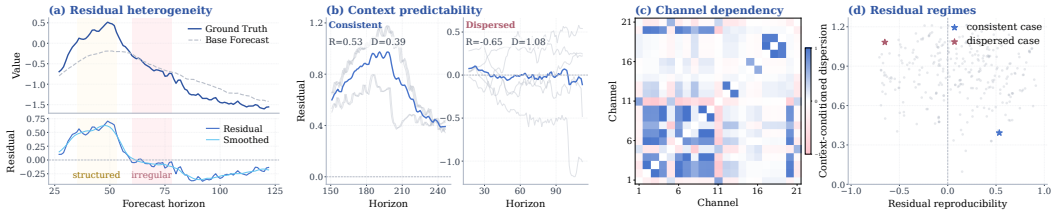

Figure 1: Empirical heterogeneity of forecasting residuals on Weather, showing diverse residual structures, context-dependent behaviors, and cross-variable dependencies.

2021). Therefore, an important question arises: which forecasting errors contain reliable correction signals, which provide reusable historical evidence under similar forecasting conditions, and which remaining errors should instead be represented as predictive uncertainty?

To address this question, we propose EUD-RM (Error–Uncertainty Decoupling with Residual Memory), a unified framework that handles forecasting failures through a progressive reduce–correct–reuse–quantify process. The reduce stage improves forecasting representations before prediction errors are generated, while subsequent stages handle failures according to empirical correctability, historical reproducibility, and remaining predictive risk. Rather than treating all errors as homogeneous residuals, EUD-RM progressively separates failures that admit reliable parametric correction, provide reusable historical evidence, or are better represented as predictive uncertainty.

Specifically, EUD-RM first introduces CAM into the base forecaster to strengthen cross-variable representations and reduce potential systematic biases in the initial prediction. LGEC then identifies prediction positions with higher empirical correction utility and applies bounded parametric corrections. For residual failures that persist after parametric correction but recur under similar forecasting states, RARM retrieves the model's own historical residuals and reuses them as non-parametric correction evidence. Finally, LGEC-U quantifies the remaining predictive risk after these correction stages without modifying the final point forecast. The resulting pipeline is Base/CAM $\rightarrow$ LGEC $\rightarrow$ RARM $\rightarrow$ LGEC-U.

Our main contributions are:

1. **An error-centric forecasting perspective.** We characterize forecasting failures by empirical correctability and historical reproducibility, distinguishing structurally correctable errors, historically reusable failures, and remaining predictive risk.
2. **A unified forecasting failure-handling framework.** EUD-RM integrates pre-forecast representation enhancement, selective parametric correction, residual-memory reuse, and post-hoc risk quantification; notably, RARM explicitly reuses forecasting residuals produced by the model itself.
3. **Mechanism-oriented empirical evaluation.** We examine whether LGEC gates predict realized correction utility, whether RARM retrieval evidence supports effective correction, and whether LGEC-U scales characterize the remaining predictive risk.

2 Related Work

Time series forecasting and multivariate dependency modeling. Long-horizon forecasting has been extensively studied with Transformer-based architectures, including Informer, Autoformer, FEDformer, Pyraformer, Non-stationary Transformer, and PatchTST (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022b; Liu et al., 2021; 2022; Nie et al., 2022). Competitive alternatives include linear and MLP-based models (Zeng et al., 2023; Chen et al., 2023; Das et al., 2023), frequency-aware and multi-scale representations (Zhou et al., 2022a; Yi et al., 2023; Wu et al., 2022; Wang et al., 2023; Challu et al., 2023; Wang et al., 2024; Murad et al., 2025), and convolutional or dynamical formulations (Luo & Wang, 2024; Liu et al., 2023). For multivariate forecasting, PatchTST adopts channel-independent modeling (Nie et al., 2022), whereas Crossformer, iTransformer, and CARD explicitly model cross-variable dependencies (Zhang & Yan, 2023; Liu et al., 2024; Zhou et al., 2024). EUD-RM also incorporates channel-aware modeling, but focuses instead on which

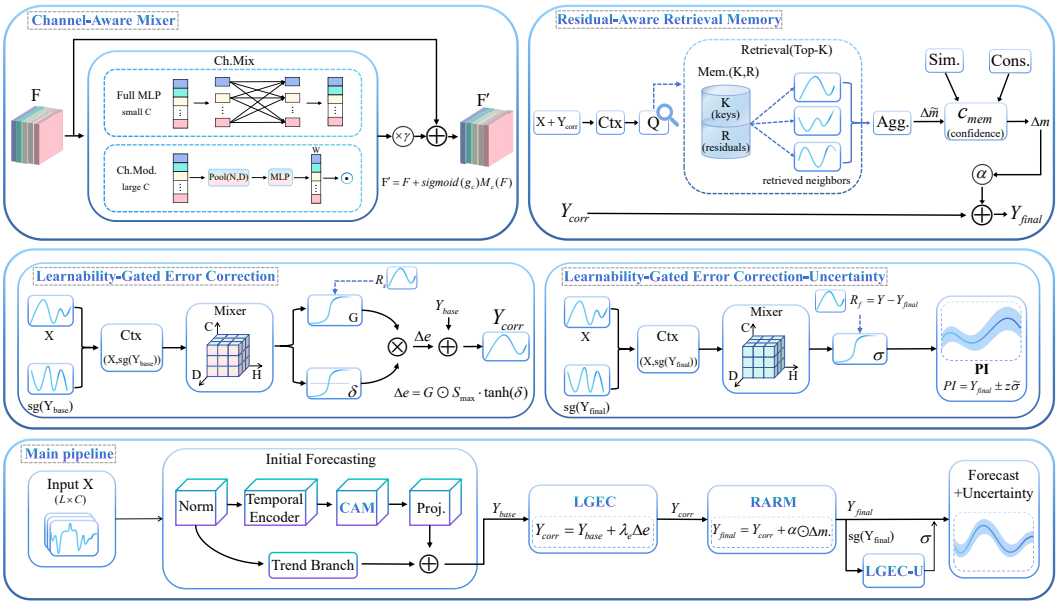

Figure 2: Overview of EUD-RM. CAM strengthens cross-variable representations before forecasting; LGEC performs gated parametric correction; RARM reuses consistent historical residuals as non-parametric correction evidence; and post-RARM LGEC-U quantifies remaining predictive risk. The bottom panel shows the complete pipeline.

forecasting failures remain reliably correctable or historically reusable after prediction. **Residual modeling and predictive uncertainty.** Forecasting residuals have long been used to improve predictions. Classical hybrid methods combine statistical and neural predictors to model structures unexplained by a primary model (Zhang, 2003; Khashei & Bijari, 2011), while recent methods construct secondary models for residual correction (Júnior et al., 2023). N-BEATS also employs iterative backcast residuals within its architecture (Oreshkin et al., 2019). However, these approaches generally do not distinguish failures by empirical correctability, historical reproducibility, and remaining predictive risk. Probabilistic methods such as DeepAR, MQ-RNN, and Temporal Fusion Transformer model predictive distributions or conditional quantiles (Salinas et al., 2020; Wen et al., 2017; Lim et al., 2021), while conformal methods construct calibrated prediction intervals from forecasting errors or conditional quantiles (Romano et al., 2019; Xu & Xie, 2021; Stankeviciute et al., 2021), including adaptive variants under distribution shift (Gibbs & Candes, 2021). EUD-RM connects these directions by separating failures supported by correction evidence, handled by LGEC and RARM, from the remaining predictive risk modeled by LGEC-U.

Retrieval-augmented forecasting and residual memory. Retrieval-augmented forecasting introduces non-parametric evidence from relevant historical instances. MQRetNN retrieves related entities for multi-horizon forecasting (Yang et al., 2022), while RAFT retrieves similar historical sequences and subsequent trajectories (Han et al., 2025); related retrieval-based forecasting methods have also been explored (Tire et al., 2024). EUD-RM differs primarily in what is stored and retrieved: RARM stores forecasting residuals previously produced by the model itself, rather than historical inputs, entities, or future trajectories. Residuals retrieved under similar input and prediction states serve as non-parametric correction evidence when sufficiently consistent and are attenuated otherwise. RARM thus extends retrieval augmentation to the model's own past forecasting failures, while LGEC-U separately models the remaining predictive risk.

3 METHOD

EUD-RM handles forecasting failures through a progressive reduce–correct–reuse–quantify pipeline. Figure 2 summarizes the architecture and information flow.

3.1 Problem Formulation and Overall Framework

Given a multivariate historical sequence $X \in \mathbb{R}^{B\times L\times C_{\text{in}}}$, where B, L, and C_{in} denote batch size, input length, and number of input variables, the model predicts $Y \in \mathbb{R}^{B\times H\times C_{\text{out}}}$ over the next H steps.

A base forecaster first produces $\hat{Y}^{(0)} = f_\theta(X)$, with CAM incorporated to strengthen cross-variable interactions inside the forecasting backbone. Unlike subsequent error-handling modules, CAM operates before the forecast is generated and does not rely on forecasting residuals or future observations. Details of the base forecaster and last-value residualization are provided in Appendix A.

LGEC applies a bounded parametric correction Δ_e:

$$\hat{Y}^{(1)} = \hat{Y}^{(0)} + \lambda_e \Delta_e, \tag{1}$$

where λ_e controls the correction strength and LGEC selectively activates corrections where reliable evidence is available.

RARM then retrieves historical forecasting residuals using keys that jointly encode the historical input and prediction state, yielding

$$\hat{Y}^{(2)} = \hat{Y}^{(1)} + \alpha \odot \Delta_m. \tag{2}$$

We denote $\hat{Y}_{\text{final}} \equiv \hat{Y}^{(2)}$. The coefficient α is calibrated on the validation set, while unreliable retrieval evidence is automatically attenuated.

Finally, LGEC-U estimates the remaining predictive risk from the stop-gradient final prediction:

$$\sigma = g_\phi\left(X, \text{sg}\left(\hat{Y}^{\text{final}}\right)\right), \tag{3}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. LGEC-U is isolated from the point-forecasting path and therefore does not alter the final point prediction.

EUD-RM uses post-RARM LGEC-U by default, so uncertainty corresponds to the actual final forecast. After validation-set scale calibration, with calibrated scale $\tilde{\sigma}$ and Gaussian quantile z for the target coverage, the prediction interval is $[\hat{Y}_{\text{final}} - z\tilde{\sigma}, \hat{Y}_{\text{final}} + z\tilde{\sigma}]$. The complete pipeline is Base/CAM $\rightarrow$ LGEC $\rightarrow$ RARM $\rightarrow$ LGEC-U.

3.2 Reducing Systematic Errors with Channel-Aware Mixing

Insufficient cross-variable interaction can leave channel-correlated biases in multivariate forecasts. We therefore insert a Channel-Aware Mixer (CAM) into each multi-scale branch after temporal and hidden-feature mixing. For $F \in R^{B\times C\times P\times D}$, CAM applies

$$F' = F + \text{sigmoid}(g_c) M_c(F), \tag{4}$$

where $M_c(\cdot)$ performs channel interaction and g_c is a learnable gate that introduces cross-variable information conservatively.

CAM uses explicit channel mixing for moderate-dimensional inputs and lightweight channel modulation for high-dimensional series. The mode is selected automatically according to variable dimensionality; full implementations are provided in Appendix B.

3.3 Learning When to Correct: Learnability-Gated Error Correction

Uniform residual compensation may overfit unstable forecasting errors. We therefore introduce Learnability-Gated Error Correction (LGEC), which selectively applies bounded parametric corrections according to empirical correction evidence.

