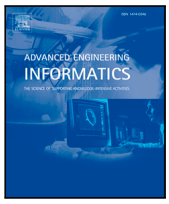
Advanced Engineering Informatics 72 (2026) 104442

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

KFS: KAN based adaptive frequency selection learning architecture for long-term time series forecasting

Changning Wu [a], Gao Wu [a], Rongyao Cai [a], Yong Liu [a,*], Kexin Zhang [a,b,*]

[a] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310013, Zhejiang, China
[b] Huzhou Institute of Zhejiang University, Huzhou, 313000, Zhejiang, China

ARTICLE INFO

Keywords:
Time series forecasting
Kolmogorov-Arnold networks
Fast Fourier transform
Multiscale mixing

ABSTRACT

Multi-scale decomposition architectures have emerged as predominant methodologies in time series forecasting. However, real-world time series exhibit noise interference across different scales, while heterogeneous information distribution among frequency components at varying scales leads to suboptimal multi-scale representation. Inspired by Kolmogorov-Arnold Networks (KAN) and Parseval's theorem, we propose a KAN based adaptive frequency Selection learning architecture (KFS) to address these challenges. This framework tackles prediction challenges stemming from cross-scale noise interference and complex pattern modeling through its FreK module, which performs energy-distribution-based dominant frequency selection in the spectral domain. Simultaneously, KAN enables sophisticated pattern representation while timestamp embedding alignment synchronizes temporal representations across scales. The feature mixing module then fuses scale-specific patterns with aligned temporal features. Extensive experiments across multiple real-world time series datasets demonstrate that KFS achieves state-of-the-art performance as a simple yet effective architecture. Our code is available on this website: https://github.com/wcnExplosion/KFS-main.

1. Introduction

Time series data is collected through sensors or other means from various sources such as traffic flow [1], industrial health monitoring [2–5], weather forecasting [6,7], fault diagnosis, energy calculation, and financial analysis [8]. As one of the most critical tasks in time series data analysis, time series forecasting has garnered significant attention from researchers. By studying time series to predict future data trends, it assists humans in making many important decisions.

Traditional time series forecasting techniques primarily rely on statistical learning methods, such as the Autoregressive Integrated Moving Average (ARIMA) [9] model and exponential smoothing. These models are typically based on strong assumptions like sequence stationarity or linear trends, which limits their performance when handling complex nonlinear relationships. In recent years, deep learning models have gradually become mainstream and achieved state-of-the-art accuracy due to their powerful automatic feature extraction and complex pattern recognition capabilities. These deep learning approaches mainly encompass methods based on CNNs [10,11], MLPs [12], and Transformers [13–15].

Due to real-world and engineering environments' complexities, observed time series often exhibit intricate and diverse patterns. These interwoven patterns result in complex dependencies with substantial noise contamination, making it challenging to establish connections between historical data and future variations. To capture complex temporal patterns, increasing research focuses on leveraging prior knowledge to decompose time series into simpler components as the foundation for forecasting. For example, Autoformer [16], DLinear [13], and FEDformer [17] decompose time series into trend and seasonal components. Building on this, TimeMixer [18] further introduces multi-scale seasonal-trend decomposition, highlighting the importance of multi-scale data. Recent models like TimesNet [19] and SparseTFT [20] concentrate on decomposing long sequences into multiple shorter subsequences based on periodicity length, thereby enabling the separate modeling of inter-period and intra-period dependencies within temporal patterns. While these methods extract subsequences from diverse perspectives to capture critical information, the subsequences split directly from the original series inevitably retain substantial noise, leading to suboptimal problems. Moreover, the frequency-domain structure of real-world time series is often subject to varying degrees of noise interference across multiple scales, with significant differences in the distribution of frequency components, which limits the effectiveness of directly learned multi-scale representations. This phenomenon can be

* Corresponding authors.
E-mail addresses: changning_wu@zju.edu.cn (C. Wu), wu_gao@zju.edu.cn (G. Wu), rycai@zju.edu.cn (R. Cai), yongliu@iipc.zju.edu.cn (Y. Liu), zhangkexin@zju.edu.cn (K. Zhang).

https://doi.org/10.1016/j.aei.2026.104442
Received 5 November 2025; Received in revised form 27 January 2026; Accepted 7 February 2026
Available online 16 February 2026

attributed to the Heterogeneous Information Distribution across different scales, where information exhibits a non-uniform and inconsistent distribution among frequency or scale components.

What is more, although multi-scale frameworks and trend-seasonal decomposition methods mitigate the impact of high-frequency noise to some extent through techniques like sliding window averaging, thereby achieving better forecasting performance. These approaches primarily operate on the temporal scale and overlook noise embedded in other patterns that is difficult to eliminate, ultimately limiting model effectiveness. It is worth noting that time series contain multiple frequency components, including noise that is unique to the frequency domain and that is difficult to mitigate in other domain. This inherent noise affects different frequencies unevenly, causing lower signal-to-noise ratios at lower-amplitude frequencies and consequently impairing model predictive performance. Mitigating noise interference while blending diverse frequency components makes forecasting particularly challenging. The aforementioned decomposition methods inspire us to design a multi-scale frequency denoising hybrid framework capable of isolating different frequency components while filtering high signal-to-noise ratio data. However, heterogeneous frequency patterns introduce complex representational challenges, often yielding suboptimal results. Fortunately, Kolmogorov-Arnold Network (KAN) [21] has recently gained significant attention in deep learning for its powerful data-fitting capability and flexibility, demonstrating potential to replace traditional MLPs. Compared to MLPs, KAN employs learnable activation functions that control its fitting capacity by adjusting basis functions. Moreover, TimeKAN [22], a KAN based method, has achieved SOTA performance in multiple datasets, demonstrating the remarkable potential of KAN for temporal feature representation. These considerations motivate us to explore KAN for representing patterns across different frequencies, thereby providing more information for forecasting.

Inspired by these observations, we propose KAN based adaptive frequency Selection learning architecture (KFS) to address forecasting challenges arising from noise and mixed data pattern. Specifically, KFS first decomposes components within the data via moving averages. Subsequently, the FreK module performs frequency selection at multiple scales to denoise the data, utilizing KAN to learn scale-specific temporal features from the denoised data. Finally, the hybrid module aligns and fuses timestamp embeddings from the look back window with corresponding scale representations, achieving temporal representation alignment and integration across scales to precisely model temporal features. Features from different scales are aggregated via averaging and projected to the desired forecast horizon through simple linear mapping. With our meticulously designed architecture, KFS achieves state-of-the-art performance in long-term time series forecasting tasks across multiple real-world datasets.

Our contributions can be summarized as follows:

- We designed an energy-distribution-based frequency selection method that effectively extracts components with higher signal-to-noise ratios. The resulting FreK module reduces noise impact and enables efficient modeling.
- We introduced a simple yet effective forecasting model KFS, and developed a Mixing Block that aligns and fuses multi-scale time series with corresponding timestamps.
- Comprehensive experiments demonstrate that our KFS achieves state-of-the-art performance in long-term forecasting tasks across multiple datasets while exhibiting exceptional efficiency.

2. Related works

2.1. Time series forecasting

In recent years, deep learning approaches for TSF have gained significant attention, mainly including CNN-based, MLP-based, and Transformer-based methodologies.

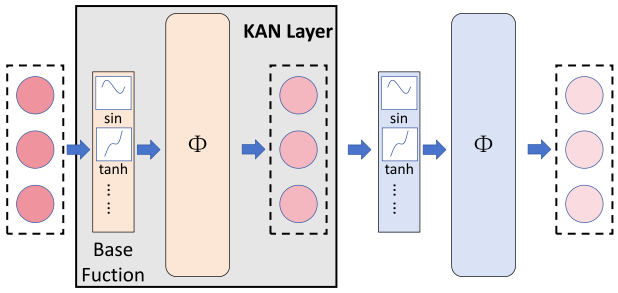


Fig. 1. The schematic diagram of a single KAN layer is shown as the gray part.

CNN-based methods focus on extracting temporal feature representations through convolutional operations. For example, MICN [23] and TimesNet [19] enhance the accuracy of sequence modeling by strategically adjusting the receptive fields of their architectures. Transformer-based approaches, while contrasting with CNN methods, exhibit substantially larger receptive fields. PatchTST [12] improves the capture of local patterns by segmenting input data into patches, while Crossformer [24] specializes in mining cross-variable dependencies. However, Transformer-based models face challenges stemming from computational complexity due to their massive parameterization. In this situation, MLP-based methods secure their position in TSF through lightweight architectures. FITS [25] introduces novel linear projections to reduce input complexity, requiring merely 10 K parameters. However, constrained by their parameterization, MLP-based approaches struggle to effectively extract and fuse diverse data modalities.

Unlike the aforementioned methods, this paper enhances data quality through spectral filtering strategies and integrate a multi-scale framework to extract temporal representations, achieving significantly improved accuracy in long term Time Series Forecasting.

Moreover, the emergence of time series foundation models such as Chronos [26], Timer [27], and Moirai [28] has shifted research focus toward zero-shot generalization via large-scale pre-training. In contrast to these general-purpose architectures, KFS is designed as a task-specific specialist for long-term forecasting that prioritizes structural frequency selection to maintain non-linear stability. By integrating a multi-scale framework with spectral filtering strategies to enhance data quality, KFS effectively extracts intrinsic temporal representations and achieves significantly improved accuracy in specialized long-term time series forecasting tasks.

2.2. Multi-scale architecture for TSF

In the field of TSF, extensive research has explored multi-scale architectures. TimeMixer [18] pioneered their application in TSF through decomposing multi-scale time series. MICN [23] extended multi-scale processing to convolutional layers, enabling efficient representation of seasonal patterns. Building on these advances, this work leverages a multi-scale framework to capture hierarchical information, proposing KFS's novel multi-pathway integration framework. By distinctly capturing temporal representations and physical timestamp embeddings, then fusing these components, KFS achieves enhanced precision in time series forecasting.

2.3. Kolmogorov-Arnold network

The Kolmogorov-Arnold representation theorem establishes that any multivariate continuous function can be expressed as a composition of univariate functions and additive operations. Using this theorem, KAN [21] introduces a novel network architecture that supplants traditional MLPs. Unlike MLPs with fixed activation functions, KAN incorporates learnable activation functions(Fig. 1). This flexibility positions KAN as a promising alternative to MLPs.

