TailFix: Mitigating Error Accumulation and Correcting Tail Deterioration in Long-Horizon Forecasting

Hua Wang
School of Computer and Artificial Intelligence
Ludong University
Yantai 264025, China
hwa229@163.com

Haijing Gao
School of Computer and Artificial Intelligence
Ludong University
Yantai 264025, China
gaohaijing@ldu.edu.cn

Fan Zhang*
School of Computer Science and Technology
Shandong Technology and Business University
Yantai 264005, China
zhangfan@sdtbu.edu.cn

Abstract

Long-term time-series forecasting (LTSF) underpins applications such as energy-load prediction and traffic-flow forecasting, yet model performance often deteriorates as the horizon extends. In particular, errors accumulate with lead time and systematically amplify toward the end of the predicted sequence, a phenomenon known as tail deterioration. We posit that this instability is driven by weakly constrained fusion of cross-temporal and cross-scale context: aggressive mixing can propagate noise and early-stage bias into distant horizons, ultimately degrading tail accuracy. To address this issue, we propose TailFix, a controllable cross-scale mixing framework for long-horizon forecasting. TailFix selectively gates interactions among multi-scale representations, derives scale-level correction terms from contextual summaries, and injects them via gated residual connections to suppress error diffusion and improve long-horizon stability. We further incorporate adaptive channel enhancement and a lightweight tail-correction module to explicitly refine the error-prone tail region. Finally, we introduce the Tail Amplification Factor (TAF) to quantify tail error amplification relative to the early portion of the sequence. Across diverse benchmarks, TailFix consistently mitigates tail error amplification and improves performance on multiple long-horizon forecasting tasks.
Code Repository: https://github.com/Xbocai1103/TailFix

1 Introduction

With rapid advances in deep learning for time-series modeling, neural forecasting methods have delivered strong performance in short-horizon prediction and have been deployed across diverse domains, including energy demand [23], traffic flow [7], financial markets [32], and climate/weather systems [13]. Representative neural forecasting paradigms range from autoregressive recurrent models [3, 14] to attention-based multi-horizon predictors [8], and more recent long-sequence backbones built upon Transformers [6, 17].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

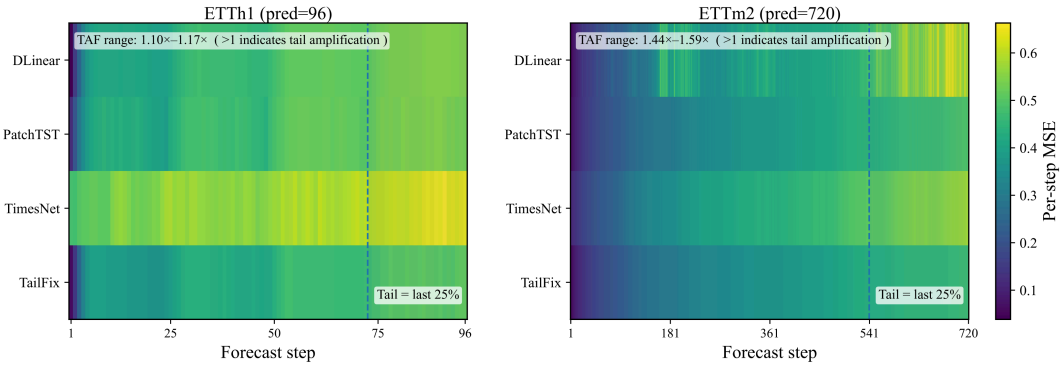


Figure 1: Tail deterioration in long-horizon forecasting. Color shows per-step MSE. The shaded area denotes the tail region.

However, when forecasting is extended from short to long horizons, performance often degrades markedly, and predictions become increasingly unstable and unreliable as the horizon grows [6, 30]. This observation suggests that the central challenge in long-horizon forecasting is not simply refitting historical patterns, but limiting error propagation during multi-step extrapolation [6, 16]. In long prediction sequences, errors are not uniformly distributed; instead, they tend to concentrate and amplify toward the end of the horizon, degrading overall forecast quality and reducing the reliability of downstream decision-making. This phenomenon is related to the broader distribution-shift gap in multi-step prediction [4]; in our direct multi-step (DMS) setting, the amplification is not caused by autoregressive roll-forward, but by error diffusion during long-range extrapolation and insufficiently controlled information fusion across time. Therefore, elucidating the mechanisms underlying error formation and amplification in long-horizon forecasting, and designing architectures that improve predictive stability, have become pressing priorities in time-series research. As illustrated in Figure 1, errors accumulate over the horizon and concentrate toward the tail, destabilizing long-range extrapolation. We hypothesize that insufficiently controlled cross-temporal fusion contributes to tail deterioration by propagating local noise and early-stage biases toward distant horizons.

Tail Amplification Factor (TAF). Given horizon H and tail ratio ρ, TAF is defined as the ratio of the mean per-step error in the tail (last $\lceil \rho H \rceil$ steps) to that in the head (first $H - \lceil \rho H \rceil$ steps):

$$\mathrm{TAF} = \frac{\frac{1}{\lceil \rho H \rceil} \sum_{t=H-\lceil \rho H \rceil+1}^{H} e_t}{\frac{1}{H-\lceil \rho H \rceil} \sum_{t=1}^{H-\lceil \rho H \rceil} e_t}, \quad \rho = 0.25, \tag{1}$$

where e_t denotes the per-step prediction error (e.g., MSE or MAE). This issue is especially prominent in practical applications. For instance, in regional power load forecasting during high summer temperatures, models often capture daily electricity consumption patterns accurately in the early stages of prediction [6, 23]. However, even a slight underestimation of load in the early stages can be amplified over subsequent days of extrapolation, ultimately leading to a significant underestimation of peak load, thereby impacting power grid scheduling and safety decisions. Similarly, in forecasting urban transportation demand before and after major holidays, the model may accurately capture conventional commuting patterns in the short term [7]. However, as the prediction extends into the post-holiday period, demand changes not fully characterized in the early stages will accumulate over time, leading to significant flow drift or even structural distortion at the end of the prediction horizon.

The main contributions of this work are as follows:

- **Phenomenon and metric.** We identify and visualize tail deterioration (Figure 1) and propose TAF to quantify tail error amplification relative to the head for assessing long-horizon stability.
- **Controllable cross-scale hybrid framework.** We propose TailFix, which selectively gates and controllably fuses cross-temporal and cross-scale contextual interactions. TailFix preserves long-range dependencies while suppressing noise and the propagation of early bias

over long horizons, thereby structurally mitigating error accumulation during multi-step extrapolation.

- **Improved tail stability.** We combine adaptive channel enhancement with a dedicated tail Refiner module to explicitly correct the error-sensitive end of the forecast window, improving the stability and reliability of long-horizon predictions. Extensive experiments on multiple long-horizon forecasting benchmarks validate the effectiveness of our approach.

2 Related work

Long-term time-series forecasting seeks to predict extended future horizons from limited historical observations; however, a central challenge is that forecasting errors tend to accumulate over time, increasingly degrading predictions at more distant horizons [5, 15, 30]. Existing Transformer-based methods, including Informer [33], Autoformer [24], FEDformer [36], ETSformer [22], FiLM [35], Crossformer [31], and iTransformer [10], mainly improve long-range dependency modeling through sparse attention, sequence decomposition, frequency-domain analysis, periodicity modeling, or variable-wise representation learning. Although these approaches have substantially improved overall predictive accuracy, most remain optimized for average error reduction and do not explicitly model error amplification near the tail of long prediction windows [15]. In contrast, TailFix shifts the emphasis from average predictive performance to tail reliability by explicitly addressing error diffusion and amplification toward the end of the prediction horizon in long-term extrapolation.

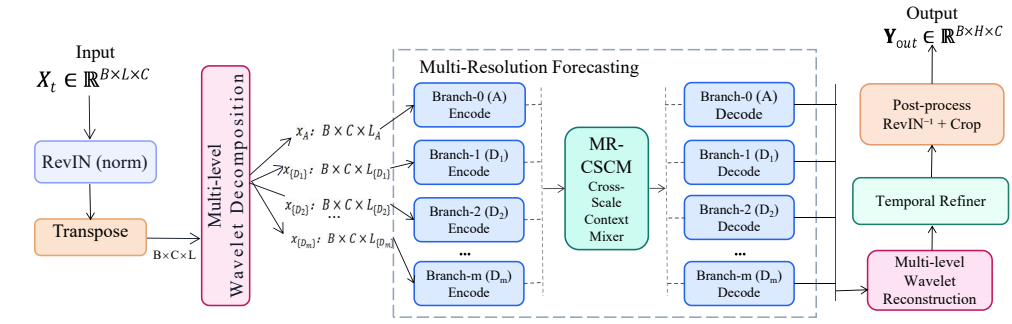
Beyond attention-based models, numerous lightweight architectures have recently been developed for long-term forecasting [1, 9, 21, 30]. Linear models such as DLinear emphasize stable and efficient trend extrapolation [30], whereas multi-scale convolutional, periodicity-based, and frequency-domain methods enhance temporal pattern extraction through structural inductive biases [9, 18, 19, 25, 26, 36]. Patch- and mixer-based paradigms, including PatchTST [12], TSMixer [1], TimeMixer [21], and WPMixer [11], improve the efficiency of long-sequence modeling through sequence patching, feature mixing, and multi-scale processing. These approaches typically offer low computational overhead and strong benchmark performance; however, they often rely on uniform global mixing strategies for cross-patch, cross-channel, or cross-scale interactions, without selectively controlling the intensity or direction of information injection [21]. To address this limitation, TailFix introduces a controllable cross-scale contextual mixing mechanism that formulates cross-scale interactions as gated, incremental residual corrections, thereby preserving multi-scale information sharing while suppressing the unconstrained propagation of noise and early-stage biases in long-term forecasting. Related studies in other domains have explored semantic guidance, structured memory, selective attention, sample valuation, and data harmonization [2, 27–29, 34].

Overall, existing research on long-term forecasting has primarily focused on average predictive accuracy, long-range dependency modeling, and computational efficiency [10, 12, 15, 20, 21, 24, 33, 36], whereas mechanism-oriented designs for improving tail stability remain underexplored. In direct multi-step forecasting, tail degradation does not primarily arise from autoregressive error accumulation; instead, it is more likely driven by error diffusion during long-range extrapolation, uncontrolled cross-scale fusion, and insufficient targeted correction near the prediction endpoint [5, 15]. To address these issues, TailFix further incorporates channel-adaptive enhancement and a lightweight temporal-domain tail-refinement module: the former adaptively reweights variable-wise contributions to reduce interference from redundant channels, whereas the latter applies stable residual corrections to error-sensitive tail regions within the prediction window. Through these designs, TailFix aims not only to reduce overall prediction error but also to mitigate tail-error amplification with minimal additional overhead, thereby enhancing the stability and reliability of long-term forecasting.

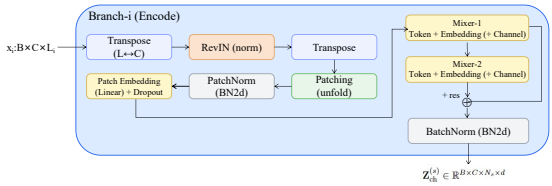
3 Method

3.1 Problem definition

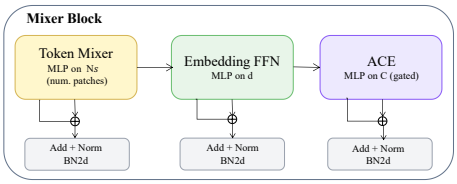
Let L denote the history length, H the prediction horizon (i.e., the number of forecast steps), C the number of variables, S the number of scales, N_s the number of patches at scale s, and d the hidden dimension.