Given the historical context and base prediction $\hat{Y}^{(0)}$, LGEC uses only inference-available, stop-gradient features to produce a raw correction δ and gate logit a:

$$G = \text{sigmoid}(a), \qquad \Delta_e = G \odot s_{\max} \tanh(\delta). \tag{5}$$

where Δ_e denotes the gated parametric error correction generated by LGEC, G is the correction activation gate, and $s_{\max}$ bounds the correction magnitude. The corrected forecast is $\hat{Y}^{(1)} = \hat{Y}^{(0)} + \lambda_e \Delta_e$.

Although both G and L_{learn} are bounded in $[0, 1]$, they represent different quantities: G is the inference-time correction activation predicted by LGEC, whereas L_{learn} is a training-only empirical learnability target.

During training, the gate is supervised by a learnability target $L_{\text{learn}} \in [0, 1]$ derived from persistent and sufficiently pronounced structure in the base residual. This target represents empirical correctability under the current correction mechanism, rather than a deterministic-error proportion or a strict deterministic–stochastic decomposition. Residual-teacher supervision, safety constraints, and regularization further stabilize the correction. Ground-truth-dependent residual signals are used only for training; at inference, both G and Δ_e depend solely on the historical input and base forecast. Full feature construction, supervision objectives, and training schedules are provided in Appendix C.

3.4 Reusing Past Failures: Residual-Aware Retrieval Memory

LGEC learns correction rules shared across samples, whereas some failures remain instance-specific but recur under similar forecasting conditions. We therefore introduce Residual-Aware Retrieval Memory (RARM), which stores residuals produced by the fixed LGEC forecaster and reuses reproducible past failures as non-parametric correction evidence.

For each training sample (X_i, Y_i), RARM stores $R_i = Y_i - \hat{Y}_i^{(1)}$ together with a retrieval key k_i that jointly encodes the historical input and prediction state. Given a query q, the K nearest historical failures are retrieved by cosine similarity. With neighbor set $\mathcal{N}_K$, similarity s_j, and temperature τ,

$$w_j = \frac{\exp(s_j/\tau)}{\sum_{\ell \in \mathcal{N}_K} \exp(s_\ell/\tau)}, \qquad \widetilde{\Delta}_m = \sum_{j \in \mathcal{N}_K} w_j R_j. \tag{6}$$

Thus, RARM retrieves how the model previously failed under similar forecasting states rather than historical ground-truth future trajectories.

Because similar states do not necessarily imply transferable residuals, RARM combines retrieval similarity with residual consistency and dispersion to obtain confidence C_m. The final memory correction is

$$\Delta_m = C_m \odot \widetilde{\Delta}_m, \qquad \hat{Y}_{\text{final}} = \hat{Y}^{(1)} + \alpha \odot \Delta_m, \tag{7}$$

where α is calibrated on the validation set and unreliable evidence is attenuated. Importantly, residual dispersion in RARM measures the reliability of historical correction evidence rather than predictive uncertainty. Retrieval-key construction, evidence gating, clipping, calibration, and memory safeguards are detailed in Appendix D.

3.5 Quantifying Remaining Predictive Risk: LGEC-U

After deterministic correction, some residual errors may remain without reliable evidence for further correction. We therefore introduce LGEC-U, an independent uncertainty branch that quantifies the remaining predictive risk without modifying the point forecast.

LGEC-U uses a fixed stop-gradient reference prediction: $\hat{Y}^{(1)}$ for controlled pre-RARM analysis and $\hat{Y}^{(2)}$ for the complete post-RARM model. From an independent history–prediction context encoder, it predicts a positive position-dependent scale

$$\sigma = \sigma_{\min} + \text{Softplus}(s), \tag{8}$$

where larger σ indicates greater remaining predictive risk. Its parameters and gradients are isolated from the point-forecasting path.

LGEC-U is trained post hoc from the realized residual of the fixed reference prediction, with greater emphasis on regions of lower empirical learnability. Validation-set conformal-style scale calibration (Romano et al., 2019; Xu & Xie, 2021) then adjusts the overall interval scale without changing the point forecast. The full model uses post-RARM LGEC-U so that uncertainty corresponds to the actual final forecast. Architecture, supervision objectives, pre-/post-RARM settings, and calibration details are provided in Appendix E.

3.6 Training and Inference Procedure

EUD-RM is trained in three stages to separate point-forecast correction from remaining-risk modeling.

Stage I: Point forecasting and parametric correction. The base forecaster, CAM, and LGEC are jointly optimized with $\mathcal{L}_{\text{stage I}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{LGEC}}$. LGEC correction is introduced gradually after warm-up, yielding the fixed point forecaster $\hat{Y}^{(1)}$. The full objective and correction-strength schedule are given in Appendix C.

Stage II: Residual-memory construction and calibration. The Stage-I forecaster is frozen, and RARM is constructed exclusively from training-set residuals. Its correction strength α is calibrated on the validation set, so RARM acts as a post-training non-parametric correction module without updating the point-forecasting parameters.

Stage III: Post-hoc remaining-risk modeling. After fixing Base/CAM, LGEC, and RARM, only LGEC-U is trained using $\text{sg}(\hat{Y}^{\text{final}})$ as its reference prediction. Validation-set scale calibration is then applied. Because the complete point forecaster remains frozen, uncertainty optimization cannot modify the final point predictions or their MSE/MAE.

At inference, EUD-RM follows Base/CAM $\rightarrow$ LGEC $\rightarrow$ RARM $\rightarrow$ LGEC-U, producing the final point forecast together with its calibrated prediction interval.

4 Experiments

4.1 Experimental Setup

We evaluate EUD-RM on seven widely used long-horizon multivariate forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic, covering both standard and high-dimensional settings. Following standard long-horizon forecasting protocols (Zhou et al., 2021; Wu et al., 2021; Nie et al., 2022), we evaluate $H \in \{96, 192, 336, 720\}$ against representative recent methods. Complete horizon-wise results and dataset statistics, splits, and implementation details are provided in Appendices G.1 and F.1, respectively.

Point forecasts are evaluated by MSE and MAE, while predictive uncertainty is assessed by PICP, MPIW, ACE, and NLL at 90% and 95% target coverage.

All experiments follow the staged training protocol in Section 3.6. RARM memory uses only training-set residuals; the validation set is used for correction-strength and scale calibration, and test residuals are never used for memory construction or parameter selection. The full model uses post-RARM LGEC-U so that uncertainty corresponds to the final point forecast. Unless otherwise stated, results use a fixed random seed; seed stability and computational efficiency analyses are reported in Appendices H.3 and H.1.

4.2 Overall Forecasting Performance

Table 1 reports average MSE and MAE over four prediction horizons; complete horizon-wise results are provided in Appendix G.1. EUD-RM achieves the lowest average MSE on ETTh1, ETTh2, ETTm1, Weather, Traffic, and Electricity, and ties for the best on ETTm2. It also achieves the lowest average MAE on the four ETT datasets and Weather, while competing methods achieve lower MAE on Traffic and EUD-RM ties for the best on Electricity. Overall, EUD-RM provides consistent improvements without dominating every metric.

Compared with the Base forecaster, EUD-RM further reduces forecasting errors, suggesting exploitable structure remains after the initial prediction. CAM strengthens cross-variable representations, LGEC performs selective parametric correction, and RARM reuses recurring historical residuals, corresponding to the reduce–correct–reuse stages.

The reported MSE and MAE are from the fixed final point forecaster. Post-RARM LGEC-U is trained afterward and therefore does not alter these results; its remaining-risk modeling is evaluated separately. We next examine the individual and complementary contributions of CAM, LGEC, and RARM through controlled ablations.

Table 1: Average point-forecasting results over $H \in \{96, 192, 336, 720\}$ against recent representative methods (Lyu et al., 2026; Niu et al., 2026; Wang et al., 2025; Fei et al., 2025; Nematirad et al., 2025; Han et al., 2025; Hu et al., 2025; Goswami et al., 2024; Liu et al., 2024). Lower MSE/MAE is better; best results are bold and second-best underlined.

Dataset	H	Ours		2026				2025										2024			
		EUD-RM		OccamVTS		PhaseFormer		TimeMixer++		Amplifier		Times2D		RAFT		AMD		MOMENT		iTransformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Avg	**0.371**	**0.403**	0.403	0.421	0.403	0.415	0.419	0.432	0.430	0.428	0.409	0.430	0.420	0.436	0.407	0.424	0.418	0.436	0.454	0.448
ETTh2	Avg	**0.311**	**0.368**	0.336	0.383	0.347	0.389	0.339	0.380	0.359	0.391	0.328	0.381	0.359	0.409	0.351	0.392	0.352	0.395	0.383	0.407
ETTm1	Avg	**0.333**	**0.370**	0.347	0.373	0.347	0.374	0.369	0.378	0.381	0.394	0.337	0.374	0.348	0.378	0.347	0.376	0.344	0.379	0.407	0.410
ETTm2	Avg	**0.245**	**0.305**	**0.245**	0.307	0.266	0.321	0.269	0.320	0.276	0.323	0.251	0.316	0.254	0.320	0.254	0.316	0.259	0.318	0.288	0.332
Weather	Avg	**0.219**	**0.258**	0.224	0.259	0.223	0.261	0.226	0.262	0.243	0.271	0.223	0.263	0.241	0.286	0.223	0.263	0.228	0.270	0.258	0.278
Traffic	Avg	**0.381**	0.265	0.407	0.279	0.387	**0.250**	0.416	0.264	0.482	0.315	0.387	0.265	0.401	0.282	0.394	0.272	0.415	0.293	0.428	0.282
Electricity	Avg	**0.153**	**0.249**	0.162	0.259	0.161	0.250	0.165	0.253	0.171	0.265	0.158	**0.249**	0.160	0.259	0.159	0.254	0.165	0.260	0.178	0.270

Table 2: Average ablation results over $H \in \{96, 192, 336, 720\}$. Lower MSE/MAE is better; best results are bold.

Dataset	Full		w/o CAM		w/o LGEC		w/o RARM		w/o CAM+RARM		w/o LGEC+RARM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.371**	**0.403**	0.375	0.405	0.377	0.410	0.374	0.406	0.377	0.407	0.384	0.415
ETTh2	**0.311**	**0.368**	0.319	0.372	0.317	0.371	0.313	0.369	0.321	0.371	0.319	0.372
ETTm1	**0.333**	**0.370**	0.342	0.375	0.339	0.374	0.335	**0.370**	0.345	0.376	0.344	0.375
ETTm2	**0.245**	**0.305**	0.253	0.313	0.249	0.309	0.247	0.306	0.255	0.314	0.254	0.313

4.3 Module Ablation Analysis

We remove CAM, LGEC, and RARM individually and additionally evaluate w/o CAM+RARM and w/o LGEC+RARM, while keeping all other settings unchanged. Table 2 reports results averaged over $H \in \{96, 192, 336, 720\}$ on the four ETT datasets; complete horizon-wise results are in Appendix G.2.

The full model achieves the lowest average MSE on all four datasets and the lowest or tied-lowest MAE. Removing CAM or LGEC consistently degrades performance, while removing RARM causes a smaller but consistent MSE increase, supporting their respective roles in representation enhancement, parametric correction, and residual reuse.

Combined ablations generally cause larger degradation. For example, removing LGEC and RARM increases ETTh1 MSE/MAE from $0.371/0.403$ to $0.384/0.415$, while removing CAM and RARM increases ETTm1 MSE from 0.333 to 0.345. Individual horizons may occasionally show tied or better MAE under an ablation; full results are provided in Appendix G.2. Sections 4.4 and 4.5 further examine whether LGEC and RARM operate as intended.

4.4 Learning When to Correct: Analysis of the LGEC Gating Mechanism

To test whether the LGEC gate reflects correction utility, we analyze its relationship with realized correction gain in the pre-RARM setting, thereby isolating the effect of LGEC. Test positions are grouped into High-G, Mid-G, and Low-G regions by inference-time gate values. Ground-truth futures are used only post hoc to measure MAE reduction and never to generate the gate or correction.