Initial implementations of KAN faced computational bottlenecks due to the excessive complexity of B-spline sampling, hindering broader adoption. To address this limitation, subsequent research explored alternative basis functions, rKAN [29] investigates rational functions as basis functions, FastKAN [30] accelerates computation using Gaussian radial basis functions to approximate third-order B-spline functions.

Furthermore, KAN has been adopted across diverse domains as a substitute for MLP. Convolutional KAN [31] replaces conventional kernels with learnable spline functions. KAT [32] integrates KAN layers into Transformer architectures, demonstrating impressive accuracy in multiple computer vision tasks. This paper proposes to introduce KAN to TSF and explore its potential in representing temporal data patterns.

While both KFS and TimeKAN [22] leverage KANs for temporal representation, our approach differs fundamentally in its motivation and theoretical basis. Grounded in Parseval's theorem, KFS explicitly addresses data noise by separating signal and noise from an energy-distribution perspective, assuming that noise predominantly resides in low-energy frequency bands. In contrast, TimeKAN focuses on decomposing frequency components to enhance learning efficiency for different temporal variations.

3. Preliminary

3.1. Motivation

In the physical world, time series data originate from sensors on physical devices or recordings of real-world relationships. These measurements inherently contain varying levels of noise interference due to factors including acquisition methods, mechanical transmission processes, and recording mechanisms. This noise significantly compromises the results of time series analysis tasks, particularly forecasting and anomaly detection. Consequently, developing methodologies to mitigate noise-induced distortions becomes imperative to enhance the representation of temporal patterns. This paper addresses this challenge through the view of multivariate time series forecasting.

We formally decompose the forecasting problem into two fundamental questions.

1. **How can we effectively reduce noise impact on both data and predictive models?**
2. **How can we explicitly extract intrinsic information from given time series?**

For the first question, we begin by assuming that the data primarily contains channel-wise independent additive white Gaussian noise. The primary mitigation approach for such noise traditionally requires prior knowledge of the noise distribution, which introduces additional domain-specific assumptions and hinders real-world applicability. However, the spectral uniformity of Gaussian white noise in the frequency domain motivates our solution: By selecting frequency bands with concentrated energy as the dominant temporal features, we reconstruct the time series within a bounded error margin, effectively attenuating noise. This principle is formalized in the following theorems.

Theorem 1 (*Parseval's Theorem*). *For a discrete signal $y \in \mathbb{R}^L$ and its DFT $Y \in \mathbb{C}^{L/2+1}$, the energy satisfies:*

$$\sum_{t=0}^{L-1} |y(t)|^2 = \frac{1}{L} \sum_{k=0}^{L/2} |Y[k]|^2. \tag{1}$$

Theorem 1 states that the total energy of a time series is equivalent in the frequency domain and the time domain. Therefore, by processing the time series in the frequency domain and converting it back to the time domain, the information of the original time series can be preserved. This foundation allows us to formalize Theorem 2, with its complete proof detailed in Appendix.

Theorem 2. *Let observed time series $y = y_0 + n$, where $n \sim \mathcal{N}(0, \sigma^2 I)$, and y_0 denotes original times series. After DFT, there exist $K \in \mathbb{N}^+$ and $\epsilon > 0$ such that the sparse reconstruction $\tilde{y}$ from the top-K frequencies of Y satisfies:*

$$\|\tilde{y} - y_0\|_2 < \epsilon. \tag{2}$$

Theorem 2 proposes that by filtering the dominant frequency bands of a time series, the proportion of noise can be reduced, thereby enhancing the quality of time series.

For the second question, we draw upon existing research to carefully design a KAN-based network architecture under the channel independence assumption, integrated with a multi-scale time series mixing framework.

3.2. Multiscale time series processing

In long time series forecasting, temporal sequences can capture information from multiple scales by down sampling, thereby enhancing prediction accuracy. For an input time series $X \in \mathbb{R}^{L \times C}$, we generate multi-scale sequences through down sampling. Specifically, for each coarser-grained subsequence X_{i+1}, it is derived from the finer-grained subsequence X_i at the preceding level by applying average pooling. We then sequentially obtain a collection of time series $\mathbb{X} = \{x_0, x_1, \ldots, x_m\}$ across m scales, where each $x_i \in \mathbb{R}^{\frac{L}{D^i} \times C}$ and D denotes the window size of average pooling. The down sampling process used in our work is shown as below:

$$x_{i+1} = \text{AvgPool}(x_i) \tag{3}$$

This technique has been extensively adopted in time series forecasting models and has demonstrated improved predictive accuracy along with enhanced modeling capabilities.

4. Method

4.1. Overview of the architecture

The core challenge lies in resolving sequence modeling for channel-independent information while effectively reducing the influence of noise. To address this, we propose a simple yet effective architecture, the **K**AN based adaptive **F**requency **S**election learning architecture (KFS), which improves prediction accuracy by organically integrating KAN to capture multiscale channel-independent features and temporal representation. The overall architecture of KFS is depicted in Fig. 2. Specifically, it consists of two key components: a **Fre**quency **K**-top Selection (**FreK**) Module and a Mixing Block. Detailed descriptions of each module are deferred to the following sections.

4.2. Frequency K-top selection

In real-world scenarios, a vast number of multivariate time series exhibit complex and diverse frequency components. Moreover, among the numerous frequency constituents within these time series, not all contribute meaningfully to the representation. These sequences commonly contain noise that reduces the signal-to-noise ratio of the time series, thereby leading to suboptimal performance.

To address this, we designed Frequency K-top Selection (FreK) module, which reduces noise through multi-scale principal frequency selection while comprehensively capturing temporal representation from the time series.

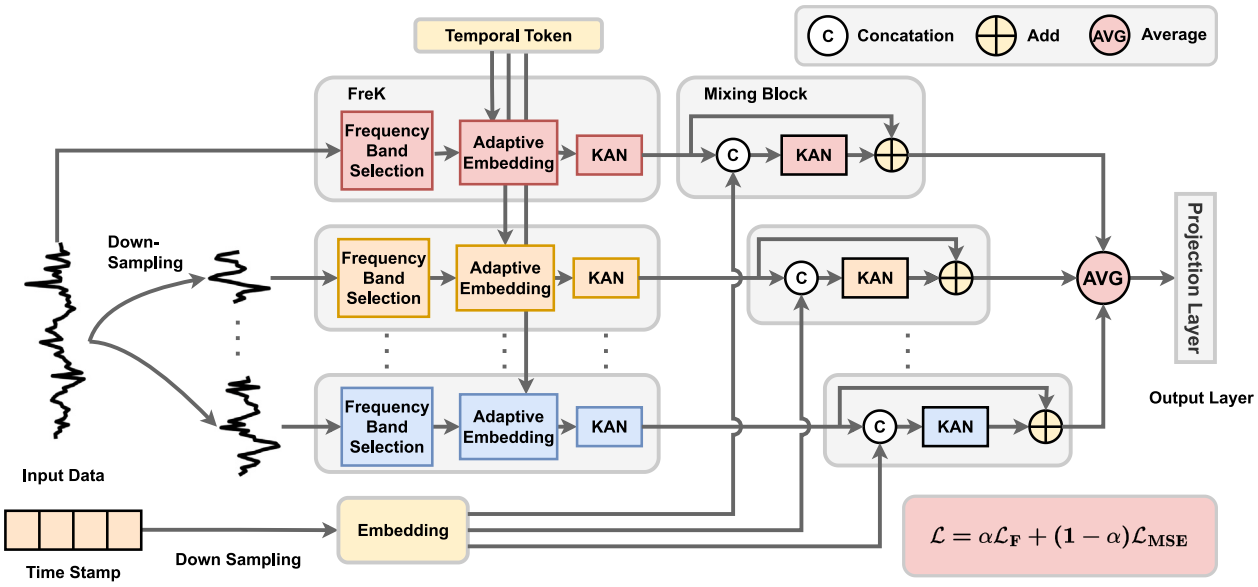


Fig. 2. Overall structure of the proposed KFS. Multi-scale architecture decomposes time series. The KANs are seamlessly integrated within the model framework. FreK selects the dominant frequency based on energy distribution and represent the temporal pattern. Mixing Block align temporal representation with its time stamp.

4.2.1. Frequency band selection

The FreK module first employs its Frequency Band Selection (FBS) block to screen primary components of time series through energy-distribution-based filtering. Since multivariate time series exhibit complex energy distributions that are difficult to extract directly, inspired by Theorem 1, we transform the time series into the spectral domain, initiating processing from the distribution of frequency components. Furthermore, to mitigate noise interference in time series and enhance the signal-to-noise ratio, we rank frequency bands in descending order of spectral energy and select the top-K bands as primary constituents of the time series. These selected bands are then inversely transformed back to the temporal domain to reconstruct the time series. As demonstrated in Theorem 2, controlling the energy distribution threshold enables reconstruction of time series that optimally approximates the noise-free sequence. Here, the reconstructed series $\tilde{x}(t)$ comes as followed:

$$\tilde{x}(t) = rFFT(TopK(FFT(x(t)))) \tag{4}$$

where K is the minimum value conducted as followed:

$$\frac{\sum_{i=1}^{K} X[i]^2}{\sum_{i=1}^{L/2+1} X[i]^2} > \delta \tag{5}$$

$$\{X(k)\}_{k=1}^{L/2+1} = sorted[FFT(x(t))] \tag{6}$$

where $sorted[\cdot]$ denotes sorting by magnitude in descending order, δ denotes the threshold of energy percentage.

At this stage, $\tilde{x}(t)$ consists predominantly of channel-independent temporal information with lower noise. Subsequently, FreK performs Adaptive Embedding(AE) of x along the temporal dimension and employs KAN for representation learning of intrinsic information. This process can be represented by the following formula:

$$E_1 = KAN(AE(\tilde{x}(t))) \tag{7}$$

where $AE(\cdot)$ denotes the adaptive embedding, E_1 denotes the temporal representation by FreK.

4.2.2. Adaptive embedding

In contemporary state-of-the-art time series forecasting models, the integration of adaptive modules into embeddings is frequently addressed. We also introduce an adaptive parameter $P \in \mathbb{R}^D$ to improve prediction efficacy, here D denotes the dimension of embedding space. However, unlike these approaches [33], the adaptive parameter in adaptive embedding serves to learn distinct characteristics unique to each dataset. The usage of P with one input series $x_i \in \mathbb{R}^{L\times C}$ is as follows:

$$E_i^j = concat([P, Linear(x_i^j)]) \tag{8}$$

where j denotes the index of variate. Thus, the whole embedding is expressed as follows:

$$E_i = AE(x_i) = [E_i^1, E_i^2, \ldots, E_i^{d_{model}}] \tag{9}$$

4.2.3. Group-rational KAN

Compared to traditional MLPs, KAN replaces fixed activate functions with learnable univariate functions, allowing complex nonlinear relationships to be modeled with fewer parameters and greater interpretability.

