MDK-MoE: Multi-view Decomposed Kalman Mixture of Experts Framework for Non-stationary Time Series Forecasting

Rui Hou [1], Yao Liu [2,*], Ruilin Jiang [2], Mengyao Lu [2], Jingbo Wang [2], Lanyi Zhang [2], Qiao Liu [1]
[1] School of Computer Science and Engineering,
University of Electronic Science and Technology of China
[2] School of Information and Software Engineering,
University of Electronic Science and Technology of China
hours0126@gmail.com, liuyao@uestc.edu.cn

Abstract

Non-stationary time series forecasting (**NTSF**) requires modeling evolving trends and seasonal patterns. Although linear decomposition operators (e.g., FFT) help isolate these patterns, they often introduce operator-dependent biases and idiosyncratic noise. To address this issue, we propose a novel Multi-view Decomposed Kalman Mixture of Experts (**MDK-MoE**) framework, which *improves forecasting by strategically integrating multiple complementary, yet biased, linear decompositions*. Specifically, a Multi-view Decomposition (**MVD**) extracts frequency components from three complementary views. To mitigate view-specific noise, a Maximizing Mutual Information (**MMI**) enhances cross-view consistency, maximizes task-relevant information across diverse views, and suppresses view-dependent task-irrelevant information. To address the conflicts inherent in multi-source observations, the efficient **Kalman MoE** adaptively fuses distinct multi-view features. Instead of adopting traditional $O(d^3)$ covariance inversion, our learnable Kalman gain design enhances numerical stability and enables reliable state estimation under non-stationary scenarios. Extensive experiments on 13 real-world datasets demonstrate that **MDK-MoE** consistently achieves state-of-the-art performance. Code is available at `https://github.com/HRFarsighted/MDK-MOE`.

1 Introduction

Non-stationary time series forecasting (**NTSF**) aims to model trend and other patterns to predict future states [15]. In 1919, Warren M. Persons [26] proposed that fluctuations in time series arise from four major factors: secular trends, cyclical fluctuations, seasonal variation, and irregular variation. Consequently, real-world time series are jointly affected by these sources of fluctuation, giving rise to the fundamental challenge of non-stationarity [27].

Deep neural networks such as Transformer [34] have achieved promising predictive performance, but the accumulation of biases [23] caused by non-stationarity across different network layers may lead to gradient explosion in the model. A common strategy to alleviate this phenomenon is to adopt learnable normalization mechanisms [13, 20] during the preprocessing stage. While these methods focus on mitigating distributional drift in the original data, they often fail to account for the critical importance of modeling seasonal patterns [17]. In contrast, decomposition-based methods directly extract intrinsic components, such as seasonal and trend patterns, from non-stationary sequences using

*Corresponding author: Yao Liu, email: liuyao@uestc.edu.cn

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

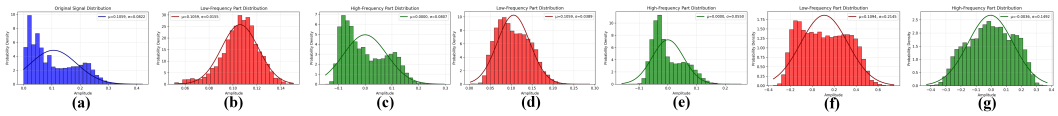


Figure 1: Operator-dependent distributional biases. (a) Original PDD on the Traffic dataset; (b)-(g) present high and low-frequency components decomposed by FFT, MA and Wavelet, respectively.

specific linear operators. For instance, DLinear [38] applies a time-domain low-pass moving average (MA) operator to separate long-term trends and residual seasonal patterns. Koopa [18] integrates Koopman operators with neural networks for non-stationary forecasting. AMD [10] employs adaptive multi-scale decomposition via average pooling. Moreover, frequency-domain techniques, such as FFT [43, 37] and wavelet transforms [29, 42], are widely used to extract high and low-frequency components to enhance representation.

Despite the success of decomposition-based methods, the single-view decomposition operators possess inherent inductive biases and are susceptible to view-specific noise. As shown in Fig. 1, (b), (d), and (f) show the probability density distributions (PDD) of the high-frequency components extracted by FFT, MA, and wavelet operators, respectively (more empirical study are provided in Appendix A.1). These distributions all deviate systematically from that of the original features (a). Similarly, the corresponding low-frequency components (c), (e), and (g) also exhibit inherent distributional biases with respect to (a). Recently, based on the time-frequency consistency principle [40], TimesNet [35] and WFTNet [16] aim to decompose complementary time-frequency information from distinct views to mitigate view-specific noise. However, these methods simply aggregate features across different views, which may amplify redundant biases rather than eliminate them.

Based on these observations, we aim to address two fundamental challenges. *First, the extraction of task-relevant information from multiple biased decomposition systems while suppressing idiosyncratic noise*. To overcome this limitation, we propose a novel **MDK-MoE** which leverages three complementary (the quantitative analysis of the complementarity among the three operators is presented in Appendix A.2) linear decomposition operators in the Multi-view Decomposition (MVD) to generate high and low frequency components. According to the multi-view information bottleneck principle [5, 9], the cross-view Maximizing Mutual Information (MMI) mechanism that effectively minimizes the upper bound of view-specific biases. *Another key challenge is to robustly fuse multi-view observations into a unified state estimate*. While the Kalman filter [32] is capable of achieving optimal state estimation from multiple linearly biased observation systems, conventional Kalman fusion strategies [36] generally fail to adaptively evaluate the reliability of multi-view decomposition systems and rely on complex covariance calculations. To address this issue, we propose the Kalman MoE, which treats multi-view features as distinct observation sources and adaptively aggregates them via the MoE structure. We provide a theoretical proof that this mechanism can efficiently and adaptively fuse multi-view observations into a unified state estimate. The contributions are summarized as follows:

- We propose a novel **MDK-MoE** for NTSF that achieves effective feature fusion and accurate prediction by capturing task-relevant information from multi-view biased decomposition linear systems while suppressing view-specific noise.
- We introduce the cross-view MMI mechanism that minimizes operator-specific biases by filtering out redundant noise. This process extract task-relevant features, establishing a refined, denoised latent foundation for multi-view feature fusion.
- We design a Kalman MoE that treats multi-view features as diverse observation sources. By employing a learnable Kalman gain instead of traditional $O(d^3)$ covariance inversions, our approach improves computational efficiency in non-stationary environments.

2 Related Works

2.1 Time-Domain Stationarization Methods

Traditional statistical learning methods, such as ARMA [33], treat TSF as a stochastic process, but those methods rely on the assumption of stationarity. To adapt the non-stationary of real-world data, ARIMA [1] introduces differencing operations to achieve stationarity. Recently, normalization-based

stationarization approaches [23, 25] have attracted attention. These methods aim to alleviate non-stationarity through model-agnostic normalization techniques, thereby providing deep models with more stable input distributions [4]. For example, RevIN [13] applies Z-score normalization with learnable affine parameters. SAN [20] further introduces a fine-grained normalization scheme. In addition, some methods focus on extracting intrinsic non-stationary components using predefined linear operators. DLinear [38] employs MA operators to decompose the input into trend and seasonal components. AMD [10] and TimeMixer [30] uses adaptive average pooling to obtain multi-scale decompositions. Koopa [18] integrates linear Koopman operators with neural networks to model non-stationary dynamics. However, linear predictive systems may suffer from inherent biases, which limit their representation capacity. To address this issue, K^2VAE [36] incorporates Kalman filtering to derive improved state estimates under biased linear dynamics.

2.2 Frequency-Domain Decomposition Methods

From the view of frequency-domain decomposition, the high-frequency components donate short-term dynamics, while the low-frequency components reflect long-term seasonal patterns. This observation has motivated the integration of frequency-domain decomposition with deep neural networks. Such as, SFM [39] employs discrete FFT to decompose the hidden states of LSTM into frequency components. FEDformer [43] incorporates frequency-enhanced attention based on discrete FFT to compute attention weights from frequency components. TimesNet [35] models multi-period patterns by applying multi-scale convolutions to the dominant frequency components, while FilterTS [31] introduces a dynamic cross-variable filtering module, which dynamically leverages other variables as filters to extract and reinforce shared variable frequency components. Unlike FFT-based methods [8] that emphasize global frequency representations, wavelet focus on capturing localized time–frequency characteristics. WHEN [29] designs a wavelet-based attention mechanism to analyze dynamic frequency components in non-stationary series. In addition, WFTNet [16] utilizes both FFT and wavelet transforms to extract comprehensive temporal-frequency information from the signal, where FFT captures the global periodic patterns and wavelet captures the local patterns.

3 Methodology

NTSF involves predicting future values based on historical values. The historical observations, $\mathbf{X} = \{x_1, \ldots, x_L\} \in \mathbb{R}^{N \times L}$, serve as the input and consist of data from N variables over L time steps. The prediction sequence, $\mathbf{Y} = \{x_{L+1}, \ldots, x_{L+T}\} \in \mathbb{R}^{N \times T}$, serves as the output and comprises data from N variables over T time steps.

3.1 Structure Overview

Existing studies have demonstrated the effectiveness of decomposition-based methods for tackling NTSF [35, 10, 31], but the use of specific linear decomposition operators inevitably introduces operator-dependent linear biases [36]. Considering the time-frequency consistency and complementarity of decomposed components [40], we adopt the novel **MDK-MoE** to achieve accurate state prediction via multi-view biased decomposition, as shown in Fig. 2, **MDK-MoE** consists of four key parts: (a) Embedding, which normalizes the raw inputs and maps them into high-dimensional representations; (b) MVD, which decomposes the input features from multiple views to obtain corresponding low and high frequency components; (c) MMI, which enforces consistency between high and low frequency representations across different views; and (d) Kalman MoE, which efficiently integrates multi-view decomposed components to estimate more accurate latent representations. Finally, a Projection layer nonlinearly decodes the learned representations into forecasting outputs.

3.2 Embedding Layer

In the Embedding layer, to address the potential bias and instability caused by fluctuations in MTS data during model training, we apply Z-score normalization to the raw data, enhancing the stability of the training process, as follows:

$$x_t^i = \left(x_t^i - \text{mean}\left(x_t^i\right)\right) / \sigma_i. \tag{1}$$

where $t = \{1, ..., L\}$, $i = \{1, ..., N\}$, and σ_i is the standard deviation for variable i.

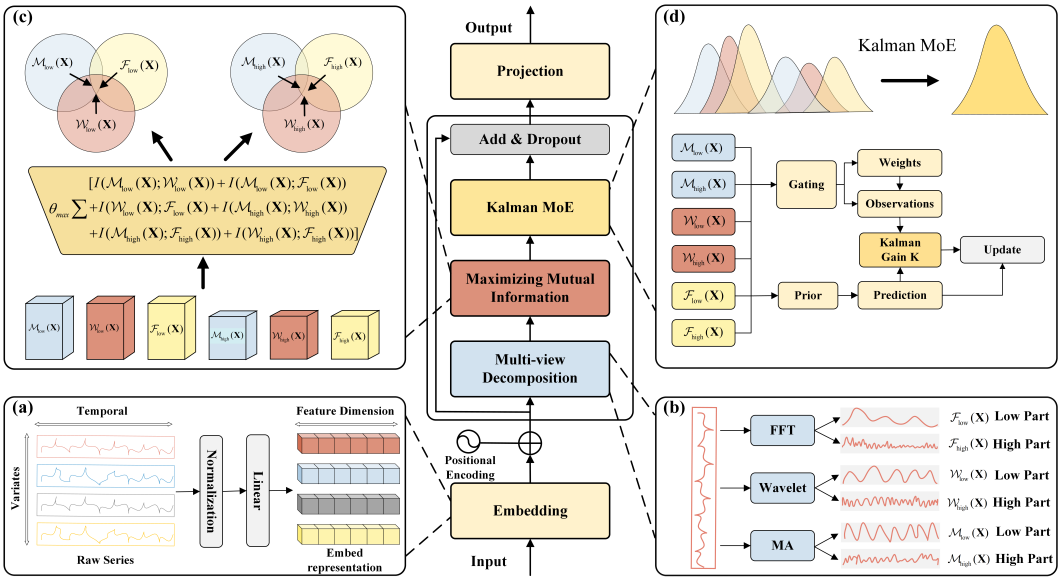


Figure 2: Overall architecture of ours **MDK-MoE** composition by four main parts.

Following normalization, input $\mathbf{X}$ mapped to embedding representation (embedding dimension is D), which is calculated by following equation:

$$X = \text{Linear}\,(X) + P_{Emb}. \tag{2}$$

where $X \in \mathbb{R}^{L\times D}$, $P_{Emb} \in \mathbb{R}^{L\times D}$ represents the positional embedding.

3.3 Multi-view Decomposition (MVD)

We aim to construct a complementary multi-view linear decomposition system in the MVD. We select three linear decomposition operators, the MA [6], Wavelet [21] and FFT [2], which decompose the original data into corresponding high-frequency components ($M_{\text{high}}(X), W_{\text{high}}(X), F_{\text{high}}(X)$) and low-frequency components ($M_{\text{low}}(X), W_{\text{low}}(X), \mathcal{F}_{\text{low}}(X)$), respectively. The complementarity of these operators is reflected in the time-frequency domain and global-local decomposition complementarity. Specifically, the MA operator, as a time-domain low-pass filter, enables smooth filtering within local windows; the Wavelet operator emphasizes the time-varying characteristics of data and provides a joint representation of temporal and frequency-domain features; the FFT operator, by contrast, focuses on transforming time-domain signals into a global frequency-domain representation.

(1) **MA** decompose the original data into corresponding high and low frequency components:

$$M_{\text{low}}(x_t^i) = \frac{1}{K}\sum_{j=-(K-1)/2}^{(K-1)/2} x_{t+j}^i, M_{\text{high}}(x_t^i) = x_t^i - M_{\text{low}}(x_t^i) \tag{3}$$

where K denotes the window size, and x_t^i represents the value of the i-th variable at time t.

(2) **Wavelet** [7] decompose the original data into corresponding high and low frequency components:

$$W_{\text{low}}(x_t^i) = x_t^i * \psi(t) = \sum_{\tau} x_{t-\tau}^i \cdot \psi(\tau). \tag{4}$$

where $\psi(t) = \frac{1}{\sqrt{s\sqrt{\pi}}} e^{-t^2/(2s^2)} \cos(2\pi f_0 t)$ is the wavelet basis, and $W_{\text{high}}(x_t^i) = x_t^i - W_{\text{low}}(x_t^i)$. s is the scale parameter, f_0 is the central frequency, and $*$ denotes convolution.

(3) **FFT** decompose the original data into corresponding high and low frequency components. By applying a low-frequency mask M_k, we isolate the low- and high-frequency component:

$$F_{\text{low}}(x_t^i)^R = \sum_{k=0}^{L-1} \hat{x}_k \cdot M_k \cdot e^{j2\pi kt/L}, F_{\text{high}}(x_t^i) = x_t^i - F_{\text{low}}(x_t^i)^R. \tag{5}$$

where R is the low-frequency threshold ratio, and $\hat{x}_k$ denotes the Fourier coefficients.

3.4 Maximizing Mutual Information (MMI)

Besides their complementarity, the decomposed features also exhibit an intrinsic cross-view consistency [40] between high (short-term trends) and low-frequency components (long-term seasonality). However, Multi-view decomposition across the time, frequency, and wavelet domains tends to introduce view-specific noise. According to the multi-view information bottleneck principle [5, 9], by MMI between high and low frequency components, we ensure that the model captures cross-view shared features, thereby mitigating learning biases (**Theorem 1**). The objective function is defined as:

$$\max_{\theta} \sum I_{low}(Z_{1,i}; Z_{2,i}) + I_{high}(Z_{1,i}; Z_{2,i}). \tag{6}$$

where $I_{low}(Z_{1,i}; Z_{2,i})$ denotes the MMI between every pair of low-frequency components - $M_{\text{low}}(X)$, $W_{\text{low}}(X)$ and $F_{\text{low}}(X)$ - treated as positive pairs. Similarly, $I_{high}(Z_{1,i}; Z_{2,i})$ represents the MMI among the corresponding high-frequency feature components - $M_{\text{high}}(X)$, $W_{\text{high}}(X)$ and $F_{\text{high}}(X)$.

Essentially, the MMI objective can be reformulated as minimizing a tractable variational lower bound, we further using the InfoNCE objective [24], the objective function is defined as follows:

$$\mathcal{L}_{\text{mi}} = -\mathbb{E}\left[\log \frac{\exp\left(\frac{s(z_{1,i}, z_{2,i})}{\tau}\right)}{\sum_{j=1}^{C} \exp\left(\frac{s(z_{1,i}, z_{2,j})}{\tau}\right)}\right]. \tag{7}$$

where $s(z_{1,i}, z_{2,i})$ denotes the similarity between the positive pairs, and τ is a temperature parameter. The denominator sums the similarities between all samples, and C denotes the number of samples.

Theorem 1. *Maximizing the mutual information $I(Z_1; Z_2)$ between different views minimizes the upper bound of the view-specific bias. Under the multi-view consistency principle, the task-relevant information $I(Z; Y)$ is maximized when the redundant view-specific noise is discarded through cross-view alignment.*

This theorem constrains the latent space by cross-view MMI, making it possible to mitigate view-specific biases across multiple complementary operators. Specifically, MMI encourages agreement between corresponding representations from different decomposition views. Under the sampling assumptions of the standard InfoNCE formulation, minimizing the contrastive loss maximizes a lower bound on cross-view mutual information. This provides a motivation for representation alignment, but does not by itself guarantee that all shared information is task-relevant or that view-specific prediction bias is eliminated.

