ARBR: ADAPTIVE RADIAL BASIS REPRESENTATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The temporal location and extent of predictive patterns are important for time series forecasting. Recent basis-based forecasting methods often combine shared temporal bases with input-dependent coefficients. However, adapting coefficients alone cannot directly relocate or rescale individual basis functions, so changes in pattern location and extent must be represented through coordinated changes across multiple shared bases. To model such geometric variation explicitly, we propose Adaptive Radial Basis Representation (ARBR), a structured temporal representation built from input-adaptive Gaussian radial bases. For each channel-wise input window, ARBR predicts bounded adjustments to radial centers and scales around a shared temporal scaffold, preserving the temporal ordering of components. Radial responses weight input-conditioned vector-valued coefficients to form latent features aligned with temporal patches. On seven multivariate forecasting benchmarks, ARBR achieves the lowest MSE on five datasets and the lowest MAE on four, averaged over four forecasting horizons. Controlled comparisons show that ARBR yields lower forecasting errors than variants with fixed or globally learned radial geometry or non-radial dynamic mixing. The code is available at https://anonymous.4open.science/r/ARBR/.

1 INTRODUCTION

Time series forecasting supports decision-making in areas such as energy management (Hong & Fan, 2016), weather forecasting (Lam et al., 2023), and traffic planning (Vlahogianni et al., 2014). The representation of temporal structure plays an important role in forecasting. Recent forecasting models use mechanisms such as attention (Zhou et al., 2021; Liu et al., 2024), patching (Nie et al., 2023), decomposition (Wu et al., 2021), and multi-scale processing (Wu et al., 2023; Wang et al., 2024) to model temporal structure. Predictive patterns do not always occur at the same temporal location or have the same temporal extent across input windows. This motivates representations that capture predictive content together with its temporal location and extent.

Structured basis representations describe temporal structure through basis components and their associated coefficients. N-BEATS uses basis expansion (Oreshkin et al., 2020), and N-HiTS incorporates hierarchical interpolation into this framework (Challu et al., 2023). BasisFormer combines shared temporal bases with input-dependent coefficients (Ni et al., 2023), and TimeBase extracts compact temporal basis components (Huang et al., 2025). However, many basis-based forecasting methods adapt their coefficients to each input while keeping the basis functions shared across inputs. For a localized basis component, we refer to its location and temporal extent as its temporal geometry. Changing coefficients can redistribute content across shared basis functions, but cannot relocate or rescale an individual basis function. Input-adaptive geometry makes these adjustments explicit, allowing each radial component to change its temporal location and extent.

Input-adaptive geometry raises the question of how to retain a common temporal reference across input windows. Independently generated centers need not preserve the temporal ordering of basis components. We therefore constrain input-specific changes in radial centers and scales around a shared temporal scaffold. This preserves a consistent temporal ordering of component indices across inputs while allowing their locations and scales to adapt.

We introduce Adaptive Radial Basis Representation (ARBR), a structured temporal representation built from input-adaptive Gaussian radial bases. The centers and scales control the temporal loca-

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.

tion and extent of the radial responses. For each channel-wise input window, ARBR also generates vector-valued coefficients that encode the latent content of each radial component. Radial responses weight these coefficients to form latent features aligned with temporal patches. A local residual pathway supplements the radial representation with patch-level features. Across seven multivariate forecasting benchmarks, ARBR achieves the lowest average MSE on five datasets over four forecasting horizons. Controlled comparisons show lower forecasting errors with input-adaptive radial geometry than with fixed geometry, globally learned geometry, or non-radial dynamic mixing. Under controlled temporal perturbations, centers respond systematically to temporal shifts, and scales respond to compression and dilation. Our contributions are as follows:

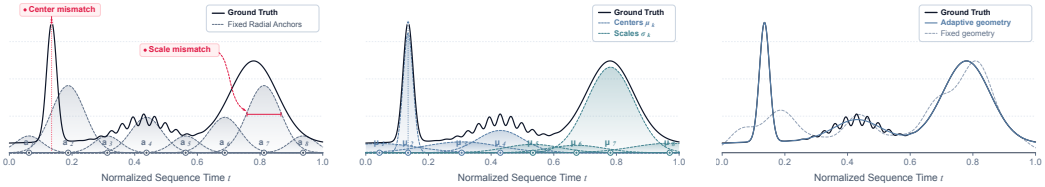
- We propose ARBR, a temporal representation built from input-adaptive Gaussian radial components. Their centers and scales control temporal location and extent, while vector-valued coefficients encode latent content.
- We parameterize radial centers and scales through bounded deformations of a shared temporal scaffold. This allows their geometry to adapt to each input while preserving the temporal ordering of radial components.
- We evaluate ARBR on seven multivariate forecasting benchmarks and four forecasting horizons. Controlled comparisons support the benefits of input-adaptive geometry, and temporal perturbations reveal systematic responses in radial centers and scales.

2 Related Work

Basis-Based Temporal Representations. N-BEATS constructs forecasts through basis expansion (Oreshkin et al., 2020), and N-HiTS uses hierarchical interpolation to assemble forecasts at multiple temporal resolutions (Challu et al., 2023). TimeBase extracts compact temporal basis components for segment-level forecasting (Huang et al., 2025). FWBNet integrates multi-scale temporal features through adaptive basis selection and frequency–wavelet modeling (Lai & You, 2025). BasisFormer learns shared temporal bases and computes input-dependent coefficients to select and combine them for forecasting (Ni et al., 2023). With this shared basis set, changes in pattern location or duration must be represented through different combinations of existing basis functions. This can require coordinating multiple components to represent a shifted or stretched local pattern. ARBR instead allows each radial component to adjust its own temporal location and extent through input-conditioned centers and scales, alongside input-conditioned vector-valued coefficients.

Adaptive Radial Geometry. Earlier work on time series prediction has explored adjusting the centers and scales of radial basis functions (Whitehead & Choate, 1996; Liu et al., 2020). In the GRBF predictor of Liu et al. (2020), Gaussian responses depend on distances between differenced input vectors and centers in the same feature space. TimeGS constructs composite kernels at fixed positions in two-dimensional period–phase coordinates by predicting mixing weights over a frozen Gaussian basis bank and scalar kernel intensities (Wang et al., 2026). It renders these kernels over the forecasting horizon and aggregates the rendered outputs into predictions. ARBR instead uses Gaussian radial responses to represent the historical window along its one-dimensional temporal axis. Evaluated at patch-aligned coordinates, these responses weight vector-valued coefficients to form latent features for a forecasting head.

Input-Adaptive Temporal Modeling. Forecasting models address input variation through instance-specific normalization and non-stationarity modeling (Kim et al., 2022; Liu et al., 2022). Other approaches adapt computational pathways (Chen et al., 2024), filter spatial-temporal dependencies (Hu et al., 2025), or learn query-specific attention metrics (Kim et al., 2026). These approaches differ in which aspects of the forecasting model are conditioned on the input. Conditional parameter generation offers a way to make this choice explicit: hypernetworks generate network parameters (Ha et al., 2017), and dynamic filter networks generate input-conditioned filters (Jia et al., 2016). ARBR applies this principle to a structured radial representation, generating input-conditioned centers, scales, and vector-valued coefficients that specify temporal location, extent, and latent content, respectively.



(a) Fixed geometry. (b) Adaptive geometry. (c) Fit comparison.

Figure 1: Illustration of adaptive radial geometry. (a) Fixed radial components exhibit center and scale mismatches with local temporal structure. (b) Input-conditioned centers and scales adapt the locations and temporal extents of the radial responses. (c) Adaptive geometry provides a closer fit to the illustrated signal than fixed geometry.

3 Adaptive Radial Basis Representation

For an input window x from a single channel, ARBR represents temporal structure using K Gaussian radial components. Each component k has an input-conditioned center $\mu_k(x)$, a positive scale $\sigma_k(x)$, and a vector-valued coefficient $V_k(x) \in \mathbb{R}^D$. The center and scale define the component's temporal geometry, while the coefficient carries its latent content. Centers and scales adapt through bounded adjustments around a shared temporal scaffold. Figure 1 illustrates the effect of adapting radial geometry.

Let $q_p \in [0, 1]$ denote the normalized temporal coordinate of the center of patch p, for $p = 1, \ldots, P$, where P is the number of patches. The radial response of component k at q_p is

$$B_{pk}(x) = \exp\left(-\frac{(q_p - \mu_k(x))^2}{2\sigma_k^2(x)}\right). \tag{1}$$

Collecting these responses gives $B_{\mathrm{rad}}(x) \in \mathbb{R}^{P\times K}$, with $[B_{\mathrm{rad}}(x)]_{pk} = B_{pk}(x)$. ARBR combines the vector-valued coefficients using these unnormalized radial responses as weights:

$$G_p(x) = \sum_{k=1}^{K} B_{pk}(x)V_k(x), \qquad p = 1, \ldots, P. \tag{2}$$

Let $V(x) \in \mathbb{R}^{K\times D}$ collect the component coefficients as rows. Stacking the patch representations $G_p(x)$ as rows gives

$$G(x) = B_{\mathrm{rad}}(x)V(x) \in \mathbb{R}^{P\times D}. \tag{3}$$

In this factorization, radial geometry determines how each component contributes across temporal positions, while its coefficient specifies the latent content distributed over those positions.

4 Methodology

Figure 2 presents the forecasting architecture built on the radial representation defined in Section 3. For each input window from a single channel, a sequence encoder produces a conditioning feature shared by the geometry and coefficient generators. Adaptive Radial Geometry (ARG) generates input-conditioned centers and scales and evaluates the resulting radial responses, while Radial Coefficient Generation (RCG) produces the vector-valued coefficients. Radial rendering combines the responses and coefficients into the patch-aligned representation G. Constrained Local Residual Correction (CLRC) adds local patch features to G, and a forecasting head maps the resulting representation to future values.

4.1 Temporal Input Encoding

Given a mini-batch $X \in \mathbb{R}^{B\times L\times C}$ with batch size B, input length L, and C channels, we apply RevIN to normalize each channel of each input window (Kim et al., 2022). We collect the $N = BC$ normalized channel windows as rows of $x \in \mathbb{R}^{N\times L}$ and process them independently with shared

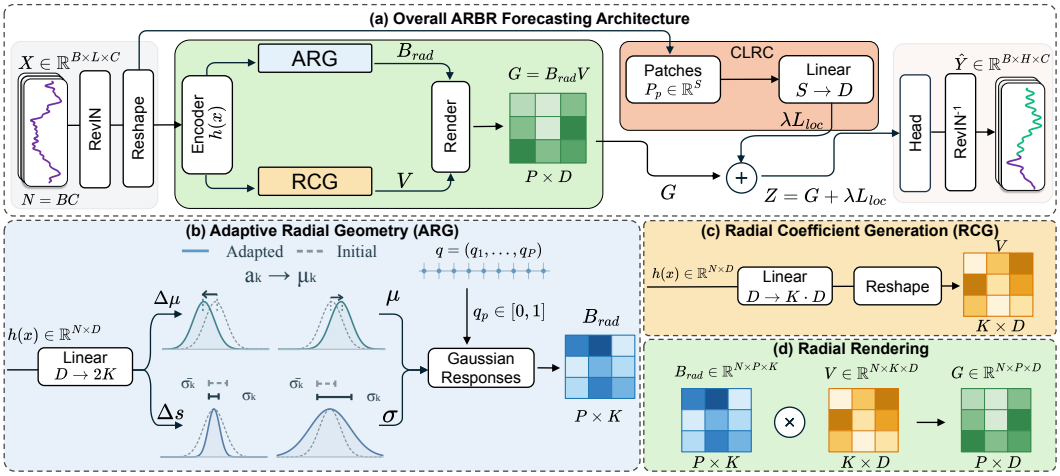


Figure 2: Overview of the ARBR architecture. (a) Overall architecture: ARBR constructs an adaptive radial representation from each input window, integrates local residual features, and maps the resulting representation to future values. (b) Adaptive Radial Geometry (ARG): Adapts radial centers and scales around a shared temporal scaffold and evaluates the resulting Gaussian responses. (c) Radial Coefficient Generation (RCG): Generates component-wise vector-valued coefficients from the shared conditioning feature. (d) Radial rendering: Combines radial responses and coefficients into patch-aligned latent features.

forecasting backbone parameters. Here and below, x denotes the batched form of the single-window input in Section 3.