In our approach, the selection of KAN was motivated by the following considerations: The design of classical KAN leads to a computational complexity significantly higher than that of MLPs. Among the various methods aimed at reducing the computational complexity of KAN, we required an approach capable of performing representation learning from the perspective of time series mechanisms or state modeling. The rational polynomial-based variant of KAN, Group-Rational KAN [32], meets our requirements. Its computational complexity does not significantly exceed that of traditional MLPs, and compared to using trigonometric functions as basis functions, polynomials offer more flexible and diverse representational capabilities by adjusting their degrees.

The rational base functions are constructed by Q(x) and P(x) of order m,n.

$$\phi(x) = wF(x) = w\frac{P(x)}{Q(x)} = w\frac{\sum_{i=0}^{n} a_i x^i}{\sum_{i=0}^{m} b_i x^i} \tag{10}$$

where a_i and b_i are coefficient of the rational function and w is the scaling factor.

To integrate rational functions as base functions within KANs while mitigating the instability caused by poles, which occurs when Q(x) = 0, Group-KAN employs a modified formulation of the standard rational function.

$$F(x) = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + |b_1 x + \cdots + b_m x^m|} \tag{11}$$

Thus, Group-Rational KAN incorporates rational functions and constructs its processing architecture through group separation and sharing base function within group. For an input variable $X \in \mathbb{R}^{d_{in}}$, let i denotes its channel index. With g groups, each group in GR-KAN contains $d_g = d_{in}/g$ channels, where $\lfloor i/dg \rfloor$ represents the group index. The operation of GR-KAN on x can be expressed as:

$$GR\text{-}KAN(\mathbf{x}) = \phi \circ x = W\mathbf{F}(x) \tag{12}$$

To simplify it, we express it in matrix form as the product of a weight matrix $W \in \mathbb{R}^{d_{in} \times d_{out}}$ and a rational function $\mathbf{F}$:

$$W = \begin{bmatrix} w_{1,1} & \cdots & w_{1,d_{\text{in}}} \\ \vdots & \ddots & \vdots \\ w_{d_{\text{out}},1} & \cdots & w_{d_{\text{out}},d_{\text{in}}} \end{bmatrix} \tag{13}$$

$$\mathbf{F}(x) = \begin{bmatrix} F_{\lfloor 1/d_g \rfloor}(x_1) & \cdots & F_{\lfloor d_{\text{in}}/d_g \rfloor}(x_{d_{\text{in}}}) \end{bmatrix}^T \tag{14}$$

In our implemented rational KAN, we set the group parameter of the rational activation function $\mathbf{F}$ to $g = 8$, with the numerator order of the rational function set to 4 and the denominator order to 5. For each KAN unit, we simply prefix the rational function to a linear layer whose input and output dimensions are d_{model}, and the hidden layer dimension is d_{ff}. Moreover, the entire KAN we use consists of two KAN units connected in series.

$$KAN_i(x) = linear(F(x)) \tag{15}$$

where i denotes the layer index in our KAN.

4.3. Time stamp embedding

Additionally, we introduce linear embeddings for timestamps. In the real world, physical quantities closely associated with time series, such as mechanical load and electricity consumption, exhibit daily, monthly, yearly, and other levels of periodicity along the temporal dimension. By aligning timestamp information with the latent representations learned by the model, we can further enhance the model's ability to understand time series data. While existing approaches [33] incorporate timestamp information to boost model performance, they neglect the critical synchronization of temporal markers with multi-scale sequence patterns. Our methodology resolves this through time stamp down sampling, where temporal embeddings are progressively coarsened to maintain alignment with corresponding resolution levels in the input sequence hierarchy.

4.4. Multiscale feature mixing

After specifically learning temporal information from time series at different scales, we need to organically integrate the feature representations learned by the model. Here, we refer to the widely adopted feed-forward network. In contrast, we incorporate timestamp information from the time series and replace the MLP with a KAN.

As such, the multiscale feature mixing module can be represented by the following formula:

$$FM(E_1, E_s) = E_1 + KAN([E_1, E_s]) \tag{16}$$

where E_s denotes the linear embedding of time stamps.

For the fused multiscale data, we employ average aggregation followed by a simple linear projection layer to generate the predicted output $\tilde{y}(t)$:

$$\tilde{y}(t) = linear(FM_{avg}) \tag{17}$$

where FM_{avg} denotes the mean FM output on different input scales.

Table 1
Detailed dataset descriptions.

Dataset	Dim	Dataset size	Frequency
ETTh1, ETTh2	7	(8545, 2881, 2881)	Hourly
ETTm1, ETTm2	7	(34 465, 11 521, 11 521)	15 min
Weather	21	(36 792, 5271, 10 540)	10 min
ECL	321	(18 317, 2633, 5261)	Hourly
Traffic	862	(12 185, 1757, 3509)	Hourly

4.5. Loss function

Incorporating frequency domain alignment terms into loss functions is not novel. However, unlike previous work [34], our approach enforces alignment exclusively on the dominant frequencies of the data. While this method reduces fine-grained fitting precision, we maintain that frequency-domain signals primarily serve as coarse-grained representations for capturing macro-level trend shifts. Intuitively, fine-grained information modeling can be sufficiently handled by the MSE loss function alone. The specific formulation is shown as follows.

$$\mathcal{L}_F = \frac{1}{K} \sum_{i}^{K} \| \mathcal{F}\{\tilde{y}(t)\}_i - \mathcal{F}\{y(t)\}_i \| \tag{18}$$

By combining the hybrid loss L_F with the MSE loss, we arrive at our final loss function as follows:

$$\mathcal{L} = \alpha \mathcal{L}_F + (1 - \alpha)\mathcal{L}_{\text{MSE}} \tag{19}$$

where α is a hyperparameter, $\tilde{y}(t)$ denotes the prediction of KFS, K denotes the index of top-K frequency prediction data with the highest amplitudes. Unlike FreK, the K here is set to a fixed value of 32. This loss function accounts for both temporal discrepancies and introduces alignment of the principal frequencies in the time series.

5. Experiments

5.1. Datasets and baselines

We conducted long-term forecasting experiments on six real-world datasets: ETT-Series [35], Electricity [36], Weather [35] and Traffic [19]. Following established protocols from previous studies [19,37], we split the datasets of the ETT series into training, validation, and test sets according to a 6: 2: 2 ratio. For the remaining datasets, the ratio is 7:1:2. Detailed descriptions of the datasets are shown in Table 1.

We carefully selected representative models as baselines in field of time series forecasting, including: (1) Transformer-based models: TimeXer [33], iTransformer [38]. (2) CNN-based models: TimesNet [19], MICN [23]. (3) MLP-based models: TimeMixer [18], DLinear [13]. (4) KAN-based models: TimeKAN [21]. (5) Frequency-based methods: FiLM [39]. (6) Time series foundation model: Time-FFM [40]

5.2. Experimental settings

To ensure fair comparisons, we adopt the same look-back window length $T = 96$ and the same prediction length $F = \{96, 192, 336, 720\}$. We use Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics to evaluate the performance of each method. All experiments are conducted using PyTorch on an NVIDIA 4090 24 GB GPU and are repeated three times for consistency. Detailed hyperparameter search strategies can be found in Appendix A.2.

Table 2
Full results of the multivariate long-term forecasting result comparison (Seed = 2021). The input sequence length is set to 96 for all baselines and the prediction lengths $F \in \{96, 192, 336, 720\}$. Avg means the average results from all four prediction lengths. – indicates that the data results on that dataset were not reported in the official paper of the method.