3.5 Kalman Mixture of Experts (Kalman MoE)

The classic Kalman filter [32] updates the estimation of the current state based on the previous state and the current observation, the state update step as follows:

$$Z_{new} = Z_{pre} + K \odot (Z_{obs} - Z_{pre}). \tag{8}$$

where Z_{obs} denotes the historical observation, Z_{pre} denotes the prior prediction, K is the Kalman gain, and $\odot$ denotes the Hadamard product operation.

We further analyze the Z_{pre}, Z_{obs} and K. Specifically, Z_{pre} is calculated using a learnable state transition matrix. For Z_{obs}, we interpret the high and low frequency components decomposed by the MVD (Z^i for i={1,...,6}) as multiple observations of the same data for the Kalman filter, and denoised via the MMI. This necessitates the construction of an effective multi-view observation fusion mechanism to obtain Z_{obs}. Furthermore, the traditional Kalman gain $K = \frac{PH^T}{HPH^T+R}$ requires the recursive update of the observation H, prediction error covariance P, and measurement noise covariance R. In the case of high-dimensional time series data, the computational cost of these covariance estimations is substantial, and the estimations are highly sensitive to noise. Thus, we aim to construct a learnable Kalman gain K^* to approximate and simplify these computational processes.

In summary, we follow Eq. 8 to design a more efficient Kalman MoE. Kalman MoE is a Kalman-inspired fusion module that retains an innovation-correction structure. It constructs a reference representation from the decomposition views and an observation-like representation through MoE weighting. A learned, bounded coefficient controls the correction between them. The connection to Kalman filtering is structural: the module does not propagate covariance estimates or compute a Bayesian filtering posterior. Kalman MoE is rephrased into following three computational processes.

(1) **MoE-Based Observation Fusion**: We design an adaptive MoE gating mechanism to fuse the observations Z_{obs}. First, the mean value Z^m of the six types of features is calculated to generate the learnable gating logic matrix $G_{raw} = Z^m w_g$, where w_g is the learnable gating matrix. Then, to enhance the model's generalization capability in selecting expert weights, learnable Gaussian noise is added during the training phase to obtain the final gating logic matrix G:

$$G = G_{raw} + \mathcal{N}(0,1) \odot \sigma. \tag{9}$$

where $\sigma = \text{Softplus}(G_{raw} \cdot w_n) + \epsilon$ represents the noise standard deviation. $G \in \mathbb{R}^{N \times E}$ with N and E denoting the number of variables and experts, while w_n denotes the learnable parameter matrix for noise. The Softplus function is employed to ensure the non-negativity of the noise standard deviation, $\mathcal{N}(\cdot)$ denotes the Gaussian noise distribution.

Next, a sparse gating mechanism is designed to enhance expert selection efficient. We compute the threshold k_{th}, construct a binary $mask$, and apply Softmax normalization:

$$Z_{obs} = \sum_{i=1}^{E} \text{Softmax}(\text{mask}(G_i)) \odot Z^i. \tag{10}$$

where G_i donates the routing weight for i-th decomposition features, and $mask = (G_i < k_{th})$. To prevent expert collapse and promote balanced expert utilization, we introduce a routing-balance loss:

$$\mathcal{L}_{\text{moe}} = E \cdot \sum_{e=1}^{E} \left(\bar{g}_e - \frac{1}{E} \right)^2. \tag{11}$$

where $\bar{g}_e = \frac{1}{N} \sum_{j=1}^{N} \text{G}[j, e]$ denotes the average routing weight assigned to e-th expert. When all experts receive equal average weight $1/E$, the loss = 0.

Theorem 2. *The MoE-based fusion mechanism $\sum w_i Z_i$ yields the best linear unbiased estimator (BLUE) for the observation state. The learnable weights w_i converge to the precision-proportional distribution $w_i \propto 1/\sigma_i^2$, which minimizes the fused observation error covariance.*

For Theorem 2, the inverse-variance estimator is BLUE only for scalar observations with zero-mean, mutually uncorrelated errors of known positive variances and deterministic weights summing to one. Tn addition, gaussianity is not required for the BLUE[11, 12] result; it is used only for the additional MLE interpretation.

(2) **Prior-Based State Prediction**: the mean Z_m of the six decomposed features is used as the prior state, which is propagated through a learnable state transition matrix A to obtain Z_{pre}:

$$Z_{pre} = A \cdot \min\left(1, \frac{\alpha}{||A||_2}\right) \cdot Z_m. \tag{12}$$

where spectral norm $||A||_2$ is constrained to ensure system stability. The α is a hyperparameter.

(3) **State Update with Learnable Kalman Gain**: A learnable Kalman gain K^* is used to balance the contribution of the prior prediction Z_{pre} and the MoE-fused observation Z_{obs} (**Theorem 3**). The refined posterior state Z_{new} is obtained by applying residual correction:

$$K^* = k_{min} + (k_{max} - k_{min}) \cdot \sigma(K_{raw}), \tag{13}$$

$$Z_{new} = Z_{pre} + K^* \odot (Z_{obs} - Z_{pre}). \tag{14}$$

where K_{raw} is a learnable parameter initialized to zero, and $\sigma(.)$ denotes the sigmoid function. The sigmoid parameterization keeps each innovation coefficient between the prescribed bounds. These bounds are a modeling constraint rather than a general property of matrix-valued Kalman gains. The coefficient is optimized jointly with the forecasting model and is not derived from an uncertainty estimate. Therefore, (Eq. 13) guarantees that K^* is constrained within the interval $[k_{min}, k_{max}] \subset (0, 1)$, which is consistent with the theoretical requirement that Kalman gains remain between 0 and 1.

Theorem 3. *The end-to-end learnable gain K^* provides a robust approximation of the data distribution by sidestepping the $O(d^3)$ matrix inversion instability.*

Theorem 3 proves that the learnable gain K^* leverages global distribution information across the entire training set. Mathematically, this sidesteps the estimation bias of local sample covariances

and inherently regularizes the system against numerical instability. When the prior and observation estimators are held fixed, the scalar population-MSE objective is minimized by K^* only when their errors are zero-mean and uncorrelated. Under our bounded parameterization, the corresponding oracle coefficient is the projection of this value onto the permitted interval. This result does not establish convergence of the complete non-convex network, a covariance-calibrated Kalman gain, or steady-state Kalman optimality. We provide the proof of **Theorem 1-3** in Appendix B, and present the sensitivity of model performance to violations of ideal assumptions in Appendix C.4.

3.6 Projection

The fused Z_{new} from the Kalman MoE is combined with the original input X via a residual connection, followed by dropout. Next, MLP maps Z_{new} to the prediction $\hat{Y}$, and a denormalization step restores the original data scale:

$$\hat{Y} = \text{Dnorm}(\text{MLP}(\text{Dropout}(Z_{\text{new}} + X)). \tag{15}$$

3.7 Overall Learning Objective

The overall learning objective is formulated as a weighted combination of the $\mathcal{L}_{\text{mi}}$, $\mathcal{L}_{\text{moe}}$, and $\mathcal{L}_{\text{pre}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pre}} + \lambda_{\text{mi}} \cdot \mathcal{L}_{\text{mi}} + \lambda_{\text{moe}} \cdot \mathcal{L}_{\text{moe}}, \tag{16}$$

$$\mathcal{L}_{\text{pre}} = \text{Smooth L1}(\hat{Y}, Y) + \text{MSE}(\hat{Y}, Y). \tag{17}$$

where Y denotes the ground-truth values for the future horizon. The coefficient λ_{mi} controls the weight of the $\mathcal{L}_{\text{mi}}$, while λ_{moe} regulates the $\mathcal{L}_{\text{moe}}$. The prediction loss $\mathcal{L}_{\text{pre}}$, which combines Smooth L1 and MSE, provides a trade-off between robustness to outliers and prediction smoothness.

4 Experiments

In this section, we present extensive empirical results to demonstrate the SOTA performance of **MDK-MoE** on multivariate long and short term forecasting tasks, compared with other baselines. In addition, the appendix provides detailed theoretical analysis, comparisons with baseline models, ablation studies, sensitivity analysis, significance analysis, and more.

4.1 Experimental Setup

Datasets. We evaluate our model on 13 real-world datasets, including Traffic, ECL, Exchange, Solar, Weather, and the ETT benchmarks (ETTh1-2, ETTm1-2) for long-term forecasting. PEMS datasets (PEMS03, PEMS04, PEMS07, and PEMS08) for short-term forecasting (see Appendix C.1).

Baselines. We carefully select 13 representative models to serve as baselines in the field of time series forecasting, including (1) Decomposition on time-domain: K^2VAE [36], AMD [10], TimeMixer [30], MTS-Mixers[14] and DLinear [38]. (2) Decomposition on frequency-domain: FilterTS [31], WFTNet [16], TimesNet [35], FEDformer and [43]. (3) Other sota models: iTransformer [19], TiDE [3], PatchTST [22] and Crossformer [41]. Where K^2VAE, AMD, TimeMixer, DLinear, FilterTS, WFTNet, TimesNet, FEDformer and iTransformer for long-term forecasting. AMD, MTS-Mixers, DLinear, TimesNet, FEDformer, TiDE, PatchTST, and Crossformer for short-term forecasting.

4.2 Main Results

Long-term forecasting task are shown in Tab. 1, we carefully select 9 representative models to serve as baselines. Avg denotes the average value across all prediction lengths T, with values highlighted in bold red* indicating the best performance for each metric, and values marked with blue underlining representing the second-best performance for each metric, ♠ representing reproduced results. The lower the MSE/MAE represent more accurate. To assess run-to-run variability, we compare MDK-MoE with AMD using ten matched seeds for each dataset-horizon configuration. Appendix C.2 reports the paired results and the statistical protocol. The differences vary across datasets and horizons; we therefore interpret the evidence as an overall rather than a universal advantage. Overall, **MDK-MoE** achieves the best performance in **14*** cases and the second-best performance in 3

Table 1: Long-term forecasting task.The results for MSE and MAE are averaged across all prediction lengths $\{T = 96, 192, 336, 720\}$. To the best for a fairer comparison for all baselines, the input sequence length L is searched among $\{96, 192, 336, 512, 672, 720\}$.

Models		MDK-MoE		k^2VAE ♠		AMD		FilterTS ♠		WFTNet♠		TimesNet		DLinear		FEDformer		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	Avg	**0.387***	**0.267***	0.601	0.323	0.393	0.271	0395	0.277	0.617	0.333	0.619	0.335	0.433	0.295	0.603	0.372	0.428	0.282
ECL	Avg	**0.157***	0.252	0.215	0.296	**0.157***	**0.250***	0.163	0.257	0.198	0.298	0.192	0.295	0.176	0.274	0.207	0.321	0.178	0.269
Exchange	Avg	0.319	0.382	0.539	0.485	0.328	0.387	0.349	0.396	0.528	0.486	0.416	0.443	**0.296***	**0.377***	0.477	0.477	0.360	0.403
Solar	Avg	**0.190***	0.248	0.235	0.325	0.191	**0.239***	0.199	0.243	0.227	0.313	0.244	0.334	0.329	0.390	0.243	0.350	0.233	0.261
Weather	Avg	**0.219***	**0.257***	0.259	0.287	0.221	0.261	0.228	0.266	0.248	0.277	0.259	0.386	0.240	0.300	0.310	0.357	0.257	0.278
ETTm1	Avg	**0.345***	**0.374***	0.410	0.410	0.346	**0.374***	0.354	0.382	0.389	0.408	0.399	0.405	0.357	0.378	0.382	0.422	0.407	0.409
ETTm2	Avg	**0.251***	**0.312***	0.295	0.339	0.253	0.315	0.259	0.318	0.288	0.330	0.313	0.332	0.267	0.331	0.291	0.343	0.288	0.332
ETTh1	Avg	**0.404***	**0.422***	0.493	0.475	0.406	0.423	0.418	0.428	0.450	0.448	0.463	0.451	0.422	0.437	0.428	0.453	0.454	0.447
ETTh2	Avg	**0.341***	**0.388***	0.410	0.426	0.350	0.392	0.351	0.392	0.387	0.416	0.414	0.427	0.431	0.446	0.387	0.434	0.383	0.406
1st count		**8**	**6**	**0**	**0**	**1**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Table 2: Short-term forecasting task. The input length L is 96 and the prediction length T is 12.

Models	(Ours)		AMD		PatchTST		Crossformer		FEDformer		TimesNet		TiDE		DLinear		MTS-Mixers	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	**0.062***	**0.164***	0.084	0.198	0.099	0.216	0.090	0.203	0.126	0.251	0.085	0.192	0.178	0.305	0.122	0.243	0.117	0.232
PEMS04	**0.073***	**0.175***	0.083	0.198	0.105	0.224	0.098	0.218	0.138	0.262	0.087	0.195	0.219	0.340	0.148	0.272	0.129	0.267
PEMS07	**0.059***	**0.154***	0.074	0.180	0.095	0.150	0.094	0.200	0.109	0.225	0.082	0.181	0.173	0.304	0.115	0.242	0.134	0.278
PEMS08	**0.074***	**0.173***	0.093	0.206	0.168	0.232	0.165	0.214	0.173	0.273	0.112	0.212	0.227	0.343	0.154	0.276	0.186	0.286
Avg	**0.067***	**0.166***	0.083	0.195	0.116	0.205	0.111	0.208	0.136	0.252	0.091	0.195	0.199	0.323	0.134	0.258	0.141	0.265
1st count	**5**	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

cases out of 18 total evaluations, significantly outperforming the second-best method, AMD, which attains only **4*** best cases. This demonstrates the strong generalization ability of **MDK-MoE** across datasets from different domains and with varying numbers of variables. Notably, **MDK-MoE** shows substantial improvements in terms of MSE. Compared with K^2VAE and DLinear, it reduces the average MSE by 24.48% and 11.59% across 9 datasets, respectively. In particular, the inferior performance of K^2VAE suggests that iterating a single Koopman operator may introduce cumulative prediction bias. In addition, models such as FilterTS, TimesNet, and FEDformer primarily emphasize partial frequency-domain representations and fail to achieve optimal performance. Although WFTNet incorporates both FFT and wavelet features, its lack of an effective feature selection and fusion mechanism leads to inferior accuracy and low computational efficiency (see Section 4.4). These results further validate the necessity and effectiveness of the proposed **Kalman MoE** for efficient feature fusion. Moreover, **MDK-MoE** consistently outperforms the advanced iTransformer.

Short-term forecasting task are shown in Tab. 2, we carefully select 8 representative models to serve as baselines. Avg denotes the average value across all PEMS datasets. **MDK-MoE** achieves the best results on all 10 MSE and MAE metrics, demonstrating its superiority in short-term forecasting. Specifically, compared with the second-best method AMD, **MDK-MoE** reduces the average MSE from 0.083 to 0.067 (a 19.27% improvement) and the average MAE from 0.195 to 0.166 (a 14.87% improvement). Forecasting result visualization and detailed analyses are provided in Appendix C.2.

Table 3: Component ablation of **MDK-MoE** on 9 datasets across all prediction lengths.

Models		MDK-MoE		w/o f		w/o m		w/o w		w/o fm		w/o fw		w/o mw		w/o mi		w/o kal		w/o mi_kal	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	Avg	**0.387**	**0.267**	0.388	0.268	**0.387**	0.268	**0.387**	0.268	0.387	**0.268**	**0.387**	0.268	**0.387**	0.268	0.389	0.271	0.388	0.268	**0.387**	0.268
ECL	Avg	**0.157**	**0.252**	0.159	0.255	0.160	0.256	0.158	0.254	0.160	0.254	0.160	0.254	0.160	0.254	0.159	0.255	0.160	0.255	0.160	0.254
Exchange	Avg	**0.319**	**0.382**	0.352	0.401	0.347	0.399	0.408	0.436	0.328	0.387	0.331	0.389	0.328	0.387	0.360	0.407	0.335	0.392	0.330	0.388
Solar	Avg	**0.190**	**0.248**	0.195	0.256	0.195	0.254	0.194	0.255	0.194	0.251	0.195	0.251	0.195	0.251	0.196	0.256	0.194	0.252	0.195	0.252
Weather	Avg	0.219	0.257	**0.218**	0.257	**0.218**	**0.256**	**0.218**	**0.256**	0.220	0.258	0.220	0.258	0.220	0.258	0.223	0.262	0.219	0.257	0.220	0.258
ETTm1	Avg	**0.345**	**0.374**	0.346	**0.374**	**0.345**	**0.374**	0.346	**0.374**	0.346	0.375	0.347	0.375	0.346	0.375	0.347	0.375	0.346	0.375	0.346	0.375
ETTm2	Avg	**0.251**	**0.312**	0.254	0.315	0.253	0.315	0.253	0.314	0.253	0.314	0.253	0.314	0.253	0.314	0.253	0.314	0.252	0.313	0.252	0.313
ETTh1	Avg	**0.404**	**0.422**	0.409	0.427	0.408	0.426	0.411	0.427	0.419	0.431	0.414	0.429	0.416	0.430	0.415	0.430	0.411	0.426	0.413	0.428
ETTh2	Avg	**0.341**	**0.388**	0.348	0.393	0.347	0.392	0.351	0.397	0.350	0.397	0.354	0.396	0.352	0.393	0.351	0.395	0.350	0.395	0.358	0.400

4.3 Ablation Studies

We conduct 9 ablation sets on Tab. 3. First, removing a single decomposition operator—FFT (w/o f), MA (w/o m), or Wavelet (w/o w)—such that only two operators remain, leads to an average increase

of 2.21% in MSE and MAE across all datasets. In particular on the Exchange dataset, where the error increases by 11.40%. Second, we evaluate settings that retain only a single operator by removing two operators: Wavelet-only (w/o fm), MA-only (w/o fw), and FFT-only (w/o mw). Using a single operator for forecasting results in an average performance degradation of 1.33% in MSE and MAE across all datasets. When all three operators are jointly used but the MMI module is removed (w/o mi), the average MSE and MAE increase by 2.57% across all datasets. Furthermore, removing both the MMI and Kalman MoE (w/o mi_kal) leads to an average performance drop of 1.43%. Overall, across all ablation settings and datasets, removing key components results in an average increase of 1.63% in MSE and MAE, indicating varying degrees of performance degradation. Although combining only two operators yields slightly better results on the Weather dataset, this behavior further highlights the benefit of the Kalman MoE in adaptively selecting and fusing multi-view features.