A full-sequence encoder maps each normalized window to a D-dimensional conditioning feature:

$$h(x) = \mathrm{Dropout}_{p_{\mathrm{gs}}}\big(\mathrm{GELU}\big(xW_{\mathrm{seq}}^{\top} + b_{\mathrm{seq}}\big)\big) \in \mathbb{R}^{N\times D}, \qquad W_{\mathrm{seq}} \in \mathbb{R}^{D\times L}. \tag{4}$$

Throughout, W and b denote trainable weight matrices and bias vectors, respectively. Here, $b_{\mathrm{seq}} \in \mathbb{R}^D$, and p_{gs} is the dropout rate. Both the Geometry Generator in ARG and the Coefficient Generator in RCG are conditioned on $h(x)$.

We also extract patches of length S with stride r from each normalized window, denoting patch p by $P_p \in \mathbb{R}^S$ for an individual window. The patch centers determine the normalized temporal queries $q = (q_1, \ldots, q_P)$ defined in Section 3. ARG evaluates radial responses at these queries, while CLRC uses the patches to construct local residual features. Replication padding and the exact query construction are detailed in Appendix E.3.

4.2 Adaptive Radial Geometry (ARG)

ARG computes the radial response tensor B_{rad} from the conditioning feature $h(x)$ and the patch-aligned queries q.

Shared temporal scaffold. We define a fixed temporal scaffold shared across inputs. Its center anchors are uniformly spaced on the normalized temporal axis:

$$a_k = \frac{k - \frac{1}{2}}{K}, \qquad k = 1, \ldots, K. \tag{5}$$

For $K \geq 2$, the reference scales form a geometric progression from $\sigma_{\min}$ to $\sigma_{\max}$, with $0 < \sigma_{\min} \leq \sigma_{\max}$:

$$\bar{\sigma}_k = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{k-1}{K-1}}. \tag{6}$$

Component k pairs center anchor a_k with reference scale $\bar{\sigma}_k$. The reported configuration uses $\sigma_{\min} = 0.05$ and $\sigma_{\max} = 0.4$.

Geometry Generator. The linear projection in Figure 2(b) maps each conditioning feature to $2K$ raw outputs:

$$[\Delta\mu(x), \Delta s(x)] = h(x)W_{\text{geo}}^{\top} + b_{\text{geo}}, \qquad W_{\text{geo}} \in \mathbb{R}^{2K\times D}, \quad b_{\text{geo}} \in \mathbb{R}^{2K}. \tag{7}$$

The two output blocks $\Delta\mu(x), \Delta s(x) \in \mathbb{R}^{N\times K}$ parameterize center displacements and log-scale changes, respectively. We initialize both W_{geo} and b_{geo} to zero.

Bounded geometry adaptation. For $0 < \rho \leq 0.5$ and $\beta > 0$, the raw outputs define bounded adjustments to the shared scaffold. Applied to each input window, the center and scale transforms are

$$\mu_k(x) = a_k + \frac{\rho}{K}\tanh\big(\Delta\mu_k(x)\big), \tag{8}$$

and

$$\sigma_k(x) = \bar{\sigma}_k \exp\big(\beta\tanh\big(\Delta s_k(x)\big)\big). \tag{9}$$

Zero initialization gives $\mu_k(x) = a_k$ and $\sigma_k(x) = \bar{\sigma}_k$ for every input. The adjustments satisfy

$$|\mu_k(x) - a_k| < \frac{\rho}{K}, \qquad \left|\log\frac{\sigma_k(x)}{\bar{\sigma}_k}\right| < \beta.$$

Thus, ρ bounds center displacement relative to the anchor spacing, while β bounds the multiplicative scale change:

$$e^{-\beta}\bar{\sigma}_k < \sigma_k(x) < e^{\beta}\bar{\sigma}_k.$$

Order-preserving adaptation. The constraint $0 < \rho \leq 0.5$ preserves the temporal ordering of component centers. Since adjacent anchors are separated by $1/K$, the center bounds imply

$$\mu_{k+1}(x) - \mu_k(x) > \frac{1-2\rho}{K} \geq 0, \qquad k = 1, \ldots, K-1. \tag{10}$$

The reported setting $\rho = 0.5$ is the largest value that guarantees this ordering for arbitrary finite generator outputs. Appendix E.4 provides the detailed derivation.

Query-aligned basis evaluation. The Basis Evaluator, shown as Gaussian Responses in Figure 2(b), applies Equation 1 at the patch-aligned queries q using the generated centers and scales. This yields $B_{\text{rad}} \in \mathbb{R}^{N\times P\times K}$, the batched radial response tensor defined in Section 3. This evaluation introduces no additional learned parameters.

4.3 Radial Coefficient Generation (RCG)

Coefficient Generator. In Figure 2(c), RCG linearly projects each conditioning feature and reshapes the output into K vector-valued coefficients of dimension D:

$$V(x) = \text{reshape}_{K\times D}\big(h(x)W_{\text{coef}}^{\top} + b_{\text{coef}}\big) \in \mathbb{R}^{N\times K\times D}, \qquad W_{\text{coef}} \in \mathbb{R}^{KD\times D}. \tag{11}$$

Here, $b_{\text{coef}} \in \mathbb{R}^{KD}$, and the reshape groups the output features into K coefficient vectors for each input window.

Radial rendering. Applying Equation 3 independently to each input window gives the patch-aligned representation $G = B_{\text{rad}}V \in \mathbb{R}^{N\times P\times D}$, where the product denotes batched matrix multiplication.

4.4 Constrained Local Residual Correction (CLRC)

Local residual integration. CLRC supplements the radial representation with local features projected directly from each input patch:

$$L_{\text{loc},p} = W_{\text{local}}P_p, \qquad W_{\text{local}} \in \mathbb{R}^{D\times S}.$$

Collecting these features across windows and patch positions gives $L_{\text{loc}} \in \mathbb{R}^{N\times P\times D}$. A shared learnable scalar a defines the mixing coefficient

$$\lambda = \lambda_{\max}\,\text{sigmoid}(a), \qquad \lambda_{\max} > 0,$$

so $0 < \lambda < \lambda_{\max}$. This bounds the mixing coefficient, but not the magnitude of the local correction λL_{loc}. We set $\lambda_{\max} = 0.5$ and initialize a so that $\lambda = 0.1$. The resulting representation is

$$Z = G + \lambda L_{\text{loc}}. \tag{12}$$

Table 1: Long-term forecasting results on seven multivariate benchmarks for seed 2021, averaged over four forecasting horizons. ARBR uses an input length of $L = 512$; baselines use their model-specific recommended configurations. Results for individual horizons are reported in Appendix A.7. Best in **bold**; second best underlined.

Dataset	ARBR (Ours)		TimeGS (2026)		LGA (2026)		MixLinear (2026)		PhaseFormer (2026)		TimeBase (2025)		TimeMixer (2024)		PatchTST (2023)		BasisFormer (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.403**	0.424	0.419	0.424	0.416	0.430	0.406	0.425	0.414	**0.419**	0.422	0.443	0.456	0.445	0.408	0.425	0.432	0.438
ETTh2	0.333	0.386	0.363	0.389	0.338	0.388	0.344	0.386	0.352	0.394	0.357	0.401	0.387	0.407	**0.331**	**0.381**	0.384	0.410
ETTm1	**0.344**	**0.372**	0.372	0.388	0.359	0.388	0.396	0.412	0.350	0.375	0.358	0.384	0.385	0.399	0.352	0.382	0.408	0.411
ETTm2	**0.247**	**0.311**	0.276	0.317	0.264	0.326	0.257	0.319	0.261	0.320	0.259	0.319	0.279	0.325	0.257	0.315	0.292	0.333
Weather	**0.218**	**0.257**	0.244	0.266	0.238	0.284	0.242	0.282	0.228	0.263	0.238	0.282	0.245	0.276	0.231	0.268	0.258	0.282
Electricity	**0.159**	**0.251**	0.181	0.265	0.168	0.264	0.165	0.258	0.160	0.252	0.164	0.259	0.182	0.272	0.168	0.258	0.203	0.296
Traffic	0.395	0.269	0.478	0.291	0.466	0.294	0.433	0.295	**0.387**	**0.250**	0.392	0.269	0.485	0.298	0.392	0.265	0.477	0.324

Forecasting head. A linear forecasting head maps the concatenated patch features of each window in Z to an H-step prediction, yielding $\widehat{y} \in \mathbb{R}^{N \times H}$. The channel-wise predictions form $\widehat{Y} \in \mathbb{R}^{B \times H \times C}$, which is returned to the pre-RevIN scale by inverse RevIN.

5 Experiments

We evaluate ARBR through forecasting benchmarks, component ablations, controlled comparisons of representation designs, and analyses of model behavior. Results across multiple random seeds and computational measurements are reported in Appendix B.

5.1 Experimental Settings

Datasets. We evaluate ARBR on seven standard multivariate forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic. The numbers of channels and time steps, sampling frequencies, and training, validation, and test splits are reported in Appendix A.1.

Baselines. We compare ARBR with eight forecasting baselines: TimeGS (Wang et al., 2026), LGA (Kim et al., 2026), MixLinear (Ma et al., 2026), PhaseFormer (Niu et al., 2026), TimeBase (Huang et al., 2025), TimeMixer (Wang et al., 2024), PatchTST (Nie et al., 2023), and BasisFormer (Ni et al., 2023). BasisFormer and TimeBase are closely related basis-based methods, while TimeGS provides a comparison with Gaussian-based forecasting. We reproduce all baseline results in the main table using official implementations and model-specific recommended configurations. All methods use the same dataset splits, forecasting horizons, and evaluation metrics.

Forecasting protocol and metrics. We evaluate all methods at forecasting horizons $H \in \{96, 192, 336, 720\}$ using mean squared error (MSE) and mean absolute error (MAE). Additional protocol and reproducibility details are provided in Appendix A.

5.2 Main Results

Table 1 reports MSE and MAE averaged over the four forecasting horizons for each dataset. Results for individual horizons are provided in Appendix A.7 (Table 5).

ARBR achieves the lowest average MSE on five of the seven datasets and the lowest average MAE on four. On Weather, ARBR reduces average MSE and MAE by approximately 4.4% and 2.3%, respectively, relative to the best baseline for each metric. ARBR also achieves lower average MSE and MAE than BasisFormer and TimeBase on all six non-Traffic datasets. ARBR ranks second in average MSE and ties for second in average MAE on ETTh2. These results show that ARBR's forecasting gains extend across multiple datasets and comparisons with existing basis-based methods. Appendix B.1 reports additional comparisons across three random seeds.

Table 2: Ablation results averaged over the four forecasting horizons $H \in \{96, 192, 336, 720\}$. Δ denotes the mean relative degradation in MSE and MAE with respect to full ARBR. Complete horizon-specific results are reported in Appendix C.

Dataset	ARBR		w/o Local Correction			w/o Scale Adaptation			w/o Center Adaptation			w/o Coefficient Conditioning		
	MSE	MAE	MSE	MAE	Δ (%)	MSE	MAE	Δ (%)	MSE	MAE	Δ (%)	MSE	MAE	Δ (%)
ETTh1	0.403	0.424	0.408	0.429	1.1	0.412	0.433	2.1	0.413	0.429	1.8	0.420	0.438	3.7
ETTh2	0.333	0.386	0.337	0.387	0.8	0.344	0.392	2.6	0.348	0.396	3.7	0.351	0.402	4.8
ETTm1	0.344	0.372	0.349	0.375	1.0	0.355	0.379	2.5	0.354	0.378	2.1	0.360	0.389	4.4
ETTm2	0.247	0.311	0.251	0.312	0.9	0.256	0.318	2.9	0.259	0.317	3.4	0.258	0.322	3.9
Weather	0.218	0.257	0.225	0.262	2.5	0.228	0.266	4.1	0.230	0.268	4.7	0.227	0.265	3.5
Electricity	0.159	0.251	0.164	0.257	2.6	0.166	0.260	3.6	0.167	0.262	4.4	0.168	0.260	4.2
Traffic	0.395	0.269	0.406	0.283	3.9	0.405	0.276	2.4	0.403	0.274	1.9	0.411	0.279	3.9

5.3 Ablation Study

We separately ablate center adaptation, scale adaptation, coefficient conditioning, and local residual correction. Table 2 reports MSE and MAE averaged over $H \in \{96, 192, 336, 720\}$. Results for individual horizons and detailed variant definitions are provided in Appendix C. For each dataset, Δ is the mean of the relative increases in MSE and MAE over full ARBR, calculated from the unrounded four-horizon averages. We summarize each ablation by averaging Δ with equal weight across the seven datasets.

