

FreSia: Frequency-Semantic Instantiation and Alignment for Multivariate Time Series Analysis

Yubo Wang^a, Hui He^b, Hezhe Qiao^b, Guoqing Ji^a, Zhendong Niu^{a,*}

^a*School of Computer Science and Technology, Beijing Institute of Technology, No.5 Zhongguancun South Street, Beijing, 100081, China*

^b*School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, 178902, Singapore*

Abstract

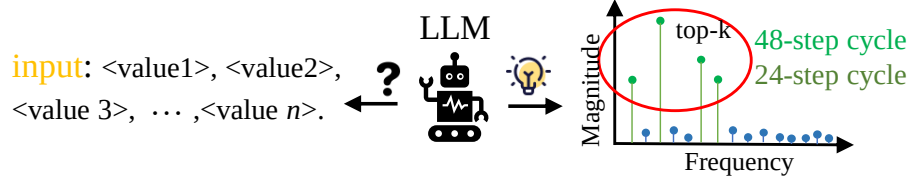
Large Language Models (LLMs) have shown strong potential in multivariate time series forecasting and anomaly detection. Existing studies predominantly inject temporal information into LLMs via direct numerical tokenization or heuristic textual descriptions. However, LLMs still face difficulty in perceiving the underlying structural patterns of numerical time series, particularly the seasonal and trend components obscured by discrete numerical tokens. To bridge this gap, we propose **FreSia**, a frequency-aware framework that establishes an effective alignment between the semantic space of LLMs and the frequency space of time series. Specifically, **FGPrompt**, a Frequency-Guided Prompt mechanism within FreSia, distills the frequency-domain structures of time series and projects them into prompts tailored to the semantic space of LLMs. Furthermore, we introduce a Global-driven Context Learning (**GCL**) component, which uses a global CLS-driven probe to generate global context to bridge the time-frequency domain gap and fuse the multi-modal information. Experiments on eight forecasting benchmarks show that FreSia achieves average improvements of 13.48% and 8.06% in MSE and MAE, respectively.

Keywords:

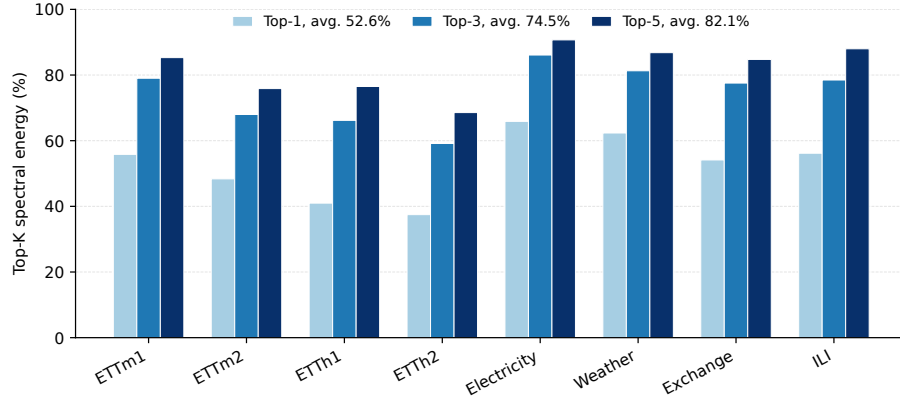
time series anomaly detection, time series forecasting, Large Language Models

*Corresponding author

Email addresses: wangyb@bit.edu.cn (Yubo Wang), zniu@bit.edu.cn (Zhendong Niu)



(a) Representation gap



(b) Spectral sparsity in real-world time series

Figure 1: (a) Representation gap between raw numerical prompts and frequency-guided semantic prompts. (b) Top-K frequency components explain a large portion of spectral energy across eight benchmark datasets. This motivates the use of dominant frequency components as compact structural primitives for frequency-guided semantic instantiation.

1. Introduction

Multivariate time series analysis plays a pivotal role in a wide range of real-world applications [1], such as traffic flow forecasting [2] and industrial anomaly detection [3]. With the rapid development of NLP, pre-trained LLMs have been widely applied to time series analysis. In particular, prompt-based LLM methods introduce textual prompts as additional inputs to aid LLMs in understanding temporal tasks. Such prompts typically contextualize the statistical features of time series with textual descriptions: either by converting data into textual representations [4] or by directly summarizing sequence information using natural language [5]. Subsequently, the prompts are encoded by a pre-trained LLM to obtain robust semantic representations for downstream tasks.

Existing LLM-based time series methods mainly inject temporal information through raw numerical serialization or time-domain statistical descriptions. For instance, Time-LLM [6] integrates domain knowledge and input statistics into descriptive prompts, which are then concatenated with temporal embeddings. TimeCMA [4] generates prompts based on historical values and trends, utilizing them as semantic anchors for cross-modal retrieval. Similarly, LangTime [7] treats time series directly as language tokens to leverage pretrained LLMs. Although these strategies make numerical data accessible to LLMs, they still suffer from an important representation mismatch.

As illustrated in Figure 1 (a), raw numerical values are dense, continuous, scale-sensitive, and tokenization-unfriendly. LLMs are primarily pretrained on discrete linguistic tokens and are effective at modeling semantic and logical relations in natural language, but such token-based representations are not naturally aligned with continuous numerical dynamics [6]. When a time series is serialized into a long sequence of numerical tokens, its underlying temporal structures, such as periodicity and seasonality, remain implicit. Frequency-domain analysis provides a structured intermediate representation by exposing periodic and harmonic components. These components can be further discretized and verbalized as symbolic cycle descriptors, such as daily cycle, half-day cycle, weekly cycle, or general x -step cycle, making them more compatible with the token-based semantic representation space of LLMs. We therefore select dominant frequency components based on their spectral energy. According to the energy-preserving property of the discrete Fourier transform, the DFT decomposes signal energy across orthogonal frequency components, making high-energy components natural candidates for capturing the main temporal variations.

As shown in Figure 1 (b), the Top-3 and Top-5 dominant frequencies explain 74.5% and 82.1% of non-DC spectral energy on average across eight benchmark datasets. This observation suggests that dense numerical sequences can be compactly summarized by a few dominant spectral primitives. It motivates us to express dominant frequency-domain structures through natural language descriptors, which provide a compact semantic abstraction of the main temporal patterns. A detailed explanation of the energy-preserving property of DFT is provided in Appendix A.

Motivated by this observation, we propose **FreSia**, a framework for MTS analy-

sis through frequency-guided prompting and global-driven contextualization. Our approach consists of two key components: **(1) Frequency-Guided Prompt (FGPrompt):** To bridge the representation gap between raw numbers and semantic reasoning, we distill the *intrinsic frequency-domain structure* of time series using FFT. By mapping Top- K frequencies into the Frequency Pool, we transform numerical observations into structured spectral primitives. **(2) Global-driven Context Learning (GCL):** To bridge the representation gap between frequency-domain structure and time-domain contextual dynamics, we introduce a Global-driven Context Learning mechanism. A global CLS-driven probe is employed to capture the context from the temporal encoder. By fusing this temporal context with spectral primitives, we can bridge time-domain and frequency-domain representations.

Our contributions are summarized as follows:

- We develop an FGPrompt to distill intrinsic spectral structures into a learnable frequency pool. This enables the LLM to retrieve frequency-domain structures rather than relying solely on numerical summary statistics.
- We propose a GCL architecture where a CLS-driven probe generates instance-specific residual offsets to calibrate the prompt space. This design bridges the gap between the time and frequency domains.
- We evaluate FreSia on both multivariate time-series forecasting and reconstruction-based anomaly detection. With task-specific heads, FreSia achieves competitive results across the evaluated forecasting and anomaly-detection benchmarks.

2. Related work

2.1. Deep Time Series Analysis

Deep learning provides an effective way to model multivariate time series by learning temporal dependencies. TimesNet [8] captures multi-period patterns through temporal 2D-variations. PatchTST [9] models long-range dependencies with patch-based representations, while iTransformer [10] captures inter-variable dependencies by multivariate-attention. Recently, SEMPO [11] further explores lightweight foundation modeling

for time-series forecasting. Frequency-domain analysis has received increasing attention in time-series forecasting. FEDformer [12] combines seasonal-trend decomposition with frequency-enhanced modeling to capture temporal patterns in the frequency domain. FredFormer [13] addresses frequency bias, FilterNet [14] learns temporal patterns through frequency filtering, and ReFocus [15] emphasizes mid- and key-frequency components. TimeKAN [16] combines frequency decomposition with multi-order KANs, capturing nonlinear dynamics across different frequency bands. FDNNet [17] disentangles informative high-frequency components from noise through information-theoretic objectives, aiming to preserve useful short-term dynamics rather than indiscriminately suppressing high-frequency information. MTCMD [18] models long- and short-term cyclical variations in a multi-dimensional temporal space and extracts seasonal and trend components across multiple scales. Despite these advances, these methods primarily operate in the numerical representation space. In anomaly detection, AdaCLIP [19] adapts CLIP with hybrid learnable prompts, while Cao et al. [20] further personalize vision-language models for zero-shot anomaly detection. More recently, a large-scale benchmark [21] was introduced to evaluate visual anomaly detection under complex view-illumination conditions. In contrast, FreSia transforms dominant spectral structures into semantic cycle descriptors and aligns them with the representation space of LLMs. Another key task in time series analysis is anomaly detection. General time-series architectures can also be adapted to anomaly detection. FPT [5] leverages pretrained language models for general time-series representation learning, while TimesNet [8] captures multi-period temporal variations and PatchTST [9] models temporal dependencies through patch-based representations. These models can be employed in reconstruction-based detection settings. In addition, there are dedicated models specifically designed for anomaly detection tasks. AnomalyTransformer [22] characterizes anomalies through association discrepancy. GCAD [23] combines graph modeling with Granger Causality to capture inter-variable relationships for anomaly detection. TimeRadar [24] introduces a domain-rotatable foundation model for time-series anomaly detection. Recently, researchers have explored combining Mixture of Experts (MoE) with graph structures. Graph-MoE [25] integrates MoE and graph learning to improve representation ability and anomaly detection performance.