In Appendix C.3, we perform balanced-parameter ablation experiments to avoid reliability issues caused by removing operator branches. To isolate the fusion design, we hold MVD and MMI fixed and replace Kalman-MoE with Concat+MLP, Attention, or Sparse-MoE. At horizon 96, these replacements increase the five-dataset mean MSE by 1.58%, 3.33%, and 1.75%, respectively. Parameter counts and individual exceptions are reported in Appendix C.3. These results support the value of the proposed fusion within the evaluated settings. Furthermore, the detailed hyperparameter configuration are reported in Appendix C.4.

Table 4: Model efficiency analysis on Weather.

Metric	MDK-MoE	K^2VAE	AMD	FilterTS	WFTNet	iTransformer
GPU Memory(MB)	556	736	734	526	5222	750
Training time(ms/iter)	39.65	18.27	183.01	31.43	197.19	18.77
MSE	0.144	0.167	0.145	0.147	0.162	0.174

4.4 Model Efficiency

We evaluate model efficiency from MSE, training time, and GPU memory. As shown in Tab. 4, we compare the efficiency of various models on the Weather dataset, where the input length and prediction horizon are set to 512 and 96, respectively, and the batch size is fixed at 16. Among the baselines, AMD, whose performance is closest to that of **MDK-MoE**, incurs a significantly higher computational cost due to its complex multi-scale decomposition mechanism. K^2VAE achieves the fastest inference speed owing to its generative probabilistic forecasting model; however, this efficiency comes at the cost of inferior predictive performance. Moreover, both **MDK-MoE** and FilterTS exhibit relatively low GPU usage, while **MDK-MoE** achieves a 2.04% reduction in MSE compared to FilterTS. These results demonstrate that **MDK-MoE** strikes an effective balance between performance and computational efficiency. More detailed analysis are reported in Appendix C.5.

Table 5: Robustness analysis on Weather.

Model	MDK-MoE		K^2VAE		AMD		FilterTS	
Metric	MSE	↑(%)	MSE	↑(%)	MSE	↑(%)	MSE	↑(%)
No_noise	0.155	-	1.67	-	0.164	-	0.161	-
Gaussian_noise(5%)	0.156	**0.64%**	0.171	2.39%	0.182	10.98%	0.172	6.83%
Gaussian_noise(10%)	0.163	**5.16%**	0.184	10.17%	0.188	14.63%	0.182	13.04%

4.5 Robustness Analysis

We inject random Gaussian noise with intensities of 5% and 10% into all samples of the Weather dataset and assess the resulting performance degradation across different models. As reported in Tab. 5, both the input and prediction horizon are fixed at 96 for all models. Under noisy conditions, AMD exhibits the most pronounced performance degradation, with MSE increases of 10.98% and 14.63% at 5% and 10% noise levels, respectively, indicating that its time-domain multi-scale decomposition mechanism is particularly sensitive to noise. FilterTS, which uses frequency-domain enhancement, also suffers notable degradation, with MSE increases of 6.83% and 13.04%, respectively. In contrast, K^2VAE demonstrates moderate noise robustness, benefiting from Koopman operator-based decomposition and Kalman filtering. Notably, **MDK-MoE** exhibits the smallest

performance degradation, with MSE increases of only 0.64% and 5.16%, respectively. These results clearly demonstrate the robustness of **MDK-MoE**, which stems from its complementary multi-view decomposition and the effective noise-mitigation capability of the **Kalman MoE** mechanism.

4.6 Biases Analysis

Bias is regarded as a structural and transferable system-level deviation. We quantitatively analyze predictive bias using *Mean Bias* and *Drift Error*. Specifically, *Mean Bias* measures the deviation of the predictive mean and indicates whether the model systematically overestimates or underestimates the target. *Drift Error* captures long-term trend deviations and evaluates whether predictions exhibit systematic temporal drift. On the ETTh1 dataset, the *Mean Bias* values obtained using FFT, MA, wavelet transform, and the proposed **MDK-MoE** are {0.1293, -0.1507, -0.1084, -0.1047}, respectively, demonstrating that Kalman MoE effectively reduces systematic mean bias. The *Drift Errors* are {835188, 957222, 721999, 695310}, respectively, further confirming the superior robustness of **MDK-MoE** in long-term forecasting. To further analyze the differences among decomposition operators and **MDK-MoE** in the representation space, we employ T-SNE to project the latent state representations produced during the testing phase into 3D space. As shown in Fig. 3, points with similar colors correspond to temporally adjacent test samples. We visualize all test samples from the ETTh1. Compared with models constructed using a single linear operator (FFT, MA, or Wavelet), the proposed **MDK-MoE** exhibits a markedly more continuous and smooth distribution. In contrast, single-operator methods display clear discontinuous regions, highlighted in red. More detailed bias analysis is provided in Appendix C.6.

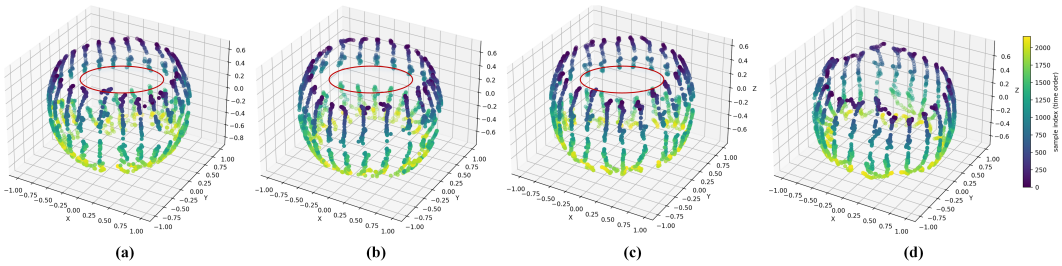


Figure 3: T-SNE on the ETTh1. (a)–(d) donate FFT, MA, Wavelet, and Kalman MoE, respectively.

5 Conclusion

In this paper, we propose a novel **MDK-MoE** framework for **NTSF**, and addresses two fundamental challenges: *the inherent distributional bias of single linear operators and the uncertainty of multi-source information fusion*. By integrating a **MVD** with a **MMI** mechanism, we theoretically and empirically demonstrated that cross-view consistency can effectively mitigate operator-specific biases and idiosyncratic noise. Furthermore, we reformulated the fusion process through a **Kalman MoE**, proving that a learnable gain mechanism provides a numerically stable and statistically optimal estimation of the underlying state in high-dimensional settings. The consistent performance gains across 13 datasets confirm that harmonizing multi-view complementary biased decomposition operators through rigorous information-theoretic and state-estimation principles effectively addresses the complexities of non-stationary dynamics. Future work will explore the extension of this **MDK-MoE** to pre-trained or multi-modal time-series tasks.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. U22B2061), the National Key Research and Development Program of China (Grant No. 2022YFB4300603), Natural Science Foundation of Sichuan, China (Grant No. 2026NSFSC0429).

References

[1] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[3] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Transactions on Machine Learning Research*, 2024.

[4] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529, 2023.

[5] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. In *International Conference on Learning Representations*, 2023.

[6] Saeed Golestan, Malek Ramezani, Josep M Guerrero, Francisco D Freijedo, and Mohammad Monfared. Moving average filter based phase-locked loops: Performance analysis and design guidelines. *IEEE transactions on power electronics*, 29(6):2750–2763, 2013.

[7] Alexander Grossmann and Jean Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736, 1984.

[8] Rui Hou, Qiao Liu, Peng He, Yao Liu, Yanwen Huang, Jun Xie, Yang Xie, and Tingting Dai. Dpem: Dual-perspective enhanced mamba for multivariate time series forecasting. *Information Fusion*, 123:103250, 2025.

[9] Shizhe Hu, Zhengzheng Lou, Xiaoqiang Yan, and Yangdong Ye. A survey on information bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5325–5344, 2024.

[10] Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17359–17367, 2025.

[11] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[12] Shiraz Khan, Raj Deshmukh, and Inseok Hwang. Optimal kalman filter with information-weighted consensus. *IEEE Transactions on Automatic Control*, 68(9):5624–5629, 2022.

[13] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *10th International Conference on Learning Representations, ICLR 2022*. International Conference on Learning Representations, ICLR, 2022.

[14] Zhe Li, Xuanxuan Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

[15] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[16] Peiyuan Liu, Beiliang Wu, Naiqi Li, Tao Dai, Fengmao Lei, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Wftnet: Exploiting global and local periodicity in long-term time series forecasting. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5960–5964. IEEE, 2024.

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

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

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

[20] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36:14273–14292, 2023.

[21] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 2002.

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

[23] Eduardo Ogasawara, Leonardo C Martinez, Daniel De Oliveira, Geraldo Zimbrão, Gisele L Pappa, and Marta Mattoso. Adaptive normalization: A novel data normalization approach for non-stationary time series. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[24] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[25] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

[26] Warren Milton Persons. Indices of business conditions: an index of general business conditions. 1919.

[27] Manel Rhif, Ali Ben Abbes, Imed Riadh Farah, Beatriz Martínez, and Yanfang Sang. Wavelet transform application for/in non-stationary time-series analysis: A review. *Applied sciences*, 9 (7):1345, 2019.

[28] Abraham Wald. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601, 1949.

[29] Jingyuan Wang, Chen Yang, Xiaohan Jiang, and Junjie Wu. When: A wavelet-dtw hybrid attention network for heterogeneous time series analysis. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2361–2373, 2023.

[30] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[31] Yulong Wang, Yushuo Liu, Xiaoyi Duan, and Kai Wang. Filterts: Comprehensive frequency filtering for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21375–21383, 2025.

[32] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[33] Herman Wold. *A study in the analysis of stationary time series*. PhD thesis, Almqvist & Wiksell, 1938.

[34] 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.

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

[36] Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. k^2 vae: A koopman-kalman enhanced variational autoencoder for probabilistic time series forecasting. *Forty-second International Conference on Machine Learning*, 2025.

[37] Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37:31350–31379, 2024.

[38] 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.

[39] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2141–2149, 2017.

[40] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in neural information processing systems*, 35:3988–4003, 2022.

[41] 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.

[42] Zichi Zhang, Tuan Dung Pham, Yimeng An, Ngoc Phu Doan, Majed Alsharari, Viet-Hung Tran, Anh-Tuan Hoang, Hans Vandierendonck, and Son T Mai. Waveletmixer: a multi-resolution wavelets based mlp-mixer for multivariate long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22741–22749, 2025.

[43] 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 Empirical Study

A.1 Analysis of inherent biases in different linear decomposition operators

To intuitively illustrate the differences among the high and low frequency components obtained by the 3 linear operators adopted in this work — MA, FFT, and wavelet (whose linearity proofs are provided Appendix B.4) — as well as the inherent distributional bias between the decomposed components and the original signal, we conduct empirical analyses on two benchmark datasets: ETTh1 and Exchange. All experiments are performed directly on the raw values of selected variables without additional preprocessing. Specifically, Fig. 4-6 present the high and low frequency components decomposed from Variable 'HULL' of the ETTh1 dataset using MA, FFT, and wavelet, respectively.

Empirical Study 1: Differences across decomposition operators. This experiment highlights the intrinsic differences between the high and low frequency components obtained using MA, FFT, and wavelet transforms.

(1) FFT is an efficient implementation of the discrete Fourier transform (DFT), which projects a time-domain signal onto a set of orthogonal complex exponential basis functions, each corresponding to a fixed frequency. We partition the spectrum into low- and high-frequency components using a 1:9 threshold ratio. The low-frequency component consists of sinusoidal components with lower frequency indices and represents globally smooth oscillatory patterns (Fig. 4, middle subplots of the upper panels). The high-frequency component is formed by summing higher-frequency sinusoidal components, capturing global rapid oscillations (Fig. 4, right subplots of the upper panels).

(2) MA is essentially a time-domain low-pass filter that suppresses rapid local fluctuations by averaging neighboring observations, thereby preserving slowly varying trends. We set the sliding window size to 15 with a stride of 1. The resulting low-frequency component captures the smoothed local trend within each window, as illustrated in the middle subplots of the upper panels in Fig. 5. The high-frequency component is computed as the residual between the original signal and the smoothed signal, representing local fluctuations, as shown in the right subplots of the upper panels.

(3) Wavelet transform performs joint time–frequency decomposition by scaling and translating a scaling function (low-frequency basis) and a wavelet function (high-frequency basis). The signal is first decomposed into approximation (low-frequency) and detail (high-frequency) components, and the approximation component is recursively decomposed to obtain multi-scale representations. We set the initial scale to 24 time steps and the central frequency of the wavelet basis to 0.04 Hz. The low-frequency components exhibit low temporal resolution but high frequency resolution, emphasizing global trends (Fig. 6, middle subplots of the upper panels), whereas the high-frequency components provide high temporal resolution and highlight local variations (Fig. 6, right subplots of the upper panels).

These results demonstrate that different operators capture complementary high–low frequency characteristics, motivating their joint exploitation for improved forecasting.

Empirical Study 2: Distributional bias between decomposed components and original signals. This experiment examines the inherent distributional bias between the decomposed components and the original data. As shown in the middle subplots of the lower panels in Fig.4–6, the low-frequency components obtained using MA, FFT, and wavelet transforms exhibit Gaussian means $\mu = 2.2242,\ 2.2242$ and 2.3180 that are equal to or close to the mean $\mu = 2.2242$ of the original signal, indicating that low-frequency components preserve stable long-term characteristics. However, the standard deviations $\sigma = 1.6881,\ 1.7672$ and 3.0848 vary across decomposition operators, leading to intrinsic distributional discrepancies among the low-frequency components. Similarly, as shown in the right subplots of the lower panels in Fig.4–6, the high-frequency components produced by all three operators have means μ close to zero, reflecting their oscillatory nature. Nevertheless, differences in standard deviation $\sigma = 1.1493,\ 0.9062$ and 1.7769 again induce inherent distributional bias across high-frequency components.

These observations indicate that, while linear decomposition operators preserve certain statistical properties, they inevitably introduce operator-dependent distributional biases. This insight motivates the need for multi-view fusion and robust state estimation mechanisms in non-stationary time series forecasting. In particular, it provides the rationale for constructing a Kalman filtering framework to learn more accurate state estimates by jointly integrating multiple biased linear systems.

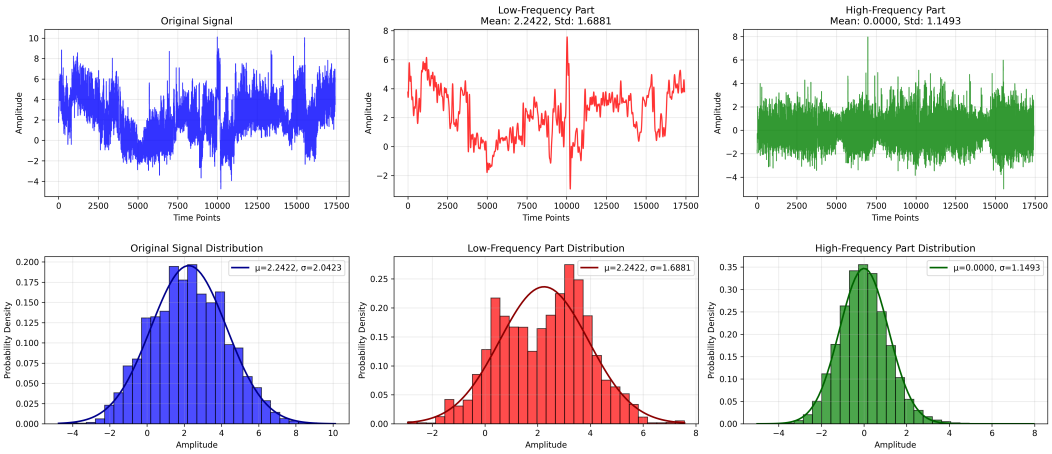


Figure 4: Decomposition results on the Variable 'HULL' of the ETTh1 dataset using the FFT operator. The upper row illustrates the original signal, low-frequency component, and high-frequency component. The lower row illustrates the corresponding Gaussian distributions of the original signal, low-frequency component, and high-frequency component.