(a) Overall framework.



(b) Branch encoder pipeline.



(c) Mixer block details.

Figure 2: Overall architecture of TailFix. (a) Following RevIN and multi-level wavelet decomposition, the input is modeled using multi-scale branches and undergoes cross-scale gated interaction via MR-CSCM. The result is then reconstructed back into the time domain and fed into the Temporal Refiner. (b) The coefficient sequences are normalized, patched, and embedded, before being passed through stacked Mixer modules for feature mixing. (c) Each Mixer consists of token/embedding mixing and may optionally incorporate gated channel mixing to enhance cross-variable modeling. In (c), ACE is implemented as a gated channel-mixing MLP over the channel dimension C.

In long-term time-series forecasting (LTSF), we consider a multivariate sequence $\{x_t\}_{t=1}^{T}$ with C variables, where $x_t \in \mathbb{R}^C$. At time t, the model takes as input a historical window of length L,

$$\mathbf{X}_t = [x_{t-L+1}, \ldots, x_t] \in \mathbb{R}^{L\times C}, \tag{2}$$

and aims to predict the subsequent H steps,

$$\mathbf{Y}_t = [x_{t+1}, \ldots, x_{t+H}] \in \mathbb{R}^{H\times C}. \tag{3}$$

We adopt a direct multi-step forecasting setting, in which the model produces the length-H prediction $\hat{\mathbf{Y}}_t$ in a single forward pass. In long-horizon forecasting, increasing the prediction horizon H amplifies error accumulation during multi-step extrapolation and leads to pronounced instability in the tail of the prediction window. Accordingly, we focus on improving the stability of long-span extrapolation, with an emphasis on predictive reliability in the tail segment.

To clarify the end-to-end data flow and the placement of each module in TailFix, Figure 2 summarizes the overall architecture and the key submodules.

We then describe, in turn, the cross-scale gated interaction module (MR-CSCM), the channel-adaptive enhancement module (ACE), and the tail-segment refinement module (Temporal Refiner). Together, these modules form a coordinated correction pathway across the scale, variable, and forecasting-step dimensions. Throughout the paper we focus on the DMS setting, where the whole horizon is predicted in a single forward pass.

3.2 Multi-Resolution Cross-Scale Context Mixer

In long-term forecasting, as the prediction horizon H increases, errors tend to propagate during multi-step extrapolation, with amplification occurring in the tail segment of the prediction. One potential contributing factor is insufficient control over cross-scale information fusion: when time-scale representations are mixed uniformly, local noise and early biases can persist, propagating into the long term and exacerbating tail instability.

To address this issue, we propose the Multi-Resolution Cross-Scale Context Mixer (MR-CSCM), a gated residual interaction mechanism for controllable cross-scale fusion, to enable lightweight, controllable, and selective cross-scale information sharing, without compromising scale-specific modeling stability, thereby suppressing error cascade propagation.

3.2.1 Cross-Scale Context Representation

Assume the model consists of S scale branches, indexed by $s \in \{1, \ldots, S\}$. For the s-th scale branch, the patch tokens after embedding are denoted as follows:

$$\mathbf{Z}^{(s)} \in \mathbb{R}^{B \times C \times N_s \times d}, \tag{4}$$

where B denotes the batch size.

To prevent patch-level noise from directly influencing cross-scale interactions, we perform average aggregation along the patch dimension to construct a scale-level context summary:

$$\mathbf{u}^{(s)} = \frac{1}{N_s} \sum_{p=1}^{N_s} \mathbf{Z}^{(s)}_{:,:,p,:} \in \mathbb{R}^{B \times C \times d}. \tag{5}$$

By stacking the summaries from all scales, we obtain the scale context tensor:

$$\mathbf{U} = \mathrm{Stack}\left(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(S)}\right) \in \mathbb{R}^{B \times C \times S \times d}. \tag{6}$$

In this representation, the third dimension of $\mathbf{U}$ corresponds to the scale index. This representation confines cross-scale interaction to a more stable, scale-level semantic space, thereby reducing the risk of noise amplification during cross-scale fusion.

3.2.2 Scale-wise Interaction and Incremental Correction

We perform lightweight interaction exclusively along the scale dimension S. For each (b, c, k) (batch, channel, and hidden index), we define the scale vector as:

$$\mathbf{v}_{b,c,k} = \mathbf{U}_{b,c,:,k} \in \mathbb{R}^S. \tag{7}$$

An incremental correction is computed using an MLP along the scale dimension:

$$\Delta\mathbf{v}_{b,c,k} = f_{cs}(\mathbf{v}_{b,c,k}) \in \mathbb{R}^S. \tag{8}$$

where f_{cs} represents a two-layer MLP, with the hidden width controlled by an expansion factor κ (corresponding to *cross_scale_expand*) . Rearranging all $\Delta\mathbf{v}_{b,c,k}$ back into tensor form yields:

$$\Delta\mathbf{U} = f_{cs}(\mathbf{U}) \in \mathbb{R}^{B \times C \times S \times d}. \tag{9}$$

This design ensures that inter-scale information is introduced gradually as a correction term, rather than disrupting the original temporal modeling structure at each scale. We explicitly model cross-scale interaction as an incremental correction rather than a replacement term, providing candidate scale information for subsequent controllable injection and avoiding overly strong mixing, which may cause scale representation drift and induce long-sequence instability.

3.2.3 Gated Residual Injection and Stable Initialization

For each scale s, we learn a scalar gating coefficient, given by:

$$\gamma_s = \sigma(\alpha_s) \in (0, 1), \tag{10}$$

where α_s is a learnable parameter and $\sigma(\cdot)$ represents the Sigmoid function. Let

$$\Delta u^{(s)} = \Delta U_{:,:,s} \in \mathbb{R}^{B \times C \times d}. \tag{11}$$

which is then broadcast back to the patch dimension and injected in a residual manner:

$$\begin{gathered} \hat{Z}^{(s)} = Z^{(s)} + \gamma_s \cdot \mathrm{Broadcast}\left(\Delta u^{(s)}\right), \\ \hat{Z}^{(s)} \in \mathbb{R}^{B \times C \times N_s \times d}. \end{gathered} \tag{12}$$

Broadcast$(\cdot)$ replicates a tensor of shape $B \times C \times d$ along the patch dimension to obtain a tensor of shape $B \times C \times N_s \times d$. This mechanism offers two key advantages:

Controllability and selectivity: γ_s enables the model to adaptively control the strength of cross-scale information injection at each scale, thus suppressing error propagation caused by indiscriminate fusion.

Stability (identity-friendly): The scale MLP is zero-initialized, ensuring that $\Delta U \approx 0$ at the beginning of training. Meanwhile, α_s is initialized to a relatively small value, ensuring that γ_s is close to 0 during early training. Consequently, the module begins learning from an approximately identity mapping, thereby reducing unstable perturbations introduced by cross-scale interactions and helping to alleviate tail deterioration.

3.3 Adaptive Channel Enhancement

Ordering. ACE is applied within each branch encoder before MR-CSCM performs cross-scale fusion. Cross-scale interaction primarily enables controllable information sharing along the scale dimension S, whereas in multivariate forecasting, the effectiveness of different variables/channels varies across samples and time periods. To address this, we introduce channel-adaptive enhancement modules at each scale branch, which lightly recalibrate the representation of the channel dimension C without altering the tensor shape, thereby emphasizing key channels and mitigating interference from redundant channels.

3.3.1 Gated Channel Enhancement

To differentiate it from subsequent modules, we use subscripts to represent the representations at different stages. For the s-th scale branch, ACE is applied within the branch encoder to recalibrate channel importance on per-scale patch tokens. Let

$$\mathbf{Z}_{\mathrm{enc}}^{(s)} \in \mathbb{R}^{B \times C \times N_s \times d} \tag{13}$$

denote the input to ACE, and let

$$\mathbf{Z}_{\mathrm{ch}}^{(s)} \in \mathbb{R}^{B \times C \times N_s \times d} \tag{14}$$

denote the channel-enhanced output.

ACE performs an incremental update along the channel dimension C. For each (b, p, k) (batch index, patch index, and latent dimension index), we define the corresponding channel vector

$$\mathbf{r}_{b,p,k,\mathrm{enc}}^{(s)} = \mathbf{Z}_{\mathrm{enc}}^{(s)}[b, :, p, k] \in \mathbb{R}^{C}. \tag{15}$$

We apply a two-layer MLP $g_{ch}(\cdot)$ along the channel dimension to generate the correction term:

$$\Delta\mathbf{r}_{b,p,k}^{(s)} = g_{ch}\left(\mathbf{r}_{b,p,k,\mathrm{enc}}^{(s)}\right) \in \mathbb{R}^{C}. \tag{16}$$

Notably, $g_{ch}(\cdot)$ is applied independently at each (b, p, k) position, without any global aggregation over temporal or patch dimensions. A gated residual form is employed for the update:

$$\mathbf{r}_{b,p,k,\mathrm{ch}}^{(s)} = \mathbf{r}_{b,p,k,\mathrm{enc}}^{(s)} + \beta \cdot \Delta\mathbf{r}_{b,p,k}^{(s)}, \quad \beta = \sigma(\beta_{\mathrm{raw}}) \in (0, 1). \tag{17}$$

Finally, reshaping $\mathbf{r}_{b,p,k,\mathrm{ch}}^{(s)}$ back to tensor form yields

$$\mathbf{Z}_{\mathrm{ch}}^{(s)}[b, :, p, k] = \mathbf{r}_{b,p,k,\mathrm{ch}}^{(s)}. \tag{18}$$

3.3.2 Stability Design

To prevent overly strong perturbations in early training, we adopt a near-identity initialization by setting the linear layer parameters of $g_{ch}(\cdot)$ to zero, such that at initialization

$$\Delta\mathbf{r} \approx 0. \tag{19}$$

The gating coefficient β further limits the injected increment, enabling the module to start from

$$\mathbf{Z}_{\mathrm{ch}}^{(s)} \approx \mathbf{Z}_{\mathrm{enc}}^{(s)}, \tag{20}$$

and gradually learn to adaptively adjust channel contributions during training.

3.4 Tail End Error Suppression and Multi Step Refinement

Although MR-CSCM and ACE enable controllable information exchange across the scale and channel dimensions, errors in multi-step prediction may still accumulate along the time dimension, particularly toward the end of the prediction window. To improve tail reliability while keeping the refinement lightweight and stable, we introduce a Temporal Refiner in the time domain. The refiner is applied on the prediction window and is initialized as a near-identity mapping with gated residual updates, such that it introduces only mild perturbations at early training stages and can effectively focus its corrections on the more unstable tail region.

Throughout the paper, *tail* refers to the end portion of the prediction window used in our analysis metrics (e.g., the last $\lceil \rho H \rceil$ steps with ρ=0.25 for TAF/TailMean/TailSlope), whereas the Temporal Refiner operates on the prediction window of length H.

3.4.1 Temporal Refinement on the prediction window

We denote the time-domain sequence reconstructed via multi-scale modeling and inverse transformation as

$$\mathbf{Y}_{\text{rec}} \in \mathbb{R}^{B\times C\times L_r}, \tag{21}$$

where B denotes the batch size, C the number of variables (channels), and L_r the reconstructed temporal length. We take the last H steps of $\mathbf{Y}_{\text{rec}}$ as the prediction window to be refined:

$$\mathbf{Y}_{\text{pred}} = \mathbf{Y}_{\text{rec}}[:, :, -H:] \in \mathbb{R}^{B\times C\times H}. \tag{22}$$

This design allows the refiner to model cross-step error couplings within the prediction window, while the near-identity gated update (detailed below) helps avoid unnecessary perturbations.