Table 3 shows consistent gate–utility stratification on both datasets: realized correction gain follows High-G $>$ Mid-G $>$ Low-G, indicating that LGEC concentrates correction on positions with greater empirical correction utility. The gate should not be interpreted as a deterministic-error proportion, and a low gate does not imply random noise; it reflects empirical correctability under the current parametric correction mechanism.

4.5 Reusing Past Failures: Analysis of the RARM Retrieval Mechanism

To examine how RARM converts historical failures into useful corrections, we analyze a representative Weather-336 case. As shown in Figure 3, the joint history–prediction retrieval key finds similar forecasting states, whose residuals provide broadly consistent correction evidence. RARM com-

Table 3: LGEC correction across predicted gate regions at $H = 336$. Lower MAE and higher correction gain are better.

Dataset	Horizon	Gate Group	Base MAE ↓	Corrected MAE ↓	Correction Gain (%) ↑
Traffic	336	High-G	0.499	0.295	**40.9**
		Mid-G	0.331	0.267	19.4
		Low-G	0.278	0.259	7.0
Electricity	336	High-G	0.360	0.318	**11.7**
		Mid-G	0.261	0.247	5.4
		Low-G	0.179	0.174	2.8

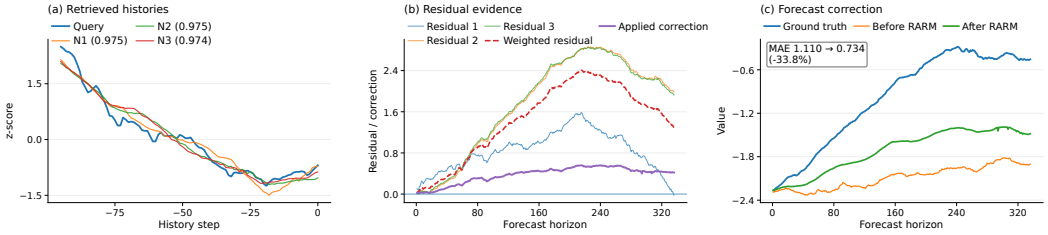

Figure 3: RARM retrieval and correction on Weather-336. (a) Query and retrieved historical contexts. (b) Retrieved residual evidence and gated correction. (c) Forecasts before and after RARM.

bines retrieval similarity with residual consistency and dispersion, attenuating unreliable evidence rather than directly copying retrieved residuals.

Figure 3(c) shows that the resulting correction reduces MAE from 1.110 to 0.734 (33.8%). This example illustrates RARM's intended mechanism: it reuses model-produced failures under similar forecasting states as confidence-controlled non-parametric correction evidence, complementing LGEC's shared parametric correction rules. The case is illustrative; RARM's aggregate contribution is evaluated by the ablation results in Table 2.

4.6 Progressive Error Handling and Functional Disentanglement

After deterministic correction, residual errors differ in empirical correctability, historical reproducibility, and remaining predictive risk. EUD-RM handles them progressively: LGEC applies corrections supported by empirical evidence, RARM reuses reproducible historical residuals, and LGEC-U quantifies the remaining risk.

Prediction interval reliability. Residual errors lacking reliable correction evidence are modeled by LGEC-U without modifying the point forecast. We evaluate uncertainty estimation in the controlled pre-RARM setting, using the fixed LGEC-corrected forecast as the stop-gradient reference. Table 4 compares a validation-calibrated global constant scale (Constant Cal.), uncalibrated position-dependent LGEC-U (LGEC-U (uncal.)), and calibrated LGEC-U (LGEC-U + Cal.) using PICP, MPIW, ACE, and NLL.

LGEC-U achieves a better coverage–sharpness trade-off than the global constant scale. At 95% coverage, compared with Constant Cal., calibrated LGEC-U reduces ACE/MPIW/NLL from 0.022/2.340/0.740 to 0.005/1.904/0.394 on Weather, and from 0.016/1.792/1.189 to 0.011/1.542/0.637 on Traffic. The uncalibrated scale already captures relative risk, while calibration mainly adjusts its overall magnitude toward the target coverage.

Progressive error handling analysis. Figure 4 analyzes the correct–reuse–quantify process on Traffic-336. In Figure 4(a), LGEC correction gain increases across gate deciles, with Spearman correlation $\rho = 0.459$, showing that larger gates correspond to higher empirical correction utility.

Figure 4(b) shows that RARM further improves LGEC using historical residual evidence: 75.1% of sample–channel cases improve, with a median relative MAE gain of 0.37%. This supports the usefulness of model-generated residuals as non-parametric correction evidence.

Table 4: Prediction-interval analysis under the controlled pre-RARM setting. PICP closer to the target and lower ACE/NLL are better; MPIW should be interpreted jointly with coverage.

Dataset	Horizon	Coverage	Method	PICP	MPIW	ACE	NLL
Traffic	336	0.90	Constant Cal.	0.869	1.125	0.031	1.727
			LGEC-U (uncal.)	**0.891**	1.094	**0.009**	**0.812**
			LGEC-U + Cal.	0.880	**1.031**	0.020	0.904
Traffic	336	0.95	Constant Cal.	0.934	1.792	0.016	1.189
			LGEC-U (uncal.)	0.919	**1.303**	0.031	0.812
			LGEC-U + Cal.	**0.939**	1.542	**0.011**	**0.637**
Weather	336	0.90	Constant Cal.	0.937	1.763	0.037	0.713
			LGEC-U (uncal.)	0.808	**0.946**	0.092	0.575
			LGEC-U + Cal.	**0.915**	1.477	**0.015**	**0.380**
Weather	336	0.95	Constant Cal.	0.972	2.340	0.022	0.740
			LGEC-U (uncal.)	0.855	**1.127**	0.095	0.575
			LGEC-U + Cal.	**0.955**	1.904	**0.005**	**0.394**

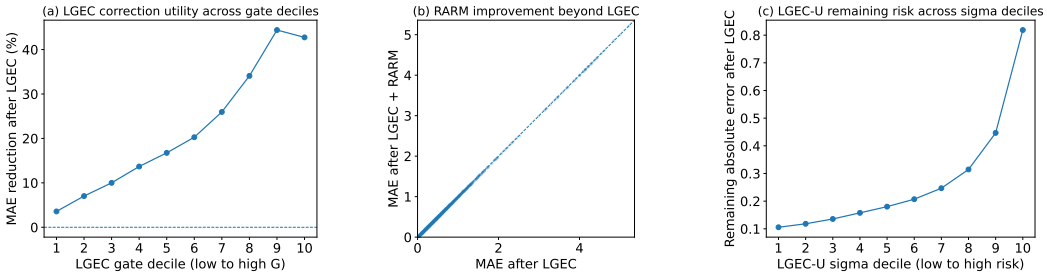

Figure 4: Progressive error handling analysis of EUD-RM on Traffic-336. (a) LGEC correction utility increases with gate values, indicating empirical correctability. (b) RARM further improves LGEC by exploiting historical residual evidence. (c) LGEC-U estimates remaining predictive risk after deterministic correction.

Figure 4(c) shows that remaining absolute error increases monotonically across LGEC-U sigma deciles, with $\rho = 0.420$, indicating effective risk ranking after deterministic correction.

Together, Table 4 and Figure 4 support the functional separation of the three stages: LGEC identifies empirically correctable errors, RARM reuses reproducible historical failures, and LGEC-U quantifies the remaining predictive risk. Further gate-controlled within-group risk analysis and mechanism studies are provided in Appendices E.9 and H.2.

5 Conclusion and Future Work

We introduced EUD-RM, an error-centric forecasting framework that handles forecasting failures through a progressive reduce–correct–reuse–quantify process. Rather than assuming a strict deterministic–random decomposition of residuals, EUD-RM handles failures according to empirical correctability, historical reproducibility, and remaining predictive risk. CAM strengthens cross-variable modeling, LGEC applies selective parametric correction, RARM reuses reproducible historical residuals, and LGEC-U quantifies the remaining risk without altering point forecasts. Experiments on long-horizon multivariate benchmarks show consistent forecasting improvements, while ablation and mechanism analyses support complementary component roles and the separation of correction from predictive risk estimation. Future work will improve residual-memory efficiency and uncertainty calibration for large-scale and latency-sensitive forecasting.

AI USE STATEMENT

Generative AI tools were used solely for translation, language polishing, and editing to improve the readability and conciseness of the manuscript. They were not used for substantive research tasks.

REPRODUCIBILITY STATEMENT

We provide detailed model formulations, training objectives, implementation details, hyperparameter configurations, dataset statistics and splits, and the complete training and inference procedure in the appendices. Full horizon-wise forecasting and ablation results, computational efficiency analysis, and random-seed stability experiments are also reported to facilitate reproducibility.

REFERENCES

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11645–11653, 2025.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

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.

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

Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 17359–17367, 2025.

Domingos S de O Santos Júnior, Paulo SG de Mattos Neto, João FL de Oliveira, and George DC Cavalcanti. A hybrid system based on ensemble learning to model residuals for time series forecasting. *Information sciences*, 649:119614, 2023.

Mehdi Khashei and Mehdi Bijari. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied soft computing*, 11(2):2664–2675, 2011.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4): 1748–1764, 2021.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36:12271–12290, 2023.

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, 2024.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *International conference on learning representations*, volume 2024, pp. 31728–31770, 2024.

Sisuo Lyu, Siru Zhong, Weilin Ruan, Qingxiang Liu, Qingsong Wen, Hui Xiong, and Yuxuan Liang. Occamvts: Distilling vision models to 1% parameters for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24216–24225, 2026.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 19581–19588, 2025.

Reza Nematirad, Anil Pahwa, and Balasubramaniam Natarajan. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19651–19658, 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.

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 8262–8285, 2026.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Kamile Stankeviciute, Ahmed M Alaa, and Mihaela Van der Schaar. Conformal time-series forecasting. *Advances in neural information processing systems*, 34:6216–6228, 2021.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. *arXiv preprint arXiv:2411.08249*, 2024.

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.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pp. 11559–11569. PMLR, 2021.

Sitan Yang, Carson Eisenach, and Dhruv Madeka. Mqretnn: Multi-horizon time series forecasting with retrieval augmentation. *arXiv preprint arXiv:2207.10517*, 2022.

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.

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.

G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

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.

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

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022b.

Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, Rong Jin, et al. Card: Channel aligned robust blend transformer for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11245–11283, 2024.

APPENDIX

A BASE FORECASTER AND LAST-VALUE RESIDUALIZATION

EUD-RM adopts a **multi-scale patch-mixing forecaster** as its base point predictor. Given the historical input

$$X \in \mathbb{R}^{B \times L \times C_{\text{in}}}, \tag{9}$$

the backbone extracts patch-level temporal representations at multiple resolutions and produces the initial point forecast

$$\hat{Y}^{(0)} = f_\theta(X). \tag{10}$$

The Channel-Aware Mixer (CAM) is inserted into the multi-scale Mixer representations to strengthen cross-variable interaction before the forecast is formed. CAM only modifies intermediate representations of the base forecaster and does not perform residual correction. In contrast, LGEC, RARM, and LGEC-U operate primarily on the resulting forecast or its associated prediction

states. CAM therefore corresponds to the **reduce** stage of EUD-RM, while the subsequent modules perform correction, residual reuse, and remaining-risk quantification.

The base forecaster further employs **last-value residualization**. Rather than directly predicting the absolute future level, the backbone learns future changes relative to the most recent observed state x_{last}. Let $\Delta\hat{Y}$ denote the predicted change; the initial point forecast can then be written as

$$\hat{Y}^{(0)} = x_{\text{last}} + \Delta\hat{Y}, \tag{11}$$

where x_{last} is broadcast along the forecast horizon as needed. This reparameterization encourages the backbone to model future deviations from the current state while returning the final prediction in the original value space.