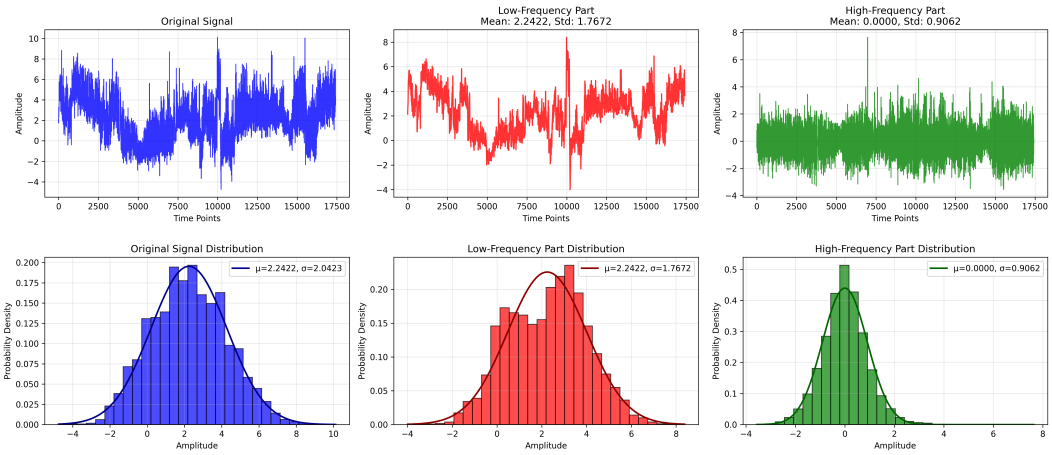


Figure 5: Decomposition results on the Variable "HULL" of the ETTh1 dataset using the MA operator. The upper row illustrates the original signal, low-frequency component, and high-frequency component. The lower row illustrates the corresponding Gaussian distributions of the original signal, low-frequency component, and high-frequency component.

A.2 Quantitative analysis of complementarity among different linear decomposition operators

To rigorously prove that the representations produced by the FFT, Wavelet, and MA operators are truly complementary rather than merely redundant, we conducted a comprehensive Information-Theoretic analysis based on Mutual Information (MI) across all datasets. Given the decomposed historical components (denoted as C) and the future prediction window Y (horizon=96), we quantified the representation properties from three dimensions: Relevance $MI(C;Y)$: The amount of predictive information a single component contains about the future. Redundancy $MI(C_1, C_2)$: The information overlap between components from two different operators. Joint MI or Complementarity $MI(C_1, C_2; Y)$: The combined information two components jointly provide about the future Y. The fundamental condition for Complementarity is that the Joint MI must strictly exceed the maximum individual MI: $MI(C_1, C_2; Y) > \max(MI(C_1; Y), MI(C_2; Y))$. This inequality guarantees that combining two views yields incremental predictive power (i.e., Conditional MI $MI(C_1; Y|C_2) > 0$) that a single operator lacks. We computed these metrics for all pairs across two datasets. Due to

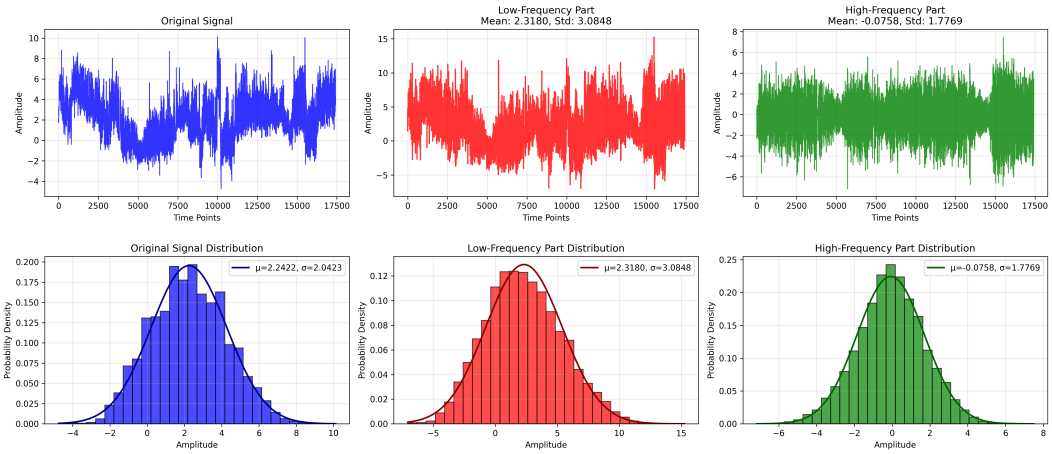


Figure 6: Decomposition results on the Variable "HULL" of the ETTh1 dataset using the wavelet operator. The upper row illustrates the original signal, low-frequency component, and high-frequency component. The lower row illustrates the corresponding Gaussian distributions of the original signal, low-frequency component, and high-frequency component.

space constraints, we report the representative results for the high-frequency component pairs and low-frequency component pairs on two datasets in the Tab. 6.

Table 6: Information theoretic quantitative analysis of multi-view complementarity

Dataset	Frequency	Pair (C_1, C_2)	Joint MI $(C_1, C_2; Y)$	Max Single MI	Complementary?
ETTh1	High-Freq	FFT & Wavelet	0.235	0.117	Yes
ETTh1	High-Freq	FFT & MA	0.321	0.149	Yes
ETTh1	High-Freq	MA & Wavelet	0.267	0.164	Yes
ETTh1	Low-Freq	FFT & Wavelet	1.990	1.012	Yes
ETTh1	Low-Freq	FFT & MA	2.052	1.047	Yes
ETTh1	Low-Freq	MA & Wavelet	1.983	1.006	Yes
Traffic	High-Freq	FFT & Wavelet	0.069	0.023	Yes
Traffic	High-Freq	FFT & MA	0.166	0.134	Yes
Traffic	High-Freq	MA & Wavelet	0.204	0.122	Yes
Traffic	Low-Freq	FFT & Wavelet	1.134	0.601	Yes
Traffic	Low-Freq	FFT & MA	1.236	0.604	Yes
Traffic	Low-Freq	MA & Wavelet	1.093	0.608	Yes

Across both High-Frequency and Low-Frequency components, the Joint MI is consistently and significantly larger than the Max Single MI, which provides direct motivation for us to construct a mutual information maximization learning mechanism. While our model maximizes inter-view consistency to suppress specific noise, the geometric priors of FFT (global periodicity), Wavelet (time-frequency localization), and MA (local trend) ensure the extraction of genuinely distinct and complementary predictive signals.

B Theoretical Analyses

B.1 Bias reduction via multi-view mutual information maximization

Multi-view feature decomposition across the time, frequency, and wavelet domains tends to introduce view-specific noise. According to the multi-view information bottleneck principle [5, 9], by maximizing the mutual information between high and low frequency components, we ensure that the model captures cross-view shared features, thereby mitigating learning biases.

Definition 1. Mutual information measures the dependency between two random variables. For any two sets of feature representations Z_1 and Z_2 , their mutual information is defined as:

$$I(Z_1; Z_2) = \mathbb{E}_{p(z_1,z_2)} \left[\log \frac{p(z_1, z_2)}{p(z_1)p(z_2)} \right]. \tag{18}$$

Following the noise-contrastive estimation framework [24]. We construct one positive pair $(z_{1,i}, z_{2,i})$ from the joint distribution $p(z_{1,i}, z_{2,i})$ and $C-1$ negative samples $(z_{1,i}, z_{2,j})$ drawn independently from the product of marginals $p(z_{1,i})p(z_{2,j})$. Let $f(\cdot, \cdot)$ be a scoring function satisfying $f(z_{1,i}, z_{2,i}) \propto p(z_{2,i} \mid z_{1,i})$. The InfoNCE loss is then defined as:

$$\mathcal{L}_{\text{mi}} = -\mathbb{E} \left[\log \frac{f(z_{1,i}, z_{2,i})}{\sum_{j=1}^{C} f(z_{1,i}, z_{2,j})} \right]. \tag{19}$$

Next, the InfoNCE loss yields a variational lower bound of mutual information:

$$I(Z_1; Z_2) \geq \log C - \mathcal{L}_{\text{mi}}. \tag{20}$$

where the term $\log C$ corresponds to the prior uncertainty of selecting the positive sample from the C candidates.

Theorem 1. *Maximizing the mutual information $I(Z_1; Z_2)$ between different views minimizes the upper bound of the view-specific bias. Under the multi-view consensus principle, the task-relevant information $I(Z; Y)$ is maximized when the redundant view-specific noise is discarded through cross-view alignment.*

Proof. Let Y be the prediction ground truth, and Z_1, Z_2 be two views generated from the underlying stable signal S. According to information theory, the joint mutual information can be decomposed using the Interaction Information $I(Z_1; Z_2; Y)$:

$$I(Z_1, Z_2; Y) = I(Z_1; Y) + I(Z_2; Y) - I(Z_1; Z_2; Y). \tag{21}$$

By maximizing the variational lower bound of $I(Z_1; Z_2)$ via InfoNCE $\mathcal{L}_{\text{mi}}$, we effectively maximize the shared information $I(Z_1; Z_2)$. According to the multi-view information bottleneck principle [5, 9], this shared component is more robust to view-specific perturbations ϵ_i.

Specifically, let $Z_i = S + \epsilon_i$, where S is the stable signal and ϵ_i is view-specific noise. Since $\epsilon_1 \perp \epsilon_2$, we have:

$$I(Z_1; Z_2) = I(S + \epsilon_1; S + \epsilon_2) \leq I(S; S). \tag{22}$$

Minimizing $\mathcal{L}_{\text{mi}}$ forces the encoder to suppress ϵ_i to increase the correlation between views, thereby reducing the learning bias introduced by single-view noise. This provides a tighter bound for the contrastive loss $\mathcal{L}_{\text{mi}}$ as $C \to \infty$.

B.2 Optimality of MoE-based adaptive observation fusion

Since the observation features consist of six features obtained through multi-view decomposition, the Kalman filtering [32] must support multi-source observation fusion. Therefore, we design an Kalman MoE adaptive observation fusion mechanism, in which the mixture weights dynamically generated from the input state features serve as an equivalent adaptive weighting of the observation noise variances. To highlight the advantages of Kalman MoE, we further prove that the MoE weighting mechanism is not merely an empirically heuristic method, but also mathematically equivalent to finding the Best Linear Unbiased Estimator (BLUE) [11, 12] for multi-source Gaussian observations.

Theorem 2. *The MoE-based fusion mechanism $\sum w_i Z_i$ yields the best linear unbiased estimator (BLUE) for the observation state. The learnable weights w_i converge to the precision-proportional distribution $w_i \propto 1/\sigma_i^2$, which minimizes the fused observation error covariance.*

Proof. Consider n independent multi-view observations $Z_i = x + v_i$ for the same underlying state x, where each view is corrupted by independent Gaussian noise $v_i \sim \mathcal{N}(0, \sigma_i^2)$. The joint likelihood function for the set $\{Z_1, \ldots, Z_n\}$ is given by:

$$P(Z_1, \ldots, Z_n | x) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(Z_i - x)^2}{2\sigma_i^2} \right). \tag{23}$$

To derive the Maximum Likelihood Estimate (MLE) [28] of x, we maximize the log-likelihood $\ell(x) = \sum_{i=1}^{n}[-\frac{1}{2}\ln(2\pi\sigma_i^2) - \frac{(Z_i-x)^2}{2\sigma_i^2}]$. Setting the derivative $\frac{d\ell(x)}{dx} = \sum \frac{Z_i-x}{\sigma_i^2} = 0$ yields:

$$\hat{x}_{\text{MLE}} = \frac{\sum_{i=1}^{n}(1/\sigma_i^2)Z_i}{\sum_{j=1}^{n}(1/\sigma_j^2)} = \sum_{i=1}^{n} w_i^* Z_i, \quad \text{where } w_i^* = \frac{1/\sigma_i^2}{\sum 1/\sigma_j^2}. \tag{24}$$

In **MDK-MoE**, the MoE gating network $g(Z_i)$ generates weights w_i via a Softmax operation. Assuming the network has sufficient capacity and is optimized to the global minimum of the negative log-likelihood, the parameterized weights w_i will adaptively converge to the theoretical optimal relative precision $\omega_i \propto 1/\sigma_i^2$. Furthermore, the fused precision follows the additive property: $\frac{1}{\sigma_{\text{fuse}}^2} = \sum_{i=1}^{n}\frac{1}{\sigma_i^2}$. Since $\sigma_i^2 > 0$, it follows that $\sigma_{\text{fuse}}^2 < \sigma_i^2$ for all i. This proves that the MoE-fused observation Z_{obs} inherently possesses higher certainty and lower noise variance than any individual view.

B.3 Efficiency and robustness of learnable kalman gain

To demonstrate that in high-dimensional, non-stationary scenarios, directly learning K^* via backpropagation is superior to explicit covariance computation in terms of sample complexity and numerical stability.

Theorem 3. *The end-to-end learnable gain K^* provides a robust approximation of the data distribution by sidestepping the $O(d^3)$ matrix inversion instability.*

proof. Standard Kalman filtering computes $K_t = P_t H^\top (H P_t H^\top + R)^{-1}$. In high-dimensional multi-view features (e.g., FFT and Wavelet), the innovation covariance matrix $A = H P_t H^\top + R$ often becomes highly ill-conditioned due to feature collinearity, meaning its condition number $\kappa(A) = \lambda_{\max}/\lambda_{\min}$ is extremely large.

According to matrix perturbation theory, if A is subject to a perturbation ΔA (e.g., floating-point errors or estimation noise), the relative error of the inversion is bounded by:

$$\frac{|(A+\Delta A)^{-1} - A^{-1}|}{|(A+\Delta A)^{-1}|} \leq \kappa(A)\frac{|\Delta A|}{|A|}. \tag{25}$$

As dimension d increases, $\kappa(A)$ grows exponentially, causing the explicit $O(d^3)$ computation of K_t to exhibit severe numerical instability. Directly learning K^* implicitly regularizes the inversion process via gradient descent, bypassing this numerical bottleneck.

Furthermore, we prove that the learned gain analytically converges to the optimal steady-state gain. Let y_t^* be the true underlying signal, and $z_t = y_t^* + v_t$ be the noisy observation with noise variance $\mathbb{E}[v_t^2] = r_t$. The prediction is updated via the learnable gain K:

$$\hat{y}_t = (1-K)\hat{y}_{t|t-1} + K z_t. \tag{26}$$

The objective of the neural network is to minimize the empirical MSE loss $\mathcal{L} = \mathbb{E}[\|y_t^* - \hat{y}_t\|^2]$. Substituting the update rule into the loss:

$$y_t^* - \hat{y}_t = y_t^* - (1-K)\hat{y}_{t|t-1} - K(y_t^* + v_t) = (1-K)(y_t^* - \hat{y}_{t|t-1}) - K v_t. \tag{27}$$

Let $e_{t|t-1} = y_t^* - \hat{y}_{t|t-1}$ be the prior estimation error with variance $\mathbb{E}[e_{t|t-1}^2] = p_t$. Since the prior error and the current observation noise are independent ($\mathbb{E}[e_{t|t-1}v_t] = 0$), the expected loss becomes:

$$\mathbb{E}[\mathcal{L}] = (1-K)^2 p_t + K^2 r_t. \tag{28}$$

Setting the derivative with respect to K to zero yields:

$$\frac{\partial \mathbb{E}[\mathcal{L}]}{\partial K} = -2(1-K)p_t + 2K r_t = 0. \tag{29}$$

Solving for K, we obtain the optimal learned parameter:

$$K_{learned} = \frac{p_t}{p_t + r_t}. \tag{30}$$

Over the entire dataset, SGD ensures K converges to the statistical expectation $K_{learned} = \frac{\bar{p}}{\bar{p}+\bar{r}}$, which exactly matches the decoupled steady-state optimal Kalman gain $K^\star$.

In non-stationary environments, explicit K_t propagates local transient noise directly into the prediction. In contrast, the backpropagation-updated K^* acts as a temporal averaging filter, distilling long-term system dynamics and inherently achieving lower generalization error.

B.4 Linearity proof for MA, FFT and Wavelet

Lemma B.1. An operator T is called a linear operator if and only if, for any two signals $x_1(t)$, $x_2(t)$ and for any scalars $a, b \in \mathbb{R}$, the following properties hold:

$$\textbf{Additivity:} \quad T\left[x_1(t) + x_2(t)\right] = T\left[x_1(t)\right] + T\left[x_2(t)\right], \tag{31}$$

$$\textbf{Homogeneity:} \quad T\left[a \cdot x(t)\right] = a \cdot T\left[x(t)\right]. \tag{32}$$

Based on Lemma B.1, the MA, FFT, and wavelet transform operators are linear operators, provided that they satisfy the additivity and homogeneity properties. The proof is given as follows.

proof. (1) The discrete MA of a signal $x(n)$ over N time points is defined as:

$$MA[x(n)] = \frac{1}{N}\sum_{k=0}^{N-1} x(n-k) \tag{33}$$

where $x(n)$ denotes the input discrete signal and n is the time index.