Removing the local residual pathway produces the smallest Δ_7 among the tested ablations, at approximately 1.8%. The Δ_7 values are approximately 2.9% without scale adaptation and 3.1% without center adaptation. Both geometry ablations increase the four-horizon average MSE and MAE on all seven datasets, with the largest Δ values on Weather and Electricity.

To remove coefficient conditioning, we replace $h(x)$ with a fixed zero vector only in the coefficient generator. The trainable bias then defines a coefficient tensor shared across inputs, while the geometry branch and the rest of the architecture are retained. This variant yields $\Delta_7 \approx 4.1\%$ across the seven datasets. Adapting either the coefficients or the geometry alone does not recover the performance of full ARBR in the tested configuration.

On Traffic, ARBR trails the best-performing baselines in Table 1, although center and scale adaptation remain beneficial within the model: removing them yields $\Delta = 1.9\%$ and 2.4%, respectively. Removing local correction or coefficient conditioning produces a larger degradation of approximately 3.9%, indicating greater sensitivity to these two components than to either geometry adaptation alone.

5.4 Representation Design Comparisons

We compare ARBR with fixed and globally learned radial geometry and a non-radial dynamic mixing control (Table 3). Horizon-specific results are provided in Appendix D.

Geometry conditioning. Fixed Scaffold fixes centers and scales at their scaffold values. Global Learned Geometry learns a single set of center and scale adjustments shared across inputs, using the same bounds as ARBR. Both variants retain the remaining architecture. Fixed Scaffold and Global Learned Geometry yield $\Delta_7 = 5.2\%$ and 3.5%, respectively. ARBR achieves lower MSE and MAE than both variants on every dataset, averaged over the four forecasting horizons. The advantage over Global Learned Geometry supports adapting geometry to individual inputs beyond learning a shared geometry.

Non-radial dynamic mixing. Dynamic Mixing retains the encoder, input-conditioned coefficients $V(x)$, the K-component factorization, the local residual pathway, and the forecasting head. It replaces the radial response matrix with a directly generated input-conditioned mixing matrix. Let B_0 be the radial response matrix evaluated at the undeformed scaffold, and define

$$\widetilde{B}_0 = \mathrm{clip}(B_0, \epsilon, 1-\epsilon) \in \mathbb{R}^{P\times K}, \qquad 0 < \epsilon < \tfrac{1}{2}.$$

A linear mixing head produces PK offsets for each input window:

$$\Delta A(x) = \mathrm{reshape}_{P\times K}\big(g_{\mathrm{mix}}(h(x))\big) \in \mathbb{R}^{N\times P\times K}.$$

Table 3: Representation-design comparisons averaged over $H \in \{96, 192, 336, 720\}$. Fixed Scaffold and Global Learned Geometry vary geometry conditioning, while Dynamic Mixing replaces radial geometry with input-conditioned non-radial mixing and retains the remaining architecture. Δ_7 denotes the mean relative MSE/MAE degradation from ARBR across the seven datasets. Best results are bold.

Parameterization	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Electricity		Traffic		Δ_7 (%)
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ARBR	**0.403**	**0.424**	**0.333**	**0.386**	**0.344**	**0.372**	**0.247**	**0.311**	**0.218**	**0.257**	**0.159**	**0.251**	**0.395**	**0.269**	**0.0**
Dynamic Mixing	0.412	0.432	0.346	0.399	0.358	0.385	0.255	0.320	0.224	0.263	0.165	0.258	0.399	0.272	2.7
Global Learned Geometry	0.416	0.438	0.347	0.401	0.357	0.386	0.256	0.321	0.226	0.264	0.166	0.259	0.410	0.278	3.5
Fixed Scaffold	0.423	0.438	0.354	0.410	0.364	0.393	0.259	0.325	0.227	0.267	0.169	0.266	0.420	0.282	5.2

The mixing matrix and resulting representation are

$$A(x) = \mathrm{sigmoid}\Big(\mathrm{logit}(\widetilde{B}_0) + \Delta A(x)\Big), \qquad G(x) = A(x)V(x), \tag{13}$$

where the shared reference matrix $\widetilde{B}_0$ is applied to each input window and the product denotes batched matrix multiplication. The mixing head's weights and bias are initialized to zero, giving $A(x) = \widetilde{B}_0$ for every input at initialization. The entries of $A(x)$ are not constrained to follow a Gaussian response over temporal coordinates. This control retains the scaffold-based initialization but removes the center–scale parameterization of input-dependent mixing.

For Dynamic Mixing, the relative increases in MSE and MAE average 2.9% and 2.5% across the seven datasets, giving $\Delta_7 = 2.7\%$. Its four-horizon-average MSE and MAE remain higher than those of ARBR on all seven datasets. Compared with Global Learned Geometry, Dynamic Mixing achieves lower MAE on all seven datasets and lower MSE on six. Three-seed results are reported in Appendix D.1. These results favor the radial parameterization over the tested non-radial mixing alternative, even when both are input-conditioned.

Parameter sensitivity. We examine sensitivity to K, ρ, and β on ETTh1, ETTm1, and Weather (Appendix F). The default $\rho = \beta = 0.5$ achieves the lowest or tied-lowest errors among the tested bound settings, averaged over the four forecasting horizons. Reducing ρ or β individually to 0.25 slightly increases the errors, while setting $\beta = 0.75$ gives results close to the default configuration.

5.5 Mechanism Analysis

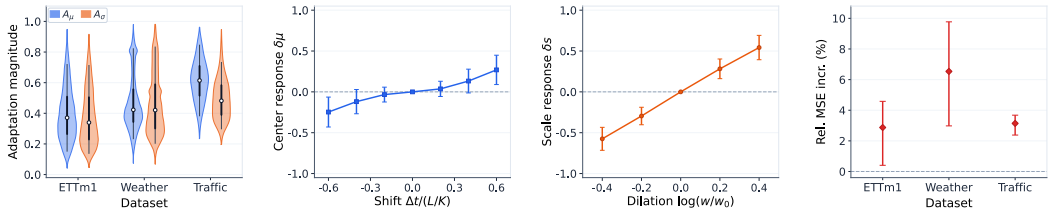
We examine how radial geometry varies across inputs and responds to controlled temporal perturbations, and whether matching the generated coefficients to their inputs matters for forecasting.

Adaptive geometry usage. For each input x, we measure the mean absolute center and log-scale deviations from the scaffold, normalized by their respective bounds:

$$A_\mu(x) = \frac{1}{K}\sum_{k=1}^{K} \frac{|\mu_k(x) - a_k|}{\rho/K}, \qquad A_\sigma(x) = \frac{1}{K}\sum_{k=1}^{K} \frac{|\log(\sigma_k(x)/\bar{\sigma}_k)|}{\beta}. \tag{14}$$

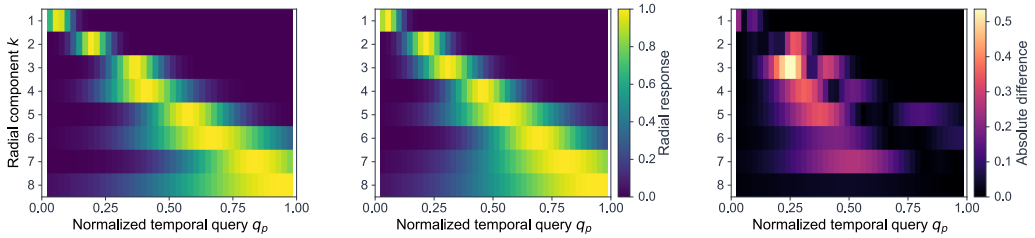
Both scores lie in $[0, 1)$, with larger values indicating greater average deviations relative to the permitted bounds. Figure 3(a) shows the distributions of $A_\mu(x)$ and $A_\sigma(x)$ across input windows on ETTm1, Weather, and Traffic.

Controlled temporal responses. We apply controlled temporal shifts, compression, and dilation to normalized input windows from Weather. Figure 3(b) shows a monotonic mean center response: shifts toward earlier times produce negative responses, while shifts toward later times produce positive responses. Figure 3(c) shows a monotonic mean scale response, with negative values under compression and positive values under dilation. These responses are consistent with centers tracking temporal location and scales tracking temporal extent under the tested perturbations. Protocol details are provided in Appendix E.6.



(a) Geometry usage. (b) Shift response. (c) Dilation response. (d) Coefficient mismatch.

Figure 3: Mechanism analysis of ARBR. (a) Normalized center (A_μ) and scale (A_σ) adaptation across channel-wise test windows. White dots mark medians; thick/thin bars span the 25th–75th/5th–95th percentiles. (b–c) Normalized center/scale responses to temporal shifts/dilation on Weather. Markers show means; error bars show standard deviations across channel-wise test windows and radial components. (d) Relative MSE increase under same-channel coefficient replacement. Diamonds show means; error bars show standard deviations across four horizons, after averaging five random reassignments per horizon.



(a) Original responses. (b) Responses after substitution. (c) Absolute difference.

Figure 4: Radial responses under geometry substitution on ETTm1. The selected window pair has a radial-response distance near the median across evaluated pairs. (a) Original radial responses B_i for window i. (b) Responses $B_{i\leftarrow j}$ after replacing all centers and scales with those generated from another window j of the same channel. (c) Element-wise absolute difference $|B_i - B_{i\leftarrow j}|$.

Test-time coefficient substitution. We replace V_i with V_j from another test window of the same channel, keeping the target-specific geometry, local representation, normalization statistics, and forecasting head unchanged. The resulting MSE increases (Figure 3(d)) show that input–coefficient matching matters for forecasting in the trained model. This test evaluates that correspondence without retraining, complementing the coefficient-conditioning ablation in Table 2.

Radial responses under geometry substitution. Figure 4 provides a qualitative comparison of radial responses using the target window's geometry and geometry from another window of the same channel. The substitution changes the temporal locations and extents of the responses while retaining the ordering of component centers.

6 Conclusion

We introduced ARBR to adapt the temporal locations and scales of radial basis components to each input. Bounded adjustments around a shared temporal scaffold preserve the ordering of these components across inputs. Input-adaptive geometry reduces forecasting errors relative to fixed and globally learned geometry in our experiments. These results support adapting basis geometry alongside input-conditioned coefficients for time series forecasting.

The current formulation limits geometric adaptation to bounded deformations of a shared temporal scaffold and processes channels independently. Future work could explore broader geometric parameterizations and incorporate cross-variable information.

AI USE STATEMENT

In this work, we used generative AI tools only to aid and polish the writing after the manuscript had been drafted. The tools were used to check grammar, spelling, word choice, and clarity. All AI-assisted edits were reviewed by the authors. We did not use generative AI tools for research ideation, methodology or experimental design, data generation, interpretation of experimental results, or mathematical proofs. We take responsibility for the final content of this work and for all AI-assisted edits.

REPRODUCIBILITY STATEMENT

Appendix A documents the dataset preprocessing, training configurations, baseline implementations, and evaluation protocol. Appendix E provides implementation details, and Appendix E.6 specifies the mechanism analysis protocols. Additional comparisons across random seeds are reported in Appendices B.1 and D.1.

REFERENCES

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.V37I6.25854.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. doi: 10.1016/j.ijforecast.2015.11.011.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24893–24911. PMLR, 2025.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. TimeBase: The power of minimalism in efficient long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26227–26246. PMLR, 2025.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, volume 29, pp. 667–675, 2016.

Dongbin Kim, Youngjoo Park, Woojin Jeong, and Jaewook Lee. Local geometry attention for time series forecasting under realistic corruptions. In *International Conference on Learning Representations*, 2026.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Qiang Lai and Yang You. Frequency-wavelet adaptive basis network for long-term time series forecasting. *Engineering Applications of Artificial Intelligence*, 161:112161, 2025. doi: 10.1016/j.engappai.2025.112161.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. doi: 10.1126/science.adi2336.