Importantly, **last-value residualization is distinct from the forecasting residuals used by LGEC and RARM**. The former is an input/output reparameterization relative to the latest observation, whereas the latter refers to the actual prediction error between the forecast and the future target, e.g.,

$$R = Y - \hat{Y}. \tag{12}$$

Thus, last-value residualization is neither an error-correction mechanism nor the residual memory used by RARM, and it should not be confused with the historical forecasting residuals stored by the proposed framework.

This design is used only to instantiate the base forecaster and is not a defining component of the proposed error-handling framework. The primary focus of EUD-RM is how forecasting failures are subsequently **reduced, corrected, reused, and quantified** after the initial forecast is formed.

B Implementation Details of the Channel-Aware Mixer

This appendix provides implementation details of CAM, including full channel mixing, lightweight channel modulation, automatic mode selection, and gated residual integration.

B.1 Full Channel Mixing

For moderate-dimensional datasets, CAM performs explicit channel mixing at each patch and hidden-feature position.

Let

$$F \in \mathbb{R}^{B\times C\times P\times D} \tag{13}$$

denote the representation after temporal and hidden-feature mixing, where B, C, P, and D denote batch size, number of variables, number of patches, and hidden dimension.

The representation is first permuted as

$$\widetilde{F} = \text{Permute}(F) \in \mathbb{R}^{B\times P\times D\times C}, \tag{14}$$

so that channels become the feature dimension. Channel mixing is then applied:

$$\widetilde{F}_c = \text{MLP}_c(\text{LN}(\widetilde{F})). \tag{15}$$

The output is mapped back:

$$\mathcal{M}_c^{\text{full}}(F) = \text{Permute}^{-1}(\widetilde{F}_c). \tag{16}$$

B.2 Lightweight Channel Modulation

For high-dimensional datasets, CAM uses a lightweight channel modulation mechanism.

Given

$$F \in \mathbb{R}^{B\times C\times P\times D}, \tag{17}$$

a global channel descriptor is computed by

$$z_c = \mathrm{Mean}_{p,d}(F) \in \mathbb{R}^{B\times C}. \tag{18}$$

Channel modulation coefficients are generated by

$$m_c = \mathrm{MLP}_{\mathrm{mod}}(z_c), \tag{19}$$

and broadcast to modulate the original representation, denoted as $\mathcal{M}_c^{\mathrm{lite}}(F)$.

B.3 Automatic Mode Selection

CAM automatically selects the channel interaction mode according to variable dimensionality:

$$\mathcal{M}_c(F) = \begin{cases} \mathcal{M}_c^{\mathrm{full}}(F), & \text{moderate-dimensional input,} \\ \mathcal{M}_c^{\mathrm{lite}}(F), & \text{high-dimensional input.} \end{cases} \tag{20}$$

For univariate forecasting or when CAM is disabled, the operation reduces to an identity mapping.

B.4 Gated Residual Integration

Both channel interaction modes use the same gated residual integration:

$$F' = F + \mathrm{sigmoid}(g_c) M_c(F), \tag{21}$$

where g_c is a learnable scalar gate.

The gate is initialized with a small activation to gradually introduce cross-variable information during training.

CAM is applied after temporal and hidden-feature mixing in each resolution branch of the multi-scale backbone. The residual integration remains unchanged for both interaction modes; only $\mathcal{M}_c(\cdot)$ differs according to input dimensionality.

C Full Training Objective and Implementation Details of LGEC

This appendix provides implementation details of LGEC, including context construction, learnability supervision, training objectives, scheduling, and gradient isolation.

C.1 Error Context and Gated Correction

LGEC receives the historical sequence X and the base forecast $\hat{Y}^{(0)}$ to construct a position-dependent correction context. The historical features and prediction state used by LGEC are stop-gradient.

Let x_{last} denote the most recent observation. We first center the historical trajectory:

$$\widetilde{X} = X - x_{\mathrm{last}}, \tag{22}$$

and project it along the temporal dimension onto the forecast horizon:

$$H_{\text{hist}} = \mathcal{P}_h(\widetilde{X}). \tag{23}$$

The base forecast offset relative to the latest observation is

$$\widetilde{Y}^{(0)} = \text{sg}(\hat{Y}^{(0)}) - x_{\text{last}}, \tag{24}$$

with first-order difference

$$\Delta\widetilde{Y}_h^{(0)} = \widetilde{Y}_h^{(0)} - \widetilde{Y}_{h-1}^{(0)}. \tag{25}$$

For each forecast position h and variable c, LGEC constructs

$$E_{h,c} = \left[\widetilde{Y}_{h,c}^{(0)}, \Delta\widetilde{Y}_{h,c}^{(0)}, H_{\text{hist},h,c}, x_{\text{last},c}\right]. \tag{26}$$

These features are processed by a shared feature projector and a lightweight sequence–channel interaction module. The correction head outputs a correction $\delta_{h,c}$ and gate logit $a_{h,c}$.

The gate is

$$G_{h,c} = \text{sigmoid}(a_{h,c}), \tag{27}$$

and the bounded correction is

$$\Delta_{e,h,c} = G_{h,c} s_{\max} \tanh(\delta_{h,c}), \tag{28}$$

where $s_{\max}$ bounds the correction magnitude.

During training, the applied correction is

$$\lambda_e \Delta_e. \tag{29}$$

Here, Δ_e denotes the local gated correction, while λ_e controls the global correction strength during training.

C.2 Learnability Proxy

The learnability target is constructed from the base forecasting residual.

The stop-gradient residual is

$$R^{(0)} = Y - \text{sg}(\hat{Y}^{(0)}). \tag{30}$$

For brevity, we write $R \equiv R^{(0)}$ below. The residual is smoothed along the prediction horizon:

$$R_s = \text{Smooth}(R), \tag{31}$$

and the rapidly varying component is

$$R_h = R - R_s. \tag{32}$$

We define the residual-persistence ratio

$$Q = \frac{|R_s|}{|R_s| + |R_h| + \epsilon}. \tag{33}$$

To avoid assigning high learnability to negligible residuals, we introduce the residual scale

$$s_R = \mathrm{Mean}(|R|) + \epsilon, \tag{34}$$

and amplitude score

$$M = \mathrm{clip}\left(\frac{|R_s|}{s_R}, 0, 1\right). \tag{35}$$

The final learnability target is

$$L_{\mathrm{learn}} = [\mathrm{clip}(Q \odot M, 0, 1)]^p, \tag{36}$$

where p controls selectivity.

After construction, L_{learn} is stop-gradient and used only as training supervision for the gate. The quantities R, R_s, R_h, and L_{learn} are not constructed during inference.

The residual decomposition above is only an empirical proxy for correctability and does not imply a strict separation between deterministic and random errors.

C.3 Residual Teacher and Learnability-Aware Weighting

We combine the base residual and its smoothed component to construct the residual teacher:

$$T_e = (1 - \beta_t)R + \beta_t R_s, \tag{37}$$

where β_t controls the contribution of persistent residual structure. We use $\beta_t = 0.7$ by default.

The learnability-aware teacher weight is

$$W_t = w_t^{\min} + (1 - w_t^{\min})L_{\mathrm{learn}}, \tag{38}$$

where $w_t^{\min} = 0.25$.

The residual-teacher loss is

$$\mathcal{L}_{\mathrm{teacher}} = \mathbb{E}\left[W_t \odot \mathrm{SmoothL1}(\Delta_e, T_e)\right]. \tag{39}$$

The objective supervises the unscaled local correction Δ_e, rather than the training-stage correction $\lambda_e \Delta_e$.

C.4 Safety Constraints and Full Training Objective

The residual teacher supervises correction direction, while additional constraints regulate correction utility, gate activation, magnitude, direction, and temporal smoothness.

Safe Correction Loss. Let $\lambda_e \Delta_e$ denote the correction applied at the current training stage. We penalize cases where correction increases the prediction error:

$$\mathcal{L}_{\mathrm{safe}} = \mathbb{E}\left[\mathrm{ReLU}\left(|R - \lambda_e \Delta_e| - |R|\right)\right]. \tag{40}$$

Gate Supervision. The learnability target supervises the predicted gate:

$$\mathcal{L}_{\mathrm{gate}} = \mathrm{BCE}(G, L_{\mathrm{learn}}). \tag{41}$$

Gate Sparsity. To avoid activating corrections at all positions:

$$\mathcal{L}_{\text{sparse}} = \mathbb{E}[G]. \tag{42}$$

Sign Consistency. For learnable residual structures, the correction direction is encouraged to follow the persistent residual component:

$$\mathcal{L}_{\text{sign}} = \mathbb{E}[L_{\text{learn}} \odot \text{ReLU}(-\Delta_e \odot R_s)]. \tag{43}$$

Over-Correction Control. Correction magnitude is constrained by the residual teacher:

$$\mathcal{L}_{\text{over}} = \mathbb{E}\left[\text{ReLU}(|\Delta_e| - \rho_{\max}|T_e|)\right], \tag{44}$$

where $\rho_{\max} = 0.80$ by default.

Horizon Smoothness. To avoid rapidly varying corrections along the forecast horizon:

$$\mathcal{L}_{\text{smooth}} = \mathbb{E}\left[|\Delta_{e,h} - \Delta_{e,h-1}|\right]. \tag{45}$$

The complete LGEC auxiliary objective is

$$\begin{aligned}\mathcal{L}_{\text{LGEC}} = \lambda_t \mathcal{L}_{\text{teacher}} + \lambda_s \mathcal{L}_{\text{safe}} + \lambda_g \mathcal{L}_{\text{gate}} \\ + \lambda_{\text{sp}} \mathcal{L}_{\text{sparse}} + \lambda_{\text{sign}} \mathcal{L}_{\text{sign}} + \lambda_{\text{over}} \mathcal{L}_{\text{over}} + \lambda_{\text{sm}} \mathcal{L}_{\text{smooth}}.\end{aligned} \tag{46}$$

The Stage-I optimization objective is

$$\mathcal{L}_{\text{stage I}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{LGEC}}. \tag{47}$$

C.5 Warm-up and Correction-Strength Scheduling

LGEC correction is gradually introduced during training to avoid unstable early corrections.

Let E_w denote the warm-up duration, E_r the ramp duration, and e the current epoch. The global correction strength is

$$\lambda_e(e) = \begin{cases} 0, & e < E_w, \\ \text{clip}\left(\frac{e - E_w}{E_r}, 0, 1\right), & e \geq E_w. \end{cases} \tag{48}$$

During warm-up, $\lambda_e = 0$. After warm-up, the correction strength increases until $\lambda_e = 1$.

The LGEC auxiliary objective is activated after the warm-up period. Validation, testing, and RARM memory construction use the full correction strength.

C.6 Gradient Isolation and Use of Future Labels

LGEC separates the inference path from the training-supervision path.

During inference, the gate and correction are generated only from historical input and the base forecast:

$$(X, \hat{Y}^{(0)}) \longrightarrow (G, \Delta_e) \longrightarrow \hat{Y}^{(1)}. \tag{49}$$

Ground-truth future values are used only during training to construct the residual teacher, learnability proxy, and auxiliary losses. These quantities are not provided as inference inputs.

Table 5: Default hyperparameters of LGEC used in our experiments.

Hyperparameter	Default Value
LGEC hidden dimension	32
Gate initial logit	-3.5
Maximum correction scale	0.05
Residual smoothing window	5
Teacher smoothing coefficient	0.7
Minimum teacher weight	0.25
Learnability power	1.0
Allowed over-correction ratio	0.80
Teacher loss weight	0.02
Safe loss weight	0.10
Gate loss weight	0.005
Sparsity loss weight	0.001
Sign-consistency loss weight	0.02
Over-correction loss weight	0.01
Smoothness loss weight	0.001
Warm-up epochs	5
Ramp epochs	5
Inference correction strength	1.0

The historical features and base forecast state used for LGEC context construction are stop-gradient, preventing auxiliary correction paths from updating the backbone through these features.

