

---

# A Dynamic Multiscale Anti-Aliasing Network for Time Series Forecasting

## Appendix

---

### CONTENTS

<b>A Preliminaries</b>	<b>3</b>
A.1 Problem Statement . . . . .	3
A.2 Problem Description: Aliasing in Multi-Scale Time Series Downsampling . . . . .	3
A.3 An intuitive explanation of aliasing-related concepts . . . . .	4
A.3.1 The Core Problem: Spectral Overlap and Aliasing . . . . .	4
A.3.2 A Deeper Dive into the Scenarios of Figure.2 . . . . .	5
<b>B Proofs of the Equivalent Sampling Rate (ESR)</b>	<b>7</b>
B.1 Notations and Signal Modeling . . . . .	7
B.1.1 Original Discrete-Time Signal . . . . .	7
B.1.2 Module Structure . . . . .	7
B.2 Downsampling and Aliasing Conditions . . . . .	7
B.3 Linear Mapping with Depth-wise Convolution and Point-wise Convolution . . . . .	7
B.3.1 Depth-wise Convolution . . . . .	7
B.3.2 Point-wise Convolution . . . . .	8
B.3.3 Unified Linear Mapping . . . . .	8
B.4 Rank Constraint and Degrees of Freedom Counting . . . . .	8
B.5 Definition of Equivalent Sampling Rate . . . . .	9
<b>C Implementation Details</b>	<b>10</b>
C.1 Datasets details . . . . .	10
C.2 Baseline details . . . . .	10
C.3 Implementation details. . . . .	10
C.4 Fair comparison settings. . . . .	10
C.5 Hyperparameter settings. . . . .	12
<b>D Full Results</b>	<b>14</b>
D.1 Error Bars . . . . .	14
D.2 Long-term Forecasting . . . . .	14
D.3 Short-term Forecasting . . . . .	14
D.4 Univariate Forecasting . . . . .	15
D.5 Results for Hyperparameter Analysis . . . . .	21
<b>E More details of Computational Costs</b>	<b>23</b>

---

054	E.1	Efficiency and Scalability Analysis on Synthetic Data . . . . .	23
055	E.2	Efficiency Comparison with State-of-the-Art Models on Real-World Datasets . . .	24
056			
057			
058	<b>F</b>	<b>More details of Pre-Sampling Filtering</b>	<b>25</b>
059			
060	<b>G</b>	<b>More details of Our Method</b>	<b>27</b>
061			
062	G.1	The Rationale for the Embedding First Architecture . . . . .	27
063			
064	G.1.1	The Pitfall of Premature Multi-Scale Decomposition . . . . .	27
065	G.1.2	Embed First: A Principled Approach in a Unified Feature Space . . . . .	27
066	G.1.3	Empirical Validation . . . . .	28
067	G.2	Principled Anti-Aliasing via Dynamic Frequency Cutoff . . . . .	28
068			
069	G.2.1	The Rationale: Focusing on Learnable Core Patterns . . . . .	28
070	G.2.2	Dynamic Adaptability and Parameter-Free Design . . . . .	28
071	G.2.3	Addressing the High-Frequency Information Trade-off . . . . .	29
072			
073			
074	<b>H</b>	<b>More details of Dependency Modeling</b>	<b>30</b>
075			
076	<b>I</b>	<b>The Use of Large Language Models</b>	<b>34</b>
077			
078	<b>J</b>	<b>REFERENCES</b>	<b>35</b>
079			

---

081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

## A PRELIMINARIES

### A.1 PROBLEM STATEMENT

**Time Series.** Time series  $\mathbf{X} \in \mathbb{R}^{C \times N}$  refers to a sequence of data points ordered by time, where  $N$  denotes the total number of timestamps and  $C$  represents the number of channels at each timestamp. Time series forecasting involves predicting future data points based on historical time series observations. The historical observations can be represented as  $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L] \in \mathbb{R}^{C \times L}$ , and  $L$  is the length of the historical look-back window. The future data for the next  $L_{\text{next}}$  time steps, denoted as  $\hat{X}_o = [\mathbf{x}_{L+1}, \mathbf{x}_{L+2}, \dots, \mathbf{x}_{L+L_{\text{next}}}] \in \mathbb{R}^{C \times L_{\text{next}}}$ , correspond to the forecast horizon. Given these, time series forecasting models are required to learn mapping functions  $\mathbf{F} : X \in \mathbb{R}^{C \times L} \rightarrow \hat{X}_o \in \mathbb{R}^{C \times L_{\text{next}}}$ .

**Aliasing.** This issue arises when different high-frequency components in a continuous signal are indistinguishably mapped to the same low-frequency components after sampling or improper downsampling. Formally, let the sampling interval be  $\Delta t$ , with the Nyquist frequency defined as  $f_{\text{Nyquist}} = \frac{1}{2\Delta t}$ . Any frequency component  $f > f_{\text{Nyquist}}$  in the signal will alias to a spurious frequency  $\tilde{f} = |f - k \cdot f_s|$  in the sampled sequence  $X$ , where  $f_s = \frac{1}{\Delta t}$  is the sampling rate, and  $k \in \mathbb{Z}^+$  ensures  $\tilde{f} \leq f_{\text{Nyquist}}$ . This may occur when the sampling rate or downsampling operations fail to meet the Nyquist criterion, that is, the sampling frequency must be at least twice the highest frequency in the original signal. If not resolved, high-frequency components would fold back into lower frequencies during downsampling, creating spurious artifacts.

### A.2 PROBLEM DESCRIPTION: ALIASING IN MULTI-SCALE TIME SERIES DOWNSAMPLING

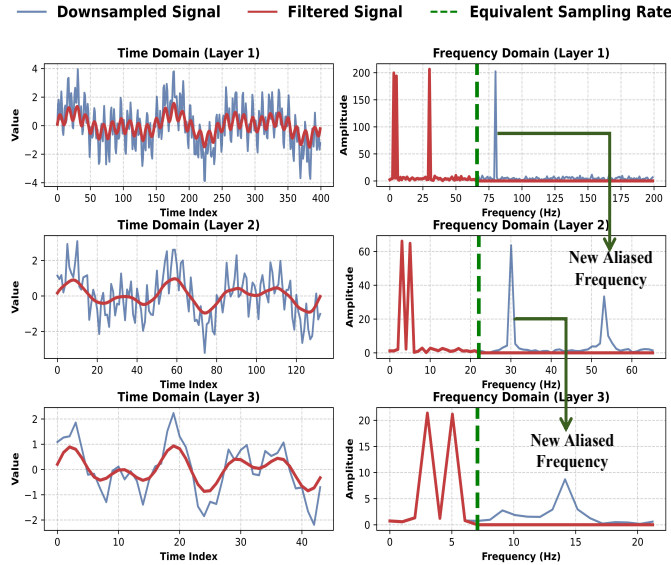


Figure 1: Left: filtering effect of the anti-aliasing filters; Right: emergence of new aliased frequencies.

In Figure.1, we present a case study exploring the critical role of anti-aliasing filters in signal preservation during multi-scale downsampling. By downsampling a synthetic signal containing both high-frequency and low-frequency components, we demonstrate the occurrence of aliasing during the reduction of the sampling rate.

The synthetic time series signal used in the study consists of several frequency components: low-frequency components (3 Hz and 5 Hz), high-frequency components (30 Hz and 80 Hz), and Gaus-

sian noise. The signal is initially sampled at a rate of 400 Hz. Subsequently, we perform multi-scale downsampling at different levels (each with a window size of 3), resulting in sampling rates of 400 Hz, 133 Hz, and 44 Hz.

According to the Nyquist sampling theorem, the Nyquist frequency is half of the sampling rate. Therefore, a 400 Hz sampling rate is sufficient to accurately sample the frequency components of the original signal. The calculation of aliasing frequencies is derived from the spectral periodicity characteristics of the Nyquist sampling theorem Nyquist (1928), based on the formula:

$$f_{\text{alias}} = |f_o - k \cdot f_s|, \quad (1)$$

where  $f_{\text{alias}}$  is the aliased frequency,  $f_o$  is the original high-frequency component,  $k$  is an integer representing the multiple mapping to the sampling frequency, and  $f_s$  is the sampling rate.