3.4.2 Multi-step Temporal Refinement

The refiner performs temporal modeling on $\mathbf{Y}_{\text{pred}}$ and generates a multi-step refinement term:

$$\Delta\mathbf{Y} = F(\mathbf{Y}_{\text{pred}}) \in \mathbb{R}^{B\times C\times H}, \tag{23}$$

where $F(\cdot)$ consists of multiple layers of 1D depthwise separable convolutions. This structure first captures local temporal dependencies within each channel, and then uses lightweight channel mixing to integrate information across channels, thereby enabling explicit modeling of error-coupling relationships among different prediction steps.

Unlike step-by-step autoregressive refinement, this refiner jointly models the entire prediction window in a single forward pass.

3.4.3 Gated Residual Update

To prevent the refiner from introducing overly strong perturbations during the early stages of training, we update the prediction using a gated residual formulation:

$$\mathbf{Y}_{\text{out}} = \mathbf{Y}_{\text{pred}} + g \cdot \Delta\mathbf{Y}, \quad g = \sigma(\delta) \in (0, 1), \tag{24}$$

where δ is a learnable scalar parameter and $\sigma(\cdot)$ denotes the Sigmoid function, which adaptively controls the correction strength.

We initialize the refiner to behave close to an identity mapping at the beginning of training, so that the update is mild and stable:

$$\mathbf{Y}_{\text{out}} \approx \mathbf{Y}_{\text{pred}}. \tag{25}$$

4 Experiment

4.1 Experimental setup

We evaluate TailFix on the standard multivariate long-term time-series forecasting task under a direct multi-step prediction setting. To ensure fair comparison, we follow the public experimental protocol of WPMixer, including its data partitioning, normalization strategy, and input configuration. Most baseline results in the main comparison are taken from published reports following the

Table 1: Average performance (Avg) for multivariate long-horizon forecasting across datasets. The look-back window length is set to $L = 512$ for all datasets except Traffic, where $L = 1200$. All baseline results except WPMixer and TimeMixer are taken from WPMixer's published reports. WPMixer and TimeMixer are reproduced by us under the same protocol. The best results are highlighted in bold, and the second-best results are underlined. Tied best results are counted for each model.

Models		**TailFix (Ours)**		**WPMixer**		**TimeMixer**		**PatchTST**		**TSMixer**		**TimesNet**		**Crossformer**		**DLinear**	
Dataset	**Metric**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	Avg	**0.378**	**0.407**	0.393	0.420	0.434	0.447	0.413	0.434	0.412	0.428	0.458	0.450	0.600	0.557	0.423	0.437
ETTh2	Avg	**0.315**	**0.368**	0.320	0.373	0.379	0.414	0.331	0.381	0.355	0.401	0.414	0.427	0.564	0.548	0.431	0.447
ETTm1	Avg	**0.343**	**0.371**	0.346	0.376	0.364	0.395	0.353	0.382	0.347	0.375	0.400	0.406	0.514	0.510	0.357	0.379
ETTm2	Avg	**0.246**	**0.307**	0.250	0.311	0.262	0.321	0.256	0.317	0.267	0.322	0.291	0.333	0.621	0.510	0.267	0.332
Weather	Avg	**0.219**	**0.253**	0.221	0.255	0.227	0.267	0.226	0.264	0.225	0.264	0.259	0.287	0.406	0.442	0.246	0.300
Electricity	Avg	0.160	**0.252**	**0.159**	**0.252**	0.165	0.260	**0.159**	0.253	0.160	0.257	0.192	0.295	0.186	0.283	0.166	0.264
Traffic	Avg	0.410	0.275	0.403	0.277	0.404	0.283	**0.391**	**0.264**	0.408	0.284	0.620	0.336	0.542	0.283	0.434	0.295
1st Count		5	6	1	1	0	0	2	1	0	0	0	0	0	0	0	0

Table 2: Ablation studies of TailFix on ETT datasets.

Models		**TailFix**		**w/o MR-CSCM**		**w/o ACE**		**w/o Temporal Refiner**	
Dataset	**Metric**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	Avg	**0.378**	**0.407**	0.385	0.414	0.382	0.412	0.385	0.413
ETTh2	Avg	**0.315**	**0.368**	0.322	0.374	0.323	0.373	0.324	0.376
ETTm1	Avg	**0.343**	**0.371**	0.347	0.375	0.346	0.375	0.350	0.378
ETTm2	Avg	**0.246**	**0.307**	0.251	0.313	0.250	0.311	0.250	0.313

Table 3: Tail-stability ablation results of TailFix. We report the proposed tail-stability metrics, including TAF, TailMean25%, and TailSlope25%, where lower values indicate better tail stability.

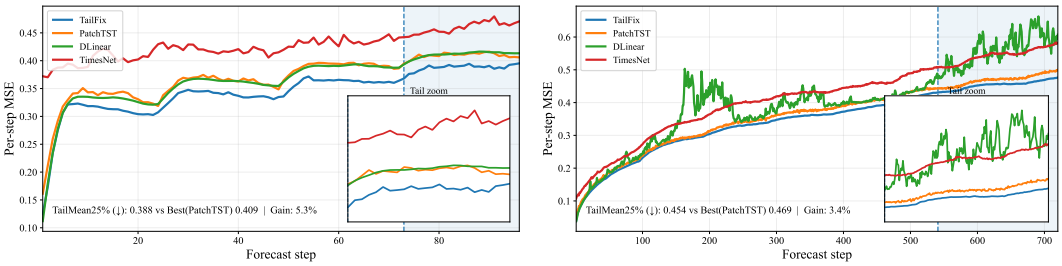
Models		**TailFix**			**w/o Cross-Scale Mixer**			**w/o ACE**			**w/o Temporal Refiner**		
Dataset	**Metric**	**TAF**	**TM25%**	**TS25%**	**TAF**	**TM25%**	**TS25%**	**TAF**	**TM25%**	**TS25%**	**TAF**	**TM25%**	**TS25%**
ETTh1	Avg	**1.165**	**0.425**	**0.000308**	1.175	0.429	0.000330	1.175	0.429	0.000371	1.189	0.437	0.000374
ETTh2	Avg	**1.370**	**0.398**	**0.000431**	1.385	0.404	0.000763	1.374	0.403	0.000570	1.383	0.403	0.000663
ETTm1	Avg	**1.194**	**0.390**	**0.000014**	1.198	0.394	0.000119	1.198	0.394	0.000101	1.195	0.393	0.000024
ETTm2	Avg	**1.448**	**0.320**	0.000471	1.457	0.323	0.000482	1.450	0.329	**0.000444**	1.450	0.323	0.000457

same experimental protocol. We use consistent data splits, preprocessing procedures, and evaluation protocols. Because published results often do not include tail-specific metrics or progressive error curves, we conduct all tail-stability analyses, ablation studies, random-seed experiments, and complexity analyses in this work.

4.2 Experimental results and analysis

Main result. Table 1 reports the average results across four forecasting horizons, $H \in \{96, 192, 336, 720\}$, on seven LTSF datasets using MSE/MAE. TailFix achieves the best MSE on 5 of 7 datasets and the best or tied-best MAE on 6, demonstrating strong overall forecasting performance. On Electricity and Traffic, TailFix remains competitive, achieving the tied-best and second-best MAE, respectively. These results indicate that TailFix improves average forecasting accuracy on most benchmarks, and we further analyze its tail-stability behavior below.

Ablation experiment. We conducted ablation studies on the ETT datasets to assess the contributions of the key components of TailFix. Specifically, we independently removed MR-CSCM, ACE, and the Temporal Refiner, and compared the resulting variants with the full model. As shown in Tables 2 and 3, TailFix achieves the best MSE, MAE, TAF, and TailMean25% across the four ETT datasets. Removing any component increases MSE and MAE, indicating that all three contribute to forecasting accuracy. These results further suggest that controllable cross-scale interaction, channel-adaptive enhancement, and temporal tail correction improve long-term forecasting performance in a complementary manner.



(a) ETTh1, $H = 96$. (b) ETTm2, $H = 720$.

Figure 3: Per-step MSE curves for tail error analysis. We plot the forecasting error at each prediction step on ETTh1 with $H = 96$ and ETTm2 with $H = 720$. The dashed line marks the start of the tail region, and the shaded region denotes the last 25% of the prediction horizon.

Table 4: Summary of tail-stability indicators. TailMean25% and TailSlope25% are computed over the final 25% of prediction steps. Gain is reported relative to the best baseline in TailMean25%.

Dataset	H	TailMean25%↓	TailSlope25%↓	Best (Gain)
ETTh1	96	0.3876	0.000585	PatchTST (0.4094), +5.33%
ETTm2	720	0.4538	0.000226	PatchTST (0.4695), +3.35%

Table 5: Model complexity and inference efficiency on ETTh1 $512 \rightarrow 96$. Latency is measured with batch size 1, and throughput is measured with batch size 16.

Method	Params (M)↓	FLOPs (G)↓	Lat. (ms)↓	Thr. (samples/s)↑
TailFix	6.966	2.015	7.870	2213.987
WPMixer	6.965	2.006	4.748	2320.651
DLinear (rep.)	0.148	0.001	0.211	69943.681
PatchTST (rep.)	0.116	0.008	2.197	7829.229

Table 6: Robustness to random seeds.

Dataset	H	seed 42		seed 43		seed 44		seed 45		seed 46		mean ± std	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.344	0.381	0.345	0.386	0.349	0.384	0.347	0.381	0.348	0.384	0.347 ± 0.002	0.383 ± 0.002
	192	0.381	0.405	0.382	0.410	0.385	0.411	0.385	0.406	0.384	0.408	0.383 ± 0.002	0.408 ± 0.003
	336	0.381	0.410	0.395	0.420	0.388	0.413	0.385	0.412	0.394	0.421	0.389 ± 0.006	0.415 ± 0.005
	720	0.408	0.435	0.410	0.437	0.415	0.448	0.412	0.440	0.426	0.443	0.414 ± 0.007	0.441 ± 0.005
Weather	96	0.146	0.194	0.145	0.195	0.150	0.199	0.148	0.198	0.151	0.200	0.148 ± 0.003	0.197 ± 0.003
	192	0.184	0.227	0.186	0.229	0.189	0.230	0.182	0.225	0.183	0.226	0.185 ± 0.003	0.227 ± 0.002
	336	0.236	0.270	0.237	0.271	0.238	0.271	0.235	0.269	0.237	0.270	0.237 ± 0.001	0.270 ± 0.001
	720	0.310	0.324	0.309	0.323	0.318	0.327	0.312	0.326	0.308	0.321	0.311 ± 0.004	0.324 ± 0.002

Tail error analysis. To validate the *tail-deterioration* phenomenon discussed above and assess whether TailFix suppresses error propagation toward distant prediction steps, we plot per-step MSE curves on ETTh1 ($H = 96$) and ETTm2 ($H = 720$). At each prediction step t, we compute the step-wise MSE by averaging across the sample and variable dimensions. We define the final 25% of the prediction window as the tail region, because this ratio captures the error-prone terminal segment while retaining sufficient prediction steps for stable estimation. A sensitivity analysis using different tail ratios is provided in the Appendix. In Figure 3, the dashed line denotes the beginning of the tail region, and the shaded area indicates the corresponding tail interval.