The corrected forecast

$$\hat{Y}^{(1)} = \hat{Y}^{(0)} + \lambda_e \Delta_e \tag{50}$$

still participates in the forecasting objective, allowing joint optimization of the backbone and LGEC.

Therefore, future-dependent quantities are training-only supervision signals and do not introduce inference-time label leakage.

C.7 Default Hyperparameters

Table 5 summarizes the default LGEC hyperparameters used in experiments.

D Implementation Details of RARM

D.1 Retrieval-Key Construction

RARM constructs retrieval keys by jointly encoding the historical input pattern and the associated prediction state.

For an arbitrary sequence

$$Z \in \mathbb{R}^{B \times T \times C}, \tag{51}$$

we first normalize each variable independently along the temporal dimension:

$$\bar{Z}_{b,t,c} = \frac{Z_{b,t,c} - \mu_{b,c}}{\max(\sigma_{b,c}, \epsilon)}, \tag{52}$$

where $\mu_{b,c}$ and $\sigma_{b,c}$ denote the temporal mean and standard deviation of channel c in sample b, respectively, and $\epsilon = 10^{-5}$ ensures numerical stability. The normalized sequence is then mapped to a fixed temporal resolution D_k through adaptive average pooling:

$$\phi(Z) \in \mathbb{R}^{B \times C \times D_k}. \tag{53}$$

For the historical input X, only the most recent L_k time steps are retained before normalization and pooling, whereas the prediction sequence is encoded over the full forecast horizon. We use

$$L_k = 48, \qquad D_k = 16 \tag{54}$$

by default.

In the channel-wise retrieval mode, the input representation and prediction state representation are concatenated along the feature dimension. The retrieval key is defined as

$$k = \mathrm{Norm}\left(\left[\phi_{L_k}(X), \gamma\phi(\hat{Y}^{(1)})\right]\right), \tag{55}$$

where $\phi_{L_k}(\cdot)$ denotes encoding based on the most recent L_k historical steps and $\mathrm{Norm}(\cdot)$ denotes L_2 normalization along the feature dimension. Prediction-state information is enabled by default with $\gamma = 0.5$.

Channel-wise versus Global Retrieval. Under automatic mode selection, channel-wise retrieval is applied when the retrieval-key channels are aligned with output channels and the number of variables does not exceed the threshold $C_{\max} = 128$. Each variable independently retrieves its corresponding historical residuals.

For high-dimensional multivariate forecasting, RARM constructs a compact global representation. Given channel-level features A, we define

$$\psi(A) = [\mathrm{Mean}_c(A_c), \mathrm{Std}_c(A_c)]. \tag{56}$$

When prediction-state information is enabled, the global key is

$$k_{\mathrm{global}} = \mathrm{Norm}\left(\left[\psi(\phi_{L_k}(X)), \psi(\gamma\phi(\hat{Y}^{(1)}))\right]\right). \tag{57}$$

This representation keeps the global key dimension independent of the number of variables and reduces retrieval overhead for high-dimensional datasets.

D.2 Residual-Memory Construction and Top-K Retrieval

For a fixed LGEC point forecaster, the historical forecasting residual of training sample (X_i, Y_i) is

$$R_i = Y_i - \hat{Y}_i^{(1)}. \tag{58}$$

Each memory entry is stored as

$$(k_i, R_i). \tag{59}$$

The point forecaster remains frozen during memory construction. The memory is bounded in size; once its capacity is reached, reservoir sampling maintains a fixed-size representative subset of historical forecasting failures.

Given a query key q, similarity to memory key k_i is computed as

$$s_i = q^\top k_i. \tag{60}$$

Because both query and memory keys are L_2 normalized, this inner product corresponds to cosine similarity. RARM retrieves the K nearest historical failures:

$$\mathcal{N}_K(q), \tag{61}$$

with $K = 16$ by default. Their aggregation weights are computed using a temperature-scaled softmax:

$$w_i = \frac{\exp(s_i/\tau)}{\sum_{j\in\mathcal{N}_K(q)} \exp(s_j/\tau)}, \tag{62}$$

where $\tau = 0.10$ by default.

The raw memory correction is

$$\widetilde{\Delta}_m = \sum_{i\in\mathcal{N}_K(q)} w_i R_i, \tag{63}$$

and the weighted residual dispersion is

$$V_m = \sum_{i \in \mathcal{N}_K(q)} w_i \left(R_i - \widetilde{\Delta}_m \right)^2 . \tag{64}$$

$\widetilde{\Delta}_m$ represents the aggregated historical residual evidence, while V_m measures residual dispersion for evidence reliability estimation. Channel-wise and global retrieval share the same Top-K weighting mechanism and differ only in retrieval granularity.

D.3 Retrieval-Evidence Gating

RARM combines retrieval similarity and residual consistency to construct a confidence score for memory correction.

The similarity confidence is computed from the mean similarity $\bar{s}$ of retrieved neighbors:

$$C_{\text{sim}} = \left[\text{clip} \left(\frac{\bar{s} - s_{\min}}{1 - s_{\min}}, 0, 1 \right) \right]^{p_s}, \tag{65}$$

where $s_{\min} = 0$ and $p_s = 1$ by default.

To measure residual consistency, we compute

$$A_m = \sum_{i \in \mathcal{N}_K(q)} w_i |R_i|, \tag{66}$$

and define

$$C_{\text{sign}} = \text{clip} \left(\frac{|\widetilde{\Delta}_m|}{A_m + \epsilon}, 0, 1 \right). \tag{67}$$

Residual dispersion consistency is defined as

$$C_{\text{disp}} = \tanh \left(\frac{|\widetilde{\Delta}_m|}{\sqrt{V_m + \epsilon_v} + \epsilon} \right), \tag{68}$$

where $\epsilon_v = 10^{-8}$ and $\epsilon = 10^{-6}$.

The residual-consistency score is

$$C_{\text{cons}} = (C_{\text{sign}} \odot C_{\text{disp}})^{1/2}. \tag{69}$$

The final memory confidence is

$$C_m = C_{\text{sim}} \odot C_{\text{cons}}^{p_c}, \tag{70}$$

where $p_c = 1$ by default.

A confidence floor is applied:

$$C_m \leftarrow c_{\min} + (1 - c_{\min}) C_m, \tag{71}$$

with $c_{\min} = 0.10$.

The final gated memory correction is

$$\Delta_m = C_m \odot \widetilde{\Delta}_m. \tag{72}$$

D.4 Correction Clipping

To prevent excessive memory corrections caused by large historical residuals, RARM applies a data-dependent clipping bound.

Let $q_{0.95}$ denote the 95th percentile of the absolute residual values stored in memory. The clipping bound is

$$b_{\text{clip}} = \kappa q_{0.95}, \tag{73}$$

and the memory correction is clipped as

$$\Delta_m \leftarrow \text{clip}(\Delta_m, -b_{\text{clip}}, b_{\text{clip}}). \tag{74}$$

We use $\kappa = 2$ by default. For large memories, the percentile is estimated from at most 10^6 approximately uniformly sampled residual elements.

D.5 Calibration of Memory-Correction Strength

The confidence score controls local correction reliability, while α controls the global influence of memory correction.

We support three calibration granularities:

$$\text{scalar}, \qquad \text{horizon}, \qquad \text{horizon-channel}. \tag{75}$$

For horizon-level and horizon-channel calibration, let Δ_m denote the uncalibrated RARM correction and

$$R = Y - \hat{Y}^{(1)} \tag{76}$$

be the validation residual of the fixed LGEC forecaster. We estimate α by minimizing

$$\left\| \hat{Y}^{(1)} + \alpha \odot \Delta_m - Y \right\|_2^2 + \lambda_\alpha \|\alpha\|_2^2. \tag{77}$$

The closed-form solution is

$$\alpha = \frac{\sum \Delta_m \odot R}{\sum \Delta_m^2 + \lambda_\alpha}. \tag{78}$$

Horizon-level calibration estimates one coefficient per forecast step, whereas horizon-channel calibration estimates separate coefficients for each forecast step and variable. We use horizon-level calibration by default.

If calibrated RARM does not improve the validation metric over the forecast without RARM, we set

$$\alpha = 0, \tag{79}$$

so that the model falls back to the LGEC point forecaster.

Residual memory is constructed only from training data. The validation set is used for calibration, while test residuals are never used for memory construction or parameter selection.

D.6 Memory Capacity and Computational Safeguards

RARM uses a bounded residual memory to control storage and retrieval costs. The effective capacity is constrained by the maximum number of stored samples and residual elements.

Once the memory reaches capacity, reservoir sampling maintains an approximately unbiased fixed-size sample of historical forecasting failures.

Memory keys are stored in FP32 for stable similarity computation, whereas historical residuals are stored in FP16 by default. The memory remains on GPU when it fits the configured budget and otherwise falls back to CPU storage.

For channel-wise retrieval, similarity computation is additionally chunked along batch and channel dimensions to reduce peak memory usage.

D.7 Self-Retrieval Protection during Post-RARM Uncertainty Training

During post-RARM LGEC-U training, training samples may retrieve their own memory entries, causing overly optimistic remaining-risk targets.

To prevent self-retrieval leakage, RARM excludes memory keys that exactly match the current query. Under normalized cosine retrieval, an exact match satisfies

$$s_i \geq 1 - \varepsilon_{\text{match}}, \tag{80}$$

and its similarity is masked before Top-K selection:

$$s_i = -\infty. \tag{81}$$

The current sample therefore cannot directly retrieve its own stored residual and must rely on residual evidence from other historical samples. Exact-match exclusion is supported in both global and channel-wise retrieval.

This protection is required only when training queries overlap with the samples used to construct residual memory. Validation and test queries do not suffer from this issue when memory is constructed exclusively from training data.

E Implementation Details of LGEC-U

E.1 Independent Uncertainty Branch

LGEC-U uses the history–prediction context while maintaining an independent parameterization from LGEC. Given historical input X and reference prediction $\hat{Y}^{\mathrm{ref}}$, the uncertainty branch constructs four position-dependent features:

1. the offset of the reference prediction relative to the latest observation;
2. the first-order difference of the reference prediction;
3. the projection of the centered historical trajectory onto the forecast horizon;
4. the latest observation context.

The resulting feature tensor is

$$E_u \in \mathbb{R}^{B\times H\times C\times 4}. \tag{82}$$

These features are processed by an independent uncertainty projector, sequence–channel mixer, and scale prediction head. The uncertainty branch and LGEC therefore have separate parameters and optimization paths.

The predictive scale is defined as

$$\sigma = \sigma_{\min} + \mathrm{Softplus}(s), \tag{83}$$

where $\sigma_{\min} > 0$ ensures a positive scale.

After validation-set calibration, the scale becomes

$$\tilde{\sigma} = m_u \sigma, \tag{84}$$

where m_u is the calibration multiplier.

E.2 Pre-RARM and Post-RARM Uncertainty

LGEC-U can operate with different reference predictions.

For the pre-RARM analysis setting,

$$\hat{Y}^{\mathrm{ref}} = \hat{Y}^{(1)}, \tag{85}$$

and the remaining residual is

$$R^{\mathrm{pre}} = Y - \hat{Y}^{(1)}. \tag{86}$$

This setting is used to analyze the relationship between LGEC correction utility and remaining risk before RARM.