In the first layer, the Nyquist frequency is 200 Hz, corresponding to a sampling rate of 400 Hz. Given that the highest frequency component of the signal is 80 Hz, which is well below the Nyquist frequency, no aliasing occurs; all frequency components can be accurately sampled and reconstructed. In practical applications, an anti-aliasing filter limits frequency components above 66 Hz, thereby preventing aliasing and removing high-frequency noise.

In the second layer, the Nyquist frequency is reduced to 66.67 Hz (corresponding to a sampling rate of approximately 133.33 Hz), which results in aliasing of the original 80 Hz high-frequency component. According to the aliasing formula (1), for  $f_o = 80$  Hz and with  $k = 1$ :

$$f_{\text{alias}} = |80 - 1 \times 133.33| \approx 53.33 \text{ Hz}. \quad (2)$$

This calculation indicates that the 80 Hz component folds into the lower frequency region, specifically within the 50–60 Hz range, thereby introducing non-original frequency components and causing spectral distortion. The anti-aliasing filter in this layer effectively removes frequencies above 22 Hz to mitigate this issue.

In the third layer, the Nyquist frequency further decreases to 22.22 Hz (with a corresponding sampling rate of approximately 44.44 Hz), leading to the aliasing of the original 30 Hz component. Using the aliasing formula with  $k = 1$ :

$$f_{\text{alias}} = |30 - 1 \times 44.44| \approx 14.44 \text{ Hz}, \quad (3)$$

indicating that the 30 Hz component folds around 14 Hz. Additionally, due to the interaction between the sampling rate and the sampling process, frequency components in the 10–15 Hz range cannot be accurately represented, even though they lie below the Nyquist frequency. This aliasing phenomenon becomes particularly significant as the frequencies approach the Nyquist limit. Nevertheless, the anti-aliasing filter is still able to extract the true frequency information with reasonable accuracy, thereby alleviating the impact of aliasing.

### A.3 AN INTUITIVE EXPLANATION OF ALIASING-RELATED CONCEPTS

This appendix provides a detailed explanation of the core signal processing concepts illustrated in Figure.2, which motivate the design of DMANet. We structure this explanation to clarify the relationship between sampling, spectral overlap, aliasing, and our proposed solution.

#### A.3.1 THE CORE PROBLEM: SPECTRAL OVERLAP AND ALIASING

The central challenge DMANet addresses is aliasing, a form of signal distortion that occurs during downsampling. To fully understand this phenomenon, it is crucial to distinguish between its **physical cause spectral overlap** and its **perceptual consequence aliasing**. These two terms describe different links in a cause-and-effect chain, where aliasing is the direct result of spectral overlap due to improper sampling Zhou et al. (2025).

- **Sampling and Spectral Replicas:** When a signal is sampled, its original spectrum is periodically replicated along the frequency axis, creating what are known as spectral replicas. Grabinski et al. (2022) The act of sampling also limits the frequency range we can observe without distortion to a new, narrower baseband (from 0 Hz to the new Nyquist frequency).



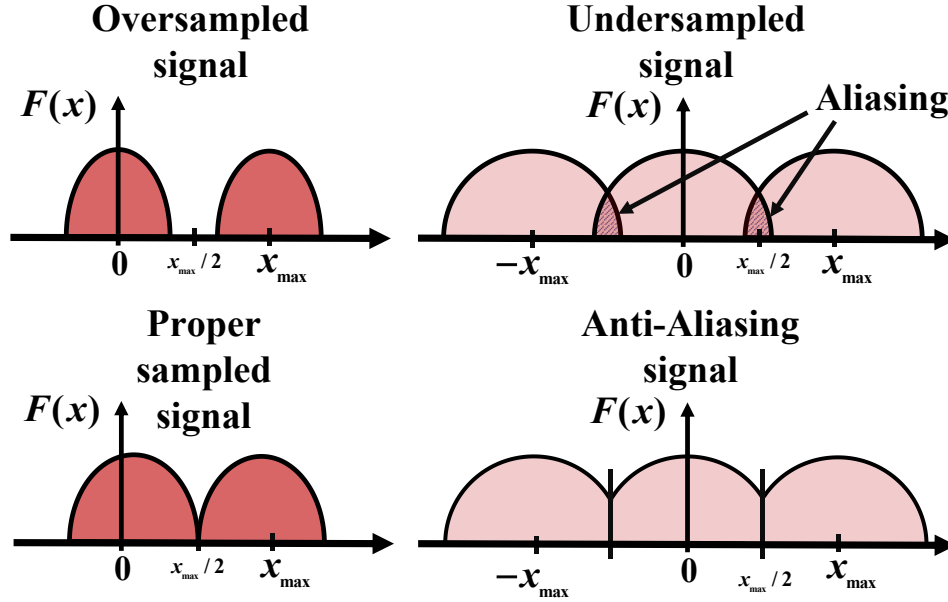


Figure 2: A conceptual illustration of the sampling process in the frequency domain. **Top-left:** Oversampling provides a wide guard band, preventing aliasing. **Top-right:** Undersampling causes spectral replicas to overlap, leading to aliasing where high frequencies (hatched areas) are misrepresented as low frequencies. **Bottom-left:** Proper (or critical) sampling meets the Nyquist criterion exactly, with replicas touching but not overlapping. **Bottom-right:** An anti-aliasing filter removes high-frequency content before sampling, ensuring that even with a lower sampling rate, no overlap occurs.

- **The Cause: Spectral Overlap.** If the sampling rate is too low to satisfy the Nyquist criterion, the separation between these spectral replicas becomes insufficient. This prevents the formation of a safety margin (Guard Band), causing them to physically overlap. This physical overlap, illustrated in the undersampled signal panel of Figure 2, is the root cause of the problem.
- **The Consequence: Aliasing.** This overlap is the direct cause of aliasing. Any high-frequency component from the original signal that exceeds the new Nyquist frequency is folded back into the new, observable low-frequency baseband. This process causes the original high-frequency information to appear as a spurious low frequency. This spurious frequency then mixes with the true low-frequency components within the baseband, becoming indistinguishable from them. Ultimately, this misrepresentation of high-frequency information as low-frequency information corrupts the signal's fidelity and is the core problem we address in our paper.

### A.3.2 A DEEPER DIVE INTO THE SCENARIOS OF FIGURE.2

Figure.2 visualizes four key scenarios:

- **The Undesirable Case (Top-Right):** The undersampled scenario is precisely the adverse outcome our work aims to prevent. The resulting aliasing (hatched areas) erroneously introduces spurious low-frequency patterns, a distortion that can severely hinder analysis and forecasting.
- **The Ideal and Safe Case (Top-Left):** The oversampled scenario is ideal because it successfully avoids aliasing. The empty space between the spectral replicas is a Guard Band, which does not imply information loss but rather a safe margin ensuring that the original spectrum can be unambiguously recovered.
- **The Theoretical Boundary (Bottom-Left):** Proper sampled (or Critical Sampling) occurs when the sampling rate is exactly twice the signal's maximum frequency ( $f_s = 2 \cdot f_{max}$ ).

---

270 The spectral replicas touch edge-to-edge without overlapping. While theoretically sound,  
271 operating at this exact boundary is risky in practice.

272

273 • **The DMANet Approach (Bottom-Right):** The Anti-Aliasing signal panel illustrates the  
274 core principle of our work. Before an operation that would otherwise cause undersam-  
275 pling, an **anti-aliasing filter** is applied. This low-pass filter removes the high-frequency  
276 components (the part of the spectrum above  $x_{max}/2$ ) that would cause overlap. After this  
277 pre-filtering, even a lower sampling rate can be safely applied without generating aliasing  
278 artifacts.

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

## B PROOFS OF THE EQUIVALENT SAMPLING RATE (ESR)

### B.1 NOTATIONS AND SIGNAL MODELING

#### B.1.1 ORIGINAL DISCRETE-TIME SIGNAL

First, we define  $X[n] \in \mathbb{R}^{C_i}$  as a discrete-time signal with  $C_i$  channels. The  $i$ -th channel of  $X[n]$ , that is,  $X_i[n]$ , is obtained by sampling a continuous-time signal  $x_i(t)$ :

$$X_i[n] = x_i(nT_s), \quad T_s = \frac{1}{f_s}, \quad i = 1, \dots, C_i, \quad (4)$$

where  $f_s$  is the original sampling rate and  $T_s$  is the sampling period. It is assumed that each continuous-time signal  $x_i(t)$  is band-limited to  $|\omega| \leq \omega_B$ .