2.2. LLM-empowered Time Series Analysis

The success of LLMs has sparked significant interest in adapting their reasoning capabilities. S²IP-LLM [26] aligns time-series representations with the semantic space of LLMs through prompt learning, while LLM4TS [27] adapts pre-trained LLMs for time-series forecasting through parameter-efficient alignment. Another models focus on prompt-based adaptation, which guides LLMs with textual descriptions. For instance, ChatTime [28] integrates processed temporal sequences with textual prompts as inputs to LLMs, while the model [29] leverages LLMs to learn textual representations of temporal patterns for forecasting. Furthermore, Time-LLM [6] incorporates domain knowledge and input statistics, while TimeCMA [4] and LLM-TPF [30] utilize historical trends and numerical extrema as instructions. Achieving seamless integration between numerical features and semantic prompts is critical for cross-modal forecasting. Existing studies adopt different strategies to integrate temporal and semantic representations. UniTime [31] uses domain instructions and a Language-TS Transformer for cross-modal alignment, GPT4MTS [32] jointly incorporates numerical and textual information, and STG-LLM [33] employs a tokenizer-adaptor architecture to bridge structured temporal data and LLMs. To facilitate more efficient temporal-textual exchange, ITFormer [34] bridges time series data with frozen LLMs via a dedicated interaction module, while Context-Alignment [35] further investigates the alignment between temporal contexts and LLM representations to better activate pretrained semantic knowledge for time-series modeling. More recent studies explore richer forms of contextual integration. TimeCMA [4] retrieves time-series information from prompt representations through cross-modality alignment, while TimeCAP [36] employs LLM agents to contextualize time-series data for event prediction.

LLMs have also been explored for time-series anomaly detection. LLMAD [37] leverages LLMs to provide interpretable anomaly analysis by combining detection results with semantic explanations, while TSINR [38] reconstructs temporal observations and identifies anomalies according to reconstruction discrepancies. These studies approach anomaly detection through semantic interpretation or reconstruction-based modeling. In contrast, FreSia incorporates frequency-derived structural information into semantic representations for reconstruction-based anomaly detection.

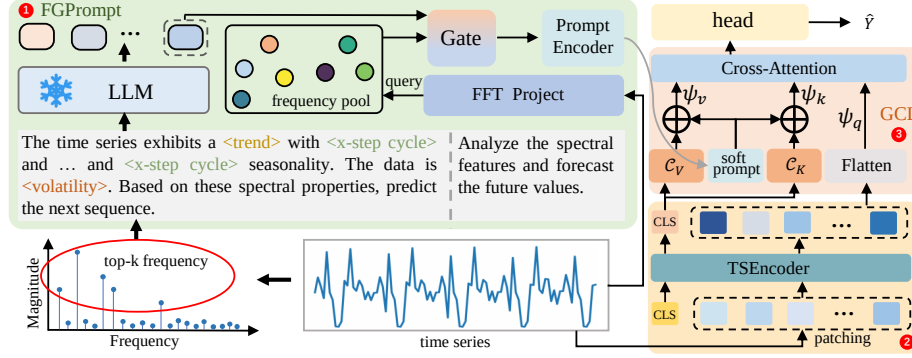


Figure 2: Framework of **FreSia**. ❶ Frequency-Guided Prompt (FGPrompt) injects frequency-domain information into prompts and the frequency pool. ❷ TSEncoder captures temporal dependencies and generates a CLS representation. ❸ Global-driven Context Learning (GCL) bridges the time-frequency domain gap and fuses multi-modal information.

3. Preliminaries

Semantic Prompt Induction. Given a multivariate time series $\mathbf{X}_L$, we derive a set of semantic instructions $\mathcal{I} = \{I_1, I_2, \dots, I_C\} \in \mathbb{R}^{C \times S}$, where each I_i corresponds to the i -th variable (channel) and has S elements.

Multivariate Time Series Forecasting. Formally, MTS forecasting uses the lookback window $\mathbf{X}_L = \{\mathbf{x}_{t-L+1}, \mathbf{x}_{t-L+2}, \dots, \mathbf{x}_t\} \in \mathbb{R}^{C \times L}$ to predict the future sequence $\mathcal{Y}_T = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \dots, \mathbf{x}_{t+T}\} \in \mathbb{R}^{C \times T}$ with $\mathcal{I}$. C is the number of variables and L represents the window length. The $\mathbf{x}_t \in \mathbb{R}^C$ represents the observed multivariate data at t .

Multivariate Time Series Anomaly Detection. Given an input $\mathbf{X}_L \in \mathbb{R}^{C \times L}$, the objective is to predict a binary sequence $\mathcal{A} = \{a_1, \dots, a_L\} \in \{0, 1\}^L$. Here, $a_i = 1$ denotes an anomaly at timestamp i , while $a_i = 0$ represents normal data. We adopt a reconstruction-based approach where the model generates $\hat{\mathbf{X}}_L = f(\mathbf{X}_L, \mathcal{I})$. An anomaly is identified if the reconstruction error $s_i = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ exceeds a specific threshold.

4. Methodology

Framework Overview. The proposed framework consists of three components: (1) **FGPrompt**, (2) **TSEncoder**, (3) **Global-driven Context Learning**. Specifically, we

leverage a frequency-driven instruction module to generate prompts. Then, the generated prompt is fused with a latent frequency pool through a gating mechanism. Concurrently, the time-series encoder extracts temporal features and global representations with a CLS token, and we use the learned CLS representation to calibrate the prompt embeddings. These multimodal representations are fused through an alignment module to forecast.

4.1. Frequency-Guided Prompt

To bridge the gap between numerical time series and the semantic space of LLMs, we propose a Frequency-Guided Prompt (FGPrompt). This module extracts time-domain and frequency-domain information from the time series to form a prompt. In this way, we can provide the model with explicit prior information about the signal’s frequency characteristics.

Property	Numerical Range	Linguistic Descriptor
Trend ($\hat{k}$)	$\hat{k} > 0.5$	strong upward trend
	$0.1 < \hat{k} \leq 0.5$	slight upward trend
	$-0.1 \leq \hat{k} \leq 0.1$	stable trend
	$-0.5 \leq \hat{k} < -0.1$	slight downward trend
	$\hat{k} < -0.5$	strong downward trend
Volatility (v)	$v > 1.0$	high volatility
	$0.5 < v \leq 1.0$	moderate volatility
	$v \leq 0.5$	smooth

Table 1: Mapping rules for linguistic descriptors in Frequency-Guided Prompts.

Feature Characterization. For each input sequence $\mathbf{x}$, FGPrompt extracts compact semantic descriptors from both frequency-domain and time-domain information. The frequency-derived cycle descriptors serve as the primary structural primitives, while trend and volatility descriptors provide auxiliary contextual cues. Specifically, dominant frequencies are identified by applying FFT to $\mathbf{x}$, and their corresponding periods

$T_i = 1/f_i$ are mapped to semantic cycle descriptors according to the sampling frequency, such as daily, weekly, or monthly cycles. Here, f_i denotes the frequency in cycles per time step. In addition, a linear fit $y = kt + b$ is used to estimate the global trend, and the coefficient of variation of first-order differences is used to characterize local volatility. These numerical features are then converted into linguistic descriptors, with the detailed quantization rules summarized in Table 1.