Tong Liu, Sheng Chen, Shan Liang, Shaojun Gan, and Chris J. Harris. Fast adaptive gradient RBF networks for online learning of nonstationary time series. *IEEE Transactions on Signal Processing*, 68:2015–2030, 2020. doi: 10.1109/TSP.2020.2981197.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022. doi: 10.52202/068431-0718.

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

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

Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. BasisFormer: Attention-based time series forecasting with learnable and interpretable basis. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71222–71241, 2023. doi: 10.52202/075280-3118.

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

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

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014. doi: 10.1016/j.trc.2014.01.005.

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

Yixin Wang, Yifan Hu, Peiyuan Liu, Naiqi Li, Tao Dai, and Shu-Tao Xia. Forecasting as rendering: A 2D gaussian splatting framework for time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3818178.

Bruce A. Whitehead and Timothy D. Choate. Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7(4):869–880, 1996. doi: 10.1109/72.508930.

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

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

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. doi: 10.1609/aaai.v35i12.17325.

APPENDIX CONTENTS

Appendix A. Reproducibility and Experimental Details 13
A.1 Dataset Statistics and Split Protocol 13
A.2 Details About Baselines 13
A.3 Training and Optimization Details 14
A.4 ARBR Model Configuration 14
A.5 Baseline Implementation Protocol 15
A.6 Evaluation Protocol 15
A.7 Full Horizon-Specific Main Results 15
Appendix B. Additional Reliability and Evaluation Controls 16
B.1 Multi-Seed Stability 16
B.2 Computational Characteristics 17
Appendix C. Extended Component Ablation 18
Appendix D. Extended Representation Design Comparisons 19
D.1 Dynamic Mixing Stability 20
Appendix E. Extended Methodological Details 21
E.1 Notation and Tensor Shapes 21
E.2 Channel-Independent Encoder and Generator Implementations 21
E.3 Patch-Aligned Query Construction 22
E.4 Properties of Bounded Adaptive Geometry 23
E.5 Complete Forward Formulation 23
E.6 Mechanism Analysis Protocol 23
Appendix F. Sensitivity Analysis 25
F.1 Number of Radial Components 25
F.2 Center- and Scale-Deformation Bounds 25
Appendix G. Additional Forecasting Visualizations 27

A Reproducibility and Experimental Details

This appendix documents the dataset preprocessing protocol, baseline set, concrete implementation settings, and evaluation rules used in the reported experiments. We first report the dataset dimensionality, sequence length, sampling frequency, and train/validation/test split ratios used in preprocessing. We then summarize the comparison methods, training and model configurations, baseline configurations, and the common evaluation protocol.

A.1 Dataset Statistics and Split Protocol

Table 4 summarizes the seven multivariate forecasting benchmarks used throughout the paper. The variable count C is dataset specific and is used as the corresponding input and output dimensionality. The *Split* column reports the train:validation:test ratio used by the data loader.

Table 4: Dataset statistics and preprocessing split ratios used in the reported experiments.

Dataset	Variables	Split	Length	Frequency	Description
ETTh1	7	6:2:2	14,400	1 hour	Electricity-transformer temperature data from one station.
ETTh2	7	6:2:2	14,400	1 hour	Electricity-transformer temperature data from a different station.
ETTm1	7	6:2:2	57,600	15 minutes	Quarter-hourly transformer data corresponding to the ETTh1 station.
ETTm2	7	6:2:2	57,600	15 minutes	Quarter-hourly transformer data corresponding to the ETTh2 station.
Weather	21	7:1:2	52,696	10 minutes	Meteorological variables including temperature, humidity, and wind speed.
Electricity	321	7:1:2	26,304	1 hour	Electricity-consumption measurements from multiple clients.
Traffic	862	7:1:2	17,544	1 hour	Road-occupancy measurements from highway sensors.

A.2 Details About Baselines

Our main comparison includes eight reproduced forecasting models chosen to cover direct basis-oriented references and complementary representation paradigms. BasisFormer and TimeBase are the closest basis-oriented comparisons, while the remaining baselines probe Gaussian, geometric, phase-aware, linear, multi-scale, and patch-based alternatives.

1. **TimeGS** represents temporal variation with Gaussian primitives in a period–phase representation (Wang et al., 2026). We include it as the closest Gaussian-based rendering reference, while its two-dimensional period–phase geometry differs from the one-dimensional adaptive temporal support used by ARBR.
2. **LGA** (Local Geometry Attention) models query-specific local geometric structure within attention (Kim et al., 2026). It provides an adaptive-geometry reference based on attention rather than radial basis rendering.
3. **MixLinear** combines time-domain trend modeling with compact spectral filtering in a linear forecasting architecture (Ma et al., 2026). We include it as a lightweight non-basis reference that tests whether the proposed representation remains competitive with recent compact linear–spectral designs.
4. **PhaseFormer** organizes forecasting around phase-aware periodic representations and cross-phase interactions (Niu et al., 2026). It serves as a recent structured temporal representation whose inductive bias is periodic phase organization rather than adaptive radial support.
5. **TimeBase** extracts compact temporal basis components for forecasting (Huang et al., 2025). It is included as a direct basis-oriented reference for evaluating whether input-adaptive support geometry provides benefits beyond learned temporal basis components.
6. **TimeMixer** performs multi-scale temporal mixing with decomposed seasonal and trend information across resolutions (Wang et al., 2024). It provides a representative multi-scale alternative to the component-based temporal organization used by ARBR.
7. **PatchTST** is a channel-independent Transformer that represents each univariate channel as a sequence of temporal patches (Nie et al., 2023). We include it as a strong patch-based

reference with the same channel-independent modeling regime used by ARBR, but without an explicit basis geometry.

8. **BasisFormer** learns a dataset-level basis set with instance-dependent coefficients (Ni et al., 2023). It is a direct comparison for the basis-and-coefficient structure of ARBR, particularly because the coefficients are input-dependent while the basis structure is shared at the dataset level.

Implementation and configuration protocols are specified in Appendix A.5; checkpoint selection, metric aggregation, and statistical reporting are specified in Appendix A.6.

A.3 Training and Optimization Details

The training configuration is summarized below. ARBR is optimized end-to-end with AdamW using MSE as the training objective; the objective is

$$\theta \leftarrow \mathrm{AdamW}\big(\theta; \eta_{\mathrm{init}}, \mathrm{weight_decay} = 0\big), \qquad \mathcal{L}_{\mathrm{MSE}} = \frac{1}{BCH} \sum_{b=1}^{B} \sum_{c=1}^{C} \sum_{t=1}^{H} \left(\widehat{Y}_{b,t,c} - Y_{b,t,c}\right)^2 .$$

MSE and MAE are both reported at validation/test time. Multi-seed stability is reported separately in Appendix B.1.

Training and optimization configuration.

Setting	Value
Optimizer	AdamW
Batch size	256 for ETT and Weather; 32 for Electricity and Traffic
Default initial learning rate	0.0012
Weight decay	0.0
Training objective	MSE
Learning-rate schedule	cosine
Initial-LR hold	first 2 epochs
Minimum learning rate	$0.01\,\eta_{\mathrm{init}}$
Maximum training epochs	30
Early-stopping patience	6
Random seed	2021

Here, batch size refers to the number of multivariate input windows B before channel-independent reshaping. Since ARBR subsequently merges the batch and channel axes into $N = BC$ channel-wise sequences, we use a smaller batch size for the high-dimensional Electricity ($C = 321$) and Traffic ($C = 862$) datasets to control memory usage.

A.4 ARBR Model Configuration

The table below summarizes the key method-specific configuration used by ARBR. The formal model definition is given in Section 4, while implementation-level tensor shapes, patch/query construction, and the complete forward pass are provided in Appendix E.

The default structural hyperparameters were selected through preliminary experiments on a small set of representative datasets and then fixed for the full benchmark suite. In particular, the reported defaults for L, D, K, patch length/stride, ρ, β, $\sigma_{\min}$, $\sigma_{\max}$, and the local-residual bound are not tuned separately for each benchmark. Appendix F reports sensitivity analyses for K, ρ, and β.

Key ARBR method-specific configuration.

Setting	Value
Main look-back length L	512
Latent dimension D	128
Patch length / stride	24 / 12
Radial components K	8
Minimum scaffold scale $\sigma_{\min}$	0.05
Maximum scaffold scale $\sigma_{\max}$	0.4
Center-deformation bound ρ	0.5
Log-scale deformation bound β	0.5
Maximum local residual weight $\lambda_{\max}$	0.5
Initial local residual weight λ	0.1
Shared-feature dropout p_{gs}	0.55
Forecasting head dropout	0.35

A.5 Baseline Implementation Protocol

All baseline results in the main comparison are reproduced using the official implementations and their recommended model-specific configurations. We do not retune the baseline architectures to a shared hyperparameter template. Dataset partitions, forecasting horizons, and evaluation metrics are kept common across methods. ARBR and all reproduced baselines use historical target-variable observations without timestamp covariates.

A.6 Evaluation Protocol

Checkpoint selection. Model checkpoints are selected exclusively according to validation-set performance; the test set is not used for checkpoint selection. For ARBR, a separate model is trained for each forecasting horizon, with its own checkpoint; early stopping and checkpoint selection use validation MSE.

Forecast horizons and metrics. All models are evaluated at forecasting horizons $H \in \{96, 192, 336, 720\}$. Forecasting accuracy is measured using mean squared error (MSE) and mean absolute error (MAE), with lower values indicating better performance. Each horizon-specific entry is computed on the corresponding test set after the model and checkpoint have been fixed using the training and validation data only. Reported MSE and MAE are computed in the dataset-standardized space; for ARBR, this is the pre-RevIN scale recovered after inverse RevIN.

Metric aggregation. For tables that summarize performance over forecasting horizons, we report the arithmetic mean of the four horizon-specific values. Specifically,

$$\overline{\mathrm{MSE}} = \frac{1}{4}\sum_{H} \mathrm{MSE}_H, \qquad \overline{\mathrm{MAE}} = \frac{1}{4}\sum_{H} \mathrm{MAE}_H.$$

The dataset-level *Avg* rows in Table 5 are computed with this four-horizon arithmetic mean, and the per-dataset summaries in the ablation, representation-design, and sensitivity studies use the same convention. The main forecasting table does not report a cross-dataset average. For the representation-design comparisons, Δ_7 first computes the relative change in MSE and MAE separately for each of the seven dataset-level four-horizon averages and then takes the arithmetic mean over the resulting fourteen relative changes. Thus, each dataset and each metric receives equal weight in Δ_7.

Random seeds and statistical reporting. The main comparison reports random seed 2021. Appendix B.1 reports the mean and sample standard deviation of four-horizon-average results across seeds $\{42, 2021, 2026\}$ for ARBR, PhaseFormer, MixLinear, and PatchTST on all seven benchmarks. Three-seed results for the Dynamic Mixing control are reported in Appendix D.1.

A.7 Full Horizon-Specific Main Results

Table 5 reports the complete horizon-specific comparison underlying the four-horizon averages in Table 1.

Table 5: Complete horizon-specific long-term forecasting results on seven multivariate benchmarks underlying the averages in Table 1. ARBR uses input length $L = 512$, while reproduced baselines follow their official/recommended configurations. All methods are evaluated at $H \in \{96, 192, 336, 720\}$. Best in **bold**; second best underlined.