#### B.1.2 MODULE STRUCTURE

The input signal  $X[n]$  passes sequentially through a depth-wise convolution and a point-wise convolution defined in the architecture of DMANet:

$$X[n] \xrightarrow{\text{DepthwiseConv1d}(K, S)} U[m] \xrightarrow{\text{PointwiseConv1d}} V[m]. \quad (5)$$

First, the DepthwiseConv1d operation, characterized by a kernel  $h_i[k]$  of length  $K$  for each  $i$ -th input channel, where  $1 \leq k \leq K$ , and a stride  $S$ , transforms  $X[n]$  into an intermediate signal  $U[m] \in \mathbb{R}^{C_i}$ . Subsequently, this intermediate signal  $U[m]$  is processed by a PointwiseConv1d operation. This second stage uses a convolution matrix  $W \in \mathbb{R}^{C_o \times C_i}$ , with elements  $w_{j,i}$ , to map the  $C_i$  channels of  $U[m]$  to the  $C_o$  output channels, producing the final output signal  $V[m] \in \mathbb{R}^{C_o}$ . Consequently, the output sampling rate of the entire module is  $f'_s = f_s/S$ .

### B.2 DOWNSAMPLING AND ALIASING CONDITIONS

As a baseline reference, consider directly downsampling the original signal  $X[n]$  by a factor  $S$  without any convolution filtering to obtain the signal  $Y[m]$ :

$$Y[m] = X[Sm]. \quad (6)$$

The new sampling rate is  $f'_s = f_s/S$ . To avoid aliasing caused directly by downsampling, the bandwidth  $B = \omega_B/(2\pi)$  of the original continuous-time signal must satisfy the Nyquist-Shannon sampling theorem requirement, which is defined as:

$$B \leq \frac{f'_s}{2} = \frac{f_s}{2S}. \quad (7)$$

Expressed in terms of normalized angular frequency  $\Omega_B = \omega_B T_s = 2\pi B T_s$ , we solve this equation and represent the condition to avoid aliasing as:

$$\Omega_B \leq \frac{\pi}{S}. \quad (8)$$

This condition applies to ideal direct downsampling, assuming that perfect anti-aliasing filtering has been performed before downsampling to remove frequency components above  $\pi/S$ , corresponding to  $\frac{f_s}{2S}$ . Our proposed DMANet contains depth-wise and point-wise modules which perform filtering in its workflow, and its behaviors are more complex.

### B.3 LINEAR MAPPING WITH DEPTH-WISE CONVOLUTION AND POINT-WISE CONVOLUTION

The operations of these two modules can be expressed as a series of linear mappings.

#### B.3.1 DEPTH-WISE CONVOLUTION

The output of the depth-wise convolution for the  $i$ -th channel,  $U_i[m]$ , is computed as follows:

$$U_i[m] = \sum_{k=0}^{K-1} h_i[k] X_i[mS + k]. \quad (9)$$

Here,  $m$  is the index for the output sequence. Due to the stride  $S$ , the output  $U_i[m]$  depends on the input  $X_i[n]$  in the range from  $n = mS$  to  $n = mS + K - 1$ .

### B.3.2 POINT-WISE CONVOLUTION

The  $j$ -th output channel  $V_j[m]$  is obtained by a linear combination of  $U_i[m]$ :

$$V_j[m] = \sum_{i=1}^{C_i} w_{j,i} U_i[m]. \quad (10)$$

Then, we substitute equation 9 into the above equation to get the expression of  $V_j[m]$ :

$$V_j[m] = \sum_{i=1}^{C_i} w_{j,i} \left( \sum_{k=1}^K h_i[k] X_i[mS + k] \right) = \sum_{i=1}^{C_i} \sum_{k=1}^K w_{j,i} h_i[k] X_i[mS + k]. \quad (11)$$

### B.3.3 UNIFIED LINEAR MAPPING

To form a unified linear transformation at each output time  $m$ , we construct an input vector  $\mathbf{x}_m$  that contains all original input samples involved in computing  $V[m]$ :

$$\mathbf{x}_m = \begin{bmatrix} X_1[mS] \\ \vdots \\ X_1[mS + K - 1] \\ \vdots \\ X_{C_i}[mS] \\ \vdots \\ X_{C_i}[mS + K - 1] \end{bmatrix} \in \mathbb{R}^{C_i K}.$$

This is a column vector formed by stacking  $K$  consecutive samples, starting from  $mS$ , from each of the  $C_i$  channels. Concurrently, a weight matrix  $G \in \mathbb{R}^{C_o \times (C_i K)}$  is constructed. For the  $j$ -th row of  $G$ , its elements correspond to  $w_{j,i} h_i[k]$  and are arranged according to the order of the elements in  $\mathbf{x}_m$ . Specifically, if the  $p$ -th element of  $\mathbf{x}_m$  is  $X_i[mS + k]$ , then the weight in  $G$  corresponds to the output  $V_j[m]$  and this input element is  $G_{j,p} = w_{j,i} h_i[k]$ . Thus, the output  $V[m]$  can be defined as:

$$V[m] = G \mathbf{x}_m, \quad V[m] \in \mathbb{R}^{C_o}. \quad (12)$$

This equation shows that at each output time  $m$ , the output vector  $V[m]$  is a linear mapping of a local window  $\mathbf{x}_m$  of the input signal.

### B.4 RANK CONSTRAINT AND DEGREES OF FREEDOM COUNTING

A fundamental property of linear algebra states that the rank of the matrix  $G$ , denoted as  $\text{rank}(G)$ , is limited by its dimensions:

$$\text{rank}(G) \leq \min\{\text{number of rows}, \text{number of columns}\} = \min\{C_o, C_i K\}. \quad (13)$$

We assume that the values of the weights  $h_i[k]$  and  $w_{j,i}$  are generic. They can be learned and are not overly sparse or linearly dependent, so that matrix  $G$  can achieve its theoretically maximum possible rank. Then, the dimension of independent information, also named the degrees of freedom, that the module extracts from the  $C_i K$ -dimensional input window  $\mathbf{x}_m$  and transmits to the  $C_o$ -dimensional output  $V[m]$  at each output time  $m$  is:

$$D = \text{rank}(G) = \min\{C_i K, C_o\}. \quad (14)$$

This  $D$  represents the maximum dimension of linearly independent information from the input segment  $\mathbf{x}_m$  that the system can distinguish or represent, without considering noise or specific signal statistics.

To relate this total degree of freedom  $D$  to each channel of the input signal, we can average it over the  $C_i$  input channels. Thus, the equivalent temporal degrees of freedom  $\alpha$  contributed by each input channel to produce one output sample  $V[m]$  is:

$$\alpha = \frac{D}{C_i} = \frac{\min\{C_i K, C_o\}}{C_i} = \min\left\{\frac{C_i K}{C_i}, \frac{C_o}{C_i}\right\} = \min\left\{K, \frac{C_o}{C_i}\right\}. \quad (15)$$

Here,  $\alpha$  can be understood as: for each input channel, its information, under the combined effect of temporal processing through kernel length  $K$  and inter-channel mapping via  $C_o/C_i$ , is refined or compressed to be equivalent to  $\alpha$  independent information units. These units contribute to the final output sample  $V[m]$ . The bottleneck here is determined by the smaller of  $K$  (temporal context length per channel) and  $C_o/C_i$  (channel transformation ratio).

## B.5 DEFINITION OF EQUIVALENT SAMPLING RATE

The actual output sampling rate of the module for each output channel is  $f'_s = f_s/S$ . At each output sampling instant, we have determined that each input channel contributes  $\alpha = \min\{K, C_o/C_i\}$  equivalent temporal degrees of freedom.

The Equivalent Sampling Rate  $f_{\text{ESR}}$  is defined as a rate such that if each of the original  $C_i$  input channels were sampled at  $f_{\text{ESR}}$ , and each sample carried one independent degree of freedom, then its total degrees of freedom throughput would match that of the current depth-wise and point-wise modules.