Semantic Prompt Formulation and Encoding. The extracted features are aggregated into a structured linguistic Prompt I , which follows a template: "The time series exhibits a [Trend] with significant [Seasonality] seasonality. The data is [Volatility]. Based on these properties, predict the next sequence." To project these semantic prompts into the latent space, we use pre-trained GPT-2 as a feature extractor. Let $\text{Tokenizer}(\cdot)$ and $\text{Enc}(\cdot)$ denote the LLM’s tokenizer and transformer backbone, respectively. The semantic embedding $\mathbf{E}_{LLM}$ is derived from the hidden state of the last token:

$$\mathbf{E}_{LLM} = \text{Enc}(\text{Tokenizer}(I)) \quad (1)$$

Frequency Pool. To bridge the gap between discrete linguistic tokens and continuous numerical features, we introduce a learnable frequency pool $\mathbf{P} = \{p_1, \dots, p_M\} \in \mathbb{R}^{M \times d}$. This pool represents the structural archetypes of time-series dynamics that can be retrieved using frequency amplitude and phase features. We use the frequency amplitude and phase features $\mathbf{f}_x$ as a query to retrieve relevant expertise from the pool:

$$\mathbf{P}_{freq} = \text{Ret}(\phi(\mathbf{f}_x), \mathbf{P}), \quad \gamma_b = \sigma\left(\mathbf{g}([\mathbf{P}_{freq} \parallel \mathbf{E}_{LLM}])\right) \quad (2)$$

Among these, ϕ is a linear mapping, and Ret denotes retrieving relevant vectors from the frequency pool via weighted methods. We concatenate the query results with the LLM-embedded prompt words, then pass this through a gated network $\mathbf{g}$ composed of a lightweight MLP to obtain the fusion weights γ_b . These are subsequently fused into $\mathcal{I}_{soft}$.

$$\mathcal{I}_{soft} = \gamma_b \mathbf{E}_{LLM} + (1 - \gamma_b) \mathbf{P}_{freq} \quad (3)$$

Then, we use a transformer-based PromptEncoder($\cdot$) to process the semantic embedding $\mathcal{I}_{soft}$. The final semantic embedding is obtained as:

$$\mathcal{I}_{soft} = \text{PromptEncoder}(\mathcal{I}_{soft}) \quad (4)$$

4.2. TSEncoder

To extract temporal dependencies and global representations from time series data, we introduce the TSEncoder. We first employ a patching strategy to process normalized time series. Subsequently, we embed the patch representations. We prepend a learnable CLS token $\mathbf{z}_{CLS}$ to the patch embeddings $\mathbf{E}_{seq}$ to aggregate global information from the input sequence. Unlike the localized patch embeddings that encode local temporal variations, the learned CLS representation summarizes information across the entire look-back window. The concatenated sequence $[\mathbf{z}_{CLS}; \mathbf{E}_{seq}]$ is then processed by the Transformer-based TSEncoder:

$$[\mathbf{E}_{Time}^{CLS}; \mathbf{E}_{Time}] = \text{TSEncoder}([\mathbf{z}_{CLS}; \mathbf{E}_{seq}]), \quad (5)$$

$$\mathbf{E}_{Time}^{Seq} = \text{FFN}(\text{Flatten}(\mathbf{E}_{Time})) \quad (6)$$

where $\mathbf{E}_{Time}^{CLS}$ denotes the learned CLS representation, and $\mathbf{E}_{Time}$ contains the contextualized patch representations. The localized features $\mathbf{E}_{Time}$ are subsequently flattened and projected to form the numerical representation $\mathbf{E}_{Time}^{Seq}$, which serves as the query ψ_q in the alignment phase.

4.3. Global-driven Context Learning

To bridge the gap between the time and frequency domains and fuse multi-modal information, we propose a dynamic contextual mechanism that modulates the prompt space based on the global feature.

We utilize the learned CLS representation $\mathbf{E}_{Time}^{CLS}$ to capture the global characteristics of the input sequence. This global representation is projected through dual transformation heads to generate instance-specific offsets, C_K and C_V , which act as residual shifts to align the generic soft prompts with the current numerical regime:

$$C_K = \text{FFN}_K(\mathbf{E}_{Time}^{CLS}), \quad C_V = \text{FFN}_V(\mathbf{E}_{Time}^{CLS}) \quad (7)$$

The calibrated prompts, ψ_k and ψ_v , are then formulated by integrating these offsets into the retrieved soft prompt embeddings $\mathcal{I}_{soft}$. In this way, we can ensure that the

semantic priors are conditioned on the specific instance’s context:

$$\psi_k = \mathcal{I}_{soft} + C_K, \quad \psi_v = \mathcal{I}_{soft} + C_V \quad (8)$$

To fuse representations from different modalities, we employ a Cross-Attention (CA) mechanism. Specifically, the flattened numerical features $\mathbf{E}_{Time}^{Seq}$ serve as the Query, while the calibrated semantic embeddings $\{\psi_k, \psi_v\}$ function as the Key and Value, respectively. This configuration allows the numerical sequence to selectively attend to the most pertinent semantic prompt. The resulting context-aware representation is integrated with the original temporal features via a residual connection and mapped through a forecasting head to yield the final prediction $\hat{\mathcal{Y}}$:

$$\hat{\mathcal{Y}} = \text{Head}(\text{CA}(\mathbf{E}_{Time}^{Seq}, \psi_k, \psi_v) + \mathbf{E}_{Time}^{Seq}) \quad (9)$$

4.4. Training and Loss function

Before model training, FreSia first converts the generated frequency-guided prompts into semantic embeddings. Specifically, for each input window, FGPrompt generates a textual prompt that describes the dominant frequency-derived cycles together with auxiliary temporal contexts such as trend and volatility. The prompt is then tokenized and fed into a frozen pretrained language model to obtain its semantic embedding. The language model is not updated during training, and the prompt embeddings can be precomputed and saved before the forecasting model is optimized. Therefore, during training and inference, FreSia only uses the cached semantic embeddings together with the time-series input, avoiding repeated online calls to the language model.

We adopt the Huber loss as the optimization objective to improve robustness against outliers in time-series observations. The Huber loss is defined as:

$$\mathcal{L} = \begin{cases} \frac{1}{2}(\mathcal{Y} - \hat{\mathcal{Y}})^2 & \text{for } |\mathcal{Y} - \hat{\mathcal{Y}}| \leq \delta \\ \delta(|\mathcal{Y} - \hat{\mathcal{Y}}| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (10)$$

In our implementation, we set $\delta = 1.0$. Under this setting, the Huber loss is equivalent to the SmoothL1 loss in PyTorch.

Models		FreSia		LangTime		TimeCMA		Time-LLM		UniTime	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.310	0.348	0.319	0.348	0.312	0.351	0.359	0.381	0.322	0.363
	192	0.352	0.374	0.368	0.375	0.361	0.378	0.383	0.393	0.366	0.387
	336	0.378	0.396	0.413	0.402	0.392	0.401	0.416	0.414	0.398	0.407
	720	0.430	0.427	0.487	0.439	0.453	0.438	0.483	0.449	0.454	0.440
	Avg	0.367	0.386	0.397	0.391	0.380	0.392	0.410	0.409	0.385	0.399
ETTm2	96	0.169	0.251	0.188	0.258	0.173	0.258	0.193	0.280	0.183	0.266
	192	0.237	0.296	0.245	0.297	0.238	0.301	0.257	0.318	0.251	0.310
	336	0.298	0.337	0.301	0.336	0.297	0.338	0.317	0.353	0.319	0.351
	720	0.392	0.395	0.402	0.393	0.393	0.394	0.419	0.411	0.420	0.410
	Avg	0.274	0.320	0.284	0.321	0.275	0.323	0.296	0.340	0.293	0.334
ETTh1	96	0.361	0.389	0.391	0.388	0.373	0.391	0.398	0.410	0.397	0.418
	192	0.406	0.418	0.429	0.419	0.427	0.421	0.451	0.440	0.434	0.439
	336	0.419	0.430	0.462	0.440	0.458	0.448	0.473	0.451	0.468	0.457
	720	0.446	0.455	0.458	0.445	0.449	0.460	0.469	0.470	0.469	0.477
	Avg	0.408	0.423	0.435	0.423	0.423	0.431	0.448	0.443	0.442	0.448
ETTh2	96	0.270	0.328	0.299	0.336	0.286	0.336	0.295	0.345	0.296	0.345
	192	0.347	0.380	0.374	0.382	0.363	0.387	0.386	0.399	0.374	0.394
	336	0.355	0.394	0.410	0.418	0.406	0.421	0.419	0.429	0.415	0.427
	720	0.400	0.428	0.418	0.426	0.417	0.438	0.425	0.442	0.425	0.444
	Avg	0.343	0.382	0.375	0.391	0.372	0.397	0.381	0.404	0.378	0.403
Electricity	96	0.145	0.243	0.181	0.266	0.146	0.245	0.172	0.265	0.196	0.287
	192	0.163	0.260	0.185	0.273	0.165	0.264	0.182	0.279	0.199	0.291
	336	0.180	0.276	0.198	0.281	0.235	0.336	0.195	0.288	0.214	0.305
	720	0.212	0.306	0.241	0.320	0.322	0.403	0.233	0.320	0.254	0.335
	Avg	0.175	0.271	0.201	0.285	0.217	0.312	0.195	0.288	0.216	0.306
Weather	96	0.150	0.192	0.178	0.202	0.167	0.211	0.198	0.235	0.171	0.214
	192	0.202	0.241	0.211	0.245	0.212	0.253	0.240	0.269	0.217	0.254
	336	0.259	0.284	0.269	0.286	0.270	0.292	0.295	0.308	0.274	0.293
	720	0.336	0.335	0.351	0.346	0.350	0.348	0.368	0.353	0.351	0.343
	Avg	0.236	0.263	0.252	0.270	0.250	0.276	0.275	0.291	0.253	0.276
Exchange	96	0.081	0.200	0.089	0.201	0.115	0.243	0.087	0.208	0.086	0.209
	192	0.177	0.299	0.175	0.298	0.209	0.330	0.178	0.302	0.174	0.299
	336	0.318	0.407	0.329	0.409	0.374	0.449	0.338	0.422	0.319	0.408
	720	0.781	0.668	0.852	0.690	1.083	0.803	0.819	0.681	0.875	0.701
	Avg	0.339	0.394	0.361	0.400	0.445	0.456	0.356	0.403	0.364	0.404
ILI	24	1.580	0.792	4.295	1.537	1.996	0.998	2.383	1.004	2.346	0.954
	36	1.373	0.717	4.272	1.588	1.906	0.915	2.390	0.993	1.998	0.912
	48	1.730	0.803	4.264	1.552	1.867	0.868	2.394	1.003	1.979	0.912
	60	1.568	0.802	4.076	1.518	1.920	0.904	2.562	1.049	2.109	0.938
	Avg	1.563	0.779	4.227	1.549	1.922	0.921	2.432	1.012	2.108	0.929

Table 2: Results of multivariate time series forecasting. The input sequence length is 36 for the ILI dataset and 96 for others. The best results are in red and the second best are blue.