Model		ARBR (Ours)		TimeGS (2026)		LGA (2026)		MixLinear (2026)		PhaseFormer (2026)		TimeBase (2025)		TimeMixer (2024)		PatchTST (2023)		BasisFormer (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.359**	0.389	0.361	**0.386**	0.373	0.399	0.374	0.409	0.367	0.389	0.368	0.404	0.381	0.401	0.365	0.392	0.397	0.415
	192	**0.394**	0.413	0.410	0.419	0.417	0.425	0.399	0.411	0.406	**0.407**	0.406	0.428	0.440	0.433	0.403	0.413	0.442	0.436
	336	0.415	0.429	0.452	0.440	0.427	0.432	0.425	**0.423**	0.444	0.432	0.440	0.451	0.501	0.462	**0.404**	0.426	0.442	0.440
	720	0.445	0.465	0.451	0.451	0.446	0.463	**0.427**	0.458	0.440	**0.447**	0.473	0.488	0.501	0.482	0.458	0.467	0.448	0.460
	Avg	**0.403**	0.424	0.419	0.424	0.416	0.430	0.406	0.425	0.414	**0.419**	0.422	0.443	0.456	0.445	0.408	0.425	0.432	0.438
ETTh2	96	**0.265**	0.332	0.281	**0.329**	0.284	0.344	0.286	0.342	0.281	0.343	0.291	0.347	0.292	0.343	0.274	0.336	0.339	0.378
	192	**0.325**	**0.377**	0.359	0.380	0.347	0.388	0.342	0.378	0.339	0.379	0.350	0.394	0.374	0.395	0.338	0.378	0.370	0.396
	336	0.352	0.401	0.401	0.416	0.334	0.389	0.364	0.400	0.379	0.412	0.374	0.415	0.428	0.433	**0.331**	**0.386**	0.384	0.415
	720	0.390	0.432	0.410	0.431	0.387	0.429	0.382	0.423	0.409	0.442	0.412	0.447	0.454	0.458	**0.379**	**0.422**	0.442	0.452
	Avg	0.333	0.386	0.363	0.389	0.338	0.388	0.344	0.386	0.352	0.394	0.357	0.401	0.387	0.407	**0.331**	**0.381**	0.384	0.410
ETTm1	96	**0.284**	**0.337**	0.307	0.347	0.297	0.353	0.354	0.389	0.301	0.345	0.304	0.350	0.328	0.363	0.293	0.344	0.343	0.377
	192	**0.321**	**0.360**	0.354	0.373	0.341	0.387	0.375	0.399	0.328	0.365	0.335	0.370	0.364	0.384	0.335	0.371	0.382	0.395
	336	0.358	0.381	0.383	0.397	0.372	0.392	0.405	0.419	**0.356**	**0.379**	0.366	0.390	0.390	0.404	0.367	0.392	0.422	0.418
	720	0.414	**0.411**	0.443	0.433	0.425	0.421	0.449	0.439	0.413	0.412	0.427	0.427	0.458	0.445	**0.412**	0.420	0.483	0.453
	Avg	**0.344**	**0.372**	0.372	0.388	0.359	0.388	0.396	0.412	0.350	0.375	0.358	0.384	0.385	0.399	0.352	0.382	0.408	0.411
ETTm2	96	**0.161**	**0.250**	0.169	0.251	0.177	0.269	0.169	0.261	0.173	0.259	0.169	0.256	0.176	0.259	0.165	0.254	0.183	0.266
	192	**0.213**	**0.287**	0.239	0.293	0.231	0.305	0.223	0.298	0.236	0.303	0.225	0.296	0.242	0.303	0.222	0.294	0.250	0.309
	336	**0.266**	**0.324**	0.298	0.332	0.283	0.338	0.276	0.332	0.280	0.335	0.279	0.333	0.304	0.342	0.277	0.329	0.315	0.348
	720	**0.349**	**0.381**	0.396	0.391	0.366	0.390	0.359	0.385	0.356	0.382	0.361	0.391	0.393	0.397	0.365	0.383	0.420	0.408
	Avg	**0.247**	**0.311**	0.276	0.317	0.264	0.326	0.257	0.319	0.261	0.320	0.259	0.319	0.279	0.325	0.257	0.315	0.292	0.333
Weather	96	**0.141**	**0.191**	0.161	0.199	0.165	0.224	0.170	0.225	0.149	0.195	0.163	0.223	0.165	0.212	0.156	0.205	0.181	0.222
	192	**0.185**	**0.234**	0.206	0.242	0.209	0.264	0.211	0.259	0.194	0.240	0.212	0.264	0.209	0.253	0.199	0.245	0.222	0.257
	336	**0.235**	**0.275**	0.263	0.286	0.257	0.301	0.260	0.298	0.248	0.283	0.257	0.297	0.264	0.293	0.249	0.285	0.275	0.296
	720	**0.311**	**0.328**	0.344	0.337	0.322	0.347	0.325	0.345	0.319	0.333	0.320	0.345	0.342	0.345	0.320	0.335	0.354	0.351
	Avg	**0.218**	**0.257**	0.244	0.266	0.238	0.284	0.242	0.282	0.228	0.263	0.238	0.282	0.245	0.276	0.231	0.268	0.258	0.282
Electricity	96	**0.128**	**0.222**	0.151	0.237	0.138	0.234	0.138	0.233	**0.128**	0.229	0.132	0.230	0.153	0.244	0.143	0.230	0.164	0.260
	192	**0.147**	0.238	0.164	0.250	0.154	0.251	0.151	0.244	**0.147**	**0.237**	0.151	0.248	0.166	0.256	0.152	0.246	0.193	0.296
	336	**0.163**	**0.256**	0.184	0.270	0.171	0.269	0.166	0.260	0.166	0.258	0.167	0.263	0.184	0.275	0.168	0.262	0.202	0.294
	720	0.199	0.289	0.224	0.304	0.208	0.301	0.205	0.293	**0.198**	**0.284**	0.204	0.296	0.226	0.313	0.209	0.295	0.254	0.335
	Avg	**0.159**	**0.251**	0.181	0.265	0.168	0.264	0.165	0.258	0.160	0.252	0.164	0.259	0.182	0.272	0.168	0.258	0.203	0.296
Traffic	96	0.367	0.255	0.457	0.283	0.425	0.279	0.406	0.282	0.362	**0.238**	**0.361**	0.252	0.462	0.285	**0.361**	0.250	0.450	0.320
	192	0.385	0.263	0.463	0.282	0.443	0.286	0.419	0.286	**0.373**	**0.243**	0.386	0.266	0.473	0.296	0.379	0.256	0.460	0.315
	336	0.395	0.270	0.478	0.288	0.479	0.299	0.437	0.296	**0.385**	**0.248**	0.393	0.270	0.498	0.296	0.395	0.267	0.478	0.320
	720	0.434	0.289	0.514	0.310	0.515	0.311	0.471	0.315	**0.428**	**0.270**	0.429	0.288	0.506	0.313	0.432	0.286	0.518	0.341
	Avg	0.395	0.269	0.478	0.291	0.466	0.294	0.433	0.295	**0.387**	**0.250**	0.392	0.269	0.485	0.298	0.392	0.265	0.477	0.324

B Additional Reliability and Evaluation Controls

This appendix reports two complementary checks: multi-seed stability and computational characteristics.

B.1 Multi-Seed Stability

We evaluate ARBR, PhaseFormer, MixLinear, and PatchTST with random seeds $\{42, 2021, 2026\}$ on all seven benchmarks. For each seed and dataset, MSE and MAE are first averaged over the four forecasting horizons $H \in \{96, 192, 336, 720\}$ using the aggregation rule in Appendix A.6. We then report the mean and sample standard deviation across the three seeds.

Three-seed comparison across seven benchmarks.

Dataset	Model	MSE mean $\pm$ std	MAE mean $\pm$ std
ETTh1	ARBR	0.4038 ± 0.0008	0.4245 ± 0.0005
	PhaseFormer	0.4147 ± 0.0005	0.4192 ± 0.0004
	MixLinear	0.4035 ± 0.0041	0.4299 ± 0.0048
	PatchTST	0.4063 ± 0.0014	0.4243 ± 0.0005
ETTh2	ARBR	0.3344 ± 0.0012	0.3858 ± 0.0004
	PhaseFormer	0.3527 ± 0.0006	0.3945 ± 0.0005
	MixLinear	0.3436 ± 0.0011	0.3865 ± 0.0007
	PatchTST	0.3305 ± 0.0003	0.3801 ± 0.0005
ETTm1	ARBR	0.3453 ± 0.0010	0.3728 ± 0.0005
	PhaseFormer	0.3495 ± 0.0003	0.3753 ± 0.0001
	MixLinear	0.3955 ± 0.0009	0.4112 ± 0.0004
	PatchTST	0.3533 ± 0.0016	0.3826 ± 0.0010
ETTm2	ARBR	0.2488 ± 0.0018	0.3111 ± 0.0006
	PhaseFormer	0.2616 ± 0.0006	0.3199 ± 0.0003
	MixLinear	0.2571 ± 0.0004	0.3190 ± 0.0003
	PatchTST	0.2578 ± 0.0008	0.3154 ± 0.0004
Weather	ARBR	0.2193 ± 0.0011	0.2581 ± 0.0009
	PhaseFormer	0.2281 ± 0.0005	0.2635 ± 0.0007
	MixLinear	0.2432 ± 0.0029	0.2821 ± 0.0006
	PatchTST	0.2313 ± 0.0003	0.2678 ± 0.0003
Electricity	ARBR	0.1601 ± 0.0010	0.2520 ± 0.0008
	PhaseFormer	0.1602 ± 0.0004	0.2528 ± 0.0007
	MixLinear	0.1652 ± 0.0003	0.2577 ± 0.0001
	PatchTST	0.1674 ± 0.0005	0.2578 ± 0.0005
Traffic	ARBR	0.3955 ± 0.0003	0.2698 ± 0.0006
	PhaseFormer	0.3875 ± 0.0004	0.2501 ± 0.0004
	MixLinear	0.4333 ± 0.0000	0.2947 ± 0.0004
	PatchTST	0.3923 ± 0.0006	0.2651 ± 0.0006

ARBR has lower three-seed mean MSE and MAE than the three baselines on ETTm1, ETTm2, Weather, and Electricity. The differences from PhaseFormer on Electricity are small. On ETTh1, MixLinear has a slightly lower mean MSE, while PhaseFormer has the lowest mean MAE. PatchTST leads on ETTh2 and PhaseFormer on Traffic. We treat this three-seed comparison as a stability check rather than an inferential significance test.

B.2 Computational Characteristics

Computational measurement protocol.

Item	Setting
Hardware	One NVIDIA GeForce RTX 5060 Ti, driver 591.86, CUDA 13.1.
Batch / precision	Batch size 256 on ETTh1; FP32; AMP disabled.
Latency	5 no-gradient warm-up forwards, then mean of 30 CUDA-Event-timed forwards with GPU synchronization.
Peak memory	Peak allocated GPU memory from `torch.cuda.max_memory_allocated`; reserved/cached memory excluded.
FLOPs	Single ETTh1 input of shape $1 \times 512 \times 7$, counted with `FlopCounterMode`.

We report a compact computational comparison on ETTh1 using the mean over the four forecasting horizons. ARBR has the lowest average FLOPs and peak memory usage, PhaseFormer uses the fewest parameters, and PatchTST has the lowest measured latency.

Average computational characteristics on ETTh1.

Table 6: Complete horizon-specific ablation results on seven benchmarks.