The total rate of generating degrees of freedom is:  $D = \min\{C_i K, C_o\} \times \frac{f_s}{S}$ . If  $C_i$  channels each operate at an equivalent sampling rate of  $f_{\text{ESR}}$ , their total degrees of freedom rate is  $C_i \times f_{\text{ESR}}$ :

$$C_i \times f_{\text{ESR}} = \min\{C_i K, C_o\} \times \frac{f_s}{S}. \quad (16)$$

Then, we can get the solve for  $f_{\text{ESR}}$ :

$$f_{\text{ESR}} = \frac{\min\{C_i K, C_o\}}{C_i} \times \frac{f_s}{S} = \min\left\{K, \frac{C_o}{C_i}\right\} \times \frac{f_s}{S}. \quad (17)$$

If we normalize the original sampling rate  $f_s$  to 1, we obtain the normalized ESR:

$$\text{ESR}_{\text{norm}} = \frac{1}{S} \min\left\{K, \frac{C_o}{C_i}\right\}. \quad (18)$$

Based on this equivalent sampling rate  $f_{\text{ESR}}$ , we can define an equivalent Nyquist frequency  $f_{\text{Nyq\_ESR}}$ . This frequency represents the maximum bandwidth that the input signal can accommodate without information loss due to module structural limitations:

$$f_{\text{Nyq\_ESR}} = \frac{f_{\text{ESR}}}{2} = \frac{f_s}{2S} \min\left\{K, \frac{C_o}{C_i}\right\}. \quad (19)$$

We can use  $f_{\text{ESR}}$  to quantify the information processing capability or information retention degree of the downsampling module consisting of depth-wise convolution and point-wise convolution relative to each input channel. It provides a useful metric to compare the effective information throughput of modules with different parameter configurations with  $K$ ,  $S$ ,  $C_i$ , and  $C_o$ . It is important to note that the anti-aliasing significance of  $f_{\text{Nyq\_ESR}}$  also depends on whether the depth-wise convolution kernel  $h_i[k]$  can effectively act as a low-pass filter to attenuate frequency components above  $f_{\text{Nyq\_ESR}}$ . If  $h_i[k]$  is not an ideal low-pass filter, the frequency components of the original signal above  $f_{\text{Nyq\_ESR}}$ , even if not completely filtered out by  $h_i[k]$ , may not be accurately represented by the output  $V[m]$  due to subsequent dimensionality reduction.

---

## C IMPLEMENTATION DETAILS

We summarized details of datasets, evaluation metrics, experiments in this section.

### C.1 DATASETS DETAILS

We evaluated the performance of different models on several well-established datasets for long-term forecasting, including Weather, Electricity, Solar-Energy, PeMS(PEMS03, PEMS04, PEMS07, PEMS08), and the ETT series (ETTh1, ETTh2, ETTm1, ETTm2). Furthermore, to demonstrate DMANet’s capability in handling highly non-stationary data, we conducted an extensive series of supplementary experiments on short-term forecasting across datasets from various domains. These include Health & Medical (ILI, COVID-19), Web Events (Wiki, Website), Finance (NASDAQ, SP500, DowJones), Market (CarSales), Energy (Power), and Society (Unemp). We detail the descriptions of the dataset in Table.1.

### C.2 BASELINE DETAILS

Acknowledging that the performance of different methods varies across scenarios, we conducted a comprehensive comparison of various approaches under three distinct settings: long-term forecasting with a lookback window of 96, long-term forecasting with a lookback window of 720, and short-term forecasting. The evaluated methods are categorized as follows:

- **Frequency-domain methods:** TimeStacker Liu et al. (2025), FilterNet Yi et al. (2024a), FITS Xu et al. (2024), Fredformer Piao et al. (2024), FEDformer Zhou et al. (2022).
- **CNN-based methods:** ModernTCN Donghao & Xue (2024), TVNet Li et al. (2025), TSLANet Eldele et al. (2024), TimesNet Wu et al. (2023), PDF Dai et al. (2024), MICN Wang et al. (2023).
- **MLP-based methods:** SOFTS Han et al. (2024), TimeMixer Wang et al. (2024a), DLinear Zeng et al. (2023), TiDE Das et al. (2023), RLinear Li et al. (2023b), MTS-Mixer Li et al. (2023c).
- **Transformer-based methods:** TimeXer Wang et al. (2024b), iTransformer Liu et al. (2024), Crossformer Zhang & Yan (2022), Pathformer Chen et al. (2024), Stationary Liu et al. (2022b), Pyraformer Liu et al. (2022a), Autoformer Wu et al. (2021).
- **LLM-based methods:** GPT4TS Zhou et al. (2023), Time-LLM Jin et al. (2024).
- **KAN-based methods:** TimeKAN Huang et al. (2025).
- **Mamba-based methods:** TimePro Ma et al. (2025).
- **Retrieval-Augmented methods:** RAFT Han et al. (2025).

### C.3 IMPLEMENTATION DETAILS.

Regarding evaluation metrics, we used mean square error (MSE) and mean absolute error (MAE) for both long-term and short-term forecasting. All experiments were conducted using PyTorch on a single NVIDIA GeForce RTX 3090 24GB GPU. We applied an early stopping strategy to all baselines when the validation loss did not decrease for three consecutive epochs. Notably, inspired by FreDF Wang et al. (2025), we argue that formulating the loss function in the frequency domain is advantageous for learning an anti-aliasing architecture. Consequently, we directly adopted the frequency-domain MAE as the loss function for both long-term and short-term forecasting. More detailed settings can be found in Appendix.C.5.

### C.4 FAIR COMPARISON SETTINGS.

To ensure a fair comparison and address challenges related to scaling laws, we maintained a consistent lookback window of 96 for all experiments in Table.4 and Table 5, and 720 for all experiments in Table.6 and Table.7. Our baseline comparisons mimic the experimental protocols established in TimesNet Wu et al. (2023), including same data processing and splitting procedures. For most

Table 1: Detailed dataset descriptions and statistics. **Dim** denotes the number of variates for each dataset. **Frequency** refers to the time interval between consecutive steps. **Split** indicates the data partitioning ratio (Train/Validation/Test). **Prediction len.** represents the prediction lengths. Our long-term forecasting employs a fixed input length of 96 or 720. For the majority of datasets, we evaluate across prediction horizons of 96, 192, 336, 720. A distinct setting is applied to the PeMS datasets, which are evaluated on shorter horizons of 12, 24, 48. For short-term forecasting, we adopt two settings: one with an input of 12 steps to predict 3, 6, 9, 12 steps, and another with an input of 36 steps to predict 24, 36, 48, 60 steps.

Dataset	Dim	Frequency	Total len.	Split	Prediction len.	Information
ETTh1, ETTh2	7	Hourly	17420	6:2:2	{96,192,336,720}	Electricity
ETTm1, ETTm2	7	15 mins	69680	6:2:2	{96,192,336,720}	Electricity
Weather	21	10 mins	52696	7:1:2	{96,192,336,720}	Weather
ECL	321	Hourly	26304	7:1:2	{96,192,336,720}	Electricity
Solar-Energy	137	10 mins	52560	7:1:2	{96,192,336,720}	Energy
PEMS03	358	5 mins	26209	6:2:2	{12,24,48}	Transportation
PEMS04	307	5 mins	16992	6:2:2	{12,24,48}	Transportation
PEMS07	883	5 mins	28224	6:2:2	{12,24,48}	Transportation
PEMS08	170	5 mins	17856	6:2:2	{12,24,48}	Transportation
ILI	7	Weekly	966	7:1:2	{24,36,48,60}	Health
COVID-19	55	Daily	335	7:1:2	{3,6,9,12}	Health
NASDAQ	12	Daily	3914	7:1:2	{24,36,48,60}	Finance
SP500	5	Daily	8077	7:1:2	{24,36,48,60}	Finance
DowJones	27	Daily	6577	7:1:2	{24,36,48,60}	Finance
CarSales	10	Daily	6728	7:1:2	{24,36,48,60}	Market
Power	2	Daily	1186	7:1:2	{24,36,48,60}	Energy
Website	4	Daily	2167	7:1:2	{3,6,9,12}	Web
Wiki	99	Daily	730	7:1:2	{3,6,9,12}	Web
Unemp	53	Monthly	531	6:2:2	{3,6,9,12}	Society

methods, we adopted the results reported in their original papers. For some methods that did not report results on the Solar-Energy dataset, we reproduced their performance using their official code repositories. The results for FITS Xu et al. (2024) and FreTSWang et al. (2025) were replicated from the FilterNet report Yi et al. (2024a); for other methods, we used the long-term prediction results provided in the iTransformer repository Liu et al. (2024). These results are based on the experimental configurations provided in the original paper or official code for each model. We verified that all hyperparameters for these baselines were selected from their respective official repositories, ensuring consistency with our fair comparison setup, where the only variations were the input and output sequence lengths.