Additivity:

$$MA[x_1(n) + x_2(n)] = \frac{1}{N}\sum_{k=0}^{N-1}[x_1(n-k) + x_2(n-k)] = MA[x_1(n)] + MA[x_2(n)] \tag{34}$$

Homogeneity:

$$MA[a \cdot x(n)] = \frac{1}{N}\sum_{k=0}^{N-1} a \cdot x(n-k) = a \cdot \frac{1}{N}\sum_{k=0}^{N-1} x(n-k) = a \cdot MA[x(n)] \tag{35}$$

proof. (2) Since FFT is a fast algorithm for computing the discrete Fourier transform (DFT), it suffices to prove the linearity of the DFT. The DFT of a discrete signal $x(n)$ of length N is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}, \; k = 0, 1, \ldots, N-1 \tag{36}$$

where, $j = \sqrt{-1}$ and k denotes the frequency index.

Additivity:

$$\begin{aligned} DFT[x_1(n) + x_2(n)] &= \sum_{n=0}^{N-1}[x_1(n) + x_2(n)] e^{-j\frac{2\pi}{N}kn} \\ &= \sum_{n=0}^{N-1} x_1(n) e^{-j\frac{2\pi}{N}kn} + \sum_{n=0}^{N-1} x_2(n) e^{-j\frac{2\pi}{N}kn} = X_1(k) + X_2(k) \end{aligned} \tag{37}$$

Homogeneity:

$$DFT[a \cdot x(n)] = \sum_{n=0}^{N-1} a \cdot x(n) e^{-j\frac{2\pi}{N}kn} = a \cdot \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} = a \cdot X(k) \tag{38}$$

proof. (3) Taking the discrete wavelet transform as an example, a signal is decomposed using translated and scaled versions of a scaling function $\phi(t)$ (low-frequency component) and a wavelet

function $\psi(t)$ (high-frequency component). The approximation coefficients A_j and detail coefficients D_j are given by:

$$A_j[k] = \sum_n x(n)\phi_{j,k}(n) \tag{39}$$

$$D_j[k] = \sum_n x(n)\psi_{j,k}(n) \tag{40}$$

where $\phi_{j,k}(n) = 2^{-j/2}\phi(2^{-j}n - k)$, j denotes the scale, and k the translation index.

Additivity:

$$\begin{aligned} A_j[x_1 + x_2] &= \sum_n [x_1(n) + x_2(n)]\phi_{j,k}(n) \\ &= \sum_n x_1(n)\phi_{j,k}(n) + \sum_n x_2(n)\phi_{j,k}(n) = A_j[x_1] + A_j[x_2] \end{aligned} \tag{41}$$

and the same holds for the detail coefficients D_j.

Homogeneity:

$$A_j[a \cdot x] = \sum_n a \cdot x(n)\phi_{j,k}(n)A = a \cdot \sum_n x(n)\phi_{j,k}(n) = a \cdot A_j[x] \tag{42}$$

and the same holds for the detail coefficients D_j.

Since the MA, FFT, and wavelet transform operators all satisfy both additivity and homogeneity, they are linear operators.

C Experiments

C.1 Experiments Setup

Datasets. As shown in Tab. 7, we extensively include 13 real-world datasets [2] in our experiments, including: (1) The Traffic collects hourly road occupancy rates measured by 862 sensors of San Francisco Bay area freeways from January 2015 to December 2016. (2) The Exchange dataset records the daily exchange rates for 8 countries from 1990 to 2010. (3) The Solar-Energy (Solar) dataset records the solar power generation data from 137 photovoltaic stations in 2006, with measurements taken every 10 minutes. (4) The Weather dataset includes 21 meteorological factors collected every 10 minutes from 1990 to 2016. (5) The ETT dataset (comprising four subsets: h1, h2, m1, m2) contains data on 7 factors of electricity transformers from 2016 to 2018, where ETTh1 and ETTh2 are used for 1-hour predictions, while ETTm1 and ETTm2 are used for 15-minute predictions. (6) The ECL records the hourly electricity consumption data of 321 clients. (7) PEMS (PEMS03, PEMS04, PEMS07, PEMS08) contains the public traffic network data in California collected by 5-minute windows. Additionally, for long-term forecasting, we rerun all baselines with different input lengths L and choose the best results to avoid underestimating the baselines. The historical window length L for datasets (ETT, Weather, ECL, Solar, Exchange, and Traffic) was searched among 96, 192, 336, 512, 672, 720, and the prediction window T lengths covered $\{96, 192, 336, 720\}$. For short-term forecasting task on PEMS (PEMS03, PEMS04, PEMS07, PEMS08), the input sequence length L is 96 and the prediction length T is 12.

Experimental Setups. Our experiments were conducted on an NVIDIA A100 80GB GPU. To ensure fair comparisons, for long-term forecasting, we rerun all baselines with different input lengths $L \in \{96, 192, 336, 512, 672, 720\}$ and choose the best results to avoid underestimating the baselines. For short-term forecasting, the input length is 96. The initial learning rate is set to 0.001, the batch size is set to 16, and the number of training epochs is set to 15. the data is divided into training, validation, and test sets with a ratio of 7:1:2. We select two common metrics in time series forecasting: Mean Absolute Error (MAE), Mean Squared Error (MSE).

$$\textbf{MAE} = \frac{1}{m}\sum_{t=1}^{m} |\hat{y}_{test} - y_{test}| \tag{43}$$

[2] https://github.com/thuml/Time-Series-Library.

Table 7: Detailed dataset descriptions. *Dim* denotes the variate number of each dataset. *Dataset Size* denotes the total number of time points in (Train, Validation, Test) split respectively. *Prediction Length* denotes the future time points to be predicted and four prediction settings are included in each dataset. *Frequency* denotes the sampling interval of time points. *Information* denotes the domain of datasets.

Dataset	Dim	Dataset Size	Prediction Length	Frequency	Information
ETTh1, ETTh2	7	(8545, 2881, 2881)	(96, 192, 336, 720)	Hourly	Electricity
ETTm1, ETTm2	7	(34465, 11521, 11521)	(96, 192, 336, 720)	15min	Electricity
Exchange	8	(5120, 665, 1422)	(96, 192, 336, 720)	Daily	Economy
Weather	21	(36792, 5271, 10540)	(96, 192, 336, 720)	10min	Weather
Solar-Energy	137	(36601, 5161, 10417)	(34465, 11521, 11521)	10min	Energy
ECL	321	(18317, 2633, 5261)	(96, 192, 336, 720)	Hourly	Electricity
Traffic	862	(12185, 1757, 3509)	(96, 192, 336, 720)	Hourly	Transportation
PEMS03	358	(15617, 5135, 5135)	12	5min	Transportation
PEMS04	307	(10172, 3375, 3375)	12	5min	Transportation
PEMS07	883	(16911, 5622, 5622)	12	5min	Transportation
PEMS08	170	(10690, 3548, 3548)	12	5min	Transportation

$$\textbf{MSE} = \frac{1}{m}\sum_{t=1}^{m}(\hat{y}_{test} - y_{test})^2 \tag{44}$$

where m represent the number of test samples, $\hat{y}_{test}$ represents the predicted value, y_{test} represents true value.

C.2 Main Results

Long-term forecasting task are shown in Tab. 8, we carefully select 9 representative models to serve as baselines. Avg denotes the average value across all prediction lengths T, with values highlighted in bold red$*$ indicating the best performance for each metric, and values marked with blue underlining representing the second-best performance for each metric, ♠ represent repeated results. The lower the MSE/MAE, the more accurate the forecast. Overall, **MDK-MoE** achieves the best performance in **61*** cases and the second-best performance in 21 cases out of 90 total evaluations, significantly outperforming the second-best method, AMD, which attains only **27** best cases. This demonstrates the strong generalization ability of **MDK-MoE** across datasets from different domains and with varying numbers of variables. Notably, **MDK-MoE** shows substantial improvements in terms of MSE. Compared with K^2VAE and DLinear, it reduces the average MSE by 24.48% and 11.59% across 9 datasets, respectively. In particular, the inferior performance of K^2VAE suggests that iterating a single Koopman operator may introduce cumulative prediction bias. In addition, models such as FilterTS, TimesNet, and FEDformer primarily emphasize partial frequency-domain representations and fail to achieve optimal performance. Although WFTNet incorporates both FFT-based frequency-domain features and wavelet-based time–frequency features, its lack of an effective feature selection and fusion mechanism leads to inferior accuracy and low computational efficiency (see Section 4.3). These results further validate the necessity and effectiveness of the proposed Kalman-based MoE for efficient feature fusion. Moreover, **MDK-MoE** consistently outperforms the advanced iTransformer.

Short-term forecasting task are shown in Tab. 2, we carefully select 8 representative models to serve as baselines. Avg denotes the average value across all datasets {PEMS03, PEMS04, PEMS07, PEMS08}. **MDK-MoE** achieves the best results on all 10 MSE and MAE metrics, demonstrating its superiority in short-term forecasting. Specifically, compared with the second-best method AMD, MDK-MoE reduces the average MSE from 0.083 to 0.067 (a 19.27% improvement) and the average MAE from 0.195 to 0.166 (a 14.87% improvement).

In summary, those experimental results demonstrate that **MDK-MoE** consistently achieves SOTA performance across both long and short term forecasting tasks, multivariate settings, and diverse application domains. Furthermore, Fig. 7 presents qualitative forecasting visualizations on the ETTh2 dataset, where both the input and prediction horizons are set to 96. The orange and blue curves denote the predicted and ground-truth values, respectively. The left, middle, and right panels correspond to **MDK-MoE**, FilterTS, and WFTNet. The red boxes highlight regions with notable performance differences, clearly illustrating the superior predictive accuracy of **MDK-MoE**.

Table 8: Long-term forecasting task.The results for MSE and MAE across all prediction lengths $\{T = 96, 192, 336, 720\}$. Avg denotes the average value, with values highlighted in bold red $*$ indicating the best performance for each metric, and values marked with blue underlining representing the second-best performance for each metric. In addition, ♠ represents a the repeated results. To the best for a fairer comparison for all baselines, the input sequence length L is searched among $\{96, 192, 336, 512, 672, 720\}$.

Models		MDK-MoE		k^2VAE♠		AMD		FilterTS♠		WFTNet♠		TimesNet		DLinear		FEDformer		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	**0.356***	**0.251***	0.575	0.305	0.366	0.259	0.370	0.267	0.590	0.315	0.593	0.321	0.410	0.282	0.576	0.359	0.395	0.268
	192	**0.373***	**0.260***	0.596	0.312	0.381	0.265	0.387	0.269	0.618	0.332	0.617	0.336	0.423	0.287	0.610	0.380	0.417	0.276
	336	**0.384***	**0.266***	0.610	0.330	0.397	0.269	0.393	0.278	0.616	0.334	0.629	0.336	0.436	0.296	0.608	0.375	0.433	0.283
	720	0.435	0.294	0.626	0.347	**0.429***	**0.292***	0.433	0.295	0.647	0.353	0.640	0.350	0.466	0.315	0.621	0.375	0.467	0.302
	Avg	**0.387***	**0.267***	0.601	0.323	0.393	0.271	0.395	0.277	0.617	0.333	0.619	0.335	0.433	0.295	0.603	0.372	0.428	0.282
ECL	96	**0.129***	0.228	0.176	0.277	**0.129***	**0.224***	**0.129***	**0.224***	0.163	0.267	0.168	0.272	0.153	0.237	0.186	0.302	0.148	0.240
	192	**0.147***	0.239	0.191	0.275	**0.147***	**0.238***	0.149	0.243	0.194	0.295	0.184	0.289	0.152	0.249	0.197	0.311	0.162	0.253
	336	0.163	0.258	0.237	0.303	**0.160***	**0.253***	0.170	0.265	0.203	0.304	0.198	0.300	0.169	0.267	0.213	0.328	0.178	0.269
	720	**0.190***	**0.284***	0.258	0.328	0.193	0.286	0.206	0.295	0.233	0.327	0.220	0.320	0.233	0.344	0.233	0.344	0.225	0.317
	Avg	**0.157***	0.252	0.215	0.296	**0.157***	**0.250***	0.163	0.257	0.198	0.298	0.192	0.295	0.176	0.274	0.207	0.321	0.178	0.269
Exchange	96	**0.081***	**0.198***	0.131	0.253	0.083	0.201	0.085	0.201	0.134	0.262	0.107	0.234	**0.081***	0.203	0.136	0.276	0.086	0.206
	192	0.170	0.295	0.264	0.366	0.171	**0.293***	0.170	0.294	0.289	0.380	0.226	0.344	**0.157***	**0.293***	0.256	0.369	0.177	0.299
	336	**0.303***	**0.398***	0.448	0.484	0.309	0.402	0.315	0.406	0.465	0.496	0.367	0.448	0.305	0.414	0.426	0.464	0.331	0.417
	720	0.725	0.637	1.316	0.838	0.750	0.652	0.828	0.685	1.224	0.808	0.964	0.746	**0.643***	**0.601***	1.090	0.800	0.847	0.691
	Avg	0.319	0.382	0.539	0.485	0.328	0.387	0.349	0.396	0.528	0.486	0.416	0.443	**0.296***	**0.377***	0.477	0.477	0.360	0.403
Solar	96	**0.170***	0.234	0.212	0.308	0.175	**0.228***	0.180	0.234	0.203	0.299	0.219	0.314	0.289	0.337	0.201	0.304	0.203	0.237
	192	0.194	0.243	0.225	0.319	**0.186***	**0.235***	0.192	0.237	0.217	0.304	0.231	0.322	0.319	0.397	0.237	0.337	0.233	0.261
	336	**0.198***	0.253	0.237	0.324	0.200	**0.246***	0.203	0.249	0.231	0.310	0.246	0.337	0.352	0.415	0.254	0.362	0.248	0.273
	720	**0.199***	0.265	0.267	0.351	0.203	**0.249***	0.221	0.254	0.257	0.340	0.280	0.363	0.356	0.412	0.280	0.397	0.249	0.275
	Avg	**0.190***	0.248	0.235	0.325	0.191	**0.239**	0.199	0.243	0.227	0.313	0.244	0.334	0.329	0.390	0.243	0.350	0.233	0.261
Weather	96	**0.144***	**0.193***	0.167	0.218	0.145	0.197	0.147	0.199	0.162	0.210	0.172	0.220	0.152	0.237	0.238	0.314	0.174	0.214
	192	**0.187***	**0.235***	0.227	0.270	**0.187***	0.238	0.192	0.244	0.211	0.254	0.219	0.261	0.220	0.282	0.275	0.329	0.221	0.254
	336	**0.240***	**0.275***	0.285	0.304	**0.240***	0.280	0.246	0.284	0.273	0.298	0.280	0.306	0.265	0.319	0.339	0.377	0.278	0.296
	720	**0.308***	**0.327***	0.359	0.357	0.315	0.330	0.330	0.338	0.349	0.347	0.365	0.359	0.323	0.362	0.389	0.409	0.358	0.349
	Avg	**0.219***	**0.257***	0.259	0.287	0.221	0.261	0.228	0.266	0.248	0.277	0.259	0.286	0.240	0.300	0.310	0.357	0.257	0.278
ETTm1	96	**0.284***	**0.337***	0.311	0.363	**0.284***	0.339	0.292	0.347	0.336	0.375	0.338	0.375	0.299	0.343	0.326	0.390	0.334	0.368
	192	0.323	**0.362***	0.392	0.397	**0.322***	**0.362***	0.345	0.377	0.373	0.396	0.371	0.387	0.335	0.365	0.365	0.415	0.377	0.391
	336	**0.359***	0.386	0.421	0.418	0.360	**0.380***	0.367	0.391	0.413	0.417	0.410	0.411	0.369	0.386	0.391	0.425	0.426	0.420
	720	0.417	**0.413***	0.519	0.464	0.421	0.416	**0.412***	0.414	0.470	0.444	0.478	0.450	0.425	0.421	0.446	0.458	0.491	0.459
	Avg	**0.345***	**0.374***	0.410	0.410	0.346	**0.374***	0.354	0.382	0.398	0.408	0.399	0.405	0.357	0.378	0.382	0.422	0.407	0.409
ETTm2	96	**0.163***	**0.253***	0.230	0.307	0.167	0.258	0.173	0.255	0.181	0.263	0.187	0.267	0.167	0.260	0.180	0.271	0.180	0.264
	192	**0.219***	**0.292***	0.251	0.314	0.221	0.294	0.225	0.298	0.247	0.306	0.249	0.309	0.224	0.303	0.252	0.318	0.250	0.309
	336	0.273	**0.326***	0.297	0.335	**0.270***	0.327	0.282	0.334	0.310	0.345	0.321	0.351	0.281	0.342	0.324	0.364	0.311	0.348
	720	**0.351***	**0.380***	0.401	0.399	0.356	0.382	0.357	0.385	0.416	0.406	0.497	0.403	0.397	0.421	0.410	0.420	0.412	0.407
	Avg	**0.251***	**0.312***	0.295	0.339	0.253	0.315	0.259	0.318	0.288	0.330	0.313	0.332	0.267	0.331	0.291	0.343	0.288	0.332
ETTh1	96	**0.366***	0.393	0.438	0.444	0.369	0.397	0.371	**0.390***	0.398	0.416	0.384	0.402	0.375	0.399	0.376	0.415	0.386	0.405
	192	**0.399***	**0.415***	0.490	0.471	0.401	0.416	0.405	0.418	0.446	0.441	0.457	0.436	0.405	0.416	0.423	0.446	0.441	0.436
	336	**0.417***	0.428	0.526	0.482	0.418	**0.427***	0.440	0.436	0.478	0.455	0.491	0.469	0.439	0.443	0.444	0.462	0.487	0.458
	720	**0.435***	0.455	0.518	0.505	0.439	**0.454***	0.457	0.468	0.478	0.481	0.521	0.500	0.472	0.490	0.469	0.492	0.503	0.491
	Avg	**0.404***	**0.422***	0.493	0.475	0.406	0.423	0.418	0.428	0.450	0.448	0.463	0.451	0.422	0.437	0.428	0.453	0.454	0.447
ETTh2	96	0.274	0.340	0.346	0.372	0.274	0.337	**0.273***	**0.336***	0.329	0.368	0.340	0.374	0.289	0.353	0.332	0.374	0.297	0.349
	192	**0.331***	0.378	0.384	0.400	0.351	0.351	0.336	**0.377***	0.367	0.410	0.402	0.414	0.383	0.418	0.407	0.446	0.380	0.400
	336	**0.359***	**0.403***	0.437	0.450	0.375	0.411	0.378	0.414	0.400	0.433	0.452	0.452	0.448	0.465	0.400	0.447	0.428	0.432
	720	**0.400***	**0.433***	0.476	0.482	0.402	0.438	0.419	0.441	0.451	0.455	0.462	0.468	0.605	0.551	0.412	0.469	0.427	0.445
	Avg	**0.341***	**0.388***	0.410	0.426	0.350	0.392	0.351	0.392	0.387	0.416	0.414	0.427	0.431	0.446	0.387	0.434	0.383	0.406
1st count		35	26	0	0	11	16	3	4	0	0	0	0	4	3	0	0	0	0