5. Experiments

5.1. Experimental Setup

Datasets and Evaluation Metrics. For MTSF, we evaluate FreSia on 8 datasets: four ETT (ETTh1/h2, ETTm1/m2) [39], Exchange [40], Weather [41], Electricity [41], and

Models		FreSia		OFA		iTransformer		TimesNet		RLinear	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.310	0.348	0.335	0.369	0.334	0.368	0.338	0.375	0.355	0.376
	192	0.352	0.374	0.374	0.385	0.377	0.391	0.374	0.387	0.387	0.392
	336	0.378	0.396	0.407	0.406	0.426	0.420	0.410	0.411	0.424	0.415
	720	0.430	0.427	0.469	0.442	0.491	0.459	0.478	0.450	0.487	0.450
	Avg	0.367	0.386	0.396	0.401	0.407	0.410	0.400	0.406	0.413	0.408
ETTm2	96	0.169	0.251	0.190	0.275	0.180	0.264	0.187	0.267	0.182	0.265
	192	0.237	0.296	0.253	0.313	0.250	0.309	0.249	0.309	0.246	0.304
	336	0.298	0.337	0.321	0.360	0.311	0.348	0.321	0.351	0.307	0.342
	720	0.392	0.395	0.402	0.406	0.412	0.407	0.408	0.403	0.407	0.398
	Avg	0.274	0.320	0.294	0.339	0.288	0.332	0.291	0.333	0.286	0.327
ETTh1	96	0.361	0.389	0.398	0.424	0.386	0.405	0.384	0.402	0.386	0.395
	192	0.406	0.418	0.449	0.427	0.441	0.436	0.434	0.429	0.437	0.424
	336	0.419	0.430	0.492	0.466	0.487	0.458	0.491	0.469	0.479	0.446
	720	0.446	0.455	0.487	0.483	0.503	0.491	0.521	0.500	0.481	0.470
	Avg	0.408	0.423	0.457	0.450	0.454	0.447	0.458	0.450	0.446	0.434
ETTh2	96	0.270	0.328	0.312	0.360	0.297	0.349	0.340	0.374	0.318	0.363
	192	0.347	0.380	0.387	0.405	0.380	0.400	0.402	0.414	0.401	0.412
	336	0.355	0.394	0.424	0.437	0.428	0.432	0.452	0.452	0.436	0.442
	720	0.400	0.428	0.433	0.453	0.427	0.445	0.462	0.468	0.442	0.454
	Avg	0.343	0.382	0.389	0.414	0.383	0.407	0.414	0.427	0.399	0.418
Electricity	96	0.145	0.243	0.197	0.290	0.148	0.240	0.168	0.272	0.201	0.281
	192	0.163	0.260	0.201	0.292	0.162	0.253	0.184	0.289	0.201	0.283
	336	0.180	0.276	0.217	0.309	0.178	0.269	0.198	0.300	0.215	0.298
	720	0.212	0.306	0.253	0.339	0.225	0.317	0.220	0.320	0.257	0.331
	Avg	0.175	0.271	0.217	0.308	0.178	0.270	0.192	0.295	0.219	0.298
Weather	96	0.150	0.192	0.203	0.244	0.174	0.214	0.172	0.220	0.192	0.232
	192	0.202	0.241	0.247	0.277	0.221	0.254	0.219	0.261	0.240	0.271
	336	0.259	0.284	0.297	0.311	0.278	0.296	0.280	0.306	0.292	0.307
	720	0.336	0.335	0.368	0.356	0.358	0.349	0.365	0.359	0.364	0.353
	Avg	0.236	0.263	0.279	0.297	0.258	0.278	0.259	0.287	0.272	0.291
Exchange	96	0.081	0.200	0.084	0.202	0.086	0.206	0.107	0.234	0.093	0.217
	192	0.177	0.299	0.177	0.298	0.177	0.299	0.226	0.344	0.184	0.307
	336	0.318	0.407	0.339	0.420	0.331	0.417	0.367	0.448	0.351	0.432
	720	0.781	0.668	0.856	0.696	0.847	0.691	0.964	0.746	0.886	0.714
	Avg	0.339	0.394	0.364	0.404	0.360	0.403	0.416	0.443	0.379	0.418
ILI	24	1.580	0.792	2.732	1.100	2.347	1.731	2.317	0.934	2.436	0.970
	36	1.373	0.717	2.664	1.063	2.468	0.998	1.972	0.920	2.362	0.953
	48	1.730	0.803	2.617	1.041	2.489	1.016	2.238	0.913	2.461	0.982
	60	1.568	0.802	2.478	1.035	2.471	1.065	2.027	0.928	2.451	0.979
	Avg	1.563	0.779	2.623	1.060	2.444	1.203	2.139	0.924	2.428	0.971

Table 3: Results of multivariate time series forecasting. The input sequence length is 36 for the ILI dataset and 96 for others. The best results are in red and the second best are blue.

the ILI [41] dataset. We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics. For anomaly detection, we evaluate on five datasets:

Dataset	Metric	FreSia	TSINR	FPT	TimesNet	Anomaly.	LightTS	DLinear
SMD	P	88.56	83.09	87.27	87.98	78.72	87.04	87.27
	R	84.07	80.46	81.08	81.54	65.43	78.39	80.99
	F1	86.25	81.76	84.06	84.64	71.46	82.49	84.01
PSM	P	97.61	99.21	98.55	98.51	98.76	98.29	98.66
	R	96.22	89.37	95.79	96.27	83.25	93.60	94.70
	F1	96.91	94.04	97.15	97.38	90.35	95.89	96.64
SWaT	P	96.08	99.31	92.12	90.75	99.73	92.36	80.91
	R	92.64	72.32	93.06	95.40	68.07	93.32	95.30
	F1	94.33	83.69	92.59	93.02	80.91	92.84	87.52
MSL	P	87.19	83.57	82.03	89.55	89.78	89.17	89.68
	R	84.09	85.40	82.01	75.29	73.66	73.64	75.31
	F1	85.61	84.47	82.02	81.80	80.93	80.66	81.87
SMAP	P	88.47	87.55	90.91	89.92	90.14	90.02	89.89
	R	70.90	55.32	61.01	56.56	54.00	53.90	54.01
	F1	78.72	67.80	73.02	69.44	67.54	67.43	67.48
Avg	P	91.58	90.55	90.18	91.34	80.25	91.38	91.55
	R	85.58	76.57	82.59	81.01	83.95	78.57	79.62
	F1	88.36	82.35	85.77	85.26	80.50	83.86	84.54

Table 4: Results of anomaly detection. The input sequence length is 100. The best results are in red and the second best are blue.

SMD [42], PSM [43], SWaT [44], MSL [45], SMAP [45], using precision (P), recall (R), and F1-score (F1) as evaluation metrics.

Baselines. For MTSF, we select eight representative forecasting models as baselines. These include (1) Prompt-based LLMs: LangTime [7], TimeCMA [4], Time-LLM [6], UniTime [31]; (2) LLM-based: OFA [5]; (3) Transformer-based model: iTransformer [10]; (4) CNN-based method: TimesNet [8]; (5) Linear-based method: RLinear [46]. For anomaly detection, we use two kinds of models, including: (1) general frameworks designed for time series modeling: FPT [5], TimesNet [8], LightTS [47] and DLinear [48]; (2) models specifically tailored for time series anomaly detection: TSINR [38], AnomalyTransformer [22]. All experiments are implemented in PyTorch 2.1.2 and on an NVIDIA RTX 4090.