Models		KFS(Ours)		TimeXer		TimeKAN		TimeMixer		iTransformer		TimeFFM		TimesNet		MICN		DLinear		FiLM	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.159	**0.205**	**0.157**	**0.205**	0.162	0.208	0.163	0.209	0.174	0.214	0.191	0.230	0.172	0.220	0.198	0.261	0.195	0.252	0.195	0.236
	192	0.207	0.249	**0.204**	**0.247**	0.207	0.249	0.211	0.254	0.221	0.254	0.236	0.267	0.219	0.261	0.239	0.299	0.237	0.295	0.239	0.271
	336	0.262	**0.288**	**0.261**	0.290	0.263	0.290	0.263	0.293	0.278	0.296	0.289	0.303	0.280	0.306	0.285	0.336	0.282	0.331	0.289	0.306
	720	0.345	0.342	0.340	0.341	**0.338**	**0.340**	0.344	0.348	0.358	0.347	0.362	0.350	0.365	0.359	0.351	0.388	0.345	0.382	0.360	0.351
	Avg	0.243	**0.271**	**0.241**	**0.271**	0.242	0.272	0.245	0.276	0.256	0.278	0.270	0.288	0.259	0.287	0.268	0.321	0.265	0.315	0.271	0.290
ETTh1	96	0.368	0.397	0.382	0.403	**0.367**	**0.395**	0.385	0.402	0.386	0.405	0.385	0.400	0.384	0.402	0.426	0.446	0.395	0.407	0.438	0.435
	192	0.425	0.426	0.429	0.435	**0.414**	**0.420**	0.443	0.430	0.441	0.436	0.439	0.430	0.436	0.429	0.454	0.464	0.446	0.441	0.494	0.466
	336	0.467	0.446	0.468	0.448	**0.445**	**0.434**	0.512	0.470	0.487	0.458	0.480	0.449	0.491	0.469	0.493	0.487	0.489	0.467	0.547	0.495
	720	0.454	**0.458**	0.469	0.461	**0.444**	0.459	0.497	0.476	0.503	0.491	0.462	0.456	0.521	0.500	0.526	0.526	0.513	0.510	0.586	0.538
	Avg	0.428	0.431	0.437	0.437	**0.417**	**0.427**	0.459	0.444	0.454	0.447	0.442	0.434	0.458	0.450	0.475	0.480	0.461	0.457	0.516	0.483
ETTh2	96	**0.280**	**0.334**	0.286	0.338	0.290	0.340	0.289	0.342	0.297	0.349	0.301	0.351	0.340	0.374	0.372	0.424	0.340	0.394	0.322	0.364
	192	**0.362**	**0.387**	0.363	0.389	0.375	0.392	0.378	0.397	0.380	0.400	0.378	0.397	0.402	0.414	0.492	0.492	0.482	0.479	0.405	0.414
	336	**0.406**	**0.421**	0.414	0.423	0.423	0.435	0.432	0.434	0.428	0.432	0.422	0.431	0.452	0.452	0.607	0.555	0.591	0.541	0.435	0.445
	720	0.423	0.435	**0.408**	**0.432**	0.443	0.449	0.464	0.464	0.427	0.445	0.427	0.444	0.462	0.468	0.824	0.655	0.839	0.661	0.445	0.457
	Avg	**0.367**	**0.394**	**0.367**	0.396	0.383	0.404	0.390	0.409	0.383	0.407	0.382	0.406	0.414	0.427	0.574	0.531	0.563	0.519	0.402	0.420
ETTm1	96	**0.314**	**0.354**	0.318	0.356	0.322	0.361	0.317	0.356	0.334	0.368	0.336	0.369	0.338	0.375	0.365	0.387	0.346	0.374	0.353	0.370
	192	0.358	**0.378**	0.362	0.383	**0.357**	0.383	0.367	0.384	0.377	0.391	0.378	0.389	0.374	0.387	0.403	0.408	0.382	0.391	0.389	0.387
	336	0.388	**0.398**	0.395	0.407	**0.382**	0.401	0.391	0.406	0.426	0.420	0.411	0.410	0.410	0.411	0.436	0.431	0.415	0.415	0.421	0.408
	720	0.460	0.446	0.452	0.441	**0.445**	**0.435**	0.454	0.441	0.491	0.459	0.469	0.441	0.478	0.450	0.489	0.462	0.473	0.451	0.481	0.441
	Avg	0.380	**0.394**	0.382	0.397	**0.376**	0.395	0.382	0.397	0.407	0.410	0.399	0.402	0.400	0.406	0.423	0.422	0.404	0.408	0.412	0.402
ETTm2	96	0.173	**0.253**	**0.171**	0.256	0.174	0.255	0.175	0.257	0.180	0.264	0.181	0.267	0.187	0.267	0.197	0.296	0.193	0.293	0.183	0.266
	192	**0.236**	**0.295**	0.237	0.299	0.239	0.299	0.240	0.302	0.250	0.309	0.247	0.308	0.249	0.309	0.284	0.361	0.284	0.361	0.248	0.305
	336	**0.291**	**0.332**	0.296	0.338	0.301	0.340	0.303	0.343	0.311	0.348	0.309	0.347	0.321	0.351	0.381	0.429	0.382	0.429	0.309	0.343
	720	0.395	0.395	**0.392**	**0.394**	0.395	0.396	**0.392**	0.396	0.412	0.407	0.406	0.404	0.408	0.403	0.549	0.522	0.558	0.525	0.410	0.400
	Avg	**0.274**	**0.318**	**0.274**	0.322	0.277	0.322	0.277	0.324	0.288	0.332	0.286	0.332	0.291	0.333	0.353	0.402	0.354	0.402	0.288	0.328
Electricity	96	0.148	**0.238**	**0.140**	0.242	0.174	0.266	0.153	0.245	0.148	0.240	0.198	0.282	0.168	0.272	0.180	0.293	0.210	0.302	0.198	0.274
	192	0.164	**0.253**	**0.157**	0.256	0.182	0.273	0.166	0.257	0.162	0.253	0.199	0.285	0.184	0.289	0.189	0.302	0.210	0.305	0.198	0.278
	336	0.181	**0.274**	**0.176**	0.275	0.197	0.286	0.185	0.275	0.178	0.269	0.212	0.298	0.198	0.300	0.198	0.312	0.223	0.319	0.217	0.300
	720	0.219	**0.306**	**0.211**	**0.306**	0.236	0.320	0.224	0.312	0.225	0.317	0.253	0.330	0.220	0.320	0.217	0.330	0.258	0.350	0.278	0.356
	Avg	0.178	**0.267**	**0.171**	0.270	0.197	0.286	0.182	0.272	0.178	0.270	0.216	0.299	0.193	0.304	0.196	0.309	0.225	0.319	0.223	0.302
Traffic	96	0.444	0.274	0.428	0.271	–	–	0.462	0.285	**0.395**	**0.268**	–	–	0.593	0.321	0.519	0.309	0.650	0.396	0.647	0.384
	192	0.452	0.284	0.448	0.282	–	–	0.473	0.296	**0.417**	**0.276**	–	–	0.617	0.336	0.537	0.315	0.598	0.370	0.600	0.361
	336	0.463	0.289	0.473	0.289	–	–	0.498	0.296	**0.433**	**0.283**	–	–	0.629	0.336	0.534	0.313	0.605	0.373	0.610	0.367
	720	0.504	0.307	0.516	0.307	–	–	0.506	0.313	**0.467**	**0.302**	–	–	0.640	0.350	0.577	0.325	0.645	0.394	0.691	0.425
	Avg	0.466	0.289	0.466	0.287	–	–	0.484	0.297	**0.428**	**0.282**	–	–	0.620	0.336	0.542	0.316	0.625	0.383	0.637	0.384
1st		**29**		20		16		1		10		0		0		0		0		0	

5.3. Main results

Comprehensive forecasting results are shown in Table 2, the best results are highlighted in **Bold** and the second-best are underlined. Lower MSE/MAE values indicate higher prediction accuracy. Through the experimental results, we can observe that KFS demonstrates highly competitive performance across multiple datasets. Overall, KFS exhibits stable leading performance in the MAE metric, reflecting its robustness against outliers and more reliable forecasting capability. Additionally, in terms of the MSE metric, KFS achieves optimal overall performance on ETTh2 and ETTm2, while also showing highly competitive results on several other datasets.

However, it is also noticeable that on the Electricity dataset, KFS underperforms compared to TimeXer. This may be attributed to the cross-attention mechanism in TimeXer's architecture, which helps extract cross-channel information, thereby enhancing its performance on high-dimensional datasets like Electricity. On the other hand, KFS performs worse than TimeKAN in MSE on ETTh1 and ETTm1. This could be because the FreK module in KFS filters out high-frequency components with lower energy in the data, leading to smoothed input data, which enhances KFS's stability when dealing with outlier inputs. Moreover, we can see that the average performance of TimeXer and KFS is very close on multiple datasets. A detailed examination of the experimental data reveals that KFS holds a clear advantage in relatively shorter forecasting windows such as 96, 192, and 336. However, when the forecasting window expands to 720, KFS is outperformed by TimeXer, resulting in a slight edge for TimeXer in the final average performance.

What is more, on the high-dimensional Traffic dataset, while iTransformer achieves the best average MSE by leveraging its inverted architecture to model global channel correlations, KFS remains highly competitive with an average MSE of 0.466, performing on par with TimeXer. The performance gap between these models stems from their architectural focus: iTransformer and TimeXer prioritize explicit cross-channel attention, whereas KFS emphasizes robust temporal-frequency features through the FreK module and KAN-based non-linear mapping. It is noteworthy that KFS has demonstrated excellent long-term stability on the Traffic dataset, achieving an MSE reduction of more than 0.01 compared to TimeXer at time horizons of 336 and 720 steps. This indicates that on high-dimensional datasets, KFS's frequency-domain TopK selection is more effective at mitigating error accumulation in extended forecasting periods than pure channel-mixing strategies.

To evaluate KFS in the context of recent trends in time series foundation models, we include the modern foundation model Time-FFM as a baseline under consistent experimental settings. As shown in Table 2, KFS consistently outperforms Time-FFM across the majority of benchmarks, such as the Weather and ETT series. These results demonstrate that frequency-domain optimized architectures remain highly competitive compared to larger pre-trained models.

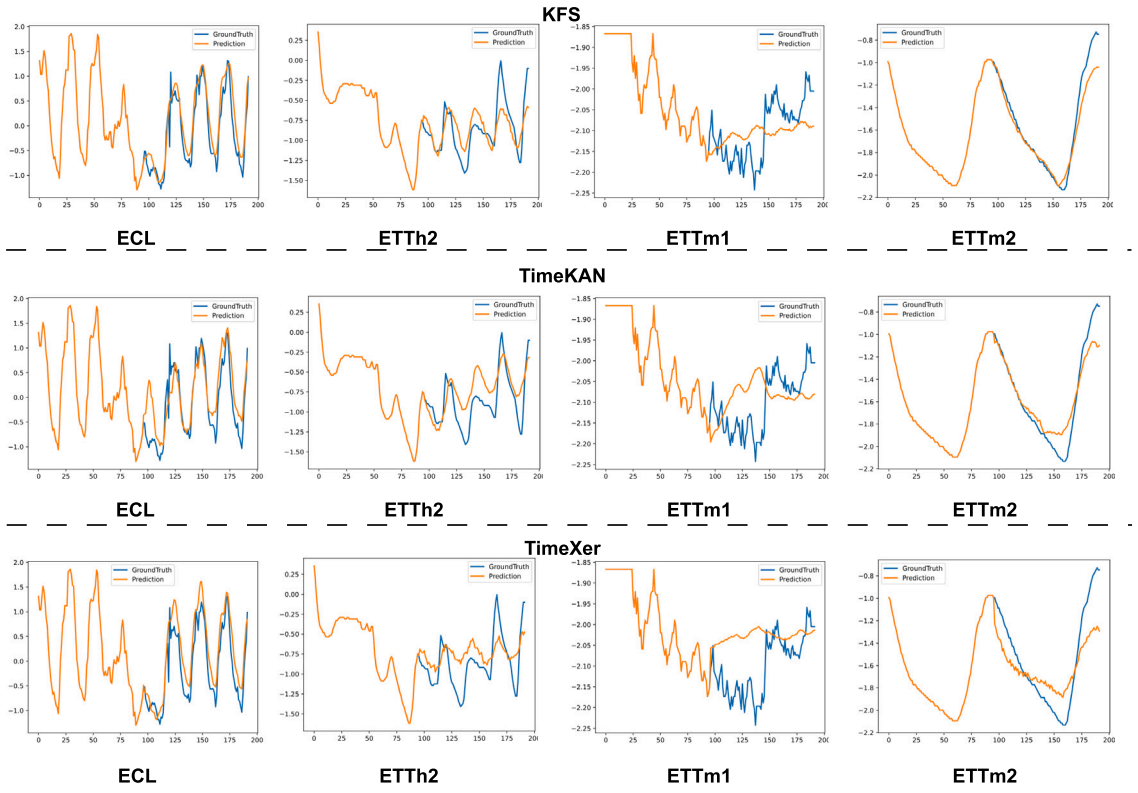


Fig. 3. Qualitative comparison of forecasting performance among KFS, TimeKAN and TimeXer on four benchmark datasets (ECL, ETTh2, ETTm1 and ETTm2). All models use an input length of 96 and predict the next 96 time steps. In each subplot, the blue line represents the ground truth, while the orange line indicates the model's prediction.