The results show that the per-step MSE of most baselines gradually increases as the prediction step advances and rises further after entering the tail region, indicating a clear pattern of *tail deterioration*. In contrast, TailFix maintains lower and smoother errors in the tail region under both settings. This effect is particularly evident in the more challenging ETTm2 setting with $H = 720$, where TailFix exhibits a slower error-growth trend. To further quantify this phenomenon, we use $\mathrm{TailMean25\%}$ and $\mathrm{TailSlope25\%}$, which measure the average error level and error-growth rate within the tail region, respectively. As reported in Table 4, TailFix achieves lower $\mathrm{TailMean25\%}$ and $\mathrm{TailSlope25\%}$ on both ETTh1 with $H = 96$ and ETTm2 with $H = 720$, indicating that it not only reduces the tail-error level but also effectively slows tail-error growth, thereby improving the stability of long-horizon forecasting.

Complexity and compute environment. We report the parameter count, FLOPs, and inference efficiency on ETTh1 under the $512 \rightarrow 96$ setting, as shown in Table 5. All efficiency measurements are conducted on an NVIDIA GeForce RTX 5060 GPU with 8 GB of memory under the same software and hardware environment; latency and throughput are measured with batch sizes of 1 and 16, respectively. TailFix introduces negligible additional model size and theoretical computational overhead, indicating that its performance gains mainly arise from more effective cross-scale modeling, channel enhancement, and temporal tail refinement.

Random-Seed robustness. To examine whether TailFix is sensitive to random initialization, we repeated its execution with five different random seeds on representative datasets. As shown in Table 6, the standard deviations of MSE and MAE remain relatively small across different prediction lengths on ETTh1 and Weather. These results indicate that TailFix exhibits stable performance across random seeds, suggesting that the improvements reported in this paper are not attributable to random initialization effects.

Boundary case analysis. On Electricity with $H = 96$, where tail degradation is relatively weak, TailFix does not achieve the lowest tail error. Its $\mathrm{TailMean}25\%$ is 0.1507, compared with 0.1506 for PatchTST and 0.1507 for DLinear. DLinear also achieves lower $\mathrm{TailSlope}25\%$ and TAF, suggesting that TailFix is more effective when tail degradation is pronounced.

Cross-benchmark evaluation. TailFix achieves the lowest average MSE and MAE on four PEMS datasets (Table 7). It ranks second on three selected fev-bench tasks and achieves the lowest MASE on three GIFT-Eval tasks, reducing average MASE by 10.6% compared with TIDE (Table 8).

Table 7: Average MSE/MAE on four PEMS datasets over $H \in \{12, 24, 48, 96\}$.

Models		TailFix (Ours)		Twins		iTransformer		PatchTST	
Dataset	**Metric**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
PEMS03	Avg	**0.088**	**0.190**	0.109	0.219	0.116	0.226	0.180	0.304
PEMS04	Avg	**0.094**	**0.198**	0.111	0.219	0.121	0.232	0.195	0.307
PEMS07	Avg	**0.068**	**0.167**	0.094	0.196	0.100	0.204	0.211	0.303
PEMS08	Avg	**0.122**	**0.196**	0.133	0.222	0.151	0.234	0.280	0.321
1st Count		4	4	0	0	0	0	0	0

Table 8: MASE on selected fev-bench and GIFT-Eval tasks. Lower values are better.

(a) fev-bench

Model	ETT-1H	Jena	BizITObs
Seasonal Naive	1.323	0.730	1.067
PatchTST	1.199	0.519	**0.452**
TiRex-2	**1.102**	**0.462**	0.774
TailFix	1.136	0.518	0.741

(b) GIFT-Eval

Model	Elec.	LOOP	Bitbrains	Avg.
TIDE	1.190	1.070	1.590	1.283
DLinear	1.240	1.170	3.470	1.960
DeepAR	2.280	1.960	7.330	3.857
Crossformer	35.200	1.090	3.100	13.130
TailFix	**1.037**	**1.063**	**1.341**	**1.147**

5 Conclusion

We examine tail deterioration in long-horizon forecasting, where prediction errors gradually amplify near the end of the prediction window. We attribute this instability to insufficient control over cross-scale feature mixing, which may propagate noise and early-stage biases to distant steps. To address this issue, we propose TailFix, a controllable cross-scale forecasting framework that regulates cross-scale interactions and improves tail reliability through stable residual refinement. We also introduce TAF as a model-independent metric for tail-error amplification. Experiments on multiple LTSF benchmarks show that TailFix improves forecasting accuracy and tail stability with comparable parameter counts and FLOPs.

Acknowledgment

This work was supported in part by the following: the National Natural Science Foundation of China under Grant Nos. U24A20328, U24A20219, 62272281, the Natural Science Foundation of Shandong Province under Grant No. ZR2026MS1059, the Youth Innovation Technology Project of Higher School in Shandong Province under Grant No. 2023KJ212, and the Yantai Natural Science Foundation under Grant No. 2024JCYJ034.

Competing Interests. The authors declare no competing interests.

References

[1] 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.

[2] Xiaohong Chen, Canran Xiao, and Yongmei Liu. Confusion-resistant federated learning via diffusion-based data harmonization on non-iid data. *Advances in Neural Information Processing Systems*, 37:137495–137520, 2024.

[3] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[4] Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. *arXiv preprint arXiv:2306.04406*, 2023.

[5] Jiaxi Hu, Yuehong Hu, Wei Chen, Ming Jin, Shirui Pan, Qingsong Wen, and Yuxuan Liang. Attractor memory for long-term time series forecasting: A chaos perspective. *Advances in Neural Information Processing Systems*, 37:20786–20818, 2024.

[6] 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 *ICML*, 2025.

[7] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, pages 11906–11917. PMLR, 2022.

[8] 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.

[9] Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024.

[10] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[11] 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, pages 19581–19588, 2025.

[12] 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.

[13] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[14] 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.

[15] Jingzhe Shi, Qinwei Ma, Huan Ma, and Lei Li. Scaling law for time series forecasting. *Advances in Neural Information Processing Systems*, 37:83314–83344, 2024.

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

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Caixia Wang, Fan Zhang, Xiaofeng Zhang, and Hua Wang. Dtfnet: A dual-modal time-frequency fusion network for non-stationary time series modeling. *Knowledge-Based Systems*, page 116022, 2026.

[19] Hua Wang, Xianhao Jiao, and Fan Zhang. PESD-TSF: A period-aware and explicit structured decomposition framework for long-term time series forecasting. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 127230–127252. PMLR, 06–11 Jul 2026. URL `https://proceedings.mlr.press/v306/wang26cs.html`.

[20] Hua Wang, Jinghao Lu, and Fan Zhang. Idealtsf: Can non-ideal data contribute to enhancing the performance of time series forecasting models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26224–26232, 2026.

[21] 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.

[22] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

[23] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.

[24] 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.

[25] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023. URL `https://arxiv.org/abs/2210.02186`.

[26] Canran Xiao and Yongmei Liu. A multifrequency data fusion deep learning model for carbon price prediction. *Journal of Forecasting*, 44(2):436–458, 2025.

[27] Canran Xiao, Jiabao Dou, Zhiming Lin, Zong Ke, and Liwei Hou. From points to coalitions: Hierarchical contrastive shapley values for prioritizing data samples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15995–16003, 2026.

[28] Canran Xiao, Zong Ke, and Puning Zhao. Focus where it matters: Llm-guided attention priors for few-shot segmentation. *Expert Systems with Applications*, page 134689, 2026.

[29] Zhen Yao, Bo Lang, Zhihao Zheng, Haoyan Xu, Zhengtao Yao, Zhikang Dong, and Mooi Choo Chuah. From semantic to reasoing: A survey of language-guided segmentation in the multimodal era. 2026.

[30] 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, pages 11121–11128, 2023.

[31] 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.

[32] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

[33] 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, pages 11106–11115, 2021.

[34] Heng Zhou, Jing Tang, Jusheng Zhang, Yanshu Li, Canran Xiao, Liwei Hou, Zong Ke, and Jiawei Yao. Comem: Compositional concept-graph memory for vision–language adaptation. In *International Conference on Learning Representations*, volume 2026, pages 20009–20032, 2026.

[35] 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, 2022.

[36] 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*, pages 27268–27286. PMLR, 2022.

A Limitations and boundary analysis of TailFix

A.1 A representative boundary case

Although TailFix improves tail stability in most long-term forecasting tasks, its full module combination may not always yield optimal results when tail degradation is weak. We use the $H = 96$ result on the Electricity dataset as a representative boundary case. As shown in Figure 4 and Table 9, the tail-region errors of TailFix, PatchTST, and DLinear are very close. TailFix obtains a $\mathrm{TailMean}25\%$ of 0.1507, whereas PatchTST achieves 0.1506, corresponding to a difference of only 0.0001. This indicates that the baseline already exhibits strong tail stability under this setting, leaving limited room for further improvement by TailFix.

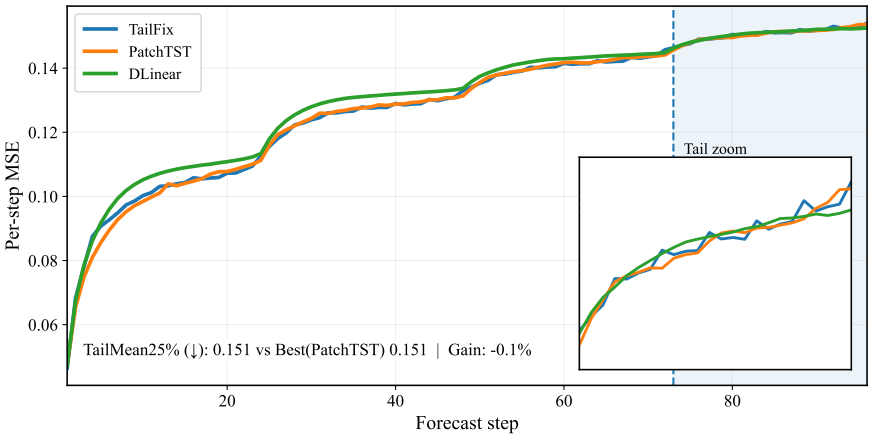


Figure 4: Boundary case on Electricity with $H = 96$. The per-step MSE curves of TailFix and strong baselines are highly close in the tail region, indicating that the improvement margin becomes limited when tail deterioration is weak.

Table 9: Tail-stability metrics for the Electricity boundary case with $H = 96$. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Model	TailMean25%	TailSlope25%	TAF
TailFix	<u>0.1507</u>	<u>0.000259</u>	<u>1.241</u>
PatchTST	**0.1506**	0.000269	1.242
DLinear	<u>0.1507</u>	**0.000227**	**1.213**

To analyze this phenomenon, we further conduct module-ablation experiments, with the results reported in Table 10. After MR-CSCM is removed, the model achieves lower MSE, MAE, $\mathrm{TailMean25\%}$, $\mathrm{TailSlope25\%}$, and TAF than the complete TailFix. This suggests that, in the short-horizon Electricity setting with $H = 96$, cross-scale contextual interaction does not provide additional benefits and may instead introduce slight interference.

Table 10: Module ablation results for the Electricity boundary case with $H = 96$. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Model Variant	MSE	MAE	TailMean25%	TailSlope25%	TAF
TailFix	0.132	0.226	0.1507	0.000259	1.241
w/o MR-CSCM	<u>0.130</u>	<u>0.225</u>	<u>0.1482</u>	<u>0.000232</u>	<u>1.240</u>
w/o ACE	0.141	0.229	0.1674	0.000526	1.310
w/o Temporal Refiner	**0.128**	**0.222**	**0.1470**	**0.000219**	**1.237**

This result may be related to the high-dimensional variable structure and strong periodicity of the Electricity dataset. The Electricity dataset contains a large number of variables, and the correlations among different users or load channels are not always uniform or stable. MR-CSCM injects corrective information across different resolutions through scale-level contextual summarization. When cross-scale patterns are consistent, this mechanism helps alleviate error diffusion over long horizons. However, in the Electricity setting with $H = 96$, the prediction length is relatively short, and the main periodic patterns can already be captured by the backbone. As a result, tail errors may not yet exhibit substantial diffusion. In this case, additional cross-scale corrections may propagate slight noise or unstable local channel information to other scales, thereby perturbing otherwise stable predictions.