5.2. Main Results

Long-term Forecasting. Table 2 and Table 3 report the forecasting results on eight datasets. At the dataset-average level, FreSia achieves the lowest MSE on all eight

datasets and the lowest or tied-lowest MAE on seven of them. In particular, the improvement is more pronounced on ILI, where FreSia obtains an average MSE/MAE of 1.563/0.779, compared with 1.922/0.921 for the strongest competing result, corresponding to reductions of approximately 18.7% and 15.4%, respectively. FreSia also achieves average MSE/MAE values of 0.343/0.382 on ETTh2, 0.236/0.263 on Weather, and 0.339/0.394 on Exchange, achieving lower average errors than the compared LLM-based and conventional forecasting models on these datasets. On Electricity, FreSia obtains the lowest average MSE of 0.175, while its MAE of 0.271 is marginally higher than the 0.270 achieved by iTransformer. Overall, these results indicate that the performance gains are observed across datasets with different sampling frequencies and temporal characteristics, rather than being concentrated on a single benchmark. We further compare FreSia with representative frequency-domain baselines, including TimeKAN, PaiFilter, TexFilter, and FredFormer. The detailed results are reported in Appendix C.

Anomaly Detection. Table 4 reports the anomaly detection results on five multivariate benchmarks. FreSia achieves the best average performance with 91.58% precision, 85.58% recall, and 88.36% F1 score, outperforming the strongest competing baseline FPT by 2.59 percentage points in average F1. At the dataset level, FreSia obtains the best F1 score on four out of five datasets, including SMD, SWaT, MSL, and SMAP. Notably, FreSia achieves the highest average recall among all methods, indicating that it can better identify anomalous patterns while maintaining competitive precision. These results suggest that the learned frequency-semantic representations are not limited to forecasting, but also generalize well to reconstruction-based anomaly detection. We further provide a diagnostic comparison of different scoring rules in Appendix B.

5.3. Sensitivity of Frequency Pool Size

We conduct parameter sensitivity analysis on the frequency pool size with ETTh2 720 horizon. As can be seen from Figure 3, the frequency pool size affects the prediction results. If the frequency pool size is too small, it may not fully represent the latent features in the time series. If it is too large, it will cause redundancy and also affect the prediction results.

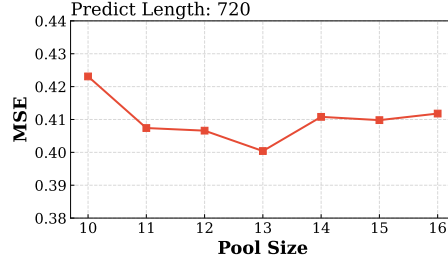


Figure 3: Sensitivity analysis of frequency pool size.

5.4. Ablation

Table 5 presents the ablation study results. We evaluate the contribution of each module by averaging metrics across all forecasting horizons.

Core Backbones. Both the LLM and Time-Series Encoder (TSE) contribute to forecasting performance. Removing the LLM (w/o LLM) increases the prediction errors, indicating the contribution of its semantic representations. Removing the TSE (w/o TSE) results in the largest degradation among the evaluated variants, highlighting the importance of temporal dependency modeling.

Datasets	ETTh2		ETTm2		ECL	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
w/o LLM	0.357	0.388	0.288	0.331	0.186	0.271
w/o TSE	0.393	0.417	0.321	0.358	0.382	0.424
w/o PE	0.353	0.387	0.279	0.325	0.177	0.274
w/o CA	0.356	0.390	0.291	0.333	0.179	0.268
w/o CLS	0.356	0.388	0.282	0.327	0.179	0.276
w/o Pool	0.352	0.387	0.279	0.324	0.183	0.278
FreSia	0.343	0.382	0.274	0.320	0.175	0.271

Table 5: Ablation studies of FreSia

Prompting and Pooling. The Prompt Encoder (w/o PE) has a relatively minor impact, yet it facilitates cross-modal alignment. In contrast, excluding the Frequency Pool

Dataset	FreSia		TimeCMA-style		w/o Freq. Path.		w/o LLM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	0.236	0.263	0.237	0.264	0.238	0.264	0.252	0.273
Exchange	0.339	0.394	0.383	0.417	0.392	0.420	0.404	0.425
ILI	1.563	0.779	1.577	0.775	1.601	0.782	1.696	0.787
Avg.	0.713	0.479	0.732	0.485	0.743	0.489	0.784	0.495

Table 6: Disentangling the effects of frequency-domain information and LLM semantic encoding.

(w/o Pool) consistently impairs forecasting precision. This suggests that the frequency-domain prototype pool contributes to effective retrieval.

Alignment Mechanisms. The alignment components provide consistent gains. Removing CLS-driven calibration (w/o CLS) reduces accuracy on several datasets, showing that global profiling helps handle regime shifts. Additionally, replacing Cross-Attention (w/o CA) yields inferior results, suggesting that attention-based fusion is more effective than simple concatenation for modeling complex interactions.

5.5. Analysis

Disentangling Frequency Information and Semantic Encoding. Table 6 reports results on representative datasets with different temporal characteristics, including Weather, Exchange, and ILI, for a compact diagnostic analysis. Removing the LLM encoder increases the average error from 0.713/0.479 to 0.784/0.495, indicating that the frozen LLM provides useful semantic representations. Meanwhile, the w/o Freq. Path. variant retains the semantic encoder but removes frequency descriptors in prompts and disables the frequency-related pool, which degrades the average performance to 0.743/0.489. This shows that the gains do not come solely from the LLM encoder; explicit frequency-domain information also contributes to performance. We further compare with a TimeCMA-style prompt, which replaces the frequency-guided prompt with a raw-value-style textual prompt. Its average performance, 0.732/0.485, is worse than FreSia, suggesting that frequency-semantic instantiation provides more effective structural cues than raw numerical prompt serialization. The degradation is more pronounced on Exchange and

ILI, while Weather shows a milder difference. Overall, these results demonstrate that FreSia benefits from the combination of frequency-domain structural cues and LLM-based semantic encoding.

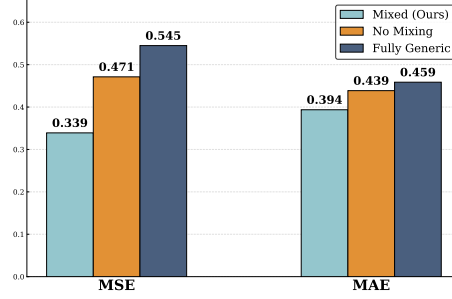


Figure 4: Effect of mixed prompting strategy on Exchange.

General Prompting Strategies. We further analyze how general prompts interact with task-specific frequency prompts. As shown in Figure 4, the mixed prompting strategy achieves the lowest errors on the Exchange dataset, with 0.339 MSE and 0.394 MAE. Compared with the no-mixing setting, it reduces MSE and MAE by 28.0% and 10.3%, respectively. Compared with the fully generic prompting setting, it reduces MSE and MAE by 37.8% and 14.2%, respectively.

These results indicate that general prompts are beneficial only when combined with task-specific frequency information. Using general prompts alone introduces overly generic semantic cues, while removing them entirely limits the contextual richness of the prompt. In contrast, the mixed prompting strategy provides both frequency-aware temporal structure and general semantic context, leading to more accurate predictions.

Mechanistic Analysis of CLS. To evaluate the impact of the CLS token, we analyze the attention within the Cross-Attention (CA) mechanism. Specifically, we visualize the residual attention map (the difference between configurations with and without CLS) and focus on the 50 channels exhibiting the most pronounced variations.

As illustrated in Figure 5 (a) and (b), the attention differences are not uniformly distributed across channels. While several channels exhibit noticeable changes after introducing the CLS-driven calibration, others remain relatively stable. This suggests

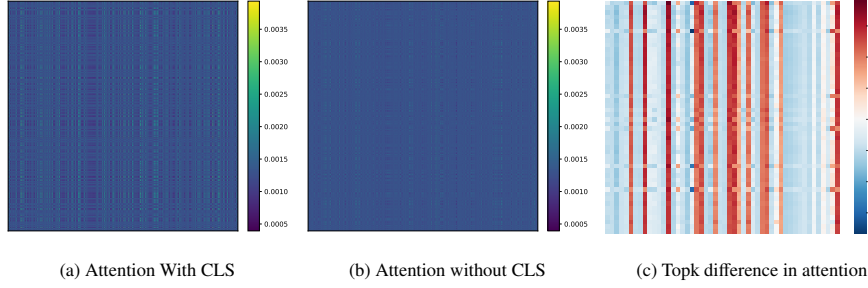


Figure 5: Attention difference analysis of CLS-driven calibration.

that the CLS token may help the model adaptively reweight channel-level responses rather than perturbing all representations equally. In Figure 5 (c), the observed local patterns of positive and negative attention changes indicate that CLS-driven calibration can modulate the interaction between semantic prompts and temporal features in a structured manner. These observations provide qualitative evidence that the CLS token contributes to context-aware alignment between global temporal information and semantic prompts, which is consistent with the performance degradation observed when removing CLS in the ablation study.