For a more in-depth analysis, we also present the quantitative results of the images generated by the three aforementioned models on four datasets, as shown in Fig. 3. When both the input length and prediction length are set to 96 steps, both KAN-based KFS and TimeKAN demonstrate better alignment with the curve shapes of ECL and ETTh2 datasets. In contrast, the Transformer-based TimeXer shows weaker fitting performance for future data patterns. Furthermore, compared to TimeKAN, KFS exhibits a superior capability in capturing future data trends, as quantitatively evidenced by a reduction in Average MAE of 6.6%, 0.3%, and 1.2% on the ECL, ETTm1, and ETTm2 datasets, respectively. Notably, on the ETTm1 dataset, KFS achieves a lower Average MAE of 0.394, demonstrating a closer numerical alignment with the ground truth mean and confirming its robustness against abrupt data fluctuations. TimeKAN, however, displays relatively noticeable mean shift issues. These analyses indicate that KAN architectures can better capture fine-grained data information, demonstrating powerful data representation capabilities. In our proposed method, the TopK selection strategy implemented in the frequency domain effectively captures the overall data trends, showcasing robust stability performance. Considering the overall performance, KFS's strong performance in the MAE metric across the majority of experiments further supports this observation.

In summary, our work demonstrates significant superiority over state-of-the-art models in terms of the MAE metric while remaining highly competitive in MSE performance. Visualization of the forecasting results confirms that our proposed KFS model accurately predicts future values of the target variables. The comparative visual analysis shows that our model outperforms baseline approaches in forecasting precision, validating its effectiveness in capturing frequency information and improving prediction accuracy. The proposed KFS model offers a promising solution for time series forecasting in real-world scenarios.

Table 3
Results of ablation study on weather, ETTh1 and ETTh2 dataset. **KAN→MLP** substitute KAN into a MLP. **w/o Stamp** replaced the time stamp with a zero matrix. **w/o AE** replaced the learnable parameter P with a zero vector.

Models		KFS		KAN→MLP		w/o Stamp		w/o AE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.159**	**0.205**	0.161	**0.205**	0.163	0.208	0.163	0.209
	192	**0.207**	**0.249**	0.208	**0.249**	0.211	0.251	0.211	0.252
	336	**0.262**	**0.288**	0.264	0.289	**0.262**	0.289	**0.262**	**0.288**
	720	0.345	0.342	**0.342**	**0.340**	0.344	0.343	0.347	0.344
	Avg	**0.243**	**0.271**	0.244	**0.271**	0.245	0.272	0.245	0.273
ETTh1	96	0.368	0.397	0.370	**0.393**	**0.367**	0.398	0.375	**0.393**
	192	**0.425**	**0.426**	0.435	0.428	0.427	0.428	0.436	0.428
	336	**0.467**	0.446	0.472	**0.445**	0.468	0.447	0.470	**0.445**
	720	**0.454**	**0.458**	0.469	0.462	0.459	0.460	0.467	0.461
	Avg	**0.428**	**0.431**	0.437	0.432	0.430	0.433	0.437	0.432
ETTh2	96	0.280	**0.334**	0.284	0.337	0.282	0.335	**0.279**	**0.334**
	192	**0.362**	0.387	0.366	0.388	0.365	**0.386**	0.364	**0.386**
	336	**0.406**	**0.421**	0.419	0.426	0.410	0.422	0.414	0.425
	720	**0.423**	**0.435**	0.435	0.443	0.431	0.444	0.431	0.440
	Avg	**0.367**	**0.394**	0.373	0.399	0.372	0.396	0.372	0.396

5.4. Model analysis

5.4.1. Ablation study

To investigate the effectiveness of each component of KFS, we conduct detailed ablation of each possible design on selected datasets. As show in Table 3, we have following observations.

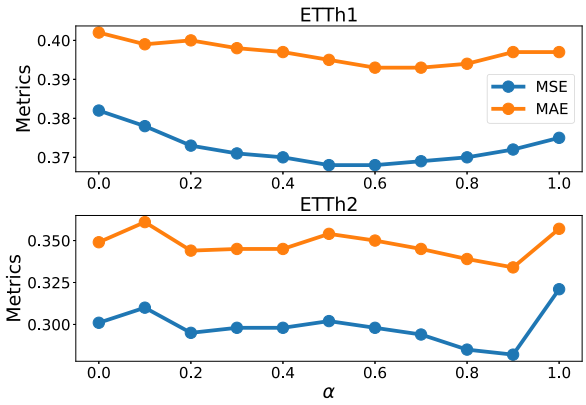


Fig. 4. The impact of α on metrics. This experiment is conducted on ETTh1 and ETTh2 datasets with both input and prediction length 96.

Table 4
Results of filter study on weather and ETTh2 dataset.

Methods		Top-K(Ours)		Avg filter		Gaussian filter	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.159**	**0.205**	0.161	0.208	0.160	0.206
	192	**0.207**	**0.249**	0.214	0.255	0.211	0.252
	336	**0.262**	**0.288**	0.272	0.294	0.265	0.289
	720	**0.345**	**0.342**	0.346	**0.342**	**0.345**	0.343
	Avg	**0.243**	**0.271**	0.248	0.274	0.245	0.272
ETTh2	96	**0.280**	**0.334**	0.292	0.345	0.283	0.336
	192	**0.362**	**0.387**	0.380	0.399	0.364	0.388
	336	**0.406**	**0.421**	0.423	0.431	0.407	0.423
	720	**0.423**	**0.435**	0.437	0.448	0.433	0.443
	Avg	**0.367**	**0.394**	0.383	0.405	0.372	0.397

For KAN (**KAN→MLP**), we substituted it into a standard MLP with matched parameterization. The consequent deterioration in error metrics substantiates that KFS's implementation of the KAN delivers substantially stronger representation learning than conventional MLPs. Furthermore, we observed that when KAN was replaced by MLP, there was an improvement of up to 0.015 in the MSE metric, and performance degradation occurred in over 75% of the experimental results. This evidence validates the functional superiority of rational basis functions for TSF.

For Time Stamp part **(w/o Stamp)**, We replaced the time stamp embedding with a zero matrix of identical dimensions. We observed performance degradation on all datasets when removing the TimeStamp component. Notably, the performance decline was more pronounced on ETTh1 and ETTh2 that are electricity equipment load datasets exhibiting stronger temporal periodicity compared to the Weather dataset. This outcome empirically validates the simple yet effective design of our TimeStamp embedding methodology.

For Embedding method **(w/o AE)**, We substituted the learnable parameter P in the Adaptive Embedding with a fixed zero matrix of identical shape and dimensions. We also observed performance degradation on all datasets.

However, we also observed that in high-dimensional and long-term forecasting scenarios, specifically on the Weather dataset, the KAN → MLP configuration exhibited signs of positive optimization. This indicates a degradation in the long-term forecasting capability of the KAN model when handling high-dimensional data. The underlying reason may lie in the rational KAN's group KAN architecture, where the shared base functions struggle to simultaneously accommodate the diverse requirements of different channels in long-horizon forecasting scenarios, thereby limiting the predictive capacity of KAN. Nevertheless, this performance decline has a negligible impact on the model's overall effectiveness.

Table 5
Results of KAN alternatives study on weather and ETTh1 dataset.

Models		KFS		FastKAN		ChebyshevKAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.159**	**0.205**	0.167	0.212	0.189	0.235
	192	**0.207**	**0.249**	0.210	0.252	0.210	0.250
	336	**0.262**	**0.288**	0.264	0.291	0.282	0.305
	720	0.345	0.342	0.346	0.344	**0.344**	**0.341**
	Avg	**0.243**	**0.271**	0.247	0.275	0.256	0.283
ETTh1	96	**0.368**	0.397	0.389	0.397	0.379	**0.396**
	192	**0.425**	0.426	0.441	**0.424**	0.427	0.429
	336	**0.467**	**0.446**	0.476	0.448	0.551	0.508
	720	**0.454**	**0.458**	0.478	0.468	0.520	0.497
	Avg	**0.428**	**0.431**	0.446	0.434	0.469	0.458

For FreK, we examined how alternative filtering methods (e.g. mean filtering and Gaussian filtering) affect model effectiveness on both the ETTh2 and Weather datasets. The results of these two experiments are presented in Table 4. Experimental results in Table 4 indicate that, compared to traditional filtering algorithms, the energy-based frequency-domain selection strategy can reduce the MSE loss by up to 0.01 and the MAE loss by up to 0.008, with an average reduction of up to 0.005. This validates the effectiveness of the proposed method in improving model performance.

When selecting KAN variants, to further validate the rationale and efficiency improvements of using Group-Rational KAN, we conducted a comparative study with common KAN variants such as FastKAN [41] and the Chebyshev KAN [42] utilized in TimeKAN. These methods were evaluated on the ETTh1 and weather datasets. The experimental results are presented in Table 5. From the experimental results, it can be observed that Group-rational KAN achieves better performance compared to FastKAN and Chebyshev KAN. This difference stems from the fact that different KAN variants employ distinct basis functions, and in time series forecasting tasks, KAN variants equipped with rational basis functions are better suited to handle abrupt changes in the data. This further demonstrates the necessity of our use of Group-rational KAN. For the loss term, we conducted a dedicated experiment on the ETTh1 dataset to investigate the impact of α on model performance, with experimental results presented in Fig. 4. The experimental results demonstrate appropriate calibration of α substantially enhances model capabilities, experimentally validating the efficacy of our proposed loss function combination.

In conclusion, our ablation studies highlight the strong potential of replacing MLPs with KANs, as this substitution maintains or improves overall performance across multiple datasets. The experiments also validate the individual contributions of each proposed module, collectively guiding the KFS model toward achieving superior and more stable forecasting performance.