In addition, short-horizon forecasting on the Electricity dataset relies more on local periodic continuation and does not necessarily require strong cross-scale semantic interaction. MR-CSCM is designed to mitigate error accumulation over long prediction intervals through multi-scale context fusion. When the task itself exhibits no substantial long-range error propagation, the benefit of cross-scale injection may diminish. In this case, preserving the relative independence of each scale branch may be more beneficial for maintaining stable predictions. This also explains why removing MR-CSCM leads to better overall and tail-specific error metrics in this boundary scenario. In contrast, removing ACE substantially degrades MSE, MAE, and $\mathrm{TailMean25\%}$, indicating that channel enhancement remains useful for high-dimensional multivariate datasets such as Electricity. This suggests that the issue is not that all newly added modules impair high-dimensional forecasting, but rather that the need for cross-scale interaction decreases in short-horizon scenarios with weak tail degradation.

Therefore, this case further clarifies the applicability boundary of TailFix: MR-CSCM is more suitable for long-horizon scenarios in which tail errors substantially diffuse as the prediction length increases, whereas in the Electricity setting with strong periodicity, a short horizon, and a stable baseline, namely $H = 96$, excessive cross-scale correction may have a slight negative impact.

A.2 General limitations

Beyond the aforementioned boundary cases, TailFix still has several general limitations. First, the benefits of TailFix depend on whether the task exhibits substantial tail-error diffusion. When the prediction length is short, the data patterns are stable, or the baseline already produces smooth and

reliable tail predictions, the additional correction space available to TailFix may decrease. In such cases, the full module combination may achieve performance close to that of a strong baseline and may not be optimal across all metrics. This is consistent with the observations from the Electricity setting with $H = 96$, suggesting that TailFix is more suitable for long-horizon scenarios with substantial tail degradation.

Second, the cross-scale interaction in MR-CSCM is not equally effective across all data structures. This module injects corrective information across different resolutions through scale-level contextual summarization, making it suitable for mitigating the gradual diffusion of cross-scale errors in long prediction windows. However, in datasets with many high-dimensional variables and unstable inter-variable correlations, cross-scale correction may propagate local noise or weakly correlated channel information to other scales. In short-horizon scenarios in particular, if the backbone has already captured the dominant periodic patterns, additional cross-scale interactions may introduce slight perturbations. Therefore, the effectiveness of MR-CSCM is closely related to the dataset-specific variable-correlation structure, prediction length, and degree of tail degradation.

Third, this work mainly uses $\mathrm{TailMean}25\%$, $\mathrm{TailSlope}25\%$, and TAF to measure tail stability, with the tail region defined by default as the final 25% of the prediction window. Although we further analyze the results under different tail ratios in the Appendix and observe a relatively stable overall trend, the tail ratio itself remains a manually specified setting. Across different application scenarios, the truly critical tail region may not always correspond to the final 25% of the prediction window. For example, power dispatch may place greater emphasis on peak-load periods, whereas traffic forecasting may focus more on specific intervals after holidays. Therefore, future work could develop more adaptive tail-risk metrics tailored to task-specific requirements.

Finally, although MR-CSCM, ACE, and the Temporal Refiner in TailFix are lightweight modules, they still introduce additional parameters and computational overhead compared with the original backbone. For scenarios with limited resources or strict real-time requirements, simple linear models or lighter backbones may remain more attractive. Therefore, TailFix is better suited as an enhancement framework for long-horizon reliability rather than as a universal alternative that is optimal under all computational budgets.

B Additional tail-stability and mechanism analyses

B.1 Sensitivity to the tail ratio

In the main text, we define the final 25% of the prediction window as the tail region. To verify that this choice does not bias our conclusions, we further examine TailMean-MSE under different tail ratios. Specifically, we set $\rho \in \{0.10, 0.20, 0.25, 0.30, 0.40\}$ and compare the average tail errors of TailFix and WPMixer across multiple ETT settings, where $\rho = 0.25$ corresponds to the Tail25% setting used in the main text. The sensitivity curves and averaged results are shown in Figure 5 and Table 11, respectively.

In Figure 5, the horizontal axis denotes the tail ratio ρ, and the vertical axis denotes the average MSE within the corresponding tail region. The blue curve represents WPMixer, the orange curve represents TailFix, and the vertical dashed line marks the main-text choice of $\rho = 0.25$. As shown in the figure, TailFix maintains lower TailMean-MSE than WPMixer across all examined ρ values and prediction horizons on ETTh1. On ETTm2, the improvement achieved by TailFix is relatively smaller, but the overall trend remains stable, with more evident gains under longer-horizon settings such as $H = 336$.

These results, together with the summary in Table 11, indicate that the tail-stability improvement achieved by TailFix does not depend on a particular choice of tail ratio. Defining the final 25% of the prediction window as the tail region captures the error-prone terminal segment while retaining sufficient prediction steps for stable estimation, making it a reasonable and representative setting.

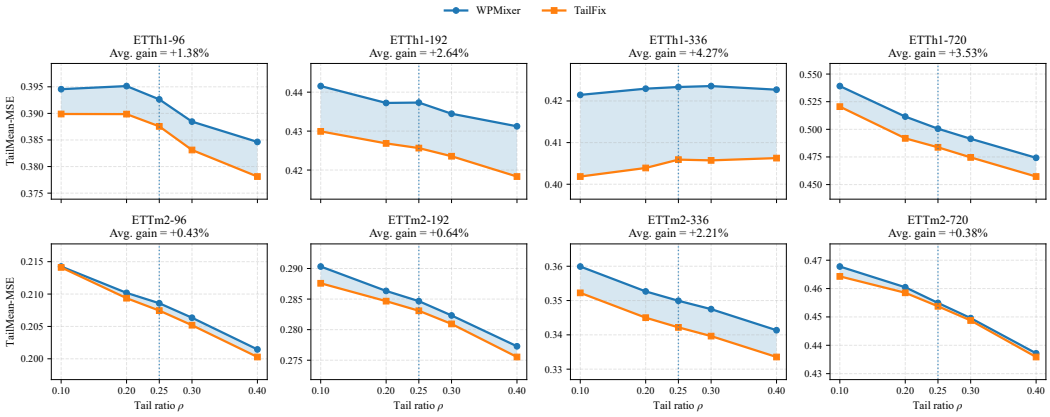


Figure 5: Sensitivity to tail-ratio definitions. We compare TailFix and WPMixer under different tail ratios $\rho \in \{0.10, 0.20, 0.25, 0.30, 0.40\}$ on ETT settings. The vertical dashed line marks the main-text choice $\rho = 0.25$. TailFix generally maintains lower TailMean-MSE across different tail-region definitions.

Table 11: Sensitivity to tail-region definitions.

Dataset	Horizon	TailMean-MSE			TAF-MSE		
		WPMixer	TailFix	Gain(%)	WPMixer	TailFix	Gain(%)
ETTh1	96	0.391	0.386	+1.37	1.168	1.165	+0.25
	192	0.436	0.425	+2.64	1.128	1.158	-2.62
	336	0.423	0.405	+4.27	1.079	1.087	-0.68
	720	0.503	0.486	+3.53	1.265	1.246	+1.55
ETTm2	96	0.208	0.207	+0.42	1.468	1.468	+0.01
	192	0.284	0.282	+0.64	1.455	1.480	-1.73
	336	0.350	0.343	+2.21	1.435	1.426	+0.64
	720	0.454	0.452	+0.39	1.453	1.447	+0.43

B.2 Validation-selected tail boundaries

The tail-stability metrics use the final 25% of the prediction window as the default evaluation region. To assess the influence of the evaluation boundary, we examine task-level tail ratios selected from validation data.

Specifically, we use the baseline model's per-step validation MSE and piecewise linear fitting to select the tail ratio from $\rho \in \{0.10, 0.20, 0.25, 0.30, 0.40\}$. The selected boundary is fixed before testing and shared by TailFix and WPMixer.

Table 12 reports the results on ETTh1 and ETTm2. TailFix achieves lower TailMean-MSE in all eight settings, while TAF improves in five settings. These results indicate that the reduction in absolute tail error is not restricted to the fixed 25% evaluation boundary.

Table 12: Tail-stability evaluation using validation-selected tail boundaries. Lower values are better.

Dataset	H	ρ^*	TailMean-MSE		TAF	
			TailFix	WPMixer	TailFix	WPMixer
ETTh1	96	0.40	**0.378**	0.385	**1.172**	1.182
	192	0.40	**0.418**	0.431	1.171	**1.140**
	336	0.40	**0.406**	0.423	1.115	**1.098**
	720	0.40	**0.457**	0.474	**1.197**	1.221
ETTm2	96	0.40	**0.200**	0.201	**1.543**	1.547
	192	0.30	**0.281**	0.282	1.508	**1.478**
	336	0.40	**0.334**	0.341	**1.503**	1.517
	720	0.40	**0.436**	0.437	**1.503**	1.510

B.3 Tail-Weighted loss versus temporal refiner

A direct way to mitigate tail errors is to increase the loss weight assigned to the tail region of the prediction window during training. To examine whether this simple loss-level adjustment is sufficient to replace the Temporal Refiner in TailFix, we construct a tail-weighted loss as a comparative baseline. Specifically, this baseline keeps the WPMixer architecture unchanged and only assigns larger training weights to the final 25% of the prediction window.

Let H denote the prediction horizon and ℓ_t denote the prediction loss at step t. The tail-weighted training objective is defined as

$$\mathcal{L}_{\mathrm{TW}} = \frac{1}{H}\sum_{t=1}^{H} w_t \ell_t, \qquad w_t = \begin{cases} 1, & t \le 0.75H, \\ \lambda, & t > 0.75H, \end{cases} \tag{26}$$

where λ controls the relative penalty assigned to the tail region. In our experiments, we set $\lambda \in \{2, 5, 10\}$ and compare these variants with the Temporal Refiner under the same datasets and prediction horizons.

Figure 6 reports both overall MSE and Tail@25% MSE. To highlight the relative differences among methods within the same setting, each row is normalized independently. The results show that tail-weighted loss is sensitive to the choice of λ. A larger tail loss weight can sometimes reduce the tail-region error, but this effect is unstable across datasets and prediction horizons, and may also lead to degradation in overall MSE.

Table 13 further summarizes the sensitivity of tail-weighted loss and the relative gain of the Temporal Refiner. In contrast, the Temporal Refiner exhibits a more stable correction effect across multiple settings. Rather than forcing the model to focus on the tail by manually amplifying the loss assigned to tail-region samples, it introduces a structural temporal residual correction over the prediction window and directly models error-coupling relationships among prediction steps. Therefore, these results suggest that tail deterioration cannot be reliably addressed by simple loss reweighting alone.