Model		ARBR		w/o Local Correction		w/o Scale Adaptation		w/o Center Adaptation		w/o Coefficient Conditioning	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.359	0.389	0.364	0.394	0.366	0.396	0.366	0.394	0.373	0.399
	192	0.394	0.413	0.397	0.415	0.400	0.419	0.400	0.416	0.410	0.424
	336	0.415	0.429	0.420	0.434	0.424	0.437	0.422	0.434	0.432	0.445
	720	0.445	0.465	0.451	0.472	0.459	0.478	0.464	0.473	0.465	0.482
	Avg	0.403	0.424	0.408	0.429	0.412	0.433	0.413	0.429	0.420	0.438
ETTh2	96	0.265	0.332	0.270	0.337	0.271	0.337	0.277	0.342	0.279	0.346
	192	0.325	0.377	0.328	0.376	0.334	0.380	0.336	0.385	0.342	0.393
	336	0.352	0.401	0.354	0.399	0.364	0.408	0.363	0.409	0.371	0.418
	720	0.390	0.432	0.396	0.437	0.408	0.444	0.417	0.449	0.411	0.450
	Avg	0.333	0.386	0.337	0.387	0.344	0.392	0.348	0.396	0.351	0.402
ETTm1	96	0.284	0.337	0.288	0.338	0.294	0.344	0.296	0.343	0.296	0.349
	192	0.321	0.360	0.328	0.364	0.334	0.367	0.330	0.364	0.334	0.375
	336	0.358	0.381	0.361	0.386	0.367	0.386	0.368	0.387	0.374	0.400
	720	0.414	0.411	0.417	0.412	0.426	0.420	0.422	0.416	0.434	0.431
	Avg	0.344	0.372	0.349	0.375	0.355	0.379	0.354	0.378	0.360	0.389
ETTm2	96	0.161	0.250	0.163	0.252	0.166	0.257	0.165	0.253	0.167	0.259
	192	0.213	0.287	0.216	0.289	0.229	0.303	0.227	0.294	0.223	0.297
	336	0.266	0.324	0.271	0.326	0.273	0.327	0.281	0.335	0.277	0.336
	720	0.349	0.381	0.352	0.380	0.357	0.383	0.362	0.386	0.364	0.395
	Avg	0.247	0.311	0.251	0.312	0.256	0.318	0.259	0.317	0.258	0.322
Weather	96	0.141	0.191	0.146	0.196	0.150	0.201	0.151	0.203	0.146	0.197
	192	0.185	0.234	0.190	0.238	0.194	0.243	0.188	0.237	0.193	0.241
	336	0.235	0.275	0.243	0.280	0.243	0.281	0.251	0.287	0.244	0.284
	720	0.311	0.328	0.319	0.334	0.325	0.340	0.328	0.344	0.323	0.338
	Avg	0.218	0.257	0.225	0.262	0.228	0.266	0.230	0.268	0.227	0.265
Electricity	96	0.128	0.222	0.133	0.230	0.135	0.231	0.136	0.233	0.134	0.229
	192	0.147	0.238	0.151	0.245	0.152	0.248	0.154	0.251	0.155	0.246
	336	0.163	0.256	0.166	0.261	0.168	0.263	0.169	0.265	0.172	0.265
	720	0.199	0.289	0.205	0.293	0.207	0.296	0.208	0.297	0.209	0.298
	Avg	0.159	0.251	0.164	0.257	0.166	0.260	0.167	0.262	0.168	0.260
Traffic	96	0.367	0.255	0.380	0.271	0.375	0.260	0.373	0.259	0.381	0.264
	192	0.385	0.263	0.395	0.277	0.393	0.269	0.391	0.267	0.399	0.272
	336	0.395	0.270	0.406	0.283	0.403	0.277	0.404	0.275	0.412	0.280
	720	0.434	0.289	0.442	0.302	0.448	0.297	0.444	0.295	0.453	0.301
	Avg	0.395	0.269	0.406	0.283	0.405	0.276	0.403	0.274	0.411	0.279

	ETTh1			
Models	**Param/M**	**FLOPs/M**	**Latency/ms**	**Peak Mem./GiB**
ARBR	2.0096	**30.4750**	2.5774	**0.2748**
PatchTST	2.2117	276.6750	**2.1012**	2.0619
PhaseFormer	**0.6301**	165.3250	3.7552	1.2788

C Extended Component Ablation

The exact substitutions used by the four ablated variants are summarized below; all unspecified components are identical to the full model.

Implementation of component ablations.

Variant	Exact substitution
Full ARBR	Adaptive $\mu_k(x)$ and $\sigma_k(x)$, input-conditioned $V_k(x)$, and the local residual pathway.
w/o Center Adaptation	Fix $\mu_k = a_k$; retain adaptive scales and all other components.
w/o Scale Adaptation	Fix $\sigma_k = \bar{\sigma}_k$; retain adaptive centers and all other components.
w/o Coefficient Conditioning	Keep the Coefficient Generator architecture unchanged, but replace its input in the coefficient branch only by the shared, non-trainable zero vector $h_{\text{const}} = \mathbf{0} \in \mathbb{R}^D$. The coefficient head remains trainable, giving $V_{\text{global}} = \text{reshape}(b_{\text{coef}})$ shared across inputs; geometry remains input-conditioned. All other components remain unchanged.
w/o Local Correction	Remove the local residual pathway and use $Z = G$.

Table 6 provides the horizon-specific values underlying the four-horizon ablation summaries in the main text.

Table 7: Complete horizon-specific representation-design comparisons on all seven benchmarks. Fixed Scaffold and Global Learned Geometry retain radial mixing; Dynamic Mixing replaces the radial branch with direct input-conditioned mixing as described in Section 5.4. The remaining components are retained. The *Avg* rows are arithmetic means over $H \in \{96, 192, 336, 720\}$, rounded after averaging, and correspond to Table 3.

Parameterization	H	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Electricity		Traffic	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ARBR (Input-Adaptive)	96	0.359	0.389	0.265	0.332	0.284	0.337	0.161	0.250	0.141	0.191	0.128	0.222	0.367	0.255
	192	0.394	0.413	0.325	0.377	0.321	0.360	0.213	0.287	0.185	0.234	0.147	0.238	0.385	0.263
	336	0.415	0.429	0.352	0.401	0.358	0.381	0.266	0.324	0.235	0.275	0.163	0.256	0.395	0.270
	720	0.445	0.465	0.390	0.432	0.414	0.411	0.349	0.381	0.311	0.328	0.199	0.289	0.434	0.289
	Avg	0.403	0.424	0.333	0.386	0.344	0.372	0.247	0.311	0.218	0.257	0.159	0.251	0.395	0.269
Dynamic Mixing	96	0.368	0.396	0.275	0.343	0.295	0.346	0.165	0.257	0.145	0.195	0.132	0.228	0.371	0.258
	192	0.404	0.422	0.338	0.391	0.330	0.370	0.220	0.296	0.191	0.240	0.153	0.245	0.384	0.263
	336	0.414	0.429	0.365	0.414	0.375	0.400	0.275	0.334	0.241	0.281	0.169	0.263	0.402	0.276
	720	0.461	0.481	0.404	0.446	0.431	0.425	0.359	0.392	0.320	0.337	0.206	0.296	0.437	0.290
	Avg	0.412	0.432	0.346	0.399	0.358	0.385	0.255	0.320	0.224	0.263	0.165	0.258	0.399	0.272
Global Learned Geometry	96	0.369	0.399	0.276	0.345	0.294	0.346	0.166	0.258	0.146	0.196	0.133	0.229	0.381	0.263
	192	0.406	0.424	0.338	0.392	0.332	0.372	0.222	0.296	0.192	0.240	0.153	0.246	0.398	0.272
	336	0.428	0.445	0.366	0.416	0.371	0.397	0.276	0.335	0.243	0.282	0.170	0.264	0.410	0.279
	720	0.461	0.482	0.406	0.449	0.431	0.427	0.361	0.393	0.322	0.337	0.207	0.298	0.449	0.298
	Avg	0.416	0.438	0.347	0.401	0.357	0.386	0.256	0.321	0.226	0.264	0.166	0.259	0.410	0.278
Fixed Scaffold	96	0.375	0.400	0.282	0.354	0.300	0.353	0.168	0.262	0.146	0.198	0.136	0.234	0.390	0.267
	192	0.412	0.425	0.346	0.402	0.338	0.380	0.224	0.300	0.193	0.243	0.157	0.252	0.409	0.276
	336	0.435	0.445	0.374	0.426	0.379	0.404	0.278	0.340	0.244	0.285	0.173	0.271	0.420	0.283
	720	0.468	0.483	0.414	0.459	0.440	0.436	0.365	0.399	0.323	0.340	0.211	0.305	0.461	0.303
	Avg	0.423	0.438	0.354	0.410	0.364	0.393	0.259	0.325	0.227	0.267	0.169	0.266	0.420	0.282

D Extended Representation Design Comparisons

Table 7 reports the seed-2021 horizon-specific results underlying Table 3. Dataset averages and relative degradations follow Appendix A.6. Additional three-seed results for Dynamic Mixing are reported below.

Dynamic Mixing substitution. For a batch of $N = BC$ channel-wise inputs, let $B_0 \in (0, 1]^{P\times K}$ denote the radial response rendered from the undeformed scaffold:

$$B_0[p, k] = \exp\left(-\frac{(q_p - a_k)^2}{2\bar{\sigma}_k^2}\right).$$

We clip this matrix as $\widetilde{B}_0 = \mathrm{clip}(B_0, \epsilon, 1-\epsilon)$ with $\epsilon = 10^{-4}$. A linear mixing head directly predicts a residual matrix

$$\Delta A(x) = \mathrm{reshape}_{P\times K}\big(g_{\mathrm{mix}}(h(x))\big),$$

and the mixing weights are

$$A(x) = \mathrm{sigmoid}\left(\mathrm{logit}(\widetilde{B}_0) + \Delta A(x)\right).$$

The mixing head is zero-initialized, so $A(x) = \widetilde{B}_0$ at initialization. During training, each entry of $A(x)$ is predicted directly and is not coupled through centers, scales, or a distance-based radial form.

The coefficient branch still produces $V \in \mathbb{R}^{N\times K\times D}$, and the batched product $G = AV \in \mathbb{R}^{N\times P\times D}$ enters the same local residual pathway and forecasting head. The shared encoder, coefficient generator, patch construction, local pathway, and forecasting head are retained. For $P = 42$, $K = 8$, and $D = 128$, the linear mixing head contains $PK(D + 1) = 43{,}344$ trainable parameters, compared with $2K(D + 1) = 2{,}064$ in the ARBR geometry head.

Training configuration. Dynamic Mixing uses AdamW with a learning rate of 0.0012, zero weight decay, and MSE loss. The learning rate is held constant for two epochs and then follows a cosine decay schedule. Training runs for at most 30 epochs with early-stopping patience of six epochs, and checkpoints are selected by validation MSE. This configuration is fixed across all datasets, horizons, and training seeds.

Table 8: Three-seed stability of Dynamic Mixing. Statistics are computed across training seeds after averaging over the four forecasting horizons.

Dataset	MSE mean $\pm$ std	MAE mean $\pm$ std
ETTh1	0.41158 ± 0.00052	0.43192 ± 0.00038
ETTh2	0.34533 ± 0.00076	0.39850 ± 0.00075
ETTm1	0.35775 ± 0.00075	0.38492 ± 0.00080
ETTm2	0.25458 ± 0.00029	0.31967 ± 0.00038
Weather	0.22400 ± 0.00043	0.26300 ± 0.00043
Electricity	0.16508 ± 0.00063	0.25792 ± 0.00014
Traffic	0.39850 ± 0.00000	0.27167 ± 0.00014

D.1 Dynamic Mixing Stability

We evaluate Dynamic Mixing with training seeds $\{42, 2021, 2026\}$ on all seven datasets. For each seed, MSE and MAE are first averaged over $H \in \{96, 192, 336, 720\}$. Table 8 reports the mean and sample standard deviation of these averages across seeds. Table 3 retains the seed-2021 results.

Dynamic Mixing has higher three-seed mean MSE and MAE than the ARBR results in Appendix B.1 on all seven datasets. These results support the consistency of the comparison across the tested training seeds; they are not an inferential significance test.

E Extended Methodological Details

This appendix gives the implementation-level forward specification used by the reported ARBR model. It expands the compact specification in Section 4 by making the channel-independent preprocessing, shared feature extractor, Geometry/Coefficient Generator implementations, basis evaluation, local projection, query alignment, forecasting head, and inverse normalization explicit.

E.1 Notation and Tensor Shapes

For a mini-batch, let $X \in \mathbb{R}^{B\times L\times C}$ denote the historical input, where B is batch size, L is look-back length, and C is the number of variables. RevIN is applied channel-wise, after which the batch and channel axes are merged to form $N = BC$ univariate sequences. All adaptive geometry and coefficient outputs are therefore conditioned per channel-wise sequence. The default latent dimension is $D = 128$, $K = 8$ radial components are used, and P denotes the number of patch-aligned temporal queries.

Tensor interfaces used by ARBR.

Symbol	Shape	Role
X	$B \times L \times C$	Historical mini-batch before channel-independent reshaping.
X^{norm}	$B \times L \times C$	Channel-wise RevIN-normalized input.
x	$N \times L$, $N = BC$	Flattened collection of univariate channel sequences.
$h(x)$	$N \times D$	Shared full-sequence feature from the linear–GELU–dropout encoder.
$\Delta\mu, \Delta s$	$N \times K$ each	Geometry Generator offsets before bounded deformation.
μ, σ	$N \times K$ each	Input-adaptive radial centers and temporal scales.
V	$N \times K \times D$	Coefficient Generator output for each channel-wise sequence.
q	P	Normalized temporal coordinates aligned with local patches.
B_{rad}	$N \times P \times K$	Query-aligned Gaussian radial responses evaluated by the ARG Basis Evaluator.
G	$N \times P \times D$	Rendered radial representation $B_{\mathrm{rad}}V$.
L_{loc}	$N \times P \times D$	Linearly projected local-patch representation.
Z	$N \times P \times D$	Representation after local residual integration with bounded scalar weight.
$\widehat{y}$	$N \times H$	Channel-wise forecast before restoring batch/channel axes.
$\widehat{Y}$	$B \times H \times C$	Final forecast after reshape and inverse RevIN.