For the complete model,

$$\hat{Y}^{\mathrm{ref}} = \hat{Y}^{\mathrm{final}}, \tag{87}$$

with

$$R^{\mathrm{post}} = Y - \hat{Y}^{\mathrm{final}}. \tag{88}$$

The default EUD-RM setting uses the post-RARM reference, so LGEC-U estimates the risk associated with the final point forecast.

E.3 Uncertainty-Scale Supervision

LGEC-U combines residual information from the base prediction and the current reference prediction.

The base residual is

$$R = Y - \hat{Y}^{(0)}, \tag{89}$$

with rapidly varying component

$$R_h = R - R_s, \tag{90}$$

where R_s is the locally smoothed residual.

The remaining residual under the current reference prediction is

$$R_{\mathrm{ref}} = Y - \hat{Y}^{\mathrm{ref}}. \tag{91}$$

The uncertainty supervision target is

$$T_u = \beta_u |R_h| + (1 - \beta_u)|R_{\mathrm{ref}}|, \tag{92}$$

where $\beta_u = 0.5$ by default.

R_h is used only as an empirical supervision signal and does not represent a strict decomposition of forecasting errors into deterministic and random components.

E.4 Learnability-Aware Uncertainty Weighting

Given the learnability proxy

$$L_{\mathrm{learn}} \in [0, 1], \tag{93}$$

we define a complementary uncertainty signal

$$U = 1 - L_{\mathrm{learn}}. \tag{94}$$

The uncertainty supervision weight is

$$W_u = w_u^{\min} + (1 - w_u^{\min})U. \tag{95}$$

We use

$$w_u^{\min} = 0.25. \tag{96}$$

Therefore, low-learnability regions receive stronger uncertainty supervision, while high-learnability regions retain non-zero weighting.

E.5 Full Uncertainty Training Objective

Let

$$R_{\mathrm{ref}} = Y - \hat{Y}^{\mathrm{ref}}, \tag{97}$$

and

$$V = \sigma^2. \tag{98}$$

LGEC-U optimizes four objectives.

The weighted Gaussian negative log-likelihood is

$$\mathcal{L}_{\mathrm{nll}} = \mathbb{E}\left[W_u \odot \left(\frac{1}{2}\frac{R_{\mathrm{ref}}^2}{V} + \frac{1}{2}\log V + \frac{1}{2}\log(2\pi)\right)\right]. \tag{99}$$

The scale regression loss is

$$\mathcal{L}_{\mathrm{tar}} = \mathbb{E}[W_u \odot \mathrm{SmoothL1}(\sigma, T_u)]. \tag{100}$$

To penalize insufficient interval width,

$$\mathcal{L}_{\mathrm{cov}} = \mathbb{E}[W_u \odot \mathrm{ReLU}(|R_{\mathrm{ref}}| - z\sigma)^2]. \tag{101}$$

The sharpness regularization is

$$\mathcal{L}_{\mathrm{sharp}} = \mathbb{E}[W_u \odot \sigma]. \tag{102}$$

The complete objective is

$$\mathcal{L}_U = \lambda_{\mathrm{nll}}\mathcal{L}_{\mathrm{nll}} + \lambda_{\mathrm{tar}}\mathcal{L}_{\mathrm{tar}} + \lambda_{\mathrm{cov}}\mathcal{L}_{\mathrm{cov}} + \lambda_{\mathrm{sharp}}\mathcal{L}_{\mathrm{sharp}}. \tag{103}$$

Default coefficients are

$$\lambda_{\mathrm{nll}} = 0.01, \quad \lambda_{\mathrm{tar}} = 0.05, \quad \lambda_{\mathrm{cov}} = 0.02, \quad \lambda_{\mathrm{sharp}} = 0.001. \tag{104}$$

E.6 Post-hoc Training and Parameter Isolation

LGEC-U is trained after fixing the point-forecasting model.

First, Base/CAM/LGEC are optimized and a fixed checkpoint is selected. RARM is then constructed and calibrated, producing $\hat{Y}^{\mathrm{final}}$.

All point forecasting parameters are frozen, and only LGEC-U parameters are updated. Therefore, uncertainty optimization cannot modify point forecasting performance.

For the potential self-retrieval issue during LGEC-U training, we apply the exact-match exclusion mechanism described in Appendix D.7.

E.7 Validation-Set Scale Calibration

After training LGEC-U, we perform empirical conformal-style scale calibration on the validation set.

For each validation position,

$$r_i = \frac{|y_i - \hat{y}_i^{\mathrm{ref}}|}{z\sigma_i}. \tag{105}$$

Given target coverage $1 - \alpha_{\mathrm{cov}}$, the scale multiplier is obtained from the empirical quantile:

$$m_u = q_{1-\alpha_{\mathrm{cov}}}. \tag{106}$$

The calibrated scale is

$$\tilde{\sigma} = m_u \sigma. \tag{107}$$

We additionally support shrink calibration:

$$m_u = 1 + \eta(q_{1-\alpha_{\mathrm{cov}}} - 1), \tag{108}$$

where $\eta \in [0, 1]$ controls calibration strength.

The resulting interval is

$$[\hat{Y}^{\mathrm{ref}} - z\tilde{\sigma}, \hat{Y}^{\mathrm{ref}} + z\tilde{\sigma}], \tag{109}$$

where z denotes the standard Gaussian quantile associated with the target coverage (e.g., 1.645 for 90% and 1.96 for 95% coverage). For the complete model, $\hat{Y}^{\mathrm{ref}} = \hat{Y}^{\mathrm{final}}$.

This procedure is referred to as conformal-style calibration rather than strict conformal prediction because it rescales interval width using empirical validation residual ratios without claiming finite-sample coverage guarantees.

E.8 Calibration on High-Dimensional Datasets

For high-dimensional datasets, validation calibration ratios may become large:

$$N_{\mathrm{val}} \times H \times C. \tag{110}$$

To reduce memory usage, we estimate empirical quantiles using at most

$$2 \times 10^6 \tag{111}$$

sampled calibration ratios.

When fewer ratios exist, all values are used. Sampling affects only quantile estimation; final calibration statistics are recomputed using the complete validation set.

E.9 Complementary Risk Analysis within LGEC Gate Groups

To analyze the complementarity between LGEC correction utility and LGEC-U risk estimation, we use the pre-RARM setting from Section 4.6.

Test positions are divided into High-G, Mid-G, and Low-G groups according to the LGEC gate. Within each group, we compute the correlation between the LGEC-U scale and remaining absolute error after LGEC correction.

Table 6 reports the average predicted scale and within-group Spearman correlation.

The positive correlations across groups indicate that LGEC-U retains additional risk-ranking information even after approximately controlling for empirical correctability.

The LGEC gate therefore characterizes correction utility, whereas the LGEC-U scale characterizes remaining predictive risk after correction.

Table 6: Remaining-risk analysis of LGEC-U within LGEC gate groups. Avg. σ denotes the mean predicted LGEC-U scale within each group, and Risk Corr. ρ denotes the within-group Spearman correlation between σ and the remaining absolute error after LGEC correction.

Dataset	Gate Group	Avg. σ	Risk Corr. $\rho \uparrow$
Traffic	High-G	0.352	**0.488**
	Mid-G	0.333	0.391
	Low-G	0.312	0.236
Electricity	High-G	0.322	0.282
	Mid-G	0.275	**0.292**
	Low-G	0.214	0.269

Table 7: Statistics and forecasting settings of the long-horizon time-series forecasting benchmarks.

Dataset	Dim.	Prediction Length H	Train / Val / Test Size	Sampling Rate	Domain
ETTh1, ETTh2	7	$\{96, 192, 336, 720\}$	(8545, 2881, 2881)	1 hour	Temperature
ETTm1, ETTm2	7	$\{96, 192, 336, 720\}$	(34465, 11521, 11521)	15 min	Temperature
Weather	21	$\{96, 192, 336, 720\}$	(36792, 5271, 10540)	10 min	Weather
Traffic	862	$\{96, 192, 336, 720\}$	(12185, 1757, 3509)	1 hour	Traffic
Electricity	321	$\{96, 192, 336, 720\}$	(18317, 2633, 5261)	1 hour	Electricity

F Experimental Details and Reproducibility

F.1 Dataset Statistics and Forecasting Protocol

We evaluate EUD-RM on seven widely used long-horizon multivariate time-series forecasting benchmarks: **ETTh1, ETTh2, ETTm1, ETTm2, Weather, Traffic, and Electricity**. These datasets cover different sampling frequencies, variable dimensionalities, and application domains, allowing us to evaluate the model under both conventional and high-dimensional multivariate forecasting settings. Table 7 summarizes the dataset statistics and forecasting settings.

Given a historical observation sequence of length L, the model predicts the subsequent H time steps. We consider four forecasting horizons for all datasets,

$$H \in \{96, 192, 336, 720\}, \tag{112}$$

to evaluate long-horizon forecasting performance across different prediction ranges. **Traffic uses an input length of $L = 720$, while all other datasets use $L = 512$.**

In terms of dimensionality, the ETT datasets contain 7 variables, Weather contains 21 variables, while Electricity and Traffic contain 321 and 862 variables, respectively. The latter two therefore additionally evaluate the applicability of EUD-RM to high-dimensional multivariate forecasting. The datasets span sampling intervals from 15 minutes and 10 minutes to 1 hour, covering multiple temporal resolutions.

For all comparisons, methods evaluated on the same dataset use the same historical input length, forecasting horizons, and train/validation/test splits. Point forecasting performance is evaluated separately at $H \in \{96, 192, 336, 720\}$ to ensure consistent comparison across methods. Detailed baseline configurations, training settings, and the staged training protocol of EUD-RM are provided in the subsequent appendices.

F.2 Hyperparameter Search and Configuration Selection

We select hyperparameter configurations of the proposed EUD-RM modules using the validation set following the standard long-horizon forecasting protocol. The test set is never used for hyperparameter selection or model configuration.

Since the primary contribution of EUD-RM lies in forecasting failure handling, the hyperparameter selection focuses on the proposed modules, including CAM, LGEC, RARM, and LGEC-U. The

Table 8: Hyperparameter configurations of EUD-RM modules.

Module	Hyperparameter	Candidates	Selected
CAM	Channel mixing strategy	fixed / automatic	automatic
LGEC	Maximum correction scale	{0.02,0.05,0.1}	0.05
LGEC	Residual smoothing window	{3,5,7}	5
LGEC	Teacher coefficient β_t	{0.5,0.7,0.9}	0.7
LGEC	Gate initial logit	fixed	-3.5
RARM	Top-K	{8,16,32,64,128}	16
RARM	Key length	fixed	48
RARM	Pooled temporal dimension D_k	fixed	16
RARM	Retrieval temperature τ	fixed	0.10
LGEC-U	Scale calibration	{none, calibrated}	calibrated

base forecasting backbone follows the configuration described in Appendix A and is not included in the module-level configuration selection.

For CAM, the channel interaction strategy is automatically selected according to the input dimensionality. Specifically, full channel mixing is applied to moderate-dimensional datasets, while lightweight channel modulation is adopted for high-dimensional multivariate forecasting settings.

For LGEC, we select correction-related configurations, including the maximum correction scale, residual smoothing window, teacher coefficient, and gate initialization. These parameters control the magnitude, stability, and selectivity of the parametric correction process.

For RARM, we consider retrieval-related configurations, including the number of retrieved neighbors, retrieval key length, pooled temporal dimension, and retrieval temperature. The selected configuration balances retrieval evidence quality and online inference overhead.

For LGEC-U, we evaluate uncertainty-scale calibration strategies and use validation-set scale calibration in the final experiments to align predictive interval magnitude with the target coverage level.

Table 8 summarizes the candidate configurations and selected values used in our experiments. Unless otherwise specified, the selected configurations are consistently applied across datasets.