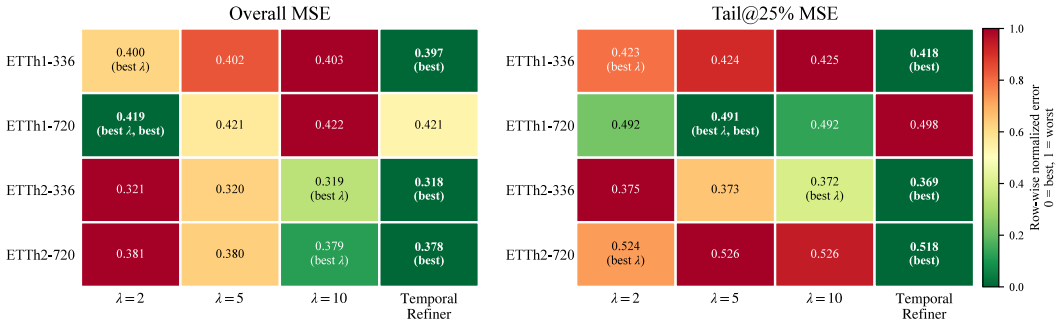


Figure 6: Tail-weighted loss versus Temporal Refiner on ETT datasets. The heatmaps show overall MSE (left) and Tail@25% MSE (right), with each row normalized independently.

Table 13: Sensitivity of tail-weighted loss and comparison with the Temporal Refiner. TWLoss span denotes the performance variation across $\lambda \in \{2, 5, 10\}$ (in $\times 10^{-3}$). Positive TR gain indicates that the Temporal Refiner achieves lower MSE than the best TWLoss setting.

Setting	TWLoss span (Overall)	TR gain (Overall)	TWLoss span (Tail)	TR gain (Tail)
ETTh1-336	2.41	+3.89	1.40	+5.49
ETTh1-720	3.08	-1.66	1.62	-7.21
ETTh2-336	1.55	+0.80	3.80	+2.49
ETTh2-720	2.71	+0.37	2.09	+5.55

B.4 Controlled cross-scale and temporal-refinement interventions

We further examine the effects of cross-scale interaction and temporal-refinement injection through controlled interventions. For cross-scale interaction, we compare learnable fusion and non-selective equal-weight fusion under different interaction spans, while keeping the model architecture, parameter count, training configuration, and random seed unchanged.

Span-0 disables cross-scale interaction, whereas Span-1 and Span-2 restrict interactions to one and two neighboring scales, respectively. Span-All allows interactions among all available scales. We additionally examine three temporal-refinement injection settings, namely Gate-0, Gate-1, and Learned Gate, on ETTh2 and ETTm2 with $H = 720$.

As shown in Table 14, extending cross-scale interaction to all scales does not consistently improve forecasting accuracy. Under non-selective equal-weight fusion, restricted interaction spans achieve lower MSE and Tail25-MSE than Span-All. The temporal-refinement results also show that the effects of different injection settings vary across datasets. These findings indicate that cross-scale interaction and temporal correction benefit from appropriate control rather than unrestricted information exchange.

Table 14: Controlled cross-scale fusion and temporal-refinement injection experiments on ETTh2 and ETTm2 with $H = 720$. Tail Slope is reported in units of 10^{-3}. Lower values are better.

Setting	MSE	Tail25-MSE	Tail/Head	Tail Slope
Learnable cross-scale fusion (ETTh2, $H = 720$)				
Span-0	0.377	0.519	1.974	0.274
Span-1	0.377	0.519	1.974	0.274
Span-2	0.377	0.519	1.974	0.274
Span-All	0.380	0.520	1.982	0.254
Non-selective equal-weight fusion (ETTh2, $H = 720$)				
Span-0	0.378	0.519	1.976	0.276
Span-1	0.374	0.514	1.991	0.258
Span-2	0.374	0.514	1.990	0.258
Span-All	0.376	0.516	1.981	0.264
Temporal-refinement injection (ETTh2, $H = 720$)				
Gate-0	0.382	0.522	1.978	0.268
Gate-1	0.382	0.528	1.972	0.293
Learned Gate	0.381	0.526	1.985	0.292
Temporal-refinement injection (ETTm2, $H = 720$)				
Gate-0	0.354	0.461	2.151	0.104
Gate-1	0.350	0.452	2.067	0.108
Learned Gate	0.347	0.448	2.070	0.103

C Generalization to a non-wavelet backbone

To examine whether TailFix relies solely on the wavelet-decomposition structure, we further integrate it into PatchTST, a non-wavelet backbone model, and conduct supplementary experiments on the ETTh1, ETTh2, ETTm1, and ETTm2 datasets. Specifically, we use the original PatchTST as the

base model and incorporate the complete TailFix module, the MR-CSCM module, and the Temporal Refiner to examine their generalizability in non-wavelet temporal forecasting frameworks. All experiments use the same prediction settings and report MSE, MAE, and Tail25% MSE. Lower values indicate better performance. The results are summarized in Table 15.

As shown in Table 15, TailFix-related modules can still provide stable performance improvements on non-wavelet backbones. Compared with the original PatchTST, incorporating TailFix modules reduces MSE, MAE, or Tail25% MSE across most datasets and prediction lengths, indicating that the improvements introduced by TailFix are not limited to wavelet-decomposition frameworks but can be transferred to other backbones as a general enhancement module for long-term forecasting.

Based on the average results in Table 15, TailFix modules generally achieve the best or second-best performance on ETTh1, ETTh2, and ETTm1, with particularly stable performance in terms of Tail25% MSE. This indicates that their ability to suppress errors in the tail region of the prediction window remains effective in non-wavelet models. On the ETTm2 dataset, MR-CSCM and the Temporal Refiner also achieve competitive results, indicating that cross-scale contextual interaction and temporal tail correction can help alleviate error accumulation over long prediction intervals.

Further analysis of the performance of different modules shows that MR-CSCM effectively reduces tail errors under certain settings, indicating that controlled contextual interaction helps mitigate error diffusion caused by unconstrained information mixing. The Temporal Refiner performs well in several long-horizon forecasting settings, suggesting that lightweight temporal correction applied directly to the prediction window can improve tail stability. The complete TailFix module maintains strong overall performance across multiple metrics, indicating a degree of complementarity among its components.

Overall, these experiments demonstrate the transferability of TailFix modules to the non-wavelet PatchTST backbone. Even without relying on the wavelet-decomposition structure, the controlled-interaction and tail-correction mechanisms in TailFix can still improve the average forecasting accuracy and tail stability of PatchTST. This further supports the central argument of this work: tail degradation in long-term forecasting can be alleviated not only by specific backbone architectures but also through general mechanisms for error-diffusion suppression and tail refinement.

Table 15: Generalization results on the non-wavelet PatchTST backbone. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Horizon	PatchTST			+ TailFix Modules			+ MR-CSCM			+ Temporal Refiner		
		MSE	MAE	Tail25%MSE	MSE	MAE	Tail25%MSE	MSE	MAE	Tail25%MSE	MSE	MAE	Tail25%MSE
ETTh1	96	0.370	0.400	0.422	0.371	**0.396**	**0.417**	0.374	0.398	0.419	**0.368**	0.399	**0.417**
	192	0.413	0.429	0.465	**0.410**	**0.418**	**0.460**	0.413	0.420	0.465	0.411	0.420	0.463
	336	**0.422**	0.440	0.471	**0.422**	**0.427**	**0.466**	0.434	0.438	0.468	0.431	0.436	0.470
	720	0.447	0.468	0.545	**0.440**	**0.456**	**0.542**	0.450	0.465	0.545	0.445	0.465	0.544
	avg	0.413	0.434	0.475	**0.410**	**0.424**	**0.471**	0.417	0.430	0.474	0.413	0.430	0.473
ETTh2	96	0.274	0.336	0.344	0.272	0.333	0.342	0.274	0.336	0.344	**0.270**	**0.331**	**0.340**
	192	0.337	0.378	0.424	**0.334**	**0.376**	**0.422**	0.339	0.380	0.425	0.338	0.379	0.425
	336	0.331	**0.380**	0.399	0.331	0.381	0.399	**0.330**	**0.380**	**0.398**	0.331	**0.380**	**0.398**
	720	0.378	0.421	0.528	0.375	**0.417**	0.525	0.377	0.421	0.527	**0.374**	0.418	**0.524**
	avg	0.330	0.379	0.424	**0.328**	**0.377**	**0.422**	0.330	0.379	0.424	**0.328**	**0.377**	**0.422**
ETTm1	96	0.289	0.343	0.322	**0.285**	0.342	**0.320**	0.290	**0.340**	0.324	0.292	0.345	0.328
	192	0.330	0.368	0.376	**0.326**	**0.364**	**0.373**	0.331	0.370	0.384	0.330	0.369	0.375
	336	0.365	0.391	0.425	**0.360**	0.387	**0.422**	0.364	**0.381**	0.427	0.363	0.388	0.423
	720	0.418	0.424	0.482	0.416	0.424	0.480	0.417	0.422	0.482	**0.414**	**0.420**	**0.479**
	avg	0.350	0.381	0.401	**0.346**	0.379	**0.398**	0.350	**0.378**	0.404	0.349	0.380	0.401
ETTm2	96	0.165	0.255	0.214	0.166	0.254	0.215	**0.163**	**0.251**	0.214	0.164	0.252	**0.213**
	192	0.220	**0.292**	0.290	**0.218**	0.293	0.290	0.220	**0.292**	**0.289**	0.221	0.293	0.291
	336	0.278	0.329	0.358	0.276	**0.328**	**0.356**	0.277	0.329	0.357	**0.275**	**0.328**	**0.356**
	720	0.367	0.385	0.484	**0.364**	**0.382**	**0.481**	**0.364**	0.383	**0.481**	0.365	**0.382**	0.482
	avg	0.257	0.315	0.336	**0.256**	0.314	**0.335**	**0.256**	**0.313**	**0.335**	**0.256**	**0.313**	**0.335**
1st Count	–	1	2	0	**13**	**11**	**14**	4	7	4	7	7	9

D Reproducibility and experimental details

D.1 Datasets and experimental configurations

Table 16: Dataset statistics and horizon-specific wavelet configurations. C denotes the number of variables, Split the train/validation/test ratio, and Wavelet/J the wavelet basis and decomposition level.

Dataset	C	Frequency	Length	Split	Wavelet / J			
					$H = 96$	$H = 192$	$H = 336$	$H = 720$
ETTh1	7	1 hour	17,420	6:2:2	db2/2	db2/2	db2/2	db4/1
ETTh2	7	1 hour	17,420	6:2:2	db2/2	db2/3	db2/5	db2/5
ETTm1	7	15 min	69,680	6:2:2	db2/1	db3/1	db5/1	db5/4
ETTm2	7	15 min	69,680	6:2:2	bior3.1/1	db2/1	db2/1	db2/1
Weather	21	10 min	52,696	7:1:2	db4/3	db3/1	db3/2	db2/1
Electricity	321	1 hour	26,304	7:1:2	sym3/2	coif5/3	sym4/1	db2/2
Traffic	862	1 hour	17,544	7:1:2	db3/1	db3/1	bior3.1/1	db3/1

Table 17: TailFix hyperparameters by dataset and horizon. t_f and d_f denote token- and embedding-mixing expansion factors, E/P denotes epochs/patience, WD denotes weight decay, and CSE/CSD denote cross-scale expansion/dropout.