For the experiments with the lookback window extended to 720, we referred to established baseline results: results in Table.6 were replicated from DUET Qiu et al. (2024), the results for GPT4TS Zhou et al. (2023) and TimeLLM Jin et al. (2024) in Table.7 were replicated from TSLANet Eldele et al. (2024), and the remaining results in Table.7 were replicated from TVNet Li et al. (2025). For short-term forecasting, we followed the results from the FreEformer repository Yue et al. (2025).

## C.5 HYPERPARAMETER SETTINGS.

**Primary Long-term Forecasting Task** For our model hyperparameter selection, in 96 lookback window long-term forecasting, we fixed  $d_{\text{model}} = 512$ , downsampling layer  $l$  to 2, depth-wise convolution kernel size  $K$  to 3, stride  $s$  to 2, and set the proportion of channel changes  $c$  to 0.5. And we only performed a limited search on the encoder layers  $E$ , learning rate  $LR$ , and batch size. Detailed configurations for each dataset can be found in Table.2.

**Other Long-term Forecasting Tasks** For long-term forecasting with an extended 720 lookback window, as well as for the 96 lookback forecasting on PEMS datasets and 336 lookback univariate forecasting tasks, we implemented a more extensive hyperparameter search. This search was conducted for each forecast horizon within a given dataset to find the optimal configuration. The search space was defined as follows:  $d_{\text{model}} \in \{256, 512\}$ , Learning Rate  $LR \in \{1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 2 \times 10^{-2}\}$ , Encoder Layers  $E \in \{1, 2, 3\}$ , Downsampling Layers  $l \in \{2, 3, 4\}$ , Batch Size  $\in \{8, 16, 32, 64\}$ . Other hyperparameters, such as the convolutional kernel size and stride, remained fixed across all experiments, consistent with the settings used in the primary 96 lookback forecasting task. In contrast to all baseline lookback windows searched from  $\{192, 336, 512, 672, 720\}$  etc., We provide long-term forecasting for the fixed 720 lookback window.

**Short-term Forecasting** We implemented a more extensive hyperparameter search like Other Long-term Forecasting Tasks. This search was conducted for each forecast horizon within a given dataset to find the optimal configuration. The search space was defined as follows: Downsampling Layers is fixed 2,  $d_{\text{model}} \in \{256, 512\}$ , Learning Rate  $LR \in \{1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 2 \times 10^{-2}\}$ , Encoder Layers  $E \in \{1, 2\}$ , Batch Size  $\in \{2, 4, 8, 16\}$ . Other hyperparameters, such as the convolutional kernel size and stride, remained fixed across all experiments, consistent with the settings used in the primary 96-lookback forecasting task.

**Ablation Study on Pre-Sampling Filtering** To validate our ESR-based filtering approach, we conducted an ablation study comparing it against alternatives that do not adhere to the Nyquist sampling theorem. Each experimental group differs from our full DMANet only in the cutoff frequency determination method within the Pre-Sampling Filtering module; all other structures and parameters remain identical. We categorize the compared methods into two groups: heuristic and classical filters.

**HEURISTIC FILTERS** These methods serve as simple, non-theoretical baselines. They are designed to mimic intuitive or simplistic approaches to filtering that one might adopt without a rigorous signal processing foundation.

- **Max:** For each time series in the batch, this filter identifies the frequency bin with the maximum amplitude and sets the cutoff frequency to twice its index. All components below this dynamic cutoff are preserved, while those above are zeroed out.
- **Random:** This filter applies a stochastic mask to the frequency spectrum, where each frequency component is independently dropped with a probability of  $p = 0.5$ .



CLASSICAL FILTERS These methods serve as benchmarks against well-established, theoretically-grounded filtering techniques. To ensure a fair comparison, a normalized cutoff frequency of 0.4 was used across all classic filter variants, preserving the lowest 80% of the frequency band.

- **Ideal:** A sharp cutoff filter where all frequency components above the cutoff frequency are set to zero.
- **Butterworth:** Known for its maximally flat passband, providing high-fidelity signal preservation. We used a 4th-order filter.
- **Gaussian:** A smooth filter often used to avoid ringing artifacts, with a sigma of 0.15.
- **Chebyshev (Type I):** Achieves a steeper rolloff than Butterworth at the cost of introducing ripples in the passband. We used a 4th-order filter with 0.5 dB of passband ripple.

Table 2: Experiment configuration of DMANet in 96 lookback window. All the experiments use the ADAM optimizer with the default hyperparameter configuration for  $(\beta_1, \beta_2)$  as (0.9, 0.999).

Dataset / Configurations	Model Hyper-parameter			Training Process			
	$E$	$l$	$d_{\text{model}}$	LR*	Loss	Batch Size	Epochs
ETTh1	1	2	512	$2 * 10^{-2}$	MAE	8	15
ETTh2	1	2	512	$1 * 10^{-2}$	MAE	8	15
ETTm1	1	2	512	$2 * 10^{-3}$	MAE	16	15
ETTm2	2	2	512	$5 * 10^{-3}$	MAE	32	15
Weather	1	2	512	$5 * 10^{-3}$	MAE	16	15
Electricity	2	2	512	$1 * 10^{-3}$	MAE	8	15
Solar-Energy	2	2	512	$5 * 10^{-3}$	MAE	16	15

\* LR means the initial learning rate.

## D FULL RESULTS

### D.1 ERROR BARS

To evaluate the performance stability and robustness of DMANet, we conducted multiple independent runs with five different random seeds and compared its performance against the second-best model, TimeMixer. The results, averaged over four prediction horizons (96, 192, 336, and 720), are presented in Table 3. We report the mean and standard deviation of the MSE and MAE metrics across the five experiments, as well as the confidence level of DMANet’s superiority over TimeMixer. This performance improvement is statistically significant, with a 99% confidence level in all evaluated scenarios.

Table 3: Standard deviation and statistical tests for our DMANet method and second-best method (TimeMixer) on five datasets.

Metric	MSE			MAE		
Dataset	DMANet	TimeMixer	Confidence	DMANet	TimeMixer	Confidence
ETTm1	<b>0.376±0.005</b>	0.386±0.003	99%	<b>0.388±0.003</b>	0.399±0.001	99%
ETTm2	<b>0.269±0.007</b>	0.278±0.001	99%	<b>0.311±0.005</b>	0.325±0.001	99%
Weather	<b>0.238±0.005</b>	0.245±0.001	99%	<b>0.263±0.005</b>	0.276±0.001	99%
Electricity	<b>0.171±0.002</b>	0.182±0.002	99%	<b>0.264±0.002</b>	0.272±0.002	99%
Solar-Energy	<b>0.228±0.003</b>	0.235±0.001	99%	<b>0.249±0.002</b>	0.292±0.001	99%

### D.2 LONG-TERM FORECASTING

Here, Table.4, Table.5, Table.6 and Table.7 present comprehensive evaluation results for long-term forecasting, including both configurations with fixed lookback windows  $L = 96$  and extended window settings  $L = 720$  designed to adhere to the scaling law inherent to TSF. In the  $L = 96$  fixed-window experiments, **consistent hyperparameters** were maintained across all forecast horizons within each dataset. By contrast, the  $L = 720$  experiments employed horizon-specific hyperparameter adjustments to enhance model adaptability while preserving scaling law compliance. Under both experimental paradigms, DMANet consistently demonstrates superior performance with statistically significant margins, thereby empirically validating its effectiveness and robustness. Notably, even when handling extended sequence lengths through augmented lookback windows, DMANet retains its inherent capability to adaptively model critical dependencies within extended temporal sequences.