G Full Experimental Results

G.1 Full Comparison with Recent Time-Series Forecasting Methods

To provide a more complete evaluation of EUD-RM across forecasting horizons, we compare it with recent representative methods on seven multivariate forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Traffic, and Electricity.

The compared methods include OccamVTS (2026), PhaseFormer (2026), TimeMixer++ (2025), Amplifier (2025), Times2D (2025), RAFT (2025), AMD (2025), MOMENT (2024), and iTransformer (2024). All methods are evaluated at $H \in \{96, 192, 336, 720\}$, and both MSE and MAE are reported. Table 9 provides the complete horizon-wise results.

G.2 Full Module Ablation Results

Table 10 reports the complete horizon-wise ablation results on the four ETT datasets. These results complement the averaged comparison in Table 2 by showing the behavior of each module at individual prediction horizons.

Across datasets and horizons, removing CAM, LGEC, or RARM generally increases forecasting error, although individual ablated settings can occasionally match or outperform the full model on a particular MAE value. The overall pattern is more stable in MSE and in the horizon-averaged results: the full model achieves the lowest average MSE on all four datasets. Combined ablations typically produce larger degradation, further supporting the complementary contributions of cross-variable representation enhancement, parametric error correction, and historical residual reuse.

Table 9: Full comparison of EUD-RM with recent time-series forecasting methods on long-horizon multivariate forecasting benchmarks. All methods are evaluated at prediction lengths $H \in \{96, 192, 336, 720\}$. Lower is better for both MSE and MAE. Avg. denotes the average over the four prediction horizons. Best results are shown in **bold**, and second-best results are underlined.

Dataset	H	Ours		2026				2025										2024			
		EUD-RM		OccamVTS		PhaseFormer		TimeMixer++		Amplifier		Times2D		RAFT		AMD		MOMENT		iTransformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.339**	**0.376**	0.360	0.388	0.359	0.382	0.361	0.403	0.371	0.392	0.362	0.395	0.367	0.397	0.369	0.397	0.387	0.410	0.386	0.405
	192	**0.379**	0.406	0.396	0.410	0.397	**0.404**	0.416	0.441	0.426	0.422	0.405	0.423	0.411	0.427	0.401	0.417	0.410	0.426	0.441	0.436
	336	**0.371**	**0.405**	0.416	0.424	0.425	0.424	0.430	0.434	0.448	0.434	0.423	0.433	0.436	0.442	0.418	0.427	0.442	0.437	0.487	0.458
	720	**0.394**	**0.426**	0.440	0.462	0.431	0.450	0.467	0.451	0.476	0.464	0.446	0.467	0.467	0.478	0.439	0.454	0.432	0.472	0.503	0.491
	Avg	**0.371**	**0.403**	0.403	0.421	0.403	0.415	0.419	0.432	0.430	0.428	0.409	0.430	0.420	0.436	0.407	0.424	0.418	0.436	0.454	0.448
ETTh2	96	**0.251**	**0.319**	0.266	0.330	0.275	0.338	0.276	0.328	0.279	0.337	0.270	0.334	0.276	0.344	0.274	0.337	0.288	0.345	0.297	0.349
	192	**0.307**	**0.361**	0.330	0.372	0.341	0.376	0.342	0.379	0.359	0.389	0.337	0.381	0.347	0.393	0.351	0.383	0.349	0.386	0.380	0.400
	336	**0.314**	**0.377**	0.357	0.400	0.360	0.405	0.336	0.398	0.377	0.406	0.329	0.385	0.376	0.425	0.375	0.411	0.369	0.408	0.428	0.432
	720	**0.373**	**0.415**	0.389	0.431	0.412	0.436	0.402	**0.415**	0.420	0.432	0.376	0.422	0.436	0.473	0.402	0.438	0.403	0.439	0.427	0.445
	Avg	**0.311**	**0.368**	0.336	0.383	0.347	0.389	0.339	0.380	0.359	0.391	0.328	0.381	0.359	0.409	0.351	0.392	0.352	0.395	0.383	0.407
ETTm1	96	**0.273**	**0.330**	0.292	0.343	0.293	0.344	0.310	0.334	0.316	0.355	0.285	0.340	0.302	0.349	0.284	0.339	0.293	0.349	0.334	0.368
	192	**0.316**	**0.358**	0.324	0.362	0.323	0.361	0.348	0.362	0.361	0.381	0.323	0.364	0.329	0.367	0.322	0.362	0.326	0.368	0.377	0.391
	336	**0.345**	0.382	0.360	**0.378**	0.358	0.381	0.374	0.391	0.393	0.402	0.350	0.381	0.355	0.383	0.361	0.383	0.352	0.379	0.406	0.420
	720	0.398	**0.408**	0.413	0.410	0.412	0.410	0.444	0.423	0.455	0.440	**0.391**	0.411	0.406	0.413	0.421	0.418	0.405	0.416	0.511	0.459
	Avg	**0.333**	**0.370**	0.347	0.373	0.347	0.374	0.369	0.378	0.381	0.394	0.337	0.374	0.348	0.378	0.347	0.376	0.344	0.379	0.407	0.410
ETTm2	96	**0.157**	**0.244**	0.160	0.248	0.167	0.256	0.170	0.245	0.176	0.258	0.165	0.258	0.164	0.260	0.167	0.258	0.169	0.260	0.180	0.264
	192	**0.213**	**0.281**	0.214	0.286	0.219	0.293	0.229	0.291	0.239	0.300	0.221	0.295	0.219	0.296	0.221	0.295	0.227	0.297	0.250	0.309
	336	0.266	0.323	**0.264**	**0.319**	0.269	0.326	0.303	0.343	0.297	0.338	0.271	0.331	0.275	0.336	0.270	0.327	0.275	0.328	0.311	0.348
	720	0.345	**0.373**	**0.341**	0.374	0.412	0.410	0.373	0.399	0.393	0.396	0.347	0.381	0.359	0.392	0.356	0.382	0.363	0.387	0.412	0.407
	Avg	**0.245**	**0.305**	**0.245**	0.307	0.266	0.321	0.269	0.320	0.276	0.323	0.251	0.316	0.254	0.320	0.254	0.316	0.259	0.318	0.288	0.332
Weather	96	**0.141**	**0.192**	0.145	0.194	0.148	0.197	0.154	0.204	0.156	0.203	0.143	0.193	0.166	0.223	0.145	0.200	0.154	0.211	0.175	0.215
	192	**0.185**	**0.232**	0.191	0.236	0.193	0.237	0.201	0.245	0.209	0.249	0.191	0.241	0.211	0.264	0.190	0.240	0.197	0.248	0.221	0.254
	336	**0.236**	0.274	0.243	0.277	0.242	0.278	0.237	**0.265**	0.264	0.290	0.243	0.282	0.260	0.302	0.242	0.280	0.246	0.285	0.278	0.296
	720	0.313	0.334	0.317	**0.329**	**0.309**	0.332	0.312	0.334	0.343	0.342	0.315	0.336	0.327	0.355	0.315	0.332	0.315	0.336	0.358	0.347
	Avg	**0.219**	**0.258**	0.224	0.259	0.223	0.261	0.226	0.262	0.243	0.271	0.223	0.263	0.241	0.286	0.223	0.263	0.228	0.270	0.258	0.278
Traffic	96	**0.351**	0.244	0.377	0.268	0.361	**0.238**	0.392	0.253	0.455	0.298	0.358	0.244	0.378	0.272	0.366	0.259	0.391	0.282	0.395	0.268
	192	**0.368**	0.258	0.396	0.276	0.373	**0.243**	0.402	0.258	0.470	0.316	0.375	0.255	0.391	0.277	0.381	0.265	0.404	0.287	0.417	0.276
	336	**0.384**	0.270	0.409	0.277	0.385	**0.248**	0.428	0.263	0.479	0.316	0.386	0.272	0.402	0.282	0.397	0.273	0.414	0.292	0.433	0.283
	720	**0.421**	0.286	0.446	0.297	0.428	**0.270**	0.441	0.282	0.523	0.328	0.427	0.290	0.434	0.297	0.430	0.292	0.450	0.310	0.467	0.302
	Avg	**0.381**	0.265	0.407	0.279	0.387	**0.250**	0.416	0.264	0.482	0.315	0.387	0.265	0.401	0.282	0.394	0.272	0.415	0.293	0.428	0.282
Electricity	96	**0.127**	**0.221**	0.136	0.235	0.133	**0.221**	0.135	0.222	0.147	0.243	0.128	**0.221**	0.133	0.232	0.129	0.225	0.136	0.233	0.148	0.240
	192	**0.141**	0.236	0.151	0.248	0.148	0.238	0.147	0.235	0.157	0.251	0.145	**0.227**	0.149	0.247	0.148	0.242	0.152	0.247	0.162	0.253
	336	**0.151**	0.246	0.166	0.263	0.165	0.257	0.164	**0.245**	0.174	0.269	0.165	0.252	0.161	0.259	0.164	0.258	0.167	0.264	0.178	0.269
	720	**0.194**	0.293	0.196	0.290	0.201	**0.285**	0.212	0.310	0.206	0.296	**0.194**	0.295	0.197	0.297	0.195	0.289	0.205	0.295	0.225	0.317
	Avg	**0.153**	**0.249**	0.162	0.259	0.161	0.250	0.165	0.253	0.171	0.265	0.158	**0.249**	0.160	0.259	0.159	0.254	0.165	0.260	0.178	0.270

H Additional Mechanism Analysis

H.1 Computational Efficiency and Runtime Analysis

Beyond predictive accuracy and error-handling behavior, we evaluate the computational and runtime cost of EUD-RM across datasets and forecasting horizons. Since each dataset–horizon pair uses its selected task-specific configuration, the results in this section characterize the practical efficiency of these dataset- and horizon-specific configurations rather than the complexity obtained by varying only the prediction length under a fixed architecture.

All efficiency experiments are conducted on an NVIDIA A800 80GB PCIe GPU using PyTorch 2.4.0, CUDA 11.8, and automatic mixed precision (AMP). We use a batch size of 256 for all configurations. Each profiling run is preceded by warm-up iterations followed by timed iterations for stable measurement.

FLOPs are reported per sample. Network latency measures the neural point-forecasting path, while RARM latency measures the additional online residual-memory retrieval and correction performed on the fixed point forecast. Full latency represents the complete online inference path including both components.

Because RARM operates outside the neural-network forward pass, its retrieval operations are not included in network FLOPs and are instead reported separately through latency and memory measurements.

Table 10: Complete horizon-wise ablation results for the core components of EUD-RM on the four ETT datasets. Full denotes the complete point forecasting model; the remaining settings remove the indicated module(s). Lower MSE and MAE are better. Best results for each dataset–horizon setting are shown in **bold**.