Dataset	H	E/P	L	d	t_f	d_f	Patch	Batch	Drop.	Emb.	WD	CSE	CSD
ETTh1	96	30/12	512	256	5	8	16	256	0.40	0.10	default	2	0.00
	192	80/25	512	256	5	8	16	256	0.30	0.05	0.0005	2	0.00
	336	80/25	512	256	5	8	16	256	0.25	0.20	0.0003	2	0.05
	720	100/20	512	256	5	4	16	256	0.25	0.30	0.0003	2	0.05
ETTh2	96	60/20	512	256	5	8	16	256	0.35	0.05	0.0001	3	0.10
	192	30/25	512	512	5	8	16	128	0.10	0.10	0.0003	2	0.00
	336	50/12	512	512	8	6	32	512	0.30	0.30	0.0003	4	0.10
	720	30/12	512	128	5	5	16	256	0.40	0.00	0.0005	2	0.10
ETTm1	96	60/20	512	256	5	3	48	128	0.30	0.05	0.00001	3	0.10
	192	80/12	512	128	3	7	48	256	0.30	0.05	0.0001	2	0.05
	336	80/12	512	256	7	7	48	256	0.40	0.00	0.0001	2	0.05
	720	80/12	512	128	3	8	48	256	0.40	0.05	0.0001	2	0.05
ETTm2	96	80/12	512	256	3	8	48	256	0.40	0.00	default	2	0.00
	192	80/12	512	256	3	5	48	256	0.40	0.00	0.0	2	0.05
	336	80/12	512	256	3	5	48	256	0.40	0.00	0.0001	2	0.05
	720	50/15	512	128	3	8	48	256	0.45	0.05	0.0001	4	0.08
Weather	96	120/20	512	1024	3	7	16	64	0.20	0.05	0.0001	3	0.05
	192	60/12	512	128	3	7	16	64	0.50	0.00	0.0003	2	0.05
	336	60/12	512	128	7	7	16	32	0.40	0.00	0.0003	2	0.05
	720	60/12	512	128	7	5	16	128	0.40	0.20	0.0003	2	0.05
Electricity	96	100/12	512	32	3	5	16	32	0.10	0.05	0.0001	3	0.05
	192	100/12	512	32	7	5	16	32	0.10	0.05	0.0001	3	0.05
	336	100/12	512	32	5	7	16	32	0.20	0.05	0.0001	3	0.05
	720	100/12	512	32	7	8	16	32	0.10	0.05	0.0001	3	0.05
Traffic	96	60/12	1200	16	3	5	16	16	0.05	0.05	0.0001	3	0.05
	192	60/12	1200	32	3	5	16	16	0.05	0.00	0.0001	3	0.05
	336	60/12	1200	32	7	7	16	16	0.00	0.00	0.0001	3	0.05
	720	60/12	1200	32	7	3	16	16	0.05	0.20	0.0001	3	0.05

D.2 Random-seed robustness across datasets

We evaluate the robustness of TailFix to random initialization across five datasets. Table 6 reports the results on ETTh1 and Weather, while Table 18 presents the results on ETTh2, ETTm1, and ETTm2.

Each dataset is evaluated at four prediction horizons using five random seeds, ranging from 42 to 46. All other experimental configurations remain unchanged.

The consistently small standard deviations of MSE and MAE indicate that TailFix maintains stable forecasting performance across different random initializations.

Table 18: Random-seed robustness on ETTh2, ETTm1, and ETTm2 across five random seeds (42–46). Results are reported using MSE and MAE, together with their mean and standard deviation.

Dataset	H	**Seed 42**		**Seed 43**		**Seed 44**		**Seed 45**		**Seed 46**		**Mean ± Std**	
		MSE	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh2	96	0.258	0.326	0.257	0.323	0.260	0.324	0.258	0.323	0.259	0.324	0.258 ± 0.001	0.324 ± 0.001
	192	0.307	0.353	0.308	0.358	0.313	0.359	0.309	0.353	0.311	0.360	0.310 ± 0.002	0.357 ± 0.003
	336	0.319	0.375	0.317	0.374	0.322	0.375	0.319	0.374	0.321	0.376	0.320 ± 0.002	0.375 ± 0.001
	720	0.378	0.418	0.379	0.420	0.383	0.422	0.390	0.427	0.387	0.424	0.383 ± 0.005	0.422 ± 0.003
ETTm1	96	0.278	0.333	0.286	0.338	0.280	0.334	0.279	0.335	0.282	0.335	0.281 ± 0.003	0.335 ± 0.002
	192	0.322	0.359	0.321	0.358	0.325	0.362	0.325	0.360	0.326	0.362	0.324 ± 0.002	0.360 ± 0.002
	336	0.360	0.384	0.358	0.383	0.361	0.385	0.355	0.382	0.367	0.384	0.360 ± 0.004	0.384 ± 0.001
	720	0.414	0.411	0.414	0.410	0.412	0.411	0.414	0.411	0.416	0.412	0.414 ± 0.001	0.411 ± 0.001
ETTm2	96	0.158	0.247	0.160	0.248	0.162	0.250	0.161	0.250	0.164	0.253	0.161 ± 0.002	0.250 ± 0.002
	192	0.214	0.286	0.217	0.287	0.215	0.287	0.216	0.288	0.217	0.289	0.216 ± 0.001	0.287 ± 0.001
	336	0.266	0.322	0.272	0.324	0.266	0.321	0.268	0.323	0.269	0.325	0.268 ± 0.002	0.323 ± 0.001
	720	0.348	0.376	0.350	0.378	0.347	0.378	0.347	0.377	0.363	0.383	0.351 ± 0.006	0.378 ± 0.002

D.3 Hyperparameter sensitivity analysis

To investigate whether TailFix relies on a narrowly tuned configuration, we conduct a sensitivity analysis of cross-scale expansion, cross-scale dropout, and Temporal Refiner depth across representative forecasting tasks. These experiments examine how different configurations affect both overall forecasting accuracy and tail stability.

As shown in Table 19, TailFix maintains relatively stable forecasting performance across the examined configurations. While cross-scale settings exhibit some task-dependent variations, increasing refinement depth does not consistently improve performance. These findings suggest that the effectiveness of TailFix does not depend on a single carefully selected hyperparameter combination. Instead, the proposed mechanisms maintain their effectiveness across multiple reasonable settings, providing further evidence of the robustness of TailFix.

Table 19: Hyperparameter sensitivity analysis of TailFix on ETTh1 ($H = 336$), ETTm2 ($H = 720$), and Weather ($H = 192$). For each group, only one hyperparameter is varied while all other settings are kept fixed. Lower values indicate better accuracy and tail stability.

Hyperparameter	Value	ETTh1, $H = 336$			ETTm2, $H = 720$			Weather, $H = 192$		
		MSE ↓	TAF ↓	TailMean25% ↓	MSE ↓	TAF ↓	TailMean25% ↓	MSE ↓	TAF ↓	TailMean25% ↓
cross_scale_expand	1	0.378	1.084	0.402	0.349	1.441	0.453	0.184	1.478	0.243
	2	0.381	1.088	0.405	0.348	1.440	0.452	0.184	1.478	0.243
	4	0.383	1.092	0.409	0.348	1.447	0.453	0.185	1.480	0.244
	8	0.382	1.092	0.408	0.348	1.446	0.454	0.184	1.476	0.242
cross_scale_dropout	0.00	0.380	1.085	0.404	0.348	1.434	0.451	0.184	1.478	0.245
	0.05	0.381	1.088	0.405	0.348	1.447	0.453	0.185	1.478	0.244
	0.10	0.381	1.088	0.405	0.348	1.447	0.453	0.184	1.478	0.243
	0.20	0.381	1.088	0.405	0.348	1.447	0.453	0.185	1.477	0.243
refiner_layers	1	0.381	1.088	0.405	0.348	1.447	0.453	0.184	1.478	0.243
	2	0.381	1.088	0.405	0.348	1.447	0.453	0.184	1.478	0.243
	3	0.381	1.088	0.405	0.348	1.447	0.453	0.184	1.478	0.243

E Detailed experimental results

E.1 Detailed main forecasting results

Table 20: Full results on multiple datasets and horizons. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Models		TailFix (Ours)		WPMixer 2025		TimeMixer 2024		PatchTST 2023		TSMixer 2023		TimesNet 2023		Crossformer 2023		DLinear 2023	
Dataset	*H*	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	96	**0.344**	**0.381**	0.349	0.384	0.379	0.407	0.370	0.400	0.361	0.392	0.384	0.402	0.418	0.438	0.375	0.399
	192	**0.381**	**0.405**	0.399	0.419	0.446	0.449	0.413	0.429	0.404	0.418	0.436	0.429	0.539	0.517	0.405	0.416
	336	**0.380**	**0.410**	0.400	0.435	0.421	0.443	0.422	0.440	0.420	0.431	0.491	0.469	0.709	0.638	0.439	0.443
	720	**0.407**	**0.435**	0.423	0.442	0.490	0.488	0.447	0.468	0.463	0.472	0.521	0.500	0.733	0.636	0.472	0.490
	Avg	**0.378**	**0.407**	0.393	0.420	0.434	0.447	0.413	0.434	0.412	0.428	0.458	0.450	0.600	0.557	0.423	0.437
ETTh2	96	**0.258**	**0.326**	0.263	0.331	0.290	0.349	0.274	0.337	0.274	0.341	0.340	0.374	0.425	0.463	0.289	0.353
	192	**0.307**	**0.353**	0.312	0.364	0.364	0.404	0.341	0.382	0.339	0.385	0.402	0.414	0.473	0.500	0.383	0.418
	336	**0.319**	**0.375**	0.321	0.376	0.393	0.425	0.329	0.384	0.361	0.406	0.452	0.452	0.581	0.562	0.448	0.465
	720	**0.378**	**0.418**	0.382	0.422	0.468	0.476	0.379	0.422	0.445	0.470	0.462	0.468	0.775	0.665	0.605	0.551
	Avg	**0.315**	**0.368**	0.320	0.373	0.379	0.414	0.331	0.381	0.355	0.401	0.414	0.427	0.564	0.548	0.431	0.447
ETTm1	96	0.278	**0.333**	**0.277**	**0.333**	0.300	0.355	0.293	0.346	0.285	0.339	0.338	0.375	0.361	0.403	0.299	0.343
	192	**0.322**	**0.359**	0.325	0.363	0.339	0.380	0.333	0.370	0.327	0.365	0.374	0.387	0.387	0.422	0.335	0.365
	336	0.360	0.384	0.370	0.393	0.377	0.406	0.369	0.392	**0.356**	**0.382**	0.410	0.411	0.605	0.572	0.369	0.386
	720	0.414	**0.411**	**0.413**	0.415	0.441	0.440	0.416	0.420	0.419	0.414	0.478	0.450	0.703	0.645	0.425	0.421
	Avg	**0.343**	**0.371**	0.346	0.376	0.364	0.395	0.353	0.382	0.347	0.375	0.400	0.406	0.514	0.510	0.357	0.379
ETTm2	96	**0.158**	**0.247**	0.159	0.248	0.168	0.258	0.166	0.256	0.163	0.252	0.187	0.267	0.275	0.358	0.167	0.260
	192	**0.214**	**0.286**	0.217	0.290	0.231	0.300	0.223	0.296	0.216	0.290	0.249	0.309	0.345	0.400	0.224	0.303
	336	**0.266**	**0.322**	0.272	0.328	0.280	0.334	0.274	0.329	0.268	0.324	0.321	0.351	0.657	0.528	0.281	0.342
	720	**0.348**	**0.376**	0.353	0.378	0.370	0.390	0.362	0.385	0.420	0.422	0.408	0.403	1.208	0.753	0.397	0.421
	Avg	**0.246**	**0.307**	0.250	0.311	0.262	0.321	0.256	0.317	0.267	0.322	0.291	0.333	0.621	0.510	0.267	0.332
Weather	96	0.146	0.194	**0.143**	**0.187**	0.149	0.202	0.149	0.198	0.145	0.198	0.172	0.220	0.232	0.302	0.176	0.237
	192	**0.184**	**0.227**	0.185	0.229	0.198	0.246	0.194	0.241	0.191	0.242	0.219	0.261	0.371	0.410	0.220	0.282
	336	**0.236**	**0.270**	0.237	0.271	0.245	0.286	0.245	0.282	0.242	0.280	0.280	0.306	0.495	0.515	0.265	0.319
	720	**0.310**	**0.324**	0.318	0.332	0.316	0.333	0.314	0.334	0.320	0.336	0.365	0.359	0.526	0.542	0.323	0.362
	Avg	**0.219**	**0.253**	0.221	0.255	0.227	0.267	0.226	0.264	0.225	0.264	0.259	0.287	0.406	0.442	0.246	0.300
Electricity	96	0.132	0.226	**0.129**	0.223	0.131	0.227	**0.129**	**0.222**	0.131	0.229	0.168	0.272	0.150	0.251	0.140	0.237
	192	**0.146**	**0.238**	**0.146**	**0.238**	0.152	0.247	0.147	0.240	0.151	0.246	0.184	0.289	0.161	0.260	0.153	0.249
	336	0.162	**0.255**	0.163	0.258	0.173	0.270	0.163	0.259	**0.161**	0.261	0.198	0.300	0.182	0.281	0.169	0.267
	720	0.201	**0.289**	0.199	**0.289**	0.205	0.296	**0.197**	0.290	**0.197**	0.293	0.220	0.320	0.251	0.339	0.203	0.301
	Avg	0.160	**0.252**	**0.159**	**0.252**	0.165	0.260	**0.159**	0.253	0.160	0.257	0.192	0.295	0.186	0.283	0.166	0.264
Traffic	96	0.384	0.263	0.373	0.253	0.373	0.268	**0.360**	**0.249**	0.376	0.264	0.593	0.321	0.514	0.267	0.410	0.282
	192	0.397	0.267	0.392	0.273	0.394	0.282	**0.379**	0.256	0.397	0.277	0.617	0.336	0.549	**0.252**	0.423	0.287
	336	0.410	0.275	0.408	0.285	0.399	0.281	**0.392**	**0.264**	0.413	0.290	0.629	0.336	0.530	0.300	0.436	0.296
	720	0.450	0.297	0.437	0.296	0.451	0.300	**0.432**	**0.286**	0.444	0.306	0.640	0.350	0.573	0.313	0.466	0.315
	Avg	0.410	0.275	0.403	0.277	0.404	0.283	**0.391**	**0.264**	0.408	0.284	0.620	0.336	0.542	0.283	0.434	0.295