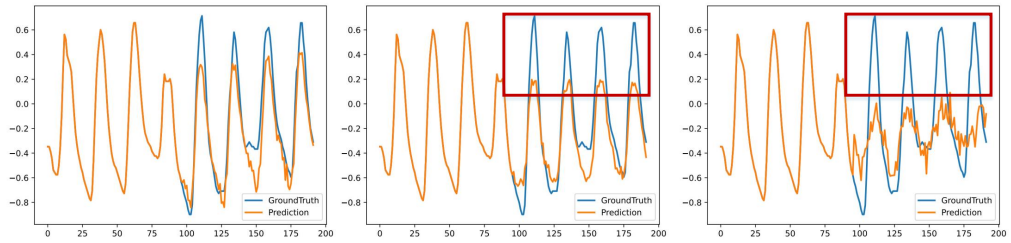


Figure 7: Visualizations on the ETTh2 dataset. The left, middle, and right panels correspond to **MDK-MoE**, FilterTS, and WFTNet. .

Furthermore, in heavily benchmarked fields like time series forecasting, rigorous verification is essential to ensure that performance gains stem from fundamental modeling capabilities rather than random variance. To address this, we present comprehensive evidence from two perspectives: first, while the absolute numerical improvement on a single dataset might seem modest, our model demonstrates overwhelming and consistent superiority across the entire benchmark suite. Out of 90 total evaluation metrics across all datasets, our proposed MDK-MoE achieved the best performance (Rank 1) 35 times and the second-best (Rank 2) 26 times. In stark contrast, the most recent strong baseline, AMD, only secured 11 first-place and 16 second-place rankings. This consistent winning rate across diverse non-stationary scenarios strongly indicates that our architectural design provides fundamentally better forecasting capability. Second, to rigorously rule out the influence of randomness and prove the stability of our model, for each dataset-horizon configuration, we evaluate MDK-MoE and AMD with matched seeds from 2024 to 2033 and report the mean and standard deviation across seeds. Paired differences are defined as the MDK-MoE error minus the AMD error. We report the dataset-level aggregate results (96,192,336,720) and corresponding p-values (each p-value corresponds to a paired comparison between MDK-MoE and AMD using matched runs and is therefore reported once per dataset and metric, bold values indicate nominal significance at $p<0.05$) below. At the nominal significance level of $p<0.05$, MDK-MoE achieves significantly lower MSE on 7/9 datasets and significantly lower MAE on 5/9 datasets. These findings support an overall, but not universal. The detailed robustness results are summarized Tab 9.

C.3 Ablation Studies

To verify the importance of the key components in our model, we conduct nine ablation studies covering all nine long-term forecasting benchmarks and all prediction horizons 96, 192, 336, 720.

Effect of removing individual operators. As shown Tab. 11, when the FFT operator is removed (w/o f), only 8 out of 90 evaluation metrics (MAE + MSE across 9 datasets) retain the best performance, corresponding to a performance degradation rate of 91.11%. Removing the MA operator (w/o m) preserves the best performance in 19 metrics, with a degradation rate of 78.89%. Removing the wavelet operator (w/o w) preserves the best performance in 13 metrics, yielding a degradation rate of 85.86%. Overall, retaining only two operators increases the average MAE and MSE by 2.21% across all datasets, with the most pronounced degradation observed on the Exchange dataset, where the error increases by 11.40%.

Effect of retaining only a single operator. When both FFT and MA are removed (w/o fm), only 8 metrics maintain the best performance, with a degradation rate of 91.11%. Removing FFT and wavelet (w/o fw) or removing wavelet and MA (w/o mw) preserves the best performance in only 10 metrics, corresponding to degradation rates of 88.89% in both cases. Overall, using a single operator for forecasting increases the average MAE and MSE by 1.33% across all datasets.

Effect of removing key modules. When all three operators are used but the mutual information maximization module is removed (w/o mi), only 10 metrics retain the best performance, with a degradation rate of 88.89%, and the average MAE and MSE increase by 2.57% across all datasets. This represents the largest performance drop among all components, highlighting the critical role of enforcing consistency between the high- and low-frequency components derived from different operators. When the Kalman MoE is removed (w/o kal), 16 metrics maintain the best performance,

Table 9: Paired statistical significance test. Each reported dataset-level test pools four horizons and ten seeds.

Dataset	Model	MSE (mean±std)	p (MSE)	MAE (mean±std)	p (MAE)
ETTh1	MDK-MoE	0.4103 ± 0.0336	0.0225	0.4262 ± 0.0276	0.0007
ETTh1	AMD	0.4129 ± 0.0314	—	0.4299 ± 0.0261	—
ETTh2	MDK-MoE	0.3482 ± 0.0491	< 0.0001	0.3931 ± 0.0378	< 0.0001
ETTh2	AMD	0.6081 ± 0.2643	—	0.5403 ± 0.1220	—
ETTm1	MDK-MoE	0.3462 ± 0.0506	< 0.0001	0.3748 ± 0.0289	< 0.0001
ETTm1	AMD	0.3514 ± 0.0508	—	0.3777 ± 0.0277	—
ETTm2	MDK-MoE	0.2542 ± 0.0706	0.7346	0.3144 ± 0.0479	0.0598
ETTm2	AMD	0.2538 ± 0.0704	—	0.3152 ± 0.0461	—
Electricity	MDK-MoE	0.1582 ± 0.0226	0.0010	0.2540 ± 0.0219	0.8105
Electricity	AMD	0.1608 ± 0.0263	—	0.2536 ± 0.0244	—
Exchange	MDK-MoE	0.4561 ± 0.3699	0.3635	0.4335 ± 0.1957	0.7704
Exchange	AMD	0.3709 ± 0.3284	—	0.4093 ± 0.2036	—
Solar	MDK-MoE	0.1925 ± 0.0105	< 0.0001	0.2489 ± 0.0119	0.9976
Solar	AMD	0.2028 ± 0.0105	—	0.2464 ± 0.0089	—
Weather	MDK-MoE	0.2192 ± 0.0622	< 0.0001	0.2581 ± 0.0504	< 0.0001
Weather	AMD	0.2245 ± 0.0626	—	0.2649 ± 0.0484	—
Traffic	MDK-MoE	0.3877 ± 0.0303	< 0.0001	0.2682 ± 0.0172	< 0.0001
Traffic	AMD	0.4105 ± 0.0213	—	0.2881 ± 0.0115	—

with a degradation rate of 82.22%, and the average MAE and MSE increase by 1.27%. When both the mutual information maximization module and the Kalman MoE are removed (w/o mi_kal), only 6 metrics retain the best performance, corresponding to a degradation rate of 93.33%, and the average MAE and MSE increase by 1.43%. Across all ablation settings and datasets, removing key components leads to an average increase of 1.63% in MAE and MSE, resulting in varying degrees of performance degradation. These results validate the effectiveness of jointly employing three complementary operators in the proposed **MDK-MoE** framework, and demonstrate that integrating mutual information maximization with the Kalman MoE mechanism yields comprehensive performance improvements.

Equal parameter ablation study. Removing one or two operators in Tab. 11 may yield counter-intuitive experimental results. After in-depth analysis, we confirm that the counter-intuitive phenomena in our operator-removal ablations (e.g., single-operator variants outperforming dual-operator variants) arise as an artifact of confounded model capacities. Specifically, removing operator branches narrows the network structure. For the single-operator variant, we discarded the MoE routing mechanism but compensated by adding an extra linear layer. This structural mismatch drastically changes the total parameter budget and optimization landscape, directly leading to the anomalous performance reversals observed.

To further validate the complementarity and necessity of our diverse operators, we perform a rigorous three-step logical analysis: First, as shown in Tab. 6, we quantitatively verify that the FFT (F), Wavelet (W), and MA (M) operators exhibit genuine information complementarity via mutual information (MI) analysis. The core criterion for complementarity is defined as: the joint mutual information of two components (C_1, C_2) with respect to the future target Y strictly exceeds their maximum individual mutual information, i.e., $MI(C_1, C_2; Y) > \max(MI(C_1; Y), MI(C_2; Y))$. Second, as reported in Tab. 10, to strictly disentangle the benefits of multi-view fusion from parameter scaling effects, we design a parameter-matched ablation experiment with fixed model capacity. We retain an architecture with exactly 3 branches (ensuring identical parameter counts and FLOPs for all variants) and only replace the operator types (e.g., FFF" denotes three repeated FFT branches; FMF" denotes two FFT branches and one MA branch).

In conclusion, when model capacity is strictly controlled, all counter-intuitive performance inconsistencies disappear entirely. Our true multi-view model (FMW) consistently outperforms both dual-operator variants (e.g., FMF) and single-operator repeated variants (e.g., FFF) across all datasets. This irrefutably demonstrates that our performance improvements originate from the genuine complementary information of multi-view fusion, rather than increased model capacity.

Table 10: Parameter matched ablation results on different datasets (prediction length = 96)

Dataset	Length	FFF(MSE)	FFF(MAE)	MMM(MSE)	MMM(MAE)	WWW(MSE)	WWW(MAE)	FMF(MSE)	FMF(MAE)	FWW(MSE)	FWW(MAE)	MMW(MSE)	MMW(MAE)
Solar	96	0.173	0.235	0.175	0.238	0.174	0.237	0.171	0.232	0.171	0.232	0.171	0.235
Exchange_rate	96	0.081	0.198	0.081	0.198	0.081	0.198	0.080	0.198	0.081	0.198	0.081	0.198
ETTm2	96	0.163	0.253	0.164	0.253	0.163	0.253	0.163	0.253	0.164	0.254	0.163	0.253
weather	96	0.144	0.194	0.143	0.193	0.144	0.193	0.143	0.193	0.143	0.193	0.143	0.193
ETTm1	96	0.283	0.336	0.282	0.336	0.283	0.337	0.283	0.336	0.283	0.336	0.282	0.336
ETTh1	96	0.367	0.392	0.367	0.393	0.367	0.392	0.368	0.394	0.366	0.393	0.367	0.393

Table 11: The input sequence length L is searched among 96, 192, 336, 512, 672, 720, and the prediction lengths are T=96, 192, 336, 720. *Avg* denotes the average value, with values highlighted in bold red indicating the best performance for each metric. *w/o f* denotes removing the FFT operator; *w/o m* denotes removing the MA operator; *w/o w* denotes removing the wavelet operator. *w/o fm* removes both FFT and MA; *w/o fw* removes both FFT and wavelet; *w/o mw* removes both wavelet and MA. *w/o mi* indicates disabling the mutual information maximization module; *w/o kal* indicates disabling the Kalman MoE; and *w/o mi_kal* disables both the mutual information maximization module and the Kalman MoE. Results highlighted in red indicate the best performance.