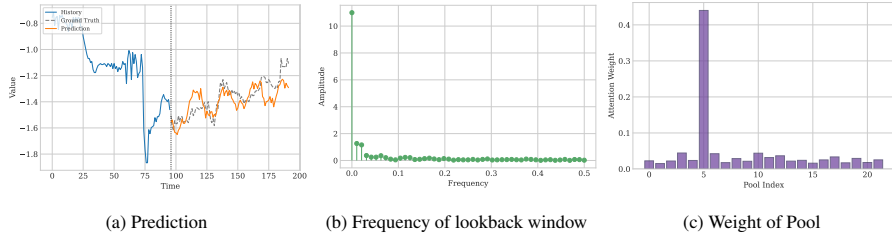


Figure 6: ETTh1 frequency pool analysis case1.

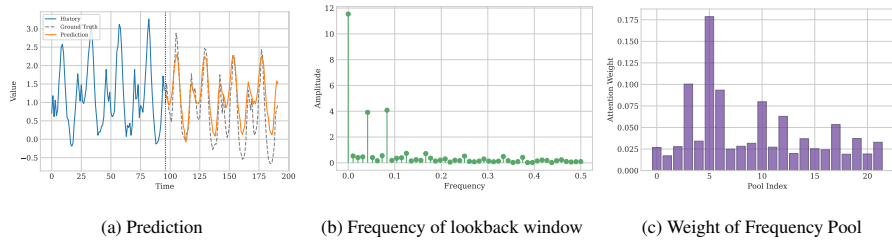


Figure 7: ETTh1 frequency pool analysis case2.

Analysis of Frequency Pool. To better understand the behavior of the Frequency Pool, we provide case studies in Figures 6 and 7. Subfigure (a) visualizes the forecasting results. Subfigure (b) shows the frequency domain distribution via FFT, while subfigure (c) displays the corresponding Frequency Pool weights. In Figure 6(b), a prominent low-frequency peak aligns with a dominant weight in Figure 6(c). Similarly, Figure 7 shows multiple high-frequency components that correlate with specific weight distributions in Figure 7(c). These observations support the effectiveness of the FFT-driven indexing mechanism. Furthermore, the results suggest that the Frequency Pool learns relatively decoupled frequency-domain prototypes.

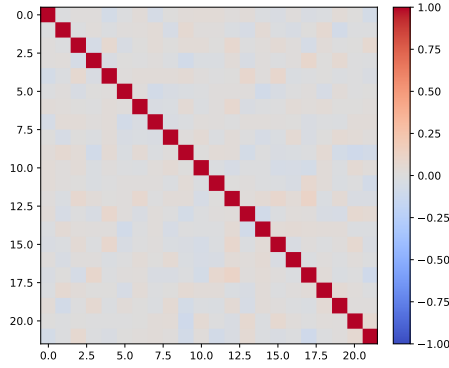


Figure 8: Frequency pool Gram matrix.

We visualize the cosine similarity matrix of the ETTh1 frequency pool to investigate its representation space. As shown in Figure 8, the matrix exhibits a distinct near-identity structure, where diagonal elements approach 1 and off-diagonal entries remain near 0. This high degree of orthogonality indicates that each prompt within the pool captures unique frequency-domain prototypes.

Impact of Loss Functions. To evaluate the influence of the optimization objective, we compare the performance of our model using Huber loss against two standard benchmarks: Mean Squared Error (MSE) and Mean Absolute Error (MAE). Figure 9 shows that using the Huber loss leads to better performance than MSE and MAE losses. This suggests that the Huber loss provides a better balance between sensitivity to small errors and robustness to large residuals.

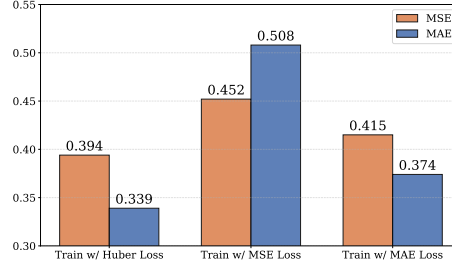


Figure 9: Comparison with different losses.

Efficiency Analysis. Table 7 reports the computational efficiency of FreSia and representative baselines on the ECL dataset with a prediction horizon of 336 and a batch size of 8. The reported test time is measured after prompt embeddings are precomputed and saved. Under the cached-prompt evaluation setting, FreSia achieves efficient inference with 11.29 seconds of test time. This is faster than TimeCMA, TimeKAN, iTransformer, and OFA. Although FreSia adopts a frozen LLM as the semantic encoder, the prompt embeddings are precomputed and cached before forecasting. Therefore, the LLM does not need to be repeatedly invoked during online testing, making the additional semantic extraction cost amortized.

In terms of memory and parameters, FreSia uses 2936 MiB GPU memory and 28.29M parameters, which is comparable to TimeCMA with 29.53M parameters and 2690 MiB memory. OFA has fewer parameters and lower GPU memory, but its test time is substantially higher in our evaluation. These results show that FreSia introduces manageable computational overhead while maintaining efficient inference.

Model	Test Time (s)	GPU Memory (MiB)	Params (M)
TimeCMA	16.78	2690	29.53
OFA	1697.66	916	3.92
iTransformer	129.98	2248	4.96
TimeKAN	44.32	3556	0.044
FreSia	11.29	2936	28.29

Table 7: Efficiency comparison. We report test time, peak GPU memory usage, and model parameters under the same evaluation setting.

Furthermore, we evaluate the computational overhead by comparing the token consumption of different prompting strategies. The TimeCMA prompt template contains approximately 226 tokens, whereas the FreSia prompt contains approximately 32 tokens under the same tokenization setting, corresponding to about a $7.1\times$ reduction in prompt length.

6. Conclusion

In this paper, we present an LLM-enhanced framework for multivariate time-series forecasting and anomaly detection. FreSia converts frequency-domain structures into semantic prompts and uses CLS-driven calibration to align these prompts with temporal representations. Experiments on eight forecasting benchmarks and five anomaly-detection benchmarks show that the resulting frequency-semantic representations are effective across both tasks.

7. Authorship contribution statement

Yubo Wang: Conceptualization, Methodology, Validation, Visualization, Writing—original draft, Writing- review & editing. Hui He: Conceptualization, Investigation, Writing- review & editing, Supervision. Hezhe Qiao: Conceptualization, Investigation, Writing- review & editing. Guoqing Ji: Investigation, Writing- review. Zhendong Niu: Conceptualization, Investigation, Writing- review & editing, Supervision.

8. Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the author(s) used ChatGPT and Gemini in order to translate and polish this paper. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Appendix A. Energy-preserving property of DFT

The theoretical motivation for selecting dominant frequency components can be further explained by the energy-preserving property of the discrete Fourier transform (DFT). Let $\mathbf{x} \in \mathbb{R}^N$ be a real-valued time series, where $n = 0, \dots, N-1$ denotes the time index. Its DFT is denoted as $\mathbf{X} = \mathcal{F}(\mathbf{x})$, where $X[k]$ is the complex Fourier coefficient at the k -th discrete frequency bin, and $k = 0, \dots, N-1$ denotes the frequency-bin index. Under the standard DFT convention,

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi kn}{N}\right), \quad (\text{A.1})$$

where j is the imaginary unit. The zero-frequency coefficient $X[0]$ is the DC component, which corresponds to the sum of the sequence values and reflects the mean level of the sequence after normalization by N .

According to Parseval's theorem [49], the signal energy is preserved between the time and frequency domains:

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2. \quad (\text{A.2})$$

This relation indicates that the magnitude spectrum provides a principled decomposition of signal energy over frequency components.

Following the implementation used in our spectral sparsity analysis, the retained energy ratio is computed on the one-sided spectrum returned by the real-valued FFT. The DC component is excluded because it mainly reflects the mean level rather than periodic variations. Let

$$\mathcal{I} = \{1, \dots, \lfloor N/2 \rfloor\} \quad (\text{A.3})$$

denote the set of one-sided positive non-DC frequency indices. For each $k \in \mathcal{I}$, its normalized frequency is $f_k = k/N$ cycles per time step, and the corresponding period is $T_k = N/k$ time steps. If the sampling interval is Δt , then $f_k = k/(N\Delta t)$ and $T_k = N\Delta t/k$.

The frequency indices in $\mathcal{I}$ are ranked by their one-sided spectral energy $|X[k]|^2$. Let $\mathcal{S}_K \subseteq \mathcal{I}$ denote the set of the K frequency indices with the largest $|X[k]|^2$. The retained non-DC spectral energy ratio is defined as

$$\rho_K = \frac{\sum_{k \in \mathcal{S}_K} |X[k]|^2}{\sum_{k \in \mathcal{I}} |X[k]|^2}. \quad (\text{A.4})$$

A large ρ_K indicates that a small number of dominant frequency components preserve most of the one-sided non-DC spectral energy. Therefore, selecting Top- K frequency components provides a compact approximation of the main temporal variations in the Fourier basis. This supports the use of dominant frequencies as structural primitives for frequency-guided semantic instantiation.

Appendix B. Reconstruction for anomaly detection

Dataset	Time Residual	Time-Freq. Residual	Freq. Residual
SMD	0.863	0.392	0.162
PSM	0.969	0.875	0.781
SWaT	0.943	0.913	0.846
MSL	0.856	0.171	0.080
SMAP	0.787	0.641	0.637
Avg.	0.884	0.598	0.501

Table B.1: F1 comparison of anomaly scoring rules. Time Residual denotes the standard reconstruction-error score used by FreSia. Time-Freq. Residual combines time-domain and frequency-domain residuals with equal weight. Freq. Residual uses only the frequency residual.