E.2 Component ablation on forecasting accuracy

Table 21: Ablation study on forecasting accuracy. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Horizon	TailFix		w/o MR-CSCM		w/o ACE		w/o Temporal Refiner	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.344**	**0.381**	0.347	0.383	0.350	0.385	0.348	0.387
	192	**0.381**	**0.405**	0.386	0.408	0.383	0.410	0.385	0.409
	336	**0.380**	**0.410**	0.393	0.428	0.383	0.413	0.387	0.414
	720	**0.407**	**0.435**	0.412	0.438	0.411	0.439	0.421	0.443
	avg	**0.378**	**0.407**	0.385	0.414	0.382	0.412	0.385	0.413
ETTh2	96	**0.258**	**0.326**	0.265	0.327	0.260	0.329	0.263	0.330
	192	**0.307**	**0.353**	0.322	0.370	0.328	0.364	0.328	0.364
	336	**0.319**	**0.375**	0.322	0.380	0.324	0.381	0.321	0.384
	720	**0.378**	**0.418**	0.380	0.421	0.381	0.420	0.383	0.424
	avg	**0.315**	**0.368**	0.322	0.374	0.323	0.373	0.324	0.376
ETTm1	96	**0.278**	**0.333**	0.281	0.337	0.283	0.338	0.284	0.340
	192	**0.322**	**0.359**	0.327	0.362	0.325	0.363	0.328	0.366
	336	**0.360**	**0.384**	0.364	0.387	0.361	0.385	0.367	0.388
	720	**0.414**	**0.411**	0.418	0.415	0.417	0.416	0.420	0.419
	avg	**0.343**	**0.371**	0.347	0.375	0.346	0.375	0.350	0.378
ETTm2	96	**0.158**	**0.247**	0.163	0.249	0.162	0.251	0.160	0.253
	192	**0.214**	**0.286**	0.219	0.292	0.217	0.289	0.218	0.291
	336	**0.266**	**0.322**	0.269	0.326	0.272	0.325	0.271	0.328
	720	**0.348**	**0.376**	0.351	0.383	0.350	0.380	0.352	0.381
	avg	**0.246**	**0.307**	0.251	0.313	0.250	0.311	0.250	0.313

E.3 Component ablation on tail stability

Table 22: Ablation study on tail stability metrics. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	H	TailFix			w/o Cross-Scale Mixer			w/o ACE			w/o Temporal Refiner		
		TAF	TailMean25%	TailSlope25%	TAF	TailMean25%	TailSlope25%	TAF	TailMean25%	TailSlope25%	TAF	TailMean25%	TailSlope25%
ETTh1	96	**1.170**	**0.387**	**0.000585**	1.177	0.389	0.000593	1.174	0.389	0.000610	1.173	0.390	0.000601
	192	**1.159**	**0.425**	**0.000217**	1.161	0.427	0.000219	1.170	0.430	0.000266	1.174	0.433	0.000261
	336	**1.088**	**0.405**	**-0.000204**	1.111	0.425	-0.000125	1.112	0.415	-0.000102	1.091	0.413	-0.000106
	720	**1.245**	0.483	**0.000636**	1.251	**0.478**	**0.000636**	1.247	0.483	0.000711	1.319	0.514	0.000740
	avg	**1.165**	**0.425**	**0.000308**	1.175	0.429	0.000330	1.175	0.429	0.000371	1.189	0.437	0.000374
ETTh2	96	1.377	**0.325**	0.000390	1.383	0.335	0.000646	**1.370**	0.330	**0.000210**	1.382	0.331	0.000287
	192	**1.297**	**0.378**	**0.000175**	1.358	0.384	0.001251	1.334	0.381	0.001013	1.338	0.384	0.001176
	336	1.239	**0.373**	0.000353	1.243	0.376	0.000414	**1.238**	0.381	**0.000313**	1.257	0.377	0.000414
	720	1.570	**0.519**	0.000807	1.558	0.521	**0.000743**	**1.555**	0.520	0.000744	1.557	0.521	0.000775
	avg	**1.370**	**0.398**	**0.000431**	1.385	0.404	0.000763	1.374	0.403	0.000570	1.383	0.403	0.000663
ETTm1	96	1.149	**0.308**	-0.000350	1.152	0.312	-0.000242	**1.145**	0.313	**-0.000421**	1.151	0.311	-0.000333
	192	1.188	0.365	-0.000190	1.190	0.369	0.000032	1.183	**0.363**	0.000121	**1.182**	0.364	**-0.000230**
	336	**1.191**	0.410	**0.000434**	1.198	0.410	0.000482	1.220	0.416	0.000543	1.199	**0.409**	0.000445
	720	1.248	**0.480**	**0.000162**	1.252	0.487	0.000205	**1.244**	0.484	0.000163	1.249	0.491	0.000214
	avg	**1.194**	**0.390**	**0.000014**	1.198	0.394	0.000119	1.198	0.394	0.000101	1.195	0.393	0.000024
ETTm2	96	1.459	**0.207**	0.000953	1.464	0.208	0.000866	**1.457**	0.221	**0.000750**	1.465	0.209	0.000837
	192	**1.473**	**0.283**	**0.000326**	1.491	0.285	0.000420	1.474	0.286	0.000386	1.483	0.285	**0.000326**
	336	**1.415**	**0.342**	**0.000381**	1.426	0.350	0.000398	1.420	0.352	**0.000381**	1.423	0.345	0.000416
	720	1.447	**0.450**	**0.000226**	1.447	0.452	0.000244	1.450	0.457	0.000261	**1.432**	0.453	0.000249
	avg	**1.448**	**0.320**	0.000471	1.457	0.323	0.000482	1.450	0.329	**0.000444**	1.450	0.323	0.000457

E.4 Horizon-wise results on PEMS datasets

To further examine the cross-benchmark performance of TailFix, Table 23 presents the horizon-wise results underlying the averages in Table 7. TailFix achieves the lowest MSE in 14 of the 16 settings and the lowest MAE in all 16 settings. These results demonstrate that its forecasting advantages extend across different traffic datasets and prediction horizons.

Table 23: Horizon-wise forecasting results on four PEMS datasets. Each entry reports MSE/MAE. Lower values are better, and the best results for each metric are highlighted in bold.

Dataset	H	TailFix (Ours)	Twins	iTransformer	PatchTST
PEMS03	12	**0.058/0.158**	0.065/0.169	0.069/0.175	0.099/0.216
	24	**0.073/0.176**	0.086/0.196	0.097/0.208	0.142/0.312
	48	**0.098/0.200**	0.121/0.234	0.131/0.243	0.211/0.319
	96	**0.126/0.224**	0.165/0.276	0.168/0.279	0.269/0.370
	Avg	**0.088/0.190**	0.109/0.219	0.116/0.226	0.180/0.304
PEMS04	12	**0.071/0.173**	0.077/0.181	0.081/0.188	0.105/0.224
	24	**0.084/0.189**	0.095/0.204	0.099/0.211	0.153/0.275
	48	**0.101/0.208**	0.120/0.231	0.133/0.247	0.229/0.339
	96	**0.118/0.223**	0.150/0.261	0.172/0.283	0.291/0.389
	Avg	**0.094/0.198**	0.111/0.219	0.121/0.232	0.195/0.307
PEMS07	12	**0.051/0.143**	0.060/0.158	0.067/0.167	0.095/0.207
	24	**0.060/0.160**	0.079/0.181	0.086/0.189	0.150/0.262
	48	**0.073/0.172**	0.104/0.209	0.110/0.214	0.253/0.340
	96	**0.089/0.192**	0.132/0.236	0.138/0.244	0.346/0.404
	Avg	**0.068/0.167**	0.094/0.196	0.100/0.204	0.211/0.303
PEMS08	12	0.079/**0.169**	**0.075**/0.174	0.080/0.183	0.168/0.232
	24	**0.090/0.181**	0.106/0.206	0.118/0.221	0.224/0.281
	48	**0.128/0.199**	0.167/0.258	0.186/0.265	0.321/0.354
	96	0.192/**0.236**	**0.184**/0.251	0.221/0.267	0.408/0.417
	Avg	**0.122/0.196**	0.133/0.222	0.151/0.234	0.280/0.321

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly define the main contributions and scope of this work. The stated claims are supported by the proposed method design and are consistent with the empirical results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and boundary cases of TailFix in Appendix A, including settings where tail deterioration is weak and the full module combination may not always yield the best results.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This work proposes a model architecture and evaluation metrics, but does not include formal theoretical results, theorems, or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the overall experimental pipeline required to reproduce the main results, including the benchmark datasets, forecasting settings, evaluation metrics, baseline sources, and unified evaluation protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary materials include anonymized code, together with scripts for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The appendix reports detailed training and implementation settings, including the optimizer, batch size, number of training epochs, random seeds, model hyperparameters, and evaluation settings used for testing.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports mean $\pm$ standard deviation results across five random seeds, providing an assessment of the robustness of the experimental results to random initialization.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the hardware configuration and estimated compute cost required to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work is conducted on publicly available time-series benchmark datasets and does not involve personally identifiable or sensitive information, human-subject experiments, or deployment in high-risk application scenarios. Therefore, the study is consistent with the NeurIPS ethics guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work studies a general-purpose time-series forecasting method and evaluates it on publicly available benchmark datasets. It does not involve practical deployment in any specific social decision-making system.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not involve such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: The paper cites the original sources of the datasets and baseline implementations, and their licenses are respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper will release documented code for reproducing the proposed method and experiments.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing experiments or human-subject research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve crowdsourcing experiments or human-subject research; therefore, IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method of this work does not use LLMs as a major, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.