Models		**MDk-MoE**		**w/o f**		**w/o m**		**w/o w**		**w/o fm**		**w/o fw**		**w/o mw**		**w/o mi**		**w/o kal**		**w/o mi_kal**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	**96**	0.356	0.251	0.357	0.252	0.356	0.252	0.357	0.252	0.356	0.252	0.356	0.251	0.357	0.252	**0.352**	**0.250**	0.357	0.252	0.356	0.252
	192	0.373	0.260	0.374	0.260	0.373	0.260	0.374	0.260	0.373	0.260	0.374	0.260	0.373	0.260	**0.369**	**0.259**	0.374	0.260	0.373	0.260
	336	**0.384**	**0.266**	0.387	0.267	0.385	0.267	0.385	0.267	0.387	0.267	0.387	0.267	0.387	0.267	0.387	0.268	0.388	0.267	0.387	0.267
	720	0.435	**0.294**	0.435	**0.294**	0.435	**0.294**	0.435	**0.294**	**0.434**	0.294	**0.434**	0.294	**0.434**	**0.294**	0.450	0.309	0.435	0.295	0.434	**0.294**
	Avg	**0.387**	**0.267**	0.388	0.268	**0.387**	0.268	**0.387**	0.268	**0.387**	0.268	**0.387**	0.268	0.387	0.268	0.389	0.271	0.388	0.268	**0.387**	0.268
ECL	**96**	**0.129**	0.228	0.131	0.230	0.132	0.231	0.131	0.229	0.129	0.226	**0.129**	**0.225**	**0.129**	**0.225**	0.131	0.231	**0.129**	**0.225**	**0.129**	**0.225**
	192	**0.147**	**0.239**	0.150	0.245	0.150	0.245	0.148	0.242	**0.147**	0.240	**0.147**	**0.239**	0.148	0.242	0.148	0.242	0.148	0.242	0.148	0.242
	336	**0.163**	**0.258**	0.164	0.259	0.167	0.263	0.165	0.259	0.167	0.260	0.167	0.261	0.167	0.260	0.166	0.262	0.167	0.262	0.167	0.260
	720	**0.190**	**0.284**	0.192	0.288	0.191	0.288	0.191	0.286	0.197	0.292	0.197	0.291	0.196	0.291	0.192	0.286	0.197	0.292	0.196	0.291
	Avg	**0.157**	**0.252**	0.159	0.255	0.160	0.256	0.158	0.254	0.160	0.254	0.160	0.254	0.160	0.254	0.159	0.255	0.160	0.255	0.160	0.254
Exchange	**96**	**0.081**	0.198	**0.081**	0.198	**0.081**	0.198	**0.081**	**0.197**	**0.081**	0.198	**0.081**	0.198	**0.081**	0.198	**0.081**	0.198	**0.081**	**0.197**	**0.081**	0.198
	192	0.170	**0.295**	0.174	0.303	0.177	0.302	0.176	0.301	0.172	0.302	0.169	0.299	**0.168**	0.299	0.173	0.304	0.169	0.300	0.169	0.299
	336	**0.303**	**0.295**	0.393	0.450	0.371	0.443	0.591	0.582	0.320	0.406	0.307	0.402	0.307	0.403	0.426	0.477	0.319	0.412	0.316	0.407
	720	**0.725**	**0.637**	0.760	0.655	0.761	0.655	0.787	0.665	0.740	0.644	0.770	0.657	0.756	0.651	0.761	0.652	0.774	0.659	0.755	0.651
	Avg	**0.319**	**0.382**	0.352	0.401	0.347	0.399	0.408	0.436	0.328	0.387	0.331	0.389	0.328	0.387	0.360	0.407	0.335	0.392	0.330	0.388
Solar	**96**	**0.170**	**0.234**	0.178	0.240	0.179	0.235	0.177	0.235	0.181	0.238	0.183	0.240	0.179	0.236	0.176	0.237	0.181	0.240	0.180	0.237
	192	**0.194**	**0.243**	0.202	0.266	0.204	0.265	0.200	0.267	0.196	0.252	0.199	0.252	0.201	0.257	0.211	0.279	0.198	0.254	0.199	0.258
	336	**0.198**	0.253	0.199	0.253	0.199	**0.252**	0.199	0.253	**0.198**	0.254	0.199	0.253	**0.198**	0.253	**0.198**	0.254	**0.198**	0.253	0.199	0.253
	720	**0.199**	0.265	0.201	0.266	0.201	0.265	0.202	0.266	0.202	0.261	0.202	0.261	0.203	0.261	0.200	**0.257**	0.202	0.262	0.202	0.261
	Avg	**0.190**	**0.248**	0.195	0.256	0.195	0.254	0.194	0.255	0.194	0.251	0.195	0.251	0.195	0.251	0.196	0.256	0.194	0.252	0.195	0.252
Weather	**96**	0.144	0.193	**0.142**	0.191	0.143	0.191	**0.142**	**0.190**	0.145	0.195	0.146	0.196	0.145	0.195	0.147	0.197	0.144	0.194	0.145	0.194
	192	0.187	**0.235**	0.187	0.236	**0.186**	0.236	0.187	**0.235**	0.188	0.237	0.188	0.237	0.188	0.237	0.192	0.241	0.188	**0.235**	0.188	0.237
	336	0.240	0.275	0.238	0.274	**0.238**	**0.273**	0.239	0.274	0.240	0.275	0.241	0.275	0.241	0.276	0.244	0.279	0.240	0.274	0.242	0.276
	720	0.308	0.327	0.307	0.327	**0.306**	**0.326**	**0.306**	**0.326**	0.307	0.327	0.307	0.327	**0.306**	**0.326**	0.310	0.333	**0.306**	**0.326**	0.307	0.327
	Avg	0.219	0.257	**0.218**	0.257	**0.218**	**0.256**	**0.218**	**0.256**	0.220	0.258	0.220	0.258	0.220	0.258	0.223	0.262	0.219	0.257	0.220	0.258
ETTm1	**96**	0.284	**0.337**	0.283	**0.337**	**0.282**	**0.337**	0.283	**0.337**	0.283	**0.337**	0.283	**0.337**	0.283	0.338	0.284	0.338	**0.282**	**0.337**	0.283	0.338
	192	0.323	**0.362**	0.324	0.363	**0.322**	**0.362**	0.323	0.363	**0.322**	**0.362**	0.324	0.363	0.323	**0.362**	0.324	**0.362**	**0.322**	**0.362**	0.323	0.363
	336	0.359	0.386	**0.358**	**0.384**	**0.358**	**0.384**	0.359	0.385	0.360	0.385	0.360	0.385	0.360	0.385	0.361	0.385	0.360	0.385	0.360	0.385
	720	**0.417**	**0.413**	0.420	0.415	0.420	0.415	0.420	0.414	0.421	0.416	0.421	0.417	0.421	0.417	0.421	0.415	0.421	0.416	0.421	0.416
	Avg	**0.345**	**0.374**	0.346	**0.374**	**0.345**	**0.374**	0.346	**0.374**	0.346	0.375	0.347	0.375	0.346	0.375	0.347	0.375	0.346	0.375	0.346	0.375
ETTm2	**96**	**0.163**	**0.253**	0.164	0.254	0.164	0.253	0.164	0.254	0.165	0.256	0.165	0.254	0.164	0.253	0.164	0.254	0.164	0.254	0.164	0.254
	192	**0.219**	0.292	0.220	0.292	0.220	0.292	0.220	0.292	0.220	0.292	0.219	**0.291**	0.220	0.292	0.220	0.292	**0.219**	**0.291**	0.220	0.292
	336	**0.273**	**0.326**	0.277	0.328	0.278	0.330	0.277	0.328	0.275	0.327	0.274	0.329	0.275	0.329	0.274	**0.326**	**0.273**	**0.326**	0.274	**0.326**
	720	**0.351**	**0.380**	0.355	0.386	0.353	0.385	0.353	0.384	0.353	0.381	0.354	0.382	0.354	0.384	0.354	0.385	0.353	0.382	0.353	0.383
	Avg	**0.251**	**0.312**	0.254	0.315	0.253	0.315	0.253	0.314	0.253	0.314	0.253	0.314	0.253	0.314	0.253	0.314	0.252	0.313	0.252	0.313
ETTh1	**96**	**0.366**	0.393	0.368	0.394	0.367	0.393	0.367	0.393	0.368	0.393	0.368	**0.392**	0.369	0.393	0.368	0.394	0.367	0.393	0.368	0.393
	192	**0.399**	**0.415**	0.400	0.417	0.400	0.416	0.400	0.417	0.400	0.416	0.400	0.416	0.400	0.416	0.400	0.416	0.400	0.416	0.400	0.416
	336	**0.417**	**0.428**	0.419	0.430	0.419	0.430	0.421	0.430	0.424	0.433	0.424	0.433	0.424	0.433	0.421	0.431	0.419	0.429	0.423	0.433
	720	**0.435**	**0.455**	0.451	0.467	0.447	0.466	0.456	0.471	0.485	0.483	0.467	0.476	0.472	0.480	0.474	0.481	0.459	0.469	0.462	0.470
	Avg	**0.404**	**0.422**	0.409	0.427	0.408	0.426	0.411	0.427	0.419	0.431	0.414	0.429	0.416	0.430	0.415	0.430	0.411	0.426	0.413	0.428
ETTh2	**96**	**0.274**	**0.340**	0.280	0.342	0.280	0.345	0.281	0.346	0.279	0.345	0.279	0.346	0.280	0.345	0.275	0.342	0.278	0.342	0.277	0.343
	192	**0.331**	**0.378**	0.335	0.377	0.338	0.379	0.340	0.384	0.347	0.391	0.349	0.387	0.345	0.383	0.350	0.387	0.339	0.380	0.346	0.389
	336	**0.359**	0.403	0.370	0.411	0.365	0.403	0.370	0.412	0.363	0.403	0.376	0.406	0.374	0.407	0.362	**0.402**	0.366	0.409	0.376	0.408
	720	**0.400**	**0.433**	0.409	0.442	0.408	0.443	0.413	0.448	0.413	0.449	0.412	0.445	0.409	0.440	0.419	0.452	0.418	0.452	0.436	0.461
	Avg	**0.341**	**0.388**	0.348	0.393	0.347	0.392	0.351	0.397	0.350	0.397	0.354	0.396	0.352	0.393	0.351	0.395	0.350	0.395	0.358	0.400
1st count		**33**	**31**	**4**	**4**	**10**	**9**	**5**	**8**	**6**	**2**	**5**	**5**	**6**	**4**	**4**	**6**	**8**	**8**	**3**	**3**

Fusion isolation ablation study. We replaced Kalman-MoE with capacity-matched Concat+MLP, Attention, and Sparse-MoE fusion modules, and additionally compared it with a standard Kalman implementation. As shown in Tab. 12, all models use the same multi-view decomposition and a prediction horizon of 96, while "Out of Time" indicates that the evaluation did not complete within the allotted time. Kalman-MoE provides the best or competitive MSE/MAE trade-off across the five datasets, replacing Kalman-MoE with Concat+MLP, Attention, or Sparse-MoE increases the average MSE by 1.58%, 3.33%, and 1.75%, respectively, showing that the improvement cannot be attributed solely to the multi-view decomposition.

Table 12: Fusion replacement with the same decomposition and MMI at horizon 96.

Dataset	Kalman-MoE	Concat+MLP	Attention	Sparse-MoE	Standard Kalman
ETTh1	0.367/0.393	0.368/0.396	0.388/0.404	0.370/0.394	0.374/0.396
ETTm2	0.163/0.252	0.164/0.253	0.166/0.255	0.165/0.256	0.163/0.254
Weather	0.143/0.195	0.147/0.198	0.145/0.196	0.144/0.194	0.145/0.195
Solar	0.173/0.235	0.185/0.241	0.179/0.237	0.186/0.244	0.198/0.253
Traffic	0.356/0.251	0.357/0.251	0.364/0.260	0.358/0.253	*Timeout*

C.4 Sensitivity Analysis

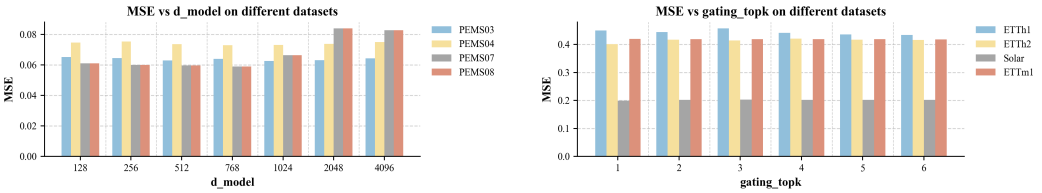
We evaluate the hyperparameter sensitivity of **MDK-MoE** with respect to 8 key factors: *seq_len* denotes the length of the historical input window T; *learning_rate* denotes the learning rate; *gating_topk* denotes the number of experts selected in the Kalman MoE mechanism; *loss_moe* corresponds to the routing balance loss weight moe λ_{moe} in Eq. (16); *d_model* denotes the embedding dimension; *batch_size* denotes the training batch size; *threshold_ratio* denotes the proportion of high-frequency components obtained by the FFT-based decomposition; and *level* denotes the sliding window length of the MA operator. To enhance the reproducibility of the experimental results reported in Tab. 8, we provide the detailed values of eight hyperparameter configurations in Tab. 13. These configurations are used for long-term forecasting with horizon L=720 and short-term forecasting with horizon L=12.

To intuitively examine the impact of key hyperparameters on model performance, Fig. 8 (a) shows that the *d_model* parameter has a significant effect on MSE/MAE performance on the PEMS03 (bule), PEMS04 (yellow), PEMS07 (gray), and PEMS08 (orange) datasets. Notably, our model does not rely on large embedding dimensions to achieve strong performance, which is an important factor contributing to its high computational efficiency. Furthermore, as shown in Fig. 8 (b), we conduct a sensitivity analysis of the *gating_topk* parameter on the ETTh1 (bule), ETTh2 (yellow), Solar (gray), and ETTm1 (orange) datasets. The performance variation with respect to *gating_topk* is relatively small. However, in conjunction with the hyperparameter settings reported in Tab. 13, setting the number of routing experts to 3 consistently yields superior performance on most datasets. This is because experts ranked higher by top-k importance weights preserve more informative features, while additional experts tend to be redundant and may introduce noise.

Table 13: The detailed values of eight hyperparameter configurations.

Metric	Dataset	seq_len	learning_rate	gating_topk	loss_moe	d_model	batch_size	threshold_ratio	level
Long-term	**ECL**	720	0.0001	3	0.05	1024	16	0.1	15
	ETTh1	96	0.001	2	1	512	8	0.1	7
	ETTh2	512	0.0005	1	1	64	128	0.1	7
	ETTm1	336	0.0001	3	10	512	128	0.1	7
	ETTm2	336	0.0001	3	10	64	128	0.1	7
	weather	720	0.0001	3	0.1	128	16	0.1	15
	Exchange	96	0.0001	3	1	32	64	0.1	15
	Traffic	720	0.0001	3	1	2048	32	0.2	15
	Solar	512	0.0005	6	0.3	512	16	0.2	15
Short-term	**PEMS03**	96	0.001	6	0.3	1024	16	0.2	15
	PEMS04	96	0.001	6	0.3	768	16	0.2	15
	PEMS07	96	0.0005	6	0.3	768	16	0.2	15
	PEMS08	96	0.001	6	0.3	512	16	0.2	15

Decomposition-parameter sensitivity. As shown in Tab. 14, we vary one decomposition parameter at a time while holding the remaining settings fixed at horizon 96. The nominal FFT ratios (corresponds



(a) Impact of *d_model* on model performance.

(b) Impact of *gating_topk* on model performance.

Figure 8: Impact of key hyperparameters on model performance.

to Eq. 5) are 0.05, 0.10, 0.20, and 0.30; wavelet kernel sizes (corresponds to Eq. 4) are 5, 7, 15, and 21; and MA windows (corresponds to Eq. 3) are 15, 21, 25, and 45. We report effective parameter values after implementation constraints and interpret the observed stability only within the evaluated ranges. The reported MSE spread is at most 0.28% for the FFT ratio, 0.63% for the wavelet kernel, and 0.21% for the MA window across the four datasets. These results concern decomposition parameters and do not imply that capacity-related choices such as d_model are equally insensitive.

Table 14: One-at-a-time decomposition sensitivity at horizon 96.

Operator	Value	ETTh1	ETTh2	Weather	Exchange
FFT ratio	0.05	0.3672/0.3935	0.2829/0.3449	0.1440/0.1949	0.0811/0.1983
FFT ratio	0.10	0.3672/0.3935	0.2829/0.3449	0.1440/0.1951	0.0811/0.1982
FFT ratio	0.20	0.3671/0.3927	0.2828/0.3444	0.1439/0.1941	0.0811/0.1982
FFT ratio	0.30	0.3666/0.3927	0.2827/0.3447	0.1436/0.1942	0.0811/0.1982
Wavelet kernel	5	0.3667/0.3938	0.2818/0.3442	0.1443/0.1934	0.0809/0.1980
Wavelet kernel	7	0.3672/0.3935	0.2829/0.3449	0.1434/0.1932	0.0809/0.1980
Wavelet kernel	15	0.3679/0.3935	0.2832/0.3442	0.1440/0.1951	0.0811/0.1982
Wavelet kernel	21	0.3674/0.3931	0.2833/0.3441	0.1437/0.1931	0.0810/0.1980
MA window	15	0.3670/0.3930	0.2824/0.3448	0.1439/0.1948	0.0811/0.1982
MA window	21	0.3669/0.3934	0.2826/0.3448	0.1439/0.1940	0.0811/0.1981
MA window	25	0.3672/0.3935	0.2829/0.3449	0.1440/0.1951	0.0811/0.1982
MA window	45	0.3665/0.3924	0.2829/0.3441	0.1437/0.1936	0.0811/0.1982

Sensitivity analysis under deviations from ideal assumptions. To directly evaluate sensitivity when the idealized assumptions of Theorems 2 and 3 are violated, we perturb the candidate branches immediately before fusion. For MDK-MoE, the perturbations are applied to the six decomposed views; for AMD, they are applied to the eight predictor outputs before weighted aggregation. Systematic Bias adds a persistent offset to one fixed branch, violating the zero-mean error assumption. Correlated Noise applies a shared perturbation to three fixed branches, inducing cross-branch dependence and violating the uncorrelated-error assumption. Impulse Noise injects sparse, large-magnitude outliers into one fixed branch, violating the Gaussian/light-tailed-noise assumption. The corresponding strengths are {0.1,0.2,0.5}, {0.1,0.3,0.5}, and {1,2,5}, respectively. MDK-MoE's six views and AMD's eight predictor outputs were perturbed using the same absolute number of branches at horizon 96 and seed 2024. Each cell reports MDK-MoE/AMD MSE; percentages denote increases over each model's own baseline. The Overall MSE is averaged across datasets, whereas Overall percentages are macro-averages of the four dataset-wise relative changes. As shown in Tab. 15, the results support comparative robustness under deliberate assumption violations.

C.5 Model Efficiency

We evaluate model efficiency from three aspects: forecasting accuracy (MSE), training time (ms/iteration), and GPU memory consumption (MB). As shown in Fig. 9 and Tab. 16, we compare the efficiency of various baseline models on the Weather dataset, including **MDK-MoE**, K^2VAE, AMD, FilterTS, WFTNet, TimesNet, iTransformer, DLinear, and FEDFormer. To ensure a fair comparison, the input length and prediction horizon are fixed to 512 and 96, respectively, and the batch size is

Table 15: Assumption violations at horizon 96 and seed 2024, each cell reports MDK-MoE/AMD MSE.

Dataset	No noise	Systematic bias	Correlated noise	Impulse noise
Weather	0.1439/0.1458	0.1439(0.02%)/0.1493(2.38%)	0.1441(0.14%)/0.1510(3.59%)	0.1440(0.09%)/0.1490(2.22%)
Solar	0.1704/0.1758	0.1735(1.82%)/0.1879(6.90%)	0.1735(1.84%)/0.1952(11.05%)	0.1735(1.80%)/0.1869(6.31%)
Electricity	0.1308/0.1296	0.1308(0.03%)/0.1303(0.57%)	0.1313(0.36%)/0.1416(9.23%)	0.1309(0.10%)/0.1308(0.93%)
ETTm1	0.2834/0.2904	0.2834(0.00%)/0.2913(0.32%)	0.2834(0.01%)/0.2968(2.19%)	0.2834(0.00%)/0.2910(0.21%)
Overall	0.1821/0.1854	0.1829(0.47%)/0.1897(2.54%)	0.1831(0.59%)/0.1962(6.52%)	0.1830(0.50%)/0.1894(2.42%)

set to 16 for all models.z In terms of GPU memory usage, the 3 most memory-efficient models are DLinear, FilterTS, and **MDK-MoE**, requiring 490 MB, 526 MB, and 556 MB, respectively. Compared with Transformer-based models (iTransformer and FEDFormer), this highlights the advantage of linear-based designs. Notably, although DLinear consumes the least memory, its performance is inferior, with an MSE that is 5.56% higher than that of **MDK-MoE**. Regarding training time, K^2VAE achieves the second-fastest inference due to its generative probabilistic forecasting mechanism; however, this comes at the cost of degraded accuracy, with an MSE 15.97% higher than **MDK-MoE**. In contrast, WFTNet exhibits the slowest training speed, as it relies on both wavelet transforms and FFT-based frequency-domain enhancement. This further emphasizes the advantage of our multi-view joint decomposition strategy and the effectiveness of the Kalman MoE mechanism in improving computational efficiency. From the view of forecasting accuracy, **MDK-MoE** achieves the best MSE performance among all compared methods. Although AMD attains performance comparable to **MDK-MoE**, its reliance on complex multi-scale decomposition results in an inference time that is 4.61× longer. Overall, this comprehensive efficiency analysis demonstrates that **MDK-MoE** achieves a favorable balance between forecasting performance and computational efficiency.