Time Residual. The standard time-domain score computes the point-wise reconstruction error by averaging over variables: $s_t(i) = \frac{1}{C} \sum_{c=1}^C (x_{i,c} - \hat{x}_{i,c})^2$, where i denotes the time index. This score preserves temporal localization because each timestamp has its own anomaly score.

Freq. Residual. For the frequency-domain score, FFT is applied along the temporal dimension of each window. The DC component is removed to avoid domination by the mean level. The frequency-domain residual is then computed as the average magnitude difference between the spectra of the original and reconstructed windows:

$$s_f^{\text{win}} = \frac{1}{FC} \sum_{k=1}^F \sum_{c=1}^C \left| |\mathcal{F}(\mathbf{x})_{k,c}| - |\mathcal{F}(\hat{\mathbf{x}})_{k,c}| \right|, \quad (\text{B.1})$$

where F denotes the number of non-DC frequency bins. Since this frequency residual is computed at the window level, it is broadcast to all timestamps within the window: $s_f(i) = s_f^{\text{win}}$, $i = 1, \dots, L$. Thus, Freq. Residual captures window-level spectral mismatch but does not provide point-wise temporal localization.

Time-Freq. Residual. To avoid scale mismatch between the time-domain and frequency-domain scores, both scores are normalized using the mean and standard deviation computed from the training set: $\tilde{s}_t = \frac{s_t - \mu_t}{\sigma_t + \epsilon}$, $\tilde{s}_f = \frac{s_f - \mu_f}{\sigma_f + \epsilon}$. The combined score is then defined as $s_{tf} = \alpha \tilde{s}_t + (1 - \alpha) \tilde{s}_f$, where $\alpha = 0.5$ is used in our diagnostic experiment.

Table B.1 shows that the standard Time Residual achieves the best F1 score on all five datasets, with the highest average F1 of 0.884. In contrast, Time-Freq. Residual and Freq. Residual obtain much lower average F1 scores of 0.598 and 0.501, respectively. This suggests that directly adding a frequency-domain residual is not a robust scoring strategy for anomaly detection in this setting. One possible reason is that the frequency residual measures window-level spectral mismatch and is assigned to all timestamps in the window, which weakens point-wise anomaly localization. Therefore, the standard reconstruction-error score is retained, and anomaly detection is used to evaluate representation generalization rather than to claim a new anomaly-specific scoring rule.

Appendix C. Comparison with Frequency-domain Models

Table C.1 compares FreSia with representative frequency-domain forecasting models. Although these baselines also exploit spectral information, they mainly model frequency-domain patterns through numerical transformations or filtering. In contrast, FreSia further converts dominant frequency components into semantic cycle descriptors and aligns them with LLM representations. FreSia achieves the best average performance on 7 out of 8 datasets and ranks first in 14 out of 16 dataset-level average metrics. This shows that the gains of FreSia cannot be attributed merely to the use of frequency-domain information. Instead, the results suggest that frequency-semantic instantiation provides additional benefits over purely numerical frequency-domain modeling. TexFilter performs slightly better on ECL, indicating that frequency filtering remains effective in some settings; nevertheless, FreSia shows stronger overall performance across diverse benchmarks.

Models		FreSia		TimeKAN		PaiFilter		TexFilter		FredFormer	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.310	0.348	0.322	0.361	0.318	0.358	0.321	0.361	0.328	0.363
	192	0.352	0.374	0.357	0.383	0.364	0.383	0.367	0.387	0.367	0.382
	336	0.378	0.396	0.383	0.402	0.396	0.406	0.401	0.409	0.395	0.403
	720	0.429	0.427	0.446	0.436	0.456	0.444	0.477	0.448	0.454	0.440
	Avg	0.367	0.386	0.377	0.395	0.384	0.398	0.392	0.401	0.386	0.397
ETTm2	96	0.169	0.251	0.174	0.255	0.174	0.257	0.175	0.258	0.178	0.261
	192	0.237	0.296	0.239	0.299	0.240	0.300	0.240	0.301	0.244	0.303
	336	0.298	0.337	0.301	0.340	0.297	0.339	0.311	0.347	0.302	0.341
	720	0.392	0.395	0.395	0.396	0.392	0.393	0.414	0.405	0.397	0.396
	Avg	0.274	0.320	0.277	0.323	0.276	0.322	0.285	0.328	0.280	0.325
ETTh1	96	0.361	0.389	0.367	0.395	0.375	0.394	0.382	0.402	0.376	0.394
	192	0.406	0.418	0.414	0.420	0.436	0.422	0.430	0.429	0.440	0.426
	336	0.419	0.430	0.445	0.434	0.476	0.443	0.472	0.451	0.472	0.440
	720	0.446	0.455	0.444	0.459	0.474	0.469	0.481	0.473	0.490	0.466
	Avg	0.408	0.423	0.417	0.427	0.440	0.432	0.441	0.439	0.444	0.432
ETTh2	96	0.270	0.328	0.290	0.340	0.292	0.343	0.293	0.343	0.293	0.343
	192	0.346	0.380	0.375	0.392	0.369	0.395	0.374	0.396	0.370	0.390
	336	0.355	0.394	0.423	0.435	0.420	0.432	0.417	0.430	0.385	0.413
	720	0.400	0.428	0.443	0.449	0.430	0.446	0.449	0.460	0.419	0.439
	Avg	0.343	0.382	0.383	0.404	0.378	0.404	0.383	0.407	0.367	0.396
ECL	96	0.145	0.243	0.174	0.266	0.176	0.264	0.147	0.245	0.148	0.242
	192	0.163	0.260	0.182	0.273	0.185	0.270	0.160	0.250	0.164	0.258
	336	0.180	0.276	0.197	0.286	0.202	0.286	0.173	0.267	0.178	0.274
	720	0.212	0.306	0.236	0.320	0.242	0.319	0.210	0.309	0.213	0.301
	Avg	0.175	0.271	0.197	0.286	0.201	0.285	0.172	0.268	0.176	0.269
Weather	96	0.150	0.192	0.162	0.208	0.164	0.210	0.162	0.207	0.163	0.207
	192	0.202	0.241	0.207	0.249	0.214	0.252	0.210	0.250	0.209	0.250
	336	0.259	0.284	0.263	0.290	0.268	0.293	0.265	0.290	0.265	0.292
	720	0.336	0.335	0.338	0.341	0.344	0.342	0.342	0.340	0.342	0.341
	Avg	0.236	0.263	0.242	0.272	0.247	0.274	0.245	0.272	0.245	0.272
Exchange	96	0.081	0.200	0.087	0.206	0.083	0.202	0.091	0.211	0.089	0.208
	192	0.177	0.299	0.181	0.299	0.174	0.296	0.186	0.305	0.181	0.301
	336	0.318	0.407	0.347	0.426	0.326	0.413	0.380	0.449	0.352	0.426
	720	0.781	0.668	0.995	0.748	0.840	0.670	0.896	0.712	0.988	0.736
	Avg	0.339	0.394	0.402	0.420	0.356	0.395	0.388	0.419	0.402	0.418
ILI	24	1.581	0.792	2.392	0.945	2.500	0.970	2.095	0.852	2.379	0.947
	36	1.373	0.717	1.935	0.879	2.425	1.015	2.099	0.885	2.323	0.954
	48	1.730	0.803	2.244	0.935	2.551	1.028	2.075	0.880	2.657	1.070
	60	1.568	0.802	2.090	0.960	2.582	1.038	2.070	0.902	2.994	1.140
	Avg	1.563	0.779	2.165	0.930	2.514	1.013	2.085	0.880	2.588	1.028

Table C.1: Comparison with frequency-domain baselines. The input sequence length is 36 for the ILI dataset and 96 for others. The best results are in red and the second best are blue.

Appendix D. Robustness to Auxiliary Trend / Volatility Mappings

Table D.1 evaluates whether FreSia relies on the fixed trend and volatility thresholds in the default FGPrompt mapping rules. The adaptive-quantile variant replaces

Variant	ETTm2		ECL		Avg.	
	MSE	MAE	MSE	MAE	MSE	MAE
Adaptive-Quantile	0.276	0.321	0.177	0.274	0.227	0.298
Continuous	0.274	0.320	0.178	0.274	0.226	0.297
Freq-only	0.276	0.322	0.179	0.276	0.228	0.299
FreSia	0.274	0.320	0.175	0.271	0.224	0.296

Table D.1: Robustness to auxiliary trend/volatility mappings. The results are averaged over all horizons.

fixed thresholds with quantile-based thresholds estimated from the training set, while the continuous variant removes discrete trend/volatility labels and directly uses the corresponding continuous scores. These alternatives achieve comparable results to the default mapping, indicating that FreSia is not sensitive to a specific threshold design. Moreover, the freq-only variant removes all trend and volatility descriptors, thereby completely eliminating these hand-crafted thresholds, but still remains close to the full model. These results show that the hard-coded trend/volatility mappings are not the core source of FreSia’s performance; the main structural cues come from frequency-derived cycle descriptors.