E.2 Channel-Independent Encoder and Generator Implementations

RevIN and channel-independent reshaping. For each variable, RevIN computes channel-wise normalization statistics over the input window and produces X^{norm}. The batch and channel axes are then merged, so $x \in \mathbb{R}^{N\times L}$ with $N = BC$. All subsequent radial and local modules use shared parameters across these univariate sequences. After prediction, the $N \times H$ outputs are reshaped to $B \times H \times C$ and inverse RevIN restores the pre-RevIN scale.

Shared sequence feature extractor. The full look-back window is compressed by

$$h(x) = \mathrm{Dropout}_{p_{\mathrm{gs}}}\big(\mathrm{GELU}\big(xW_{\mathrm{seq}}^{\top} + b_{\mathrm{seq}}\big)\big), \qquad W_{\mathrm{seq}} \in \mathbb{R}^{D\times L},$$

which corresponds to $\mathrm{Linear}(L, D)$ followed by GELU and dropout. For the default $D = 128$ and $L = 512$, this is a $512 \to 128$ projection. The same $h(x)$ conditions the Geometry Generator in ARG and the Coefficient Generator in RCG.

Geometry Generator in ARG. The Geometry Generator first applies a zero-initialized linear geometry head mapping $D \to 2K$:

$$[\Delta\mu(x), \Delta s(x)] = h(x)W_{\mathrm{geo}}^{\top} + b_{\mathrm{geo}},$$

where W_{geo} and b_{geo} are initialized to zero. Thus $\Delta\mu = \Delta s = 0$ at initialization. The generator then applies the scaffold-conditioned bounded transformations in Equations 8 and 9, producing (μ, σ); consequently the initial generated geometry equals the shared scaffold.

Basis Evaluator in ARG. Given (μ, σ) from the Geometry Generator and the patch-aligned query q constructed in Appendix E.3, the Basis Evaluator evaluates Equation 1 to produce $B_{\mathrm{rad}} \in \mathbb{R}^{N\times P\times K}$. The evaluator introduces no additional learned parameters; it realizes the generated geometry at the query positions. Thus ARG terminates at the geometry-side output B_{rad}.

Coefficient Generator in RCG. The Coefficient Generator uses a linear coefficient head $D \to KD$ and reshapes its output over the radial-component axis:

$$V(x) = \mathrm{reshape}_{K\times D}\big(h(x)W_{\mathrm{coef}}^{\top} + b_{\mathrm{coef}}\big)\,.$$

For $D = 128$ and $K = 8$, the internal linear head is a $128 \to 1024$ projection. The reshaped tensor $V \in \mathbb{R}^{N\times K\times D}$ is the content-side output of RCG.

Radial rendering. Given B_{rad} and V, a batched matrix multiplication forms $G = B_{\mathrm{rad}}V$.

Local residual pathway and forecasting head. The normalized univariate sequence is also divided into local patches with length 24 and stride 12. Each patch is linearly projected to $D = 128$, producing $L_{\mathrm{loc}} \in \mathbb{R}^{N\times P\times D}$. After $Z = G + \lambda L_{\mathrm{loc}}$, the tensor is permuted to $N \times D \times P$, flattened to $N \times (DP)$, passed through forecasting head dropout, and projected by a single linear layer $DP \to H$. For the representative $L = 512$ trace with $P = 42$ and $D = 128$, the flattened width is 5376.

E.3 Patch-Aligned Query Construction

The ARG Basis Evaluator operates at exactly the same temporal resolution as the local residual pathway. Let the patch length be S and the stride be r. Rather than discarding a trailing incomplete interval, the implementation uses a ceiling-based patch count so that the observed sequence is covered through its right boundary:

$$P(L) = \left\lceil \frac{L-S}{r} \right\rceil + 1.$$

For the reported $L = 512$, $S = 24$, and $r = 12$, this gives $P = \lceil 488/12 \rceil + 1 = 42$.

The patch extractor applies no padding at the left boundary. The minimum right-padding required to materialize all P patches is

$$\mathrm{Pad}_{\mathrm{right}} = S + (P-1)r - L.$$

The added values use replication padding, i.e., every padded sample copies the last observed value. Hence the default $L = 512$, $S = 24$, $r = 12$ setting pads four positions on the right and none on the left.

Let

$$\mathcal{P}(x) = \{P_p\}_{p=1}^{P}$$

denote the ordered patches. The temporal reference assigned to each patch is its replication-aware center. Using one-based patch index p, the raw center coordinate is

$$t_p^{\mathrm{raw}} = \frac{1}{S}\sum_{i=0}^{S-1} \min\big((p-1)r + i,\ L-1\big), \qquad p = 1, \ldots, P.$$

For patches that do not touch the right boundary, $t_p^{\mathrm{raw}} = (p-1)r + (S-1)/2$. For the final padded patch, indices beyond $L-1$ are clamped to $L-1$ before averaging, matching the replication-padding semantics.

Time coordinates are normalized by the largest valid input index:

$$\mathcal{N}_t(t) = \frac{t}{L-1}.$$

The radial query associated with patch p is therefore

$$q_p = \mathcal{N}_t(t_p^{\mathrm{raw}}) = \frac{1}{S(L-1)}\sum_{i=0}^{S-1} \min\big((p-1)r + i,\ L-1\big) \in [0,1].$$

Thus q, G, and L_{loc} have the same P positions by construction. For $L = 512$, $S = 24$, and $r = 12$, the first query is $q_1 = 11.5/511 \approx 0.0225$, the center of the 41st patch is $q_{41} = 491.5/511 \approx 0.9618$, and the final replication-aware query is $q_{42} \approx 503.0833/511 \approx 0.9845$.

E.4 Properties of Bounded Adaptive Geometry

Because $|\tanh(\cdot)| < 1$, the center displacement satisfies

$$|\mu_k(x) - a_k| < \frac{\rho}{K}, \qquad a_k - \frac{\rho}{K} < \mu_k(x) < a_k + \frac{\rho}{K}.$$

For any $0 < \rho \leq 0.5$, adjacent anchors satisfy $a_{k+1} - a_k = 1/K$, and the center bound above gives

$$\mu_k(x) < a_k + \frac{\rho}{K} \leq a_k + \frac{1}{2K} = a_{k+1} - \frac{1}{2K} \leq a_{k+1} - \frac{\rho}{K} < \mu_{k+1}(x).$$

Hence the adapted radial centers remain strictly ordered for all inputs within this deformation range; the reported $\rho = 0.5$ is the largest value covered by the ordering argument. This statement applies to the centers and does not require the adapted scales themselves to remain sorted.

The scale deformation obeys

$$e^{-\beta}\bar{\sigma}_k < \sigma_k(x) < e^{\beta}\bar{\sigma}_k,$$

or equivalently $|\log(\sigma_k(x)/\bar{\sigma}_k)| < \beta$. With $\beta = 0.5$, every adapted scale lies within the multiplicative interval $(e^{-0.5}, e^{0.5})$ around its reference scale. The reference scales themselves are geometrically spaced from $\sigma_{\min} = 0.05$ to $\sigma_{\max} = 0.4$.

These bounds also explain the normalization used by the mechanism statistics in Section 5.5: dividing center displacement by ρ/K and log-scale displacement by β measures how much of the permitted deformation range is used.

E.5 Complete Forward Formulation

Algorithm 1 summarizes the complete channel-independent forward pass, including the exact patch/query construction used by the implementation.

Algorithm 1: Forward pass of ARBR.

1. Apply channel-wise RevIN to $X \in \mathbb{R}^{B\times L\times C}$ and merge batch and channel axes to obtain $x \in \mathbb{R}^{BC\times L}$.
2. Compute $h(x)$ with $\mathrm{Linear}(L, 128)$, GELU, and dropout.
3. In the ARG Geometry Generator, apply the zero-initialized $128 \to 2K$ geometry head, split its output into $\Delta\mu$ and Δs, and apply the bounded transform in Equations 8 and 9 to obtain (μ, σ).
4. In the RCG Coefficient Generator, apply the $128 \to K \cdot 128$ coefficient head and reshape its output to $V \in \mathbb{R}^{N\times K\times 128}$.
5. With patch length $S = 24$ and stride $r = 12$, compute $P = \lceil (L - S)/r \rceil + 1$, apply the required right-side replication padding, extract the P local patches, and linearly project them to 128 dimensions. Construct the one-to-one radial queries from the replication-aware patch centers using $q_p = t_p^{\mathrm{raw}}/(L-1)$ as defined in Appendix E.3.
6. In the ARG Basis Evaluator, evaluate $B_{\mathrm{rad}}(q, \mu, \sigma)$.
7. Form $G = B_{\mathrm{rad}}V$ by batched matrix multiplication.
8. Compute $\lambda = 0.5\ \mathrm{sigmoid}(a)$, initialized at $\lambda = 0.1$, and form $Z = G + \lambda L_{\mathrm{loc}}$.
9. Permute Z to $N \times 128 \times P$, flatten the last two dimensions, apply the forecasting head dropout, and linearly project $128P \to H$.
10. Reshape the channel-wise forecasts to $B \times H \times C$ and apply inverse RevIN to produce $\widehat{Y}$.

Training uses the AdamW/MSE configuration in Appendix A.3. The Geometry Generator and Basis Evaluator form ARG and produce B_{rad}, while the Coefficient Generator in RCG produces V. Their product is integrated with the local pathway by CLRC before forecasting.

E.6 Mechanism Analysis Protocol

The shift and dilation diagnostics use normalized Weather test windows. Let x denote an unperturbed channel-wise window and x' its temporally transformed counterpart. We measure changes

relative to the original window, normalized by the center displacement bound and the log-scale deformation bound:

$$\delta\mu_k(x', x) = \frac{\mu_k(x') - \mu_k(x)}{\rho/K} = \tanh\big(\Delta\mu_k(x')\big) - \tanh\big(\Delta\mu_k(x)\big),$$
$$\delta s_k(x', x) = \frac{\log\sigma_k(x') - \log\sigma_k(x)}{\beta} = \tanh\big(\Delta s_k(x')\big) - \tanh\big(\Delta s_k(x)\big).$$

Here, $\Delta\mu_k$ and Δs_k are the raw geometry-generator outputs defined in Section 4.2. Figures 3(b) and 3(c) report the means of these normalized changes, with error bars showing standard deviations across channel-wise test windows and components. The protocols for geometry usage and temporal perturbations are summarized below.

Controlled perturbation and aggregation protocol.

Item	Protocol
Perturbation data	Official Weather test split after RevIN; $L = 512$, $C = 21$, 10,444 windows, yielding 219,324 channel-wise sequences.
Temporal shift	Nominal normalized shifts $\Delta t/(L/K) \in \{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6\}$, with anchor spacing $L/K = 64$; implemented as integer shifts $\{-38, -26, -13, 0, 13, 26, 38\}$ with edge-replication padding.
Shift statistic	Normalized center change $\delta\mu_k(x', x)$, with x' the shifted window; mean and standard deviation over all channel-wise test windows and $K = 8$ components.
Temporal dilation	$\log(w/w_0) \in \{-0.4, -0.2, 0, 0.2, 0.4\}$, corresponding to scale factors $\{0.6703, 0.8187, 1, 1.2214, 1.4918\}$; 1-D linear interpolation with `align_corners=True`, followed by symmetric replicate padding for compression or center cropping for dilation.
Dilation statistic	Normalized log-scale change $\delta s_k(x', x)$, with x' the compressed or dilated window; mean and standard deviation over all channel-wise test windows and $K = 8$ components.
Geometry usage	Use seed-2021 checkpoints with $H = 96$ on ETTm1, Weather, and Traffic. Scores are averaged over components before pooling channel-wise test windows. Gaussian KDE uses Scott's bandwidth rule and is displayed only between the empirical 0.1st and 99.9th percentiles.

Test-time coefficient substitution. For each horizon, we use the seed-2021 checkpoint and perform five random derangements of test-window indices using seeds 2027–2031, preserving channel identity and excluding self-pairings. Relative MSE increases are computed against the corresponding unmodified checkpoint and averaged over assignments within each horizon. Figure 3(d) reports the mean and population standard deviation of these four horizon-level averages.