5.4.2. *Hyperparameter sensitivity analysis*

In this section, we will further discuss and analyze the impact of certain hyperparameters on the model's performance during the experiments.

Varying energy threshold δ

We investigate its impact on the model performance of KFS across the ETT series datasets. Additionally, we conduct a more fine-grained parameter sensitivity analysis specifically on the ETTm1 and ETTm2 datasets. The experimental results are shown in Fig. 5. From the experimental results shown in Fig. 5, we have the following observations and analysis:

Overall, as the energy retention ratio δ gradually increases from 0.2 to 1.0, both the MSE and MAE for all four datasets (ETTh1, ETTh2, ETTm1, ETTm2) show a gradual declining trend, indicating that retaining more frequency band information is generally beneficial

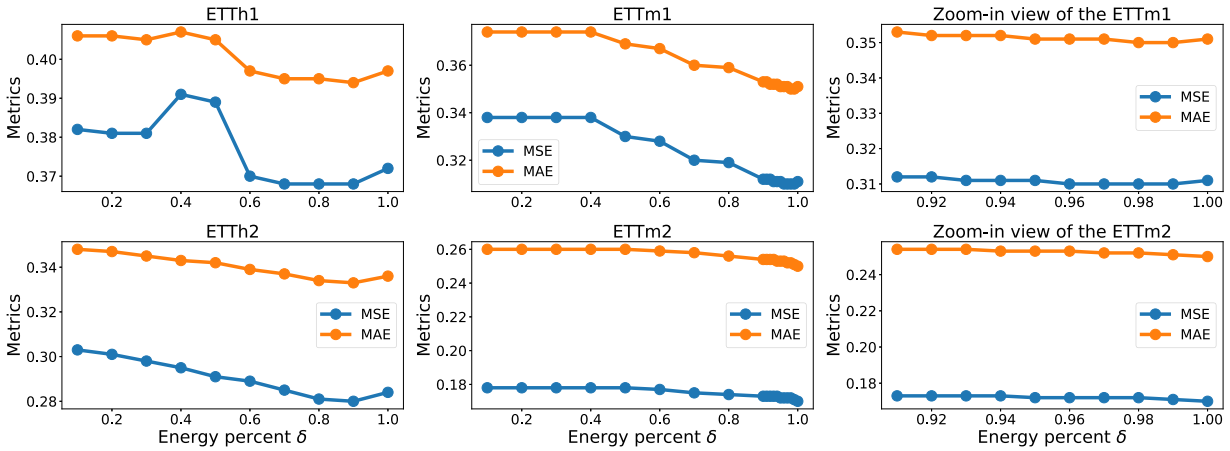


Fig. 5. The impact of δ on metrics. This experiment is conducted on ETT Series datasets with look-back window 96 and prediction length 96.

for improving the model's predictive performance. Particularly after δ reaches around 0.9, the metrics begin to converge, suggesting that model performance improvement gradually saturates when more than 90% of the energy is retained, and further increasing the frequency band information contributes little to additional gains.

Notably, the patterns of metric changes differ across datasets: in ETTh1, the MSE shows a significant drop around $\delta \approx 0.5$ before stabilizing, indicating that mid-energy frequency bands have a more pronounced impact on model performance for this dataset. In contrast, ETTm1 and ETTm2 exhibit an almost monotonic and smooth decline, suggesting a relatively uniform contribution distribution across their frequency bands. Furthermore, the overall metric values for ETTm2 are lower than those for ETTm1, implying that the ETTm2 dataset is more sensitive to frequency band selection, or that it inherently contains less noise and has more concentrated effective information. From the zoomed-in views of ETTm1 and ETTm2, it can be further observed that within the high-energy interval $\delta \in [0.92, 1.0]$, both the MSE and MAE curves flatten with minimal fluctuations. This further confirms that model performance enters a stable phase after retaining the vast majority of the energy, and also indicates that there indeed exists a portion of low-energy frequency bands in the data whose contained information contributes weakly to the prediction task and may even introduce noise.

These results consistently support our initial hypothesis: not all frequency components in the time-series data positively impact predictive performance. Some frequency bands, especially the low-energy portions, may carry noise or redundant information. Filtering out these parts can help the model focus on the key frequency features, thereby enhancing prediction accuracy. Simultaneously, this experiment validates the effectiveness of our proposed Top-K selection method based on energy analysis. This method, by preserving high-energy frequency bands and filtering out low-energy parts, achieves noise suppression and feature enhancement without excessive loss of information, providing a reliable and interpretable preprocessing strategy for frequency-domain modeling.

Varying Look-back Window

Generally, time series forecasting models tend to benefit from an increased look-back window, leading to improved performance. To validate and explore the impact of this on KFS, we present the results of KFS with varying look-back window lengths, as shown in Fig. 6. The results in Fig. 6 indicate that KFS can achieve better performance with longer look-back window lengths.

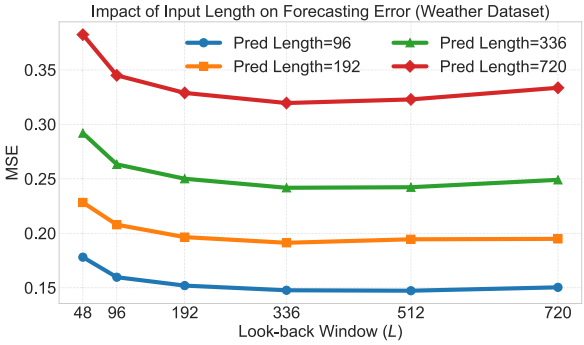


Fig. 6. Model analysis on Weather dataset with an enlarged look-back length $T \in \{48, 96, 192, 336, 512, 720\}$.

Longer Forecasting Window

In the analysis of overall experimental results presented earlier in Table 2, we observed performance degradation in KFS at a prediction length of 720. To systematically evaluate its performance under extended forecasting windows, we expanded the prediction horizon and compared the effects of varying input lengths. In this experiment, we used the TimeXer model as a control, with look-back window $L \in \{96, 336\}$, and forecasting window $T \in \{512, 720, 960, 1440, 2160.\}$. The experimental results are shown in Fig. 7.

From the experimental results, we observe that when the input length is increased to 336, KFS achieves an MSE improvement of over 0.02 across all prediction horizons, indicating that its performance benefits from a longer input sequence. Furthermore, compared to TimeXer, although KFS consistently underperforms, this performance gap can be mitigated by increasing the input length. Moreover, the disparity between the two models does not widen as the prediction horizon extends, demonstrating that KFS does not exhibit significant performance degradation in ultra-long-term forecasting tasks.

5.4.3. Robustness evaluation

To validate the predictive stability of our proposed model, we conducted three experiments under five different random seeds(2020–2024) in the ETTh1, ETTh2, ETTm1, ETTm2, ECL and Weather datasets, and calculated the mean and variance for each experiment. The results, as shown in Table 7, indicate that our model performs consistently stable in most cases, with variances below 0.01, demonstrating its robustness.

5.4.4. Statistical testing

To validate that our proposed model possesses sufficient stability surpassing existing SOTA models, we selected the best-performing baseline models and KAN-based models (namely TimeXer and TimeKAN)

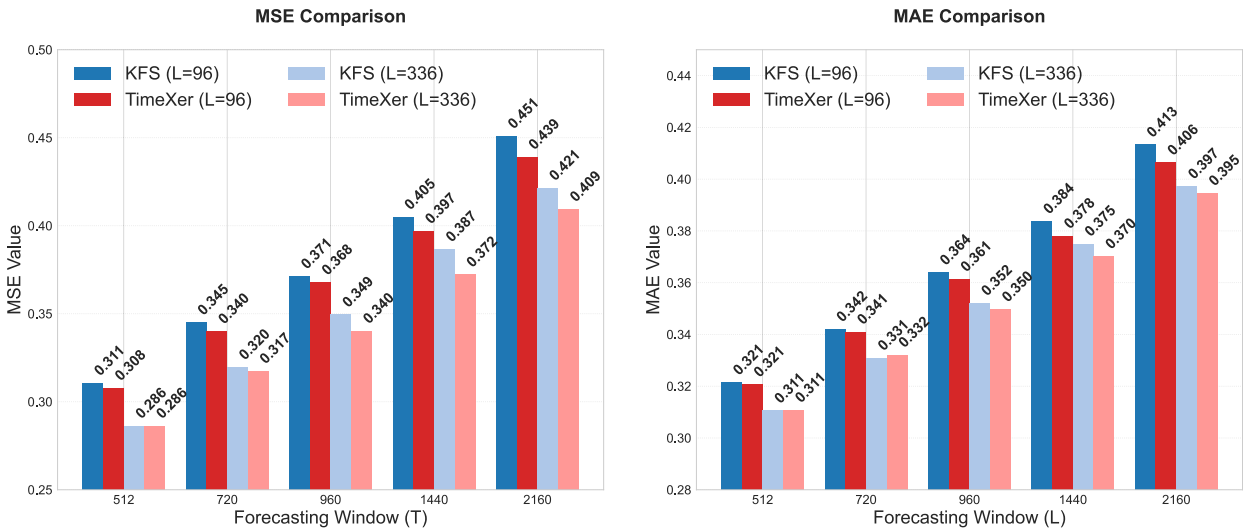


Fig. 7. Performance comparison of KFS and TimeXer in terms of MSE and MAE across various forecasting horizons T with different look-back window lengths $L \in \{96, 336\}$.

Table 6
The t-test experimental results for KFS and baseline models. We conducted it by comparing the mean performance of those models across multiple prediction windows under different datasets. ✓ indicates that with at least 95% confidence, the performance of KFS surpasses that of compared model. \ indicates that there is insufficient evidence to conclude with confidence that either model outperforms the other. * indicates that with 95% confidence, the performance of compared model surpasses that of KFS.