References

- [1] Q. Wen, L. Yang, T. Zhou, L. Sun, Robust time series analysis and applications: An industrial perspective, in: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 2022, pp. 4836–4837.
- [2] Y. Fang, Y. Liang, B. Hui, Z. Shao, L. Deng, X. Liu, X. Jiang, K. Zheng, Efficient large-scale traffic forecasting with transformers: A spatial data management perspective, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1, 2025, pp. 307–317.
- [3] L. Pemula, D. Zhang, O. Dabeer, Robust ad: A real world benchmark dataset for robustness in industrial anomaly detection, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 4047–4057.
- [4] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, R. Zhao, Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 18780–18788.
- [5] T. Zhou, P. Niu, L. Sun, R. Jin, et al., One fits all: Power general time series analysis by pretrained lm, Advances in neural information processing systems 36 (2023) 43322–43355.
- [6] M. Jin, S. Wang, L. Ma, Z. Chu, J. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, Q. Wen, Time-llm: Time series forecasting by reprogramming large language models, in: B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, Y. Sun (Eds.), International Conference on Learning Representations, 2024.
- [7] W. Niu, Z. Xie, Y. Sun, W. He, M. Xu, C. Hao, LangTime: A language-guided unified model for time series forecasting with proximal policy optimization, in: A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, J. Zhu (Eds.), Proceedings of the 42nd International Conference on Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, PMLR, 2025, pp. 46712–46734.

- [8] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, Timesnet: Temporal 2d-variation modeling for general time series analysis, in: ICLR, OpenReview.net, 2023.
- [9] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers, in: International Conference on Learning Representations, 2023.
- [10] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, iTransformer: Inverted Transformers Are Effective for Time Series Forecasting, in: International Conference on Learning Representations, 2024.
- [11] H. He, K. Yi, Y. Ma, Q. Zhang, Z. Niu, G. Pang, SEMPO: lightweight foundation models for time series forecasting, in: NeurIPS, 2025.
- [12] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International conference on machine learning, PMLR, 2022, pp. 27268–27286.
- [13] X. Piao, Z. Chen, T. Murayama, Y. Matsubara, Y. Sakurai, Fredformer: Frequency debiased transformer for time series forecasting, in: Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining, 2024, pp. 2400–2410.
- [14] K. Yi, J. Fei, Q. Zhang, H. He, S. Hao, D. Lian, W. Fan, Filtarnet: Harnessing frequency filters for time series forecasting, *Advances in Neural Information Processing Systems* 37 (2024) 55115–55140.
- [15] G. Yu, Y. Li, J. Wang, X. Guo, A. I. Avilés-Rivero, T. Yang, S. Wang, Refocus: Reinforcing mid-frequency and key-frequency modeling for multivariate time series forecasting, *CoRR abs/2502.16890* (2025).
- [16] S. Huang, Z. Zhao, C. Li, L. BAI, TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting, in: The Thirteenth International Conference on Learning Representations, 2025.

- [17] A. Hu, L. Wen, J. Duan, Y. Dai, D. Wang, S. Huang, J. Wang, Z. Xu, Fdnet: High-frequency disentanglement network with information-theoretic guidance for multivariate time series forecasting, *Pattern Recognit.* 173 (2026) 112810.
- [18] F. Zhang, L. Yuan, W. Zhang, M. Zhang, H. Wang, Multi-scale temporal correlation multi-dimensional decomposition network for time series analysis, *Pattern Recognition* (2026) 113140.
- [19] Y. Cao, J. Zhang, L. Frittoli, Y. Cheng, W. Shen, G. Boracchi, Adaclip: Adapting CLIP with hybrid learnable prompts for zero-shot anomaly detection, in: *ECCV* (35), volume 15093 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 55–72.
- [20] Y. Cao, X. Xu, Y. Cheng, C. Sun, Z. Du, L. Gao, W. Shen, Personalizing vision-language models with hybrid prompts for zero-shot anomaly detection, *IEEE Trans. Cybern.* 55 (2025) 1917–1929.
- [21] Y. Cao, Y. Cheng, Y. Zhang, X. Xu, Y. Zhang, Y. Sun, Y. Tan, X. Huang, C. Huang, W. Shen, Visual anomaly detection under complex view-illumination interplay: A large-scale benchmark, *Pattern Recognit.* 179 (2026) 113666.
- [22] J. Xu, H. Wu, J. Wang, M. Long, Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy, in: *International Conference on Learning Representations*, 2022.
- [23] Z. Liu, M. Gao, P. Jiao, Gcad: Anomaly detection in multivariate time series from the perspective of granger causality, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2025, pp. 19041–19049.
- [24] H. He, H. Qiao, Y. Chen, K. Yi, G. Pang, Timeradar: A domain-rotatable foundation model for time series anomaly detection, in: *KDD* (2), ACM, 2026, pp. 1602–1613.
- [25] X. Huang, W. Chen, B. Hu, Z. Mao, Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection, in: *Proceedings*

- of the AAAI conference on artificial intelligence, volume 39, 2025, pp. 17476–17484.
- [26] Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, D. Song, s^2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting, in: Forty-first International Conference on Machine Learning, 2024.
 - [27] C. Chang, W.-Y. Wang, W.-C. Peng, T.-F. Chen, Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters, *ACM Transactions on Intelligent Systems and Technology* 16 (2025) 1–20.
 - [28] C. Wang, Q. Qi, J. Wang, H. Sun, Z. Zhuang, J. Wu, L. Zhang, J. Liao, Chat-time: A unified multimodal time series foundation model bridging numerical and textual data, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 12694–12702.
 - [29] J. Xiong, C. Wang, H. Sun, Y. Jing, Q. Qi, Z. Zhuang, L. Zhang, J. Liao, J. Wang, Beyond statistical analysis: Multimodal framework for time series forecasting with llm-driven temporal pattern, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, 2025, pp. 6696–6704.
 - [30] Q. Pan, H. Tan, G. Shen, X. Kong, M. Wang, C. Xu, Llm-tpf: Multiscale temporal periodicity-semantic fusion llms for time series forecasting, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, 2025, pp. 6030–6038.
 - [31] X. Liu, J. Hu, Y. Li, S. Diao, Y. Liang, B. Hooi, R. Zimmermann, Unitime: A language-empowered unified model for cross-domain time series forecasting, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 4095–4106.
 - [32] F. Jia, K. Wang, Y. Zheng, D. Cao, Y. Liu, Gpt4mts: Prompt-based large language model for multimodal time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 23343–23351.
 - [33] L. Liu, S. Yu, R. Wang, Z. Ma, Y. Shen, How can large language models understand spatial-temporal data?, *CoRR* abs/2401.14192 (2024).

- [34] Y. Wang, P. Lei, J. Song, Y. Hao, T. Chen, Y. Zhang, L. Jia, Y. Li, Z. Wei, ITFormer: Bridging time series and natural language for multi-modal QA with large-scale multitask dataset, in: A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, J. Zhu (Eds.), Proceedings of the 42nd International Conference on Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, PMLR, 2025, pp. 63324–63344.
- [35] Y. Hu, Q. Li, D. Zhang, J. Yan, Y. Chen, Context-alignment: Activating and enhancing llms capabilities in time series, in: International Conference on Learning Representations, 2025.
- [36] G. Lee, W. Yu, K. Shin, W. Cheng, H. Chen, Timecap: Learning to contextualize, augment, and predict time series events with large language model agents, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 18082–18090.
- [37] J. Liu, C. Zhang, J. Qian, M. Ma, S. Qin, C. Bansal, Q. Lin, S. Rajmohan, D. Zhang, Large language models can deliver accurate and interpretable time series anomaly detection, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, 2025, pp. 4623–4634.
- [38] M. Li, K. Liu, H. Chen, J. Bu, H. Wang, H. Wang, Tsinr: capturing temporal continuity via implicit neural representations for time series anomaly detection, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1, 2025, pp. 671–682.
- [39] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 11106–11115.
- [40] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long-and short-term temporal patterns with deep neural networks, in: The 41st international ACM SIGIR conference on research & development in information retrieval, 2018, pp. 95–104.

- [41] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, *Advances in neural information processing systems* 34 (2021) 22419–22430.
- [42] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.
- [43] A. Abdulaal, Z. Liu, T. Lancewicki, Practical approach to asynchronous multivariate time series anomaly detection and localization, in: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2485–2494.
- [44] A. P. Mathur, N. O. Tippenhauer, Swat: A water treatment testbed for research and training on ics security, in: *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, IEEE, 2016, pp. 31–36.
- [45] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.
- [46] Z. Li, S. Qi, Y. Li, Z. Xu, Revisiting long-term time series forecasting: An investigation on linear mapping, *CoRR abs/2305.10721* (2023).
- [47] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, J. Li, Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures, *CoRR abs/2207.01186* (2022).
- [48] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 2023, pp. 11121–11128.
- [49] A. V. Oppenheim, *Discrete-time signal processing*, Pearson Education India, 1999.