F SENSITIVITY ANALYSIS

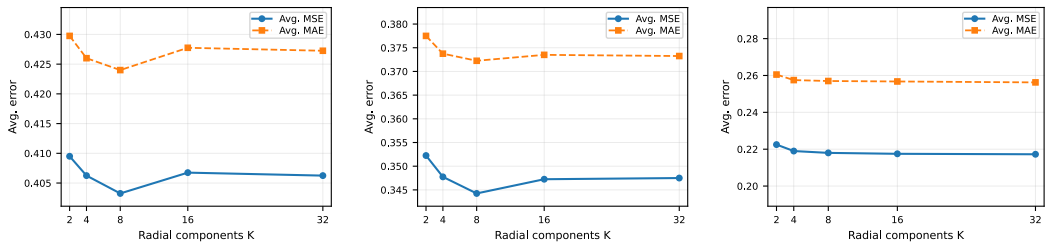
We complement the main forecasting and design studies with sensitivity analyses for the number of radial components, the center-deformation bound ρ, and the log-scale deformation bound β. All sweeps are reported on ETTh1, ETTm1, and Weather and include the four forecasting horizons together with their averages.

F.1 NUMBER OF RADIAL COMPONENTS

Table 9 varies the number of radial components over $K \in \{2, 4, 8, 16, 32\}$, and Figure 5 shows the corresponding four-horizon average MSE and MAE. $K = 8$ gives the lowest average MSE on ETTh1 and ties $K = 32$ for the lowest average MAE; on ETTm1, $K = 8$ and $K = 16$ tie on both reported averages; and on Weather, $K = 32$ gives the lowest averages (0.217 MSE and 0.256 MAE). Overall, performance remains stable across the stronger K settings rather than improving monotonically with the number of radial components. Because K changes the geometry-head and coefficient-head dimensions, these results characterize sensitivity to the radial-component configuration.

Table 9: Complete sensitivity results for the number of radial components K on ETTh1, ETTm1, and Weather. All forecasting horizons are reported. Lower is better; the best result in each dataset–horizon row is shown in bold.

Radial Components		K = 2		K = 4		K = 8		K = 16		K = 32	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.365	0.395	0.362	0.391	**0.359**	**0.389**	0.360	0.390	0.361	0.390
	192	0.402	0.420	0.395	**0.413**	**0.394**	**0.413**	0.396	0.415	0.396	0.415
	336	0.421	0.435	0.423	0.436	**0.415**	0.429	**0.415**	0.429	**0.415**	**0.428**
	720	0.450	0.469	0.445	0.464	0.445	0.465	0.446	0.466	**0.444**	**0.463**
	Avg	0.409	0.430	0.406	0.426	**0.403**	**0.424**	0.404	0.425	0.404	**0.424**
ETTm1	96	0.291	0.343	0.286	**0.337**	**0.284**	**0.337**	**0.284**	0.338	**0.284**	0.338
	192	0.336	0.369	0.332	0.366	**0.321**	**0.360**	**0.321**	**0.360**	0.322	0.362
	336	0.363	0.386	0.359	0.381	0.358	0.381	**0.357**	**0.380**	0.359	**0.380**
	720	0.419	0.412	**0.414**	**0.411**	**0.414**	**0.411**	**0.414**	0.412	**0.414**	**0.411**
	Avg	0.352	0.377	0.348	0.374	**0.344**	**0.372**	**0.344**	**0.372**	0.345	0.373
Weather	96	0.145	0.195	0.141	**0.190**	0.141	0.191	**0.140**	**0.190**	**0.140**	**0.190**
	192	0.189	0.237	0.186	0.235	0.185	0.234	0.185	0.234	**0.184**	**0.233**
	336	0.242	0.279	0.237	0.276	0.235	0.275	**0.233**	0.275	**0.233**	**0.274**
	720	0.314	0.331	0.312	0.329	**0.311**	**0.328**	0.312	**0.328**	0.312	**0.328**
	Avg	0.222	0.261	0.219	0.258	0.218	0.257	0.218	0.257	**0.217**	**0.256**



(a) ETTh1. (b) ETTm1. (c) Weather.

Figure 5: Sensitivity to the number of radial components on ETTh1, ETTm1, and Weather. Each panel reports the MSE and MAE averaged over the four forecasting horizons for $K \in \{2, 4, 8, 16, 32\}$. Lower values are better.

F.2 CENTER- AND SCALE-DEFORMATION BOUNDS

We additionally examine the two bounded-deformation hyperparameters used by the Geometry Generator. The center-deformation sweep varies $\rho \in \{0.25, 0.5\}$ while holding $\beta = 0.5$, and the scale-deformation sweep varies $\beta \in \{0.25, 0.5, 0.75\}$ while holding $\rho = 0.5$. All other settings are kept at the reported defaults. Tables 10 and 11 report the complete horizon-specific results.

Across ETTh1, ETTm1, and Weather, the default values $\rho = 0.5$ and $\beta = 0.5$ are consistently best or tied on the four-horizon averages. Reducing either bound to 0.25 produces a small but systematic degradation. Increasing β to 0.75 remains close to the default without yielding a consistent improvement. Averaged over all twelve dataset–horizon combinations, the ρ sweep gives MSE/MAE values of $0.324/0.353$ and $0.322/0.351$ for $\rho = 0.25$ and 0.5, respectively; the corresponding β sweep gives $0.323/0.353$, $0.322/0.351$, and $0.322/0.351$. These results indicate that the model is not highly sensitive to the tested changes around the default deformation bounds.

Table 10: Complete sensitivity results for the center-deformation bound ρ on ETTh1, ETTm1, and Weather. All forecasting horizons are reported. Lower is better; the best result in each dataset–horizon row is shown in bold.

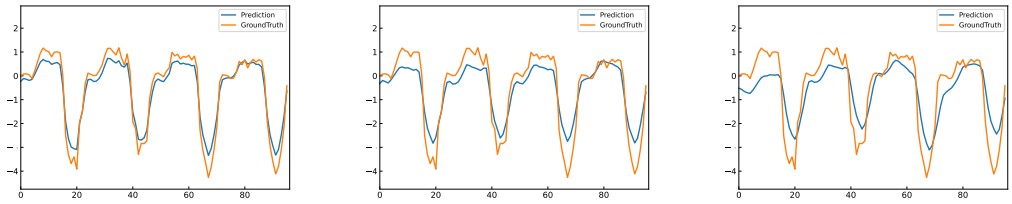
Center Bound		$\rho = 0.25$		$\rho = 0.5$	
Metric		MSE	MAE	MSE	MAE
ETTh1	96	0.361	0.391	**0.359**	**0.389**
	192	0.395	**0.413**	**0.394**	**0.413**
	336	**0.415**	**0.429**	**0.415**	**0.429**
	720	0.446	0.466	**0.445**	**0.465**
	Avg	0.404	0.425	**0.403**	**0.424**
ETTm1	96	0.286	0.338	**0.284**	**0.337**
	192	0.326	0.361	**0.321**	**0.360**
	336	0.359	**0.381**	**0.358**	**0.381**
	720	0.415	0.412	**0.414**	**0.411**
	Avg	0.346	0.373	**0.344**	**0.372**
Weather	96	0.143	0.193	**0.141**	**0.191**
	192	0.186	0.236	**0.185**	**0.234**
	336	0.242	0.282	**0.235**	**0.275**
	720	0.312	0.329	**0.311**	**0.328**
	Avg	0.221	0.260	**0.218**	**0.257**

Table 11: Complete sensitivity results for the log-scale deformation bound β on ETTh1, ETTm1, and Weather. All forecasting horizons are reported. Lower is better; the best result in each dataset–horizon row is shown in bold.

Scale Bound		$\beta = 0.25$		$\beta = 0.5$		$\beta = 0.75$	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.360	0.390	**0.359**	**0.389**	**0.359**	**0.389**
	192	0.395	**0.413**	**0.394**	**0.413**	**0.394**	**0.413**
	336	**0.415**	**0.429**	**0.415**	**0.429**	**0.415**	**0.429**
	720	**0.445**	**0.465**	**0.445**	**0.465**	**0.445**	**0.465**
	Avg	0.404	0.424	**0.403**	**0.424**	**0.403**	**0.424**
ETTm1	96	0.286	0.338	**0.284**	**0.337**	**0.284**	**0.337**
	192	0.324	0.362	**0.321**	**0.360**	**0.321**	**0.360**
	336	0.359	0.381	**0.358**	0.381	**0.358**	**0.380**
	720	0.416	0.412	**0.414**	**0.411**	0.416	0.412
	Avg	0.346	0.373	**0.344**	**0.372**	0.345	**0.372**
Weather	96	0.145	0.195	**0.141**	**0.191**	0.143	0.193
	192	**0.185**	0.236	**0.185**	**0.234**	**0.185**	0.235
	336	0.240	0.280	**0.235**	**0.275**	0.236	**0.275**
	720	**0.311**	0.329	**0.311**	**0.328**	**0.311**	**0.328**
	Avg	0.220	0.260	**0.218**	**0.257**	0.219	0.258

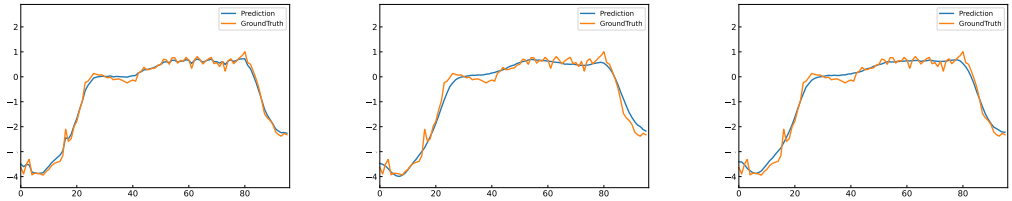
G ADDITIONAL FORECASTING VISUALIZATIONS

We show ground-truth and prediction trajectories for ARBR, PhaseFormer, and PatchTST on ETTh1, ETTm1, Weather, and Traffic at $H = 96$. All displayed predictions are produced by seed-2021 runs. Each row corresponds to one dataset, with models ordered as ARBR, PhaseFormer, and PatchTST.



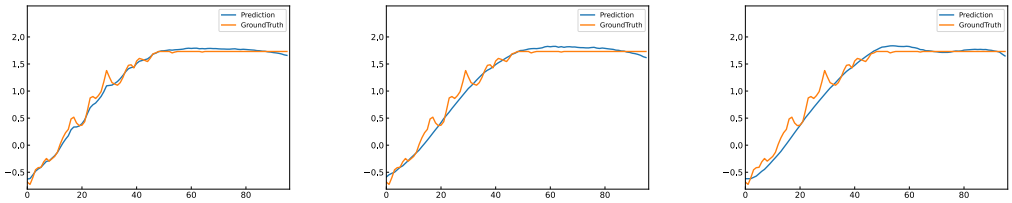
(a) ARBR (b) PhaseFormer (c) PatchTST

Figure 6: Ground-truth and prediction examples on ETTh1 at $H = 96$.



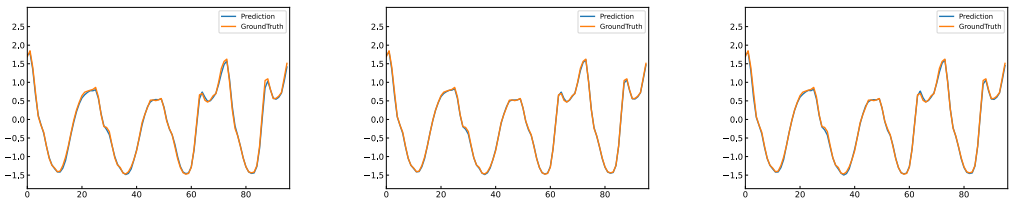
(a) ARBR (b) PhaseFormer (c) PatchTST

Figure 7: Ground-truth and prediction examples on ETTm1 at $H = 96$.



(a) ARBR (b) PhaseFormer (c) PatchTST

Figure 8: Ground-truth and prediction examples on Weather at $H = 96$.



(a) ARBR (b) PhaseFormer (c) PatchTST

Figure 9: Ground-truth and prediction examples on Traffic at $H = 96$.