The results for PEMS dataset forecasting, presented in Table.8 for a lookback window of  $L = 96$  and the forecasting horizon  $T \in \{12, 24, 48\}$ , demonstrate the exceptional capability of DMANet. Across all four PEMS datasets, DMANet consistently outperforms all baselines. This superiority is quantified by average reductions of 14.4% in MSE and 5.7% in MAE compared to a strong baseline, iTransformer. We attribute this robust performance to our convolutional architecture’s inherent proficiency in preserving localized features and mitigating the interference of high-frequency noise, which are critical for high-dimensional short-term prediction.

### D.3 SHORT-TERM FORECASTING

The short-term forecasting results, presented in Table.10, validate the superiority of DMANet in handling highly non-stationary time series. Across a diverse set of challenging datasets including ILI (health), COVID-19 (pandemic), DowJones (finance), and Unemp (society), DMANet consistently achieves state-of-the-art performance, securing the top rank in 17 out of 20 metrics. It significantly outperforms other methods, including strong frequency-domain baselines like Fredformer and FilterNet. We attribute this exceptional capability in short-term and non-stationary forecasting to DMANet’s synergistic design: its convolutional architecture excels at preserving local features, while the anti-aliasing structure effectively mitigates disruptive high-frequency noise. This robust performance on volatile, real-world data underscores the effectiveness of our approach in capturing the transient and complex patterns inherent to non-stationary signals.

---

#### D.4 UNIVARIATE FORECASTING

Here we provide the univariate forecasting results on ETT datasets. There is a target feature oil temperature within those datasets, which is the univariate time series that we are trying to forecast. As shown in Table.9 , the anti-aliasing depth-wise convolution has better temporal modelling capabilities, allowing DManet to achieve better performance than the state-of-the-art CNN-based ModerTCN in univariate forecasting tasks.

Table 4: Full results of long-term forecasting with a 96-step lookback window (Part I). The input sequence length  $L$  is set to 96 for all baselines. All results are averaged across four different forecasting horizon:  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively. Among them, - means that the code has not yet been open sourced. We will put the summary table in the appendix of the next version.

Models	DMANet Ours		TimeStacker 2025		TimeXer 2024b		iTransformer 2024		TimeMixer 2024a		FilterNet 2024a		Fredformer 2024		FITS 2024		FreTS 2024b		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh1	96	<b>0.308</b>	<u>0.343</u>	<u>0.311</u>	<b>0.337</b>	0.318	0.356	0.334	0.368	0.320	0.357	0.318	0.358	0.326	0.361	0.355	0.375	0.339	0.374
	192	<b>0.354</b>	<u>0.372</u>	0.364	<b>0.367</b>	0.362	0.383	0.377	0.391	<u>0.361</u>	0.381	0.364	0.383	0.363	0.380	0.392	0.393	0.382	0.397
	336	<b>0.384</b>	<u>0.394</u>	<u>0.389</u>	<b>0.391</b>	0.395	0.407	0.426	0.420	0.390	0.404	0.396	0.406	0.395	0.403	0.424	0.414	0.421	0.426
	720	<b>0.447</b>	<u>0.431</u>	0.460	<b>0.428</b>	<u>0.452</u>	0.441	0.491	0.459	0.454	0.441	0.456	0.444	0.453	0.438	0.487	0.449	0.485	0.462
	Avg.	<b>0.373</b>	<u>0.385</u>	<u>0.381</u>	<b>0.381</b>	0.382	0.397	0.407	0.410	0.381	0.395	0.384	0.398	0.393	0.403	0.387	0.408	0.407	0.415
ETTh2	96	<b>0.165</b>	<b>0.244</b>	<u>0.171</u>	<u>0.250</u>	<u>0.171</u>	0.256	0.180	0.264	0.175	0.258	0.174	0.257	0.177	0.259	0.183	0.266	0.190	0.282
	192	<b>0.231</b>	<b>0.288</b>	<u>0.235</u>	<u>0.292</u>	0.237	0.299	0.250	0.309	0.237	0.299	0.240	0.300	0.241	0.300	0.247	0.305	0.260	0.329
	336	<b>0.289</b>	<b>0.325</b>	<u>0.293</u>	<u>0.329</u>	0.296	0.338	0.311	0.348	0.298	0.340	0.297	0.339	0.302	0.340	0.307	0.342	0.373	0.405
	720	<b>0.385</b>	<b>0.383</b>	0.395	<u>0.391</u>	0.392	0.394	0.412	0.407	<u>0.391</u>	0.396	0.392	0.393	0.397	0.396	0.407	0.399	0.517	0.499
	Avg.	<b>0.268</b>	<b>0.310</b>	<u>0.274</u>	<u>0.316</u>	<u>0.274</u>	0.322	0.288	0.332	0.275	0.323	0.276	0.322	0.279	0.324	0.286	0.328	0.335	0.379
ETTth1	96	<b>0.370</b>	<u>0.391</u>	<b>0.379</b>	0.385	0.382	0.403	0.386	0.405	<u>0.375</u>	0.400	<u>0.375</u>	0.394	0.376	0.394	0.386	0.396	0.399	0.412
	192	<b>0.417</b>	<u>0.420</u>	<u>0.429</u>	<b>0.416</b>	0.429	0.435	0.441	0.436	0.429	0.421	0.436	0.422	0.440	0.425	0.436	0.423	0.453	0.443
	336	<b>0.457</b>	<u>0.440</u>	<u>0.459</u>	<b>0.436</b>	0.468	0.448	0.487	0.458	0.484	0.458	0.476	0.443	0.472	<u>0.440</u>	0.478	0.444	0.503	0.475
	720	<u>0.468</u>	<u>0.465</u>	<b>0.464</b>	<b>0.455</b>	0.469	0.461	0.503	0.491	0.498	0.482	0.474	0.469	0.490	0.467	0.502	0.495	0.596	0.565
	Avg.	<b>0.428</b>	<u>0.429</u>	<u>0.433</u>	<b>0.423</b>	0.437	0.437	0.454	0.447	0.447	0.440	0.440	0.432	0.445	0.432	0.447	0.448	0.488	0.474
ETTth2	96	<b>0.280</b>	<u>0.329</u>	<b>0.280</b>	<b>0.327</b>	<u>0.286</u>	0.338	0.297	0.349	0.289	0.341	0.292	0.343	0.292	0.343	0.295	0.350	0.350	0.403
	192	<b>0.349</b>	<b>0.374</b>	<u>0.373</u>	0.385	<u>0.363</u>	0.389	0.380	0.400	0.372	0.392	0.369	0.395	0.370	0.390	0.381	0.396	0.472	0.475
	336	0.393	<b>0.410</b>	0.407	0.416	0.414	0.423	0.428	0.432	<u>0.386</u>	0.414	0.420	0.432	<b>0.385</b>	<u>0.413</u>	0.426	0.438	0.564	0.528
	720	0.418	0.437	<u>0.412</u>	<b>0.431</b>	<b>0.408</b>	<u>0.432</u>	0.427	0.445	<u>0.412</u>	0.434	0.430	0.446	0.419	0.439	0.431	0.446	0.815	0.654
	Avg.	<b>0.361</b>	<b>0.388</b>	0.368	<u>0.390</u>	0.367	0.396	0.383	0.407	<u>0.364</u>	0.395	0.378	0.397	0.367	0.396	0.383	0.408	0.550	0.515
Weather	96	<b>0.148</b>	<b>0.191</b>	0.161	<u>0.198</u>	<u>0.157</u>	0.205	0.174	0.214	0.163	0.209	0.164	0.210	0.163	0.207	0.166	0.213	0.184	0.239
	192	<b>0.199</b>	<b>0.238</b>	0.207	<u>0.241</u>	<u>0.204</u>	0.247	0.221	0.254	0.208	0.250	0.214	0.252	0.211	0.251	0.213	0.254	0.223	0.275
	336	<u>0.256</u>	<u>0.282</u>	0.261	<b>0.281</b>	0.261	0.290	0.278	0.296	<b>0.251</b>	0.287	0.268	0.293	0.267	0.292	0.269	0.294	0.272	0.316
	720	<b>0.339</b>	<u>0.336</u>	0.343	<b>0.334</b>	<u>0.340</u>	0.341	0.358	0.349	<b>0.339</b>	0.341	0.344	0.342	0.343	0.341	0.346	0.343	<u>0.340</u>	0.363
	Avg.	<b>0.236</b>	<b>0.262</b>	0.243	<u>0.264</u>	0.241	0.271	0.258	0.279	<u>0.240</u>	0.271	0.248	0.278	0.246	0.272	0.249	0.276	0.255	0.363
Electricity	96	<b>0.139</b>	<b>0.234</b>	0.168	0.251	<u>0.140</u>	0.242	0.148	<u>0.240</u>	0.153	0.247	0.176	0.264	0.147	0.241	0.200	0.278	0.183	0.269
	192	<b>0.157</b>	<b>0.250</b>	0.176	0.262	<b>0.157</b>	0.256	<u>0.162</u>	<u>0.253</u>	0.166	0.256	0.185	0.270	0.165	0.258	0.200	0.280	0.187	0.276
	336	<b>0.175</b>	<b>0.269</b>	0.195	0.278	<u>0.176</u>	0.275	<u>0.178</u>	<b>0.269</b>	0.185	0.277	0.202	0.286	<u>0.177</u>	<u>0.273</u>	0.214	0.295	0.202	0.292
	720	<b>0.210</b>	<b>0.301</b>	0.235	0.310	<u>0.211</u>	0.306	0.225	0.317	0.225	0.310	0.242	0.319	0.213	<u>0.304</u>	0.255	0.327	0.237	0.325
	Avg.	<b>0.170</b>	<b>0.264</b>	0.194	0.275	<u>0.171</u>	0.270	0.178	0.270	0.182	0.272	0.201	0.285	0.175	<u>0.269</u>	0.217	0.295	0.202	0.290
Solar-Energy	96	<b>0.184</b>	<b>0.217</b>	-	-	0.215	0.295	0.203	<u>0.237</u>	<u>0.189</u>	0.241	0.224	0.264	0.200	0.275	0.328	0.396	0.252	0.319
	192	<b>0.220</b>	<b>0.242</b>	-	-	0.236	0.301	0.233	0.261	<u>0.222</u>	0.283	0.259	0.284	0.226	<u>0.259</u>	0.397	0.387	0.283	0.338
	336	<u>0.247</u>	<b>0.266</b>	-	-	0.252	0.307	0.248	<u>0.273</u>	<b>0.231</b>	0.292	0.284	0.298	0.254	0.277	0.433	0.410	0.299	0.344
	720	0.257	<b>0.270</b>	-	-	<u>0.244</u>	0.305	0.249	<u>0.275</u>	<b>0.223</b>	0.285	0.284	0.298	0.249	0.284	0.429	0.396	0.298	0.351
	Avg.	<u>0.227</u>	<b>0.249</b>	-	-	0.237	0.302	0.233	<u>0.262</u>	<b>0.216</b>	0.280	0.263	0.286	0.232	0.274	0.397	0.398	0.283	0.338