Metric	MSE			MAE		
Dataset	KFS	TimeXer	t-test	KFS	TimeXer	t-test
ETTh1	0.435 ± 0.004	0.442 ± 0.002	✓	0.434 ± 0.002	0.437 ± 0.002	✓
ETTh2	0.374 ± 0.005	0.377 ± 0.004	\	0.399 ± 0.003	0.400 ± 0.001	\
ETTm1	0.379 ± 0.001	0.389 ± 0.004	✓	0.393 ± 0.001	0.402 ± 0.002	✓
ETTm2	0.273 ± 0.001	0.276 ± 0.001	✓	0.318 ± 0.001	0.322 ± 0.001	✓
Weather	0.244 ± –	0.247 ± –	✓	0.271 ± 0.001	0.279 ± –	✓
ECL	0.177 ± –	0.171 ± 0.001	*	0.266 ± 0.001	0.273 ± 0.002	✓
Traffic	0.470 ± 0.003	0.476 ± 0.004	✓	0.289 ± 0.001	0.283 ± 0.003	*
Metric	MSE			MAE		
Dataset	KFS	TimeKAN	t-test	KFS	TimeKAN	t-test
ETTh1	0.435 ± 0.004	0.428 ± 0.008	*	0.434 ± 0.002	0.431 ± 0.004	*
ETTh2	0.374 ± 0.005	0.392 ± 0.006	✓	0.399 ± 0.003	0.411 ± 0.004	✓
ETTm1	0.379 ± 0.001	0.381 ± 0.002	\	0.393 ± 0.001	0.397 ± 0.002	✓
ETTm2	0.273 ± 0.001	0.281 ± 0.001	✓	0.318 ± 0.001	0.327 ± 0.001	✓
Weather	0.244 ± –	0.244 ± –	\	0.271 ± 0.001	0.273 ± 0.001	✓
ECL	0.177 ± –	0.199 ± 0.001	✓	0.266 ± 0.001	0.288 ± 0.001	✓

for comparison. We conducted t-test statistical experiments under five different random seeds, and the obtained results are shown in Table 6. In the table, ✓ indicates that our model achieves a 95% confidence level in overall performance superiority over the baseline model on the respective dataset. \ denotes that neither our model nor the baseline model has sufficient confidence to surpass the other, implying comparable capabilities. $*$ indicates that the baseline model surpasses our model with over 95% confidence. From the experimental results, we observe that, whether compared with TimeXer or TimeKAN, our method achieves comparable results in the majority of experiments, and in over half of the experimental outcomes, our method surpasses both models. This confirms that our approach is more effective and stable than the baselines.

5.4.5. Efficiency analysis

To evaluate the practical utility of KFS, we conduct an efficiency analysis focusing on the strategic trade-off between representational capacity and computational overhead. While recent trends have introduced extremely light-weight architectures [20,43,44] that prioritize minimal parameter counts, KFS is designed as a high-efficiency SOTA solution that preserves the non-linear modeling power necessary for complex patterns while significantly reducing the resource demands of traditional high-performance models.

As shown in Table 8, KFS achieves a superior balance compared to established heavy-parameter baselines. Specifically, KFS requires only 116 MB of memory, which represents an 85%–90% reduction compared to PatchTST (807 MB) and TimesNet (1227 MB). Despite the inclusion of FFT operations within the FreK module, the model maintains a highly competitive training efficiency with a step time of 21 ms. These results confirm that KFS provides the necessary representational depth to capture fine-grained temporal information while remaining substantially more resource-efficient than conventional Transformer-based and CNN-based architectures.

5.5. Discussion and limitation

As demonstrated in Table 2, KFS consistently achieves superior results in the MAE metric. This performance is rooted in the localized activation of KAN layers combined with the frequency filtering of the FreK module. Unlike Transformer-based attention mechanisms,

Table 7
Mean values and standard deviations of KFS. '–' means the deviation is less than 0.0005.

Dataset	ETTh1		ETTh2	
Metric	MSE	MAE	MSE	MAE
96	0.371 ± 0.002	0.397 ± 0.001	0.290 ± 0.004	0.343 ± 0.003
192	0.434 ± 0.006	0.429 ± 0.002	0.370 ± 0.004	0.391 ± 0.002
336	0.467 ± 0.002	0.446 ± 0.001	0.420 ± 0.008	0.426 ± 0.004
720	0.468 ± 0.012	0.466 ± 0.005	0.423 ± 0.006	0.435 ± 0.004
Dataset	ETTm1		ETTm2	
Metric	MSE	MAE	MSE	MAE
96	0.313 ± 0.002	0.353 ± 0.003	0.172 ± 0.001	0.253 ± 0.001
192	0.359 ± 0.001	0.378 ± 0.001	0.236 ± 0.001	0.295 ± 0.001
336	0.389 ± 0.001	0.400 ± 0.001	0.292 ± 0.001	0.331 ± 0.001
720	0.459 ± 0.004	0.445 ± 0.002	0.394 ± 0.001	0.391 ± 0.001
Dataset	Weather		ECL	
Metric	MSE	MAE	MSE	MAE
96	0.159± –	0.205 ± –	0.147 ± –	0.237 ± –
192	0.209 ± 0.001	0.250 ± 0.001	0.163 ± –	0.252 ± 0.001
336	0.262 ± –	0.288 ± –	0.181 ± –	0.272 ± 0.001
720	0.345 ± 0.002	0.343 ± 0.002	0.218 ± 0.001	0.305 ± 0.001

Table 8
A comparison of used Memory, Training time per step and FLOPs between KFS and other 4 models. To ensure a fair comparison, we fix the prediction length $F = 96$ and the input length $T = 96$, and set the input batch size to 32.

Model	Memory	Step time	FLOPs
PatchTST	807 MB	70 ms	51.28 G
FEDformer	379 MB	70 ms	5.28 G
TimesNet	1227 MB	50 ms	115.85 G
TimeMixer	132 MB	**13 ms**	**0.62 G**
KFS	**116 MB**	21 ms	1.66 G

which are often sensitive to extreme values, the rational-function-based mapping in KAN provides a more robust fit to data distributions. This effectively mitigates outlier-induced skewness, ensuring that the learned representations remain centered on the underlying structural process rather than being distorted by transient spikes. Quantitatively, this trend-capturing capability is evidenced by the reduction in prediction error variance during abrupt changes, as visualized in Fig. 3.

The performance divergence across different datasets further elucidates the architectural trade-offs of KFS. On datasets with strong periodicities (e.g., ETT and ECL), KFS leverages the spectral bias of the FreK module. By acting as a learnable filter that retains dominant frequencies while discarding stochastic noise, FreK prevents the cumulative error growth typical of attention-based models in extended forecasting windows. However, this stability necessitates a "smoothing effect" that may overlook low-energy, high-frequency variations. This accounts for the marginal MSE sacrifice on ETTh1 and ETTm1 compared to TimeKAN, suggesting that KFS prioritizes structural trend consistency over fitting fine-grained local noise. Similarly, on the Electricity dataset, TimeXer's cross-attention provides an edge by capturing dense inter-channel information that KFS's current temporal-focused block may miss.

Although the channel-independent (CI) strategy has proven its effectiveness in recent studies [12,13] and can avoid the risk of overfitting caused by spurious correlations between channels in datasets with low channel correlation, the CI strategy employed in KFS introduces specific limitations in high-dimensional scenarios, such as the Traffic dataset. By treating each variable as an individual temporal evolution, KFS avoids overfitting risks associated with spurious inter-variable correlations. However, it lacks an explicit mechanism to model the complex spatial–temporal entanglement between hundreds of sensors. This variate-agnostic approach explains why KFS may underperform models like iTransformer, which utilize inverted frameworks to aggregate cross-channel information. While the CI approach ensures stable latent representations and individual trend accuracy, it presents a performance ceiling when global, coordinate-based interactions are paramount for optimizing average MSE.

In summary, while Transformer-based models remain optimal for modeling dense global variable interactions, the KFS architecture offers a superior solution for scenarios where capturing frequency structures and maintaining non-linear stability are critical. Beyond its current task-specific application, the core FreK module, which captures dominant frequency patterns, is inherently scalable. We believe this frequency-selection mechanism could serve as a robust backbone for future foundation models, potentially enhancing zero-shot capabilities through superior noise-signal separation and more stable trend projection.

6. Conclusion

In this paper, we propose a KAN-based time series forecasting model (KFS) that integrates a frequency-selective filtering method based on Parseval's theorem, aiming to address the issues of data noise and complex temporal representations. The proposed model adopts a multi-scale architecture. At each scale, the model performs data filtering based on the energy distribution of the data in the frequency domain, while simultaneously using KAN for complex data representation learning. A novel frequency-selective strategy based on energy distribution is designed to reduce the impact of noise. By leveraging extrinsic timestamp information and a designed mixing module, the KAN combines data representations with real-world temporal information to generate future estimates, and an MLP-based decoder is applied to produce the final predictions. Extensive experiments on multiple real-world datasets, along with comparative studies covering various aspects, demonstrate the robustness and superiority of KFS's performance.

Currently, KFS primarily focuses on sequence-level data modeling and does not account for inter-channel learning, which may limit its forecasting capability. Furthermore, the frequency-selection strategy is mainly based on amplitude information in the frequency domain, without considering the mechanism of phase shifts' influence on the data, which may also restrict its applicability to noise. In future work, we will explore cross-channel mechanisms or incorporate more prior knowledge of frequency-domain data to enhance efficiency. Moving forward, we aim to extend KFS to the audio domain and integrate its spectral selection mechanism into larger generative architectures, thereby bridging the gap between task-specific precision and foundation model generality.

CRediT authorship contribution statement

Changning Wu: Writing – original draft, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Gao Wu:** Writing – original draft, Validation, Investigation, Formal analysis. **Rongyao Cai:** Writing – original draft, Validation. **Yong Liu:** Writing – review & editing, Supervision, Funding acquisition. **Kexin Zhang:** Writing – review & editing, Supervision, Project administration.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used DeepSeek in order to improve the language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62503419) and the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2026C02A1250).

Appendix

A.1. Proof of Theorem 2

Let observed time series $y = y_0 + n$, where $n \sim \mathcal{N}(0, \sigma^2 I)$, and y_0 denotes original times series. After DFT, there exist $K \in \mathbb{N}^+$ and $\epsilon > 0$ such that the sparse reconstruction $\tilde{y}$ from the top-K frequencies of Y satisfies:

$$\|\tilde{y} - y_0\|_2 < \epsilon. \tag{20}$$

Proof. To prove Theorem 2, it suffices to demonstrate that $P(\|\tilde{y} - y_0\|_2 > \epsilon)$ has an upper bound $\mu < 1$, which reduces to verifying the following inequality:

$$\mathbb{P}\left(\|\tilde{y} - y_0\|_2^2 > \epsilon\right) \leq \frac{C\sigma^2 K + \sum_{j>K} |Y_0^{(j)}|^2}{\epsilon} < 1,$$

where $Y_0^{(j)}$ is the jth largest DFT coefficient of y_0, $C > 0$ is a constant value.