Table 16: Model efficiency analysis on Weather.

Metric	MDK-MoE	K^2VAE	AMD	FilterTS	WFTNet	DLinear	iTransformer	TimesNet	FEDFormer
GPU Memory(MB)	556	736	734	526	5222	490	750	1450	2178
Training time(ms/iter)	39.65	18.27	183.01	31.43	197.19	5.26	18.77	57.73	157
MSE	0.144	0.167	0.145	0.147	0.162	0.152	0.174	0.172	0.238

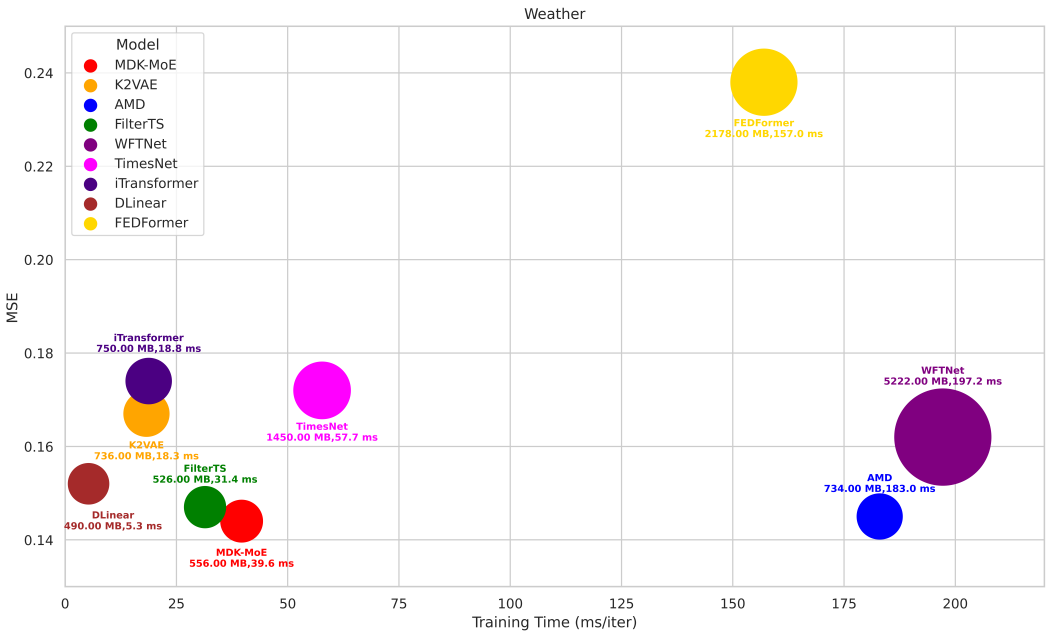


Figure 9: Visualizations on the weather dataset. The left, middle, and right panels correspond to **MDK-MoE**, FilterTS, and WFTNet. .

Furthermore, to verify the computational efficiency of our multi-view Kalman MoE, we construct an ablation variant that replaces the proposed Kalman-MoE with the standard Kalman filter (S-Kalman). We evaluate the time cost (ms/iter) on a single GPU, as reported in Tab. 17. As demonstrated, substituting Kalman-MoE with S-Kalman drastically increases computational overhead (e.g., approximately $28\times$ slower training on ETTh1) and leads to degraded MSE performance. This empirical evidence verifies our claim in Theorem 3: our learnable Kalman gain effectively avoids the prohibitive $O(d^3)$ matrix inversion bottleneck inherent to traditional Kalman filters under high-dimensional conditions. Furthermore, our framework remains exceptionally lightweight and runs faster than the SOTA baseline AMD.

Table 17: Efficiency Comparison (MSE | Train Time | Inference Time)

Model	ETTh1			Solar		
	MSE	Train	Infer	MSE	Train	Infer
Ours (Kalman-MoE)	0.366	41.46	10.57	0.170	183.00	15.41
S-Kalman	0.376	1199.40	351.64	0.206	1724.50	750.81
AMD Baseline	0.369	244.35	128.11	0.175	1134.34	270.10

C.6 Biases Analysis

To systematically investigate the bias characteristics induced by different decomposition operators, we conduct a comprehensive bias analysis from both distributional and temporal accumulation views. Model prediction bias is quantitatively evaluated using three metrics: *Mean Bias*, *Coverage Error*, and long-term trend deviation (*Drift Error*). In this study, bias is regarded as a structural and transmissible system bias.

Mean Bias measures the deviation of the predicted mean and is used to assess whether model predictions are systematically overestimated or underestimated. Let y_t denote the ground truth and $\hat{y}_t$ the corresponding prediction at time step t. The prediction residual is defined as

$$e_t = \hat{y}_t - y_t \tag{45}$$

Based on the residual sequence $\{e_t\}_{t=1}^{T}$, all bias-related metrics are defined as follows:

$$\text{Mean Bias} = \frac{1}{T}\sum_{t=1}^{T} e_t = E[e_t] \tag{46}$$

where a non-zero Mean Bias indicates persistent overestimation or underestimation. Models equipped with effective bias-correction mechanisms are expected to produce residuals with means closer to zero.

Coverage Error evaluates the bias of predictive intervals and reflects whether the estimated uncertainty is well calibrated. Assuming that the residuals approximately follow a Gaussian distribution, a 95% confidence interval can be constructed as:

$$\mathcal{I}_t = [-1.96\,\hat{\sigma},\ 1.96\,\hat{\sigma}] \tag{47}$$

$$\text{Coverage} = \mathbb{P}\left(|e_t| \le 1.96\,\hat{\sigma}\right) \tag{48}$$

$$\text{Coverage Error} = |\text{Coverage} - 0.95| \tag{49}$$

where $\hat{\sigma}$ denotes the standard deviation of the residuals. Coverage represents the empirical coverage probability, and *Coverage Error* measures its deviation from the nominal confidence level. It is worth noting that *Coverage Error* is primarily influenced by the accuracy of variance estimation and the tail behavior of the residual distribution. Consequently, models that overestimate variance may achieve lower Coverage Error by excessively widening prediction intervals, which does not necessarily imply more accurate forecasts.

Drift Error captures long-term trend deviation and is designed to quantify whether prediction errors exhibit persistent accumulation and systematic drift over extended horizons, which is particularly

critical for non-stationary time series. A smaller Drift Error indicates stronger suppression of systematic bias and better long-term stability. It is defined as:

$$E_t = \sum_{i=1}^{t} e_i \tag{50}$$

$$\text{Drift Error} = \frac{1}{T} \sum_{t=1}^{T} |E_t| \tag{51}$$

where E_t denotes the cumulative residual up to time step t.

Biases Analysis. as reported in Tab.18, on the ETTh1 and ETTh2 datasets, the *Mean Bias* values obtained using single decomposition operators — FFT, MA, and Wavelet — are $\{0.1058, -0.0856, -0.0998\}$ and $\{0.1058, -0.0856, -0.0998\}$, respectively. In contrast, the *Mean Bias* achieved by our Kalman MoE is -0.0885 on ETTh1 and -0.0456 on ETTh2, which are closer to zero. This indicates that Kalman MoE provides more effective bias correction. Similarly, the *Drift Error* values of the FFT, MA, and Wavelet-based models are $\{835118, 721999, 957222\}$ on ETTh1 and $\{646243, 475995, 500187\}$ on ETTh2, whereas Kalman MoE consistently yields the smallest *Drift Error* 566284 and 325363, respectively. A lower *Drift Error* implies stronger suppression of systematic bias accumulation and better long-term stability. Consequently, Kalman MoE significantly outperforms FFT, MA, and Wavelet-based approaches in terms of both mean bias and long-term drift, which is consistent with the conclusions of the empirical study. These results demonstrate that MDK-MoE possesses superior bias correction capability and enhanced long-term error mitigation when fusing multi-view decomposed components. Although Kalman MoE does not achieve the lowest *Coverage Error*, this is primarily due to its more compact and less conservative uncertainty estimation. In contrast, MA-based methods often reduce *Coverage Error* by inflating the residual variance, which results in excessively wide prediction intervals.

Furthermore, Fig. 10 and 11 visualize the residual density histograms, enabling a more intuitive comparison of Mean Bias and residual standard deviation across different operators. Overall, the bias analysis confirms that Kalman MoE effectively alleviates both instantaneous and accumulated biases while maintaining reasonable uncertainty calibration, making it particularly suitable for long-horizon forecasting of non-stationary time series.

Representation Space Visualization. To further analyze the differences among decomposition operators and Kalman MoE fusion strategies in the representation space, we employ T-SNE to project the latent state representations produced during the testing phase into a 3D space. This analysis aims to reveal differences in feature continuity, distributional consistency, and cross-sample smoothness. Since T-SNE preserves local neighborhood relationships in high-dimensional feature spaces, continuity in the projected sample distribution can be regarded as an important indicator of whether the model learns smooth and stable representations. As shown in Fig. 12 and 13, panels (a)–(d) correspond to FFT, MA, Wavelet, and Kalman MoE, respectively. Samples with similar colors correspond to temporally adjacent test instances. We visualize all test samples from the ETTh1 and ETTh2 datasets. Compared with models constructed using a single linear operator (FFT, MA, or Wavelet), the proposed Kalman MoE exhibits a markedly more continuous and smooth distribution in the reduced space. In contrast, single-operator methods display clear discontinuous regions, highlighted in red.

This phenomenon can be attributed to the nature of single-operator decompositions, whose latent representations are dominated by fixed linear operator–induced low- and high-frequency components. When the spectral structure or temporal scale of the signal changes abruptly, such linear decompositions are prone to introducing feature mapping bias under non-stationary conditions, leading to discontinuities in the T-SNE space. By contrast, Kalman MoE jointly models multiple biased linear decompositions, effectively mitigating the feature mapping bias introduced by any single operator. Specifically, Kalman MoE treats the low-frequency and residual components derived from FFT, MA, and Wavelet decompositions as multi-source observations and dynamically weights them through the MoE gating mechanism, preventing any single decomposition from dominating the state update. Moreover, the recursive update mechanism of Kalman filtering imposes temporal consistency constraints on the latent states, resulting in smoother state transitions across adjacent time steps and effectively suppressing feature discontinuities caused by local shifts.

Table 18: Biases analysis on ETTh1 and ETTh2.

Dataset	ETTh1				ETTh2			
Metric	Mean Bias	Bias Std	Coverage Error	Drift Error	Mean Bias	Bias Std	Coverage Error	Drift Error
Only-FFT	-0.1293	0.6752	0.0151	835188	0.1058	**0.6313**	0.0270	646243
Only-MA	-0.1084	0.6752	**0.0139**	721999	-0.0856	0.6366	**0.0237**	475995
Only-Wavelet	-0.1507	0.6803	0.0149	957222	-0.0998	0.6354	0.0258	500187
Kalman MoE	**-0.0885**	**0.6542**	0.0158	**566284**	**-0.0456**	0.6314	0.0244	**325363**

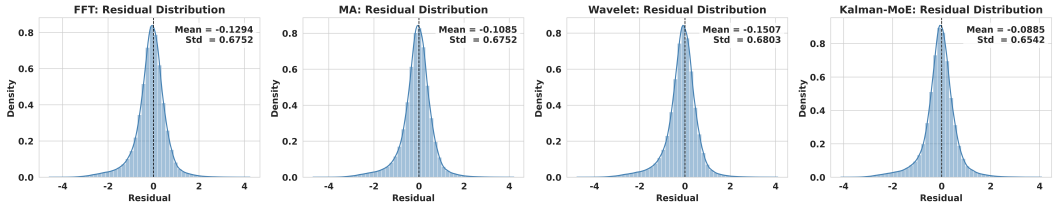


Figure 10: Residual distributions on the ETTh1 dataset. From left to right: FFT, MA, Wavelet, and MDK-MoE.

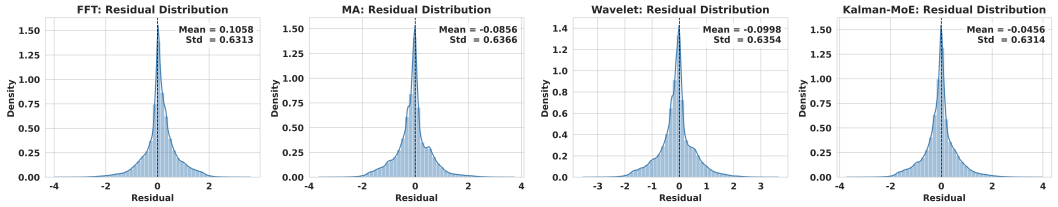


Figure 11: Residual distributions on the ETTh2 dataset. From left to right: FFT, MA, Wavelet, and MDK-MoE.

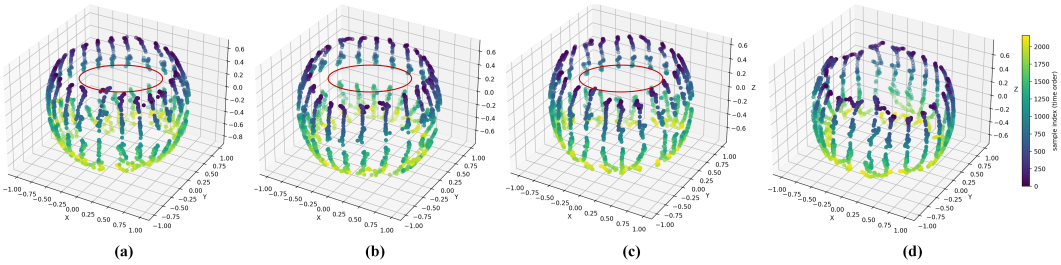


Figure 12: T-SNE visualization on the ETTh1 dataset. Panels (a)–(d) correspond to FFT, MA, Wavelet, and Kalman MoE, respectively. Red markers indicate discontinuous regions.

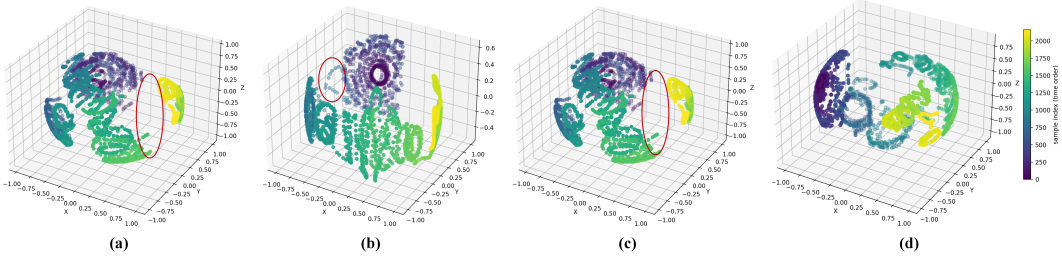


Figure 13: T-SNE visualization on the ETTh2 dataset. Panels (a)–(d) correspond to FFT, MA, Wavelet, and Kalman MoE, respectively. Red markers indicate discontinuous regions.

D Limitations

Despite its strong empirical performance, the proposed **MDK-MoE** framework has several limitations that open directions for future work. First, the current design involves a number of hyperparameters

associated with the multi-view decomposition, mutual information maximization, and Kalman MoE components, which require careful tuning across different datasets. This increases the computational overhead and may hinder the framework's out-of-the-box applicability. Future work will focus on simplifying the architecture and developing a unified, more robust hyperparameter set, potentially via neural architecture search or automatic tuning strategies. Second, while our experiments are conducted on diverse real-world forecasting benchmarks, further investigation is needed to evaluate the model's generalization to cross-domain and cross-dataset scenarios, such as transferring learned components to new domains with significantly different non-stationarity characteristics. Finally, we plan to explore extensions to related tasks, including probabilistic forecasting and multi-step prediction, to broaden the practical utility of the proposed approach.

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 accurately reflect the paper's contributions and scope, clearly stating the problem of non-stationary time series forecasting, the limitations of linear decomposition methods, and the core innovations of our MDK-MoE framework. All claims are supported by the theoretical derivations and experimental results in the main paper.

   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 mention our limitations in Appendix D.

   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: [Yes]

Justification: We proof the symmetry network components in Appendix B.

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: We disclose all information needed for reproduction in Appendix C.

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: All code is included in the Supplementary Material. The full dataset, codebase, and instructions will be made publicly available with the final paper.

   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: We provide experimental details in Appendix C.

   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: We provide the results of the p-value significance test in Tab. 9 of Appendix C.2.

   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: We provide information about computer resources used in Appendix C.5.

   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: We have read and understood the code of ethics and have made every effort to adhere to it.

   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: [N/A]

    Justification: Our work contributes to a better understanding of multi-view fusion mechanisms for mitigating prediction errors in non-stationary time series forecasting. We do not anticipate any direct negative societal impacts from this research.

    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: We do not release any new models or datasets. Our work uses standard, publicly available resources—ETTh, Weather datasets—and does not pose additional risk of misuse beyond their existing public availability.

    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: We use publicly available datasets (ETTh, Weather, Exchange), all of which are properly cited in the paper. We adhere to the licenses and terms of use as specified by their original creators.

    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: We will release our codebase along with detailed README files. The release does not include source code copied from external libraries; all code provided is our own.

    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 use crowdsourcing or human subjects.

    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 use crowdsourcing or human subjects.

    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: We used an LLM solely for editing assistance.

    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.