Table 5: Full results of long-term forecasting with a 96-step lookback window (Part II). The input sequence length  $L$  is set to 96 for all baselines. All results are averaged across four different forecasting horizon:  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively.

Models	DManet Ours		TimePro 2025		TimeKAN 2025		SOFTS 2024		FreDF 2025		PatchTST 2023		TimesNet 2023		DLinear 2023		MICN 2023	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	<b>0.308</b> <b>0.343</b>	0.326	0.364	<u>0.322</u> <u>0.361</u>	0.325	<u>0.361</u>	0.324	0.362	0.329	0.367	0.338	0.375	0.346	0.374	0.365	0.387	
	192	<b>0.354</b> <b>0.372</b>	0.367	0.383	<u>0.357</u> <u>0.383</u>	0.375	0.389	0.373	0.385	0.367	0.385	0.374	0.387	0.382	0.391	0.403	0.408	
	336	<b>0.384</b> <b>0.394</b>	0.402	0.409	<b>0.382</b> <u>0.401</u>	0.405	0.412	0.402	0.404	0.399	0.410	0.410	0.411	0.415	0.415	0.436	0.431	
	720	<u>0.447</u> <b>0.431</b>	0.469	0.446	<b>0.445</b> <u>0.435</u>	0.466	0.447	0.469	0.444	0.454	0.439	0.478	0.450	0.473	0.451	0.489	0.462	
	Avg.	<b>0.373</b> <b>0.385</b>	0.391	0.400	<u>0.376</u> <u>0.395</u>	0.393	0.403	0.392	0.399	0.415	0.400	0.400	0.406	0.404	0.408	0.423	0.422	
ETTh2	96	<b>0.165</b> <b>0.244</b>	0.178	0.260	0.174	0.255	0.180	0.261	<u>0.173</u> <u>0.252</u>	0.175	0.259	0.187	0.267	0.193	0.293	0.197	0.296	
	192	<b>0.231</b> <b>0.288</b>	0.242	0.303	<u>0.239</u> <u>0.299</u>	0.246	0.306	0.241	<u>0.298</u> <u>0.334</u>	0.241	0.302	0.249	0.309	0.284	0.361	0.284	0.361	
	336	<b>0.289</b> <b>0.325</b>	0.303	0.342	0.301	0.340	0.319	0.352	<u>0.298</u> <u>0.334</u>	0.305	0.343	0.321	0.351	0.382	0.429	0.381	0.429	
	720	<b>0.385</b> <b>0.383</b>	0.400	0.399	<u>0.395</u> <u>0.396</u>	0.405	0.401	0.398	<u>0.393</u> <u>0.402</u>	0.402	0.400	0.408	0.403	0.558	0.525	0.549	0.522	
	Avg.	<b>0.268</b> <b>0.310</b>	0.281	0.326	<u>0.277</u> <u>0.322</u>	0.287	0.330	0.278	<u>0.319</u> <u>0.328</u>	0.281	0.326	0.291	0.333	0.354	0.402	0.305	0.349	
ETT1	96	<u>0.370</u> <b>0.391</b>	0.375	0.398	<b>0.367</b> <u>0.395</u>	0.381	0.399	0.382	0.400	0.414	0.419	0.384	0.402	0.397	0.412	0.426	0.446	
	192	<u>0.417</u> <b>0.420</b>	0.427	0.429	<b>0.414</b> <b>0.420</b>	0.435	0.431	0.430	0.427	0.460	0.445	0.436	0.429	0.446	0.441	0.454	0.464	
	336	<u>0.457</u> <u>0.440</u>	0.472	0.450	<b>0.445</b> <b>0.434</b>	0.480	0.452	0.474	0.451	0.501	0.466	0.491	0.469	0.489	0.467	0.493	0.487	
	720	<u>0.468</u> <u>0.465</u>	0.476	0.474	<b>0.444</b> <b>0.459</b>	0.499	0.488	0.463	<u>0.462</u> <u>0.500</u>	0.488	0.488	0.521	0.509	0.513	0.510	0.526	0.526	
	Avg.	<u>0.428</u> <u>0.429</u>	0.438	0.438	<b>0.417</b> <b>0.427</b>	0.449	0.442	0.437	0.435	0.469	0.454	0.458	0.450	0.461	0.457	0.475	0.480	
ETT2	96	<b>0.280</b> <b>0.329</b>	0.293	0.345	0.290	0.340	0.297	0.347	<u>0.289</u> <u>0.337</u>	0.302	0.348	0.340	0.374	0.340	0.394	0.372	0.424	
	192	<b>0.349</b> <b>0.374</b>	0.367	0.394	0.375	0.392	0.373	0.394	<u>0.363</u> <u>0.385</u>	0.388	0.400	0.402	0.414	0.482	0.479	0.492	0.492	
	336	<b>0.393</b> <b>0.410</b>	0.419	0.431	0.423	0.435	<u>0.410</u> <u>0.426</u>	0.419	<u>0.426</u> <u>0.426</u>	0.426	0.433	0.452	0.452	0.591	0.541	0.607	0.555	
	720	0.418	<u>0.437</u>	0.427	0.445	0.443	0.449	<b>0.411</b> <b>0.433</b>	0.415	0.437	0.431	0.446	0.462	0.468	0.839	0.661	0.824	
	Avg.	<b>0.361</b> <b>0.388</b>	0.377	0.403	0.383	0.404	0.373	0.400	<u>0.371</u> <u>0.396</u>	0.387	0.407	0.414	0.427	0.563	0.519	0.574	0.531	
Weather	96	<b>0.148</b> <b>0.191</b>	0.166	0.207	<u>0.162</u> <u>0.208</u>	0.166	0.208	0.164	<u>0.202</u> <u>0.233</u>	0.177	0.218	0.172	0.220	0.195	0.252	0.198	0.261	
	192	<b>0.199</b> <b>0.238</b>	0.216	0.254	<u>0.207</u> <u>0.249</u>	0.217	0.253	0.220	0.253	0.225	0.259	0.219	0.261	0.237	0.295	0.239	0.299	
	336	<b>0.256</b> <b>0.282</b>	0.273	0.296	<u>0.263</u> <u>0.290</u>	0.282	0.300	0.275	0.294	0.278	0.297	0.280	0.306	0.282	0.331	0.285	0.336	
	720	<u>0.339</u> <b>0.336</b>	0.351	0.346	<b>0.338</b> <u>0.340</u>	0.356	0.351	0.356	0.347	0.354	0.348	0.365	0.359	0.345	0.382	0.351	0.388	
	Avg.	<b>0.236</b> <b>0.262</b>	0.251	0.276	<u>0.242</u> <u>0.272</u>	0.255	0.278	0.254	0.274	0.259	0.281	0.259	0.287	0.265	0.315	0.268	0.321	
Electricity	96	<b>0.139</b> <u>0.234</u>	<b>0.139</b> <u>0.234</u>	0.174	0.266	<u>0.143</u> <b>0.233</b>	0.144	<u>0.233</u> <u>0.233</u>	0.195	0.285	0.168	0.272	0.210	0.302	0.180	0.293		
	192	<u>0.157</u> <u>0.250</u>	<b>0.156</b> <u>0.249</u>	0.182	0.273	<u>0.158</u> <u>0.248</u>	0.159	<b>0.247</b> <u>0.247</u>	0.199	0.289	0.184	0.289	0.210	0.305	0.189	0.302		
	336	<u>0.175</u> <u>0.269</u>	<b>0.172</b> <u>0.267</u>	0.197	0.286	0.178	0.269	<b>0.172</b> <b>0.263</b>	0.215	0.305	0.198	0.300	0.223	0.319	0.198	0.312		
	720	0.210	0.301	<u>0.209</u> <u>0.299</u>	0.236	0.320	0.218	0.305	<b>0.204</b> <u>0.294</u>	0.256	0.337	0.220	0.320	0.258	0.350	0.217	0.330	
	Avg.	<u>0.170</u> <u>0.264</u>	<b>0.169</b> <u>0.262</u>	0.197	0.286	0.174	0.264	<u>0.170</u> <b>0.259</b>	0.216	0.304	0.193	0.295	0.225	0.319	0.196	0.309		
Solar-Energy	96	<b>0.184</b> <b>0.217</b>	0.196	0.237	0.254	0.318	<u>0.200</u> <u>0.230</u>	0.232	0.256	0.234	0.286	0.250	0.292	0.290	0.378	0.257	0.325	
	192	<b>0.220</b> <b>0.242</b>	0.231	0.263	0.285	0.326	<u>0.229</u> <u>0.253</u>	0.276	0.288	0.267	0.310	0.296	0.318	0.320	0.398	0.278	0.354	
	336	<u>0.247</u> <b>0.266</b>	0.250	0.281	0.315	0.338	<b>0.243</b> <u>0.269</u>	0.301	0.306	0.290	0.315	0.319	0.330	0.353	0.415	0.298	0.375	
	720	0.257	<b>0.270</b> <u>0.253</u>	0.285	0.313	0.340	<b>0.245</b> <u>0.272</u>	0.308	0.316	0.289	0.317	0.338	0.337	0.357	0.413	0.299	0.379	
	Avg.	<b>0.227</b> <b>0.249</b>	0.232	0.266	0.292	0.331	<u>0.229</u> <u>0.256</u>	0.279	0.292	0.270	0.307	0.301	0.319	0.330	0.401	0.283	0.358	