The reconstruction error decomposes into two parts:

$$\|\tilde{y} - y_0\|_2^2 = \underbrace{\sum_{j>K} |Y_0^{(j)}|^2}_{\text{Signal Energy Loss}} + \underbrace{\sum_{j=1}^{K} |N^{(j)}|^2}_{\text{Retained Noise Energy}}.$$

1. **Signal energy loss**: By Parseval's theorem, $\sum_{j>K} |Y_0^{(j)}|^2$ is the energy discarded from the true signal. Since y_0 has finite energy ($\|y_0\|_2^2 < \infty$), $\sum_{j>K} |Y_0^{(j)}|^2 \to 0$ as $K \to \infty$.
2. **Retained noise energy**: The coefficients $|N^{(j)}|^2$ follow a chi-square distribution χ^2_{2K} (real and imaginary parts of complex Gaussian are independent). By Markov's inequality:
$$\mathbb{P}\left(\sum_{j=1}^{K} |N^{(j)}|^2 > t\right) \leq \frac{\mathbb{E}\left[\sum_{j=1}^{K} |N^{(j)}|^2\right]}{t} = \frac{K\sigma^2}{t}.$$
3. **Joint bound**: For any $\epsilon > \sum_{j>K} |Y_0^{(j)}|^2$, set $\epsilon_1 = \epsilon - \sum_{j>K} |Y_0^{(j)}|^2$. Then:
$$\mathbb{P}\left(\|\tilde{y} - y_0\|_2^2 > \epsilon\right) \leq \mathbb{P}\left(\sum_{j=1}^{K} |N^{(j)}|^2 > \epsilon_1\right) \leq \frac{K\sigma^2}{\epsilon_1}.$$
Substituting ϵ_1 gives the result with $C = 1$.

Obviously, there exist at least one K and one ϵ satisfying $K\sigma^2 < \epsilon - \sum_{j>K} |Y_0^{(j)}|^2$, then we can get the upper bound is:

$$\frac{K\sigma^2}{\epsilon_1} = \frac{K\sigma^2}{\epsilon - \sum_{j>K} |Y_0^{(j)}|^2} < 1$$

Therefore, Theorem 2 is proven.

A.2. Hyper-parameter selection and implementation details

For the main experiments, we have the following hyperparameters. The dimension of embedding d_{model}. The hidden state of KAN d_{ff}. d_{model} and d_{ff} are set via hyperparameter searching among the range of $\{128, 256, 512, 1024\}$ for d_{model} and $\{256, 512, 1024, 2048\}$ for d_{ff}. And we set $\alpha = 0.3$ and $\delta = 80\%$ in Traffic, Weather and ECL datasets. For ETTm1, ETTm2, ETTh1 and ETTh2, we search α in the range of 0 to 1 with step 0.1. And for δ, the step is 0.05.

Data availability

Data will be made available on request.

References

[1] Weiwei Jiang, Jiayun Luo, Graph neural network for traffic forecasting: A survey, Expert Syst. Appl. 207 (2022) 117921.
[2] Helin Li, Shufeng Zheng, Yonghao Shen, Minghai Han, Rui Zhang, Huadong Zhao, Hydro-steel structure digital twins: Application in structural health monitoring and maintenance of large-scale reservoir, Adv. Eng. Inform. 62 (2024) 102922.
[3] Wenbin Cai, Dezun Zhao, Tianyang Wang, Multi-scale dynamic graph mutual information network for planet bearing health monitoring under imbalanced data, Adv. Eng. Inform. 64 (2025) 103096.
[4] Jr-Fong Dang, Tzu-Li Chen, Hung-Yi Huang, The human-centric framework integrating knowledge distillation architecture with fine-tuning mechanism for equipment health monitoring, Adv. Eng. Inform. 65 (2025) 103167.
[5] Shundi Duan, Xiao Tan, Pengwei Guo, Yurong Guo, Yi Bao, The transformative roles of generative artificial intelligence in vision techniques for structural health monitoring: A state-of-the-art review, Adv. Eng. Inform. 68 (2025) 103719.
[6] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, Peter Battaglia, Learning skillful medium-range global weather forecasting, Science 382 (6677) (2023) 1416–1421.
[7] Yi Zhou, Minghua Hu, Daniel Delahaye, Ying Zhang, Lei Yang, Collaborative strategic conflict management for 4D trajectories under weather forecast uncertainty, Adv. Eng. Inform. 65 (2025) 103293.
[8] Hongbin Huang, Minghua Chen, Xiao Qiao, Generative learning for financial time series with irregular and scale-invariant patterns, in: The Twelfth International Conference on Learning Representations, 2024.
[9] S.L. Ho, M. Xie, The use of ARIMA models for reliability forecasting and analysis, Comput. Ind. Eng. 35 (1) (1998) 213–216.
[10] Luo donghao, Wang Xue, ModernTCN: A modern pure convolution structure for general time series analysis, in: The Twelfth International Conference on Learning Representations, 2024.
[11] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, Gregory D Hager, Temporal convolutional networks for action segmentation and detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156–165.
[12] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, Jayant Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations, 2023.
[13] Ailing Zeng, Muxi Chen, Lei Zhang, Qiang Xu, Are transformers effective for time series forecasting? 2023.
[14] Yudong Zhang, Xu Wang, Zhaoyang Sun, Pengkun Wang, Binwu Wang, Limin Li, Yang Wang, Meta koopman decomposition for time series forecasting under temporal distribution shifts, Adv. Eng. Inform. 62 (2024) 102840.
[15] Jinkai Zhang, Yingying Wang, Shengbin Ma, Xinghao Ding, Xiaotong Tu, SOMA: A semantic-guided order-aware mamba architecture for multivariate time series forecasting, Adv. Eng. Inform. 68 (2025) 103788.
[16] Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, J. Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 22419–22430.
[17] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, Rong Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: Proc. 39th International Conference on Machine Learning (ICML 2022), 2022.
[18] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, JUN ZHOU, TimeMixer: Decomposable multiscale mixing for time series forecasting, in: The Twelfth International Conference on Learning Representations, 2024.

[19] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, Mingsheng Long, TimesNet: Temporal 2D-variation modeling for general time series analysis, in: International Conference on Learning Representations, 2023.
[20] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, Junjie Yang, SparseTSF: Modeling long-term time series forecasting with *1k* parameters, in: Forty-First International Conference on Machine Learning, 2024.
[21] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, Max Tegmark, KAN: Kolmogorov–Arnold networks, in: The Thirteenth International Conference on Learning Representations, 2025.
[22] Songtao Huang, Zhen Zhao, Can Li, Lei Bai, TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting, in: The Thirteenth International Conference on Learning Representations, 2025.
[23] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, Yifei Xiao, MICN: Multi-scale local and global context modeling for long-term series forecasting, in: The Eleventh International Conference on Learning Representations, 2023.
[24] Yunhao Zhang, Junchi Yan, Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: International Conference on Learning Representations, 2023.
[25] Zhijian Xu, Ailing Zeng, Qiang Xu, FITS: Modeling time series with $10k$ parameters, in: The Twelfth International Conference on Learning Representations, 2024.
[26] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, Bernie Wang, Chronos: Learning the language of time series, Trans. Mach. Learn. Res. (2024) Expert Certification.
[27] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, Mingsheng Long, Timer: Generative pre-trained transformers are large time series models, in: Forty-First International Conference on Machine Learning.
[28] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, Doyen Sahoo, Unified training of universal time series forecasting transformers, in: Forty-First International Conference on Machine Learning, 2024.
[29] Alireza Afzal Aghaei, rKAN: Rational Kolmogorov-Arnold networks, 2024.
[30] Ziyao Li, Kolmogorov-Arnold networks are radial basis function networks, 2024.
[31] Alexander Dylan Bodner, Antonio Santiago Tepsich, Jack Natan Spolski, Santiago Pourteau, Convolutional Kolmogorov-Arnold networks, 2025.
[32] Xinchao Wang Xingyi Yang, Kolmogorov-Arnold transformer, in: The Thirteenth International Conference on Learning Representations, 2025.
[33] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Yunzhong Qiu, Haoran Zhang, Jianmin Wang, Mingsheng Long, Timexer: Empowering transformers for time series forecasting with exogenous variables, Adv. Neural Inf. Process. Syst. (2024).
[34] Hao Wang, Lichen Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, Dacheng Tao, FreDF: Learning to forecast in the frequency domain, in: The Thirteenth International Conference on Learning Representations, 2025.
[35] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, Wancai Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, vol. 35, AAAI Press, 2021, pp. 11106–11115.
[36] Artur Trindade, ElectricityLoadDiagrams20112014, 2015, http://dx.doi.org/10.24432/C58C86, UCI Machine Learning Repository.
[37] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, Jianmin Wang, Deep time series models: A comprehensive survey and benchmark, 2024, ArXiv Preprint ArXiv:2407.13278.
[38] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, Mingsheng Long, Itransformer: Inverted transformers are effective for time series forecasting, in: The Twelfth International Conference on Learning Representations, 2024.
[39] Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, FiLM: Frequency improved Legendre memory model for long-term time series forecasting, in: Alice H. Oh, Alekh Agarwal, Danielle Belgrave, Kyunghyun Cho (Eds.), Advances in Neural Information Processing Systems, 2022.
[40] Qingxiang Liu, Xu Liu, Chenghao Liu, Qingsong Wen, Yuxuan Liang, Time-FFM: Towards LM-empowered federated foundation model for time series forecasting, in: The Thirty-Eighth Annual Conference on Neural Information Processing Systems, 2024.
[41] Ziyao Li, Kolmogorov-Arnold networks are radial basis function networks, 2024.
[42] Sidharth S.S., Keerthana A.R., Gokul R., Anas K.P., Chebyshev polynomial-based Kolmogorov-Arnold networks: An efficient architecture for nonlinear function approximation, 2024.
[43] Yihang Wang, Yuying Qiu, Peng Chen, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, Chenjuan Guo, LightGTS: A lightweight general time series forecasting model, in: Forty-Second International Conference on Machine Learning.
[44] Xinyu Zhang, Shanshan Feng, Jianghong Ma, Huiwei Lin, Xutao Li, Yunming Ye, Fan Li, Yew Soon Ong, FRNet: Frequency-based rotation network for long-term time series forecasting, in: Ricardo Baeza-Yates, Francesco Bonchi (Eds.), Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024, ACM, 2024, pp. 3586–3597.