Dataset	H	Full		w/o CAM		w/o LGEC		w/o RARM		w/o CAM+RARM		w/o LGEC+RARM	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.339**	**0.376**	0.344	0.379	0.342	0.380	0.343	0.380	0.348	0.383	0.346	0.384
	192	**0.379**	**0.406**	0.381	0.408	0.381	0.408	0.381	0.409	0.384	0.410	0.384	0.410
	336	**0.371**	**0.405**	0.375	0.406	0.384	0.421	0.373	0.406	0.376	0.407	0.400	0.432
	720	**0.394**	**0.426**	0.398	0.428	0.401	0.430	0.398	0.428	0.399	0.429	0.406	0.434
	Avg.	**0.371**	**0.403**	0.375	0.405	0.377	0.410	0.374	0.406	0.377	0.407	0.384	0.415
ETTh2	96	**0.251**	**0.319**	0.261	0.326	0.257	0.324	0.253	0.320	0.263	0.327	0.259	0.325
	192	**0.307**	**0.361**	0.315	0.362	0.313	**0.361**	0.308	0.362	0.318	0.363	0.315	0.362
	336	**0.314**	0.377	0.318	0.378	0.317	0.377	0.315	0.378	0.319	**0.369**	0.318	0.377
	720	**0.373**	**0.415**	0.383	0.422	0.381	0.421	0.375	0.417	0.385	0.423	0.382	0.422
	Avg.	**0.311**	**0.368**	0.319	0.372	0.317	0.371	0.313	0.369	0.321	0.371	0.319	0.372
ETTm1	96	**0.273**	**0.330**	0.287	0.339	0.278	0.335	0.274	**0.330**	0.288	0.339	0.283	0.336
	192	**0.316**	**0.358**	0.326	0.365	0.328	0.368	0.317	0.359	0.328	0.367	0.326	0.369
	336	**0.345**	0.382	0.349	0.385	0.347	**0.381**	0.346	0.383	0.350	0.386	0.354	0.382
	720	**0.398**	**0.408**	0.404	0.411	0.404	0.413	0.401	0.409	0.414	0.413	0.411	0.414
	Avg.	**0.333**	**0.370**	0.342	0.375	0.339	0.374	0.335	**0.370**	0.345	0.376	0.344	0.375
ETTm2	96	**0.157**	**0.244**	0.162	0.250	0.161	0.249	0.159	0.246	0.163	0.251	0.167	0.255
	192	**0.213**	**0.281**	0.219	0.290	0.215	0.286	0.214	0.282	0.222	0.291	0.218	0.290
	336	**0.266**	0.323	0.273	0.326	0.272	0.323	0.270	**0.322**	0.275	0.327	0.273	0.324
	720	**0.345**	**0.373**	0.356	0.384	0.349	0.379	0.346	**0.373**	0.361	0.386	0.358	0.381
	Avg.	**0.245**	**0.305**	0.253	0.313	0.249	0.309	0.247	0.306	0.255	0.314	0.254	0.313

Table 11: Computational efficiency of EUD-RM under dataset- and horizon-specific configurations. FLOPs are reported per sample, while latency is measured per batch with batch size 256. Peak GPU memory denotes the additional peak memory during full online inference.

Dataset	H	FLOPs (G/sample)	Net Lat. (ms/batch)	RARM Lat. (ms/batch)	Full Lat. (ms/batch)	Peak GPU Mem. (MB)	RARM Mem. (MB)
ETTh1	96	2.028	105.557	44.114	149.992	553.5	24.5
	192	1.333	104.442	43.825	148.978	384.1	35.0
	336	0.870	106.344	43.820	150.893	300.4	50.8
	720	0.975	62.095	43.960	106.405	638.4	92.8
ETTh2	96	1.320	96.581	43.824	140.584	382.9	24.5
	192	1.987	104.803	43.889	148.772	554.1	35.0
	336	0.265	73.956	43.992	118.098	296.3	50.8
	720	0.415	102.590	43.673	147.349	631.5	92.8
ETTm1	96	0.247	11.932	43.876	56.315	131.7	24.5
	192	0.148	20.935	43.850	63.035	170.0	35.0
	336	0.566	30.234	43.529	74.401	301.0	57.8
	720	0.159	8.864	43.739	52.581	191.3	99.8
ETTm2	96	0.610	29.847	43.556	63.880	188.7	24.5
	192	0.543	20.058	43.918	64.582	175.4	35.0
	336	0.400	29.448	43.828	63.617	134.7	50.8
	720	0.664	83.694	44.713	128.177	636.3	92.8
Electricity	96	1.352	10.723	4.542	14.832	705.2	123.0
	192	2.154	23.061	38.500	61.561	589.4	969.0
	336	2.405	16.678	52.700	69.378	1007.6	1469.9
	720	2.874	26.713	56.483	81.477	7077.1	452.4

Table 11 reports the computational efficiency of EUD-RM under different dataset and horizon configurations. The results include both standard low-dimensional forecasting datasets and the high-dimensional Electricity dataset, demonstrating the computational behavior of EUD-RM under different forecasting scales.

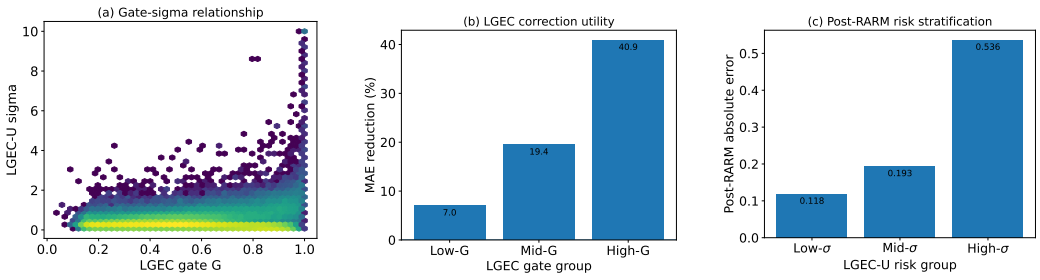

Figure 5: Additional mechanism analysis of EUD-RM. (a) Relationship between the LGEC gate and LGEC-U scale, showing the complementarity between correction utility and remaining predictive risk. (b) LGEC correction gain across different LGEC gate groups, verifying that the gate identifies positions with higher empirical correctability. (c) Remaining predictive risk across LGEC-U risk groups after deterministic correction, demonstrating the risk ranking capability of LGEC-U.

H.2 Additional Analysis of Error Correctability and Predictive Risk

This section provides additional analyses to further investigate the functional roles of the correction-oriented and uncertainty-oriented components in EUD-RM. Specifically, we study whether the LGEC gate captures empirical correction utility, whether LGEC-U provides complementary information beyond the correction decision, and whether the predicted uncertainty scale can effectively rank the remaining predictive risk after deterministic correction.

H.2.1 Complementarity Between Correction Utility and Predictive Risk

The LGEC gate and LGEC-U scale characterize different aspects of forecasting failures. The LGEC gate estimates whether a forecasting position contains reliable evidence for deterministic correction, while the LGEC-U scale measures the remaining predictive risk after applying available corrections. Therefore, these two signals are expected to provide complementary rather than redundant information.

To examine this relationship, we first analyze the correlation between the LGEC gate and LGEC-U scale. As shown in Figure 5(a), the association between the two signals is relatively weak, indicating that correction utility and predictive risk represent different properties of forecasting errors.

H.2.2 Gate-based Analysis of LGEC Correction Utility

We further evaluate whether the LGEC gate provides meaningful guidance for selective correction. Forecast positions are divided into coarse groups according to their inference-time gate values. For each group, we compute the correction gain obtained after applying LGEC.

Figure 5(b) shows that positions with larger gate values generally achieve larger correction gains. This result indicates that LGEC learns to allocate stronger corrections to forecasting failures with higher empirical correctability, instead of uniformly modifying all prediction positions.

H.2.3 Risk Ranking Ability of LGEC-U

Finally, we evaluate whether LGEC-U can characterize the remaining predictive risk after deterministic correction. Forecast positions are grouped according to their predicted LGEC-U scales, and the corresponding remaining absolute errors are compared across risk groups.

As shown in Figure 5(c), groups with larger LGEC-U scales exhibit higher remaining prediction errors. This monotonic relationship demonstrates that LGEC-U provides effective risk ranking after correction and captures uncertainty information that is not directly encoded by the LGEC gate.

Table 12: Random seed stability analysis on ETT datasets. Results are averaged over 5 experiments with different random seeds. We report the mean and standard deviation of MSE and MAE.

Horizon	ETTh1		ETTh2		ETTm1		ETTm2	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.343 ± 0.002	0.378 ± 0.001	0.258 ± 0.001	0.324 ± 0.001	0.278 ± 0.003	0.332 ± 0.001	0.159 ± 0.002	0.246 ± 0.002
192	0.382 ± 0.002	0.406 ± 0.001	0.310 ± 0.002	0.363 ± 0.003	0.323 ± 0.003	0.363 ± 0.002	0.216 ± 0.002	0.285 ± 0.002
336	0.377 ± 0.005	0.408 ± 0.003	0.317 ± 0.005	0.375 ± 0.002	0.350 ± 0.003	0.383 ± 0.002	0.271 ± 0.002	0.326 ± 0.002
720	0.404 ± 0.008	0.432 ± 0.005	0.379 ± 0.005	0.419 ± 0.003	0.402 ± 0.002	0.409 ± 0.000	0.375 ± 0.001	0.387 ± 0.001

Together, these analyses provide additional evidence that EUD-RM separates three different aspects of forecasting failures: LGEC identifies errors that are worth correcting, RARM exploits historically reproducible failures, and LGEC-U quantifies the remaining predictive risk after correction.

H.3 Random Seed Stability Analysis

Deep forecasting models can exhibit performance variations due to random initialization, data shuffling, and stochastic optimization. To evaluate the stability of EUD-RM under different random seeds, we repeat the experiments on the four ETT benchmarks using five different random seeds while keeping the dataset splits, forecasting horizons, and model configurations unchanged.

Table 12 reports the mean and standard deviation of MSE and MAE over the five independent runs. The results show that the performance variation across different seeds remains relatively small on all evaluated datasets and horizons. These observations indicate that the reported forecasting performance of EUD-RM is not dependent on a particular random initialization.

I Model Scope and Failure Modes

EUD-RM is designed for supervised long-horizon multivariate forecasting where forecasting failures exhibit exploitable empirical structure. Its error-handling mechanisms rely on empirical correctability, historical residual reproducibility, and validation-based risk calibration rather than on a strict decomposition of forecasting errors into deterministic and stochastic components. We summarize the main model boundaries and potential failure modes below.

Limited empirical correctability. LGEC is effective when residual structures observed during training provide reliable evidence for correction. Errors that are weak, rapidly varying, or not predictable from the available history and prediction state may receive low gate values and remain uncorrected. This behavior is intentional: the learnability target is an empirical proxy for correction utility rather than a guarantee that all structured errors are identifiable or correctable. The bounded correction and safety constraints further prioritize conservative correction over aggressively fitting unstable residual fluctuations.

Insufficient historical reproducibility. RARM relies on the existence of similar forecasting states whose historical residuals provide consistent correction evidence. It may offer little benefit when a query corresponds to a novel regime, when retrieved states have inconsistent residuals, or when similar input states lead to substantially different forecasting failures. RARM therefore combines retrieval similarity, residual consistency, and dispersion to attenuate unreliable evidence. Moreover, if calibrated RARM does not improve validation performance, its correction strength is set to zero, reducing the model to the LGEC point forecaster.

Dependence on validation-based calibration. Both the global influence of RARM and the scale of LGEC-U are calibrated on the validation set. Consequently, their effectiveness depends on the extent to which validation residual statistics remain representative at inference time. Under substantial distribution shift, the calibrated correction strength or interval scale may transfer imperfectly. In particular, the LGEC-U procedure is conformal-style calibration rather than strict conformal prediction and therefore does not claim finite-sample coverage guarantees.

Residual-memory cost and scalability. RARM introduces additional storage and retrieval cost beyond the neural forecasting path. The implementation bounds memory capacity, applies reser-

voir sampling, supports CPU fallback, and chunks similarity computation to control resource usage. Nevertheless, residual retrieval may become a practical bottleneck for very large memories or latency-sensitive applications. For high-dimensional series, EUD-RM therefore uses a compact global retrieval representation to reduce retrieval overhead.

Evaluation scope. Our experiments focus on seven standard long-horizon multivariate forecasting benchmarks with prediction horizons $H \in \{96, 192, 336, 720\}$. The current evaluation therefore does not establish the same behavior for all forecasting regimes, such as substantially different sampling processes, extreme distribution shifts, or application-specific deployment constraints. Extending residual-memory retrieval and uncertainty calibration to broader and more latency-sensitive settings remains an important direction for future work.