Table 6: Full results of long-term forecasting with a 720-step lookback window (Part I) The input length  $L$  is fixed 720 for optimal horizon in the scaling law of TSF Shi et al. (2024). All results are averaged across four different forecasting horizon:  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively.

Models		DMANet Ours		PDF 2024		iTransformer 2024		Pathformer 2024		FITS 2024		TimeMixer 2024a		PatchTST 2023		Crossformer 2022		TimesNet 2023		Dlinear 2023		Stationary 2022b	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.287	0.340	0.286	0.340	0.300	0.353	0.290	0.335	0.303	0.345	0.293	0.345	0.289	0.343	0.314	0.367	0.340	0.378	0.300	0.345	0.415	0.410
	192	0.322	0.364	0.321	0.364	0.341	0.380	0.337	0.363	0.337	0.365	0.335	0.372	0.329	0.368	0.374	0.410	0.392	0.404	0.336	0.366	0.494	0.451
	336	0.352	0.381	0.354	0.383	0.374	0.396	0.374	0.384	0.368	0.384	0.368	0.386	0.362	0.390	0.413	0.432	0.423	0.426	0.367	0.386	0.577	0.490
	720	0.403	0.410	0.408	0.415	0.429	0.430	0.428	0.416	0.420	0.413	0.426	0.417	0.416	0.423	0.753	0.613	0.475	0.453	0.419	0.416	0.636	0.535
	Avg.	<b>0.341</b>	<b>0.374</b>	<u>0.342</u>	0.376	0.361	0.390	0.357	<u>0.375</u>	0.357	0.377	0.356	0.380	0.349	0.381	0.464	0.456	0.408	0.415	0.356	0.378	0.531	0.472
ETTm2	96	0.158	0.246	0.163	0.251	0.175	0.266	0.164	0.250	0.165	0.254	0.165	0.256	0.165	0.255	0.296	0.391	0.189	0.265	0.164	0.255	0.210	0.294
	192	0.214	0.287	0.219	0.290	0.242	0.312	0.219	0.288	0.219	0.291	0.225	0.298	0.221	0.293	0.369	0.416	0.254	0.310	0.224	0.304	0.338	0.373
	336	0.264	0.320	0.269	0.330	0.282	0.337	0.267	0.319	0.272	0.326	0.277	0.332	0.276	0.327	0.588	0.600	0.313	0.345	0.277	0.337	0.432	0.416
	720	0.345	0.373	0.349	0.382	0.375	0.394	0.361	0.377	0.359	0.381	0.360	0.387	0.362	0.381	0.750	0.612	0.413	0.402	0.371	0.401	0.554	0.476
	Avg.	<b>0.245</b>	<b>0.307</b>	<u>0.250</u>	0.313	0.269	0.327	0.253	<u>0.308</u>	0.254	0.313	0.257	0.318	0.256	0.314	0.501	0.505	0.292	0.331	0.259	0.324	0.383	0.390
Weather	96	0.141	0.188	0.147	0.196	0.157	0.207	0.148	0.195	0.172	0.225	0.147	0.198	0.149	0.196	0.143	0.210	0.168	0.214	0.170	0.230	0.188	0.242
	192	0.189	0.237	0.193	0.240	0.200	0.248	0.191	0.235	0.215	0.261	0.192	0.243	0.191	0.239	0.198	0.260	0.219	0.262	0.216	0.273	0.241	0.290
	336	0.239	0.275	0.245	0.280	0.252	0.287	0.243	0.274	0.261	0.295	0.247	0.284	0.242	0.279	0.258	0.314	0.278	0.302	0.258	0.307	0.341	0.341
	720	0.303	0.327	0.323	0.334	0.320	0.336	0.318	0.326	0.326	0.341	0.318	0.330	0.312	0.330	0.335	0.385	0.353	0.351	0.323	0.362	0.403	0.388
	Avg.	<b>0.218</b>	<b>0.257</b>	0.227	0.263	0.232	0.270	0.225	<b>0.257</b>	0.244	0.280	0.226	0.264	<u>0.224</u>	<u>0.261</u>	0.234	0.292	0.255	0.282	0.242	0.293	0.293	0.315
Electricity	96	0.130	0.227	0.128	0.222	0.134	0.230	0.135	0.222	0.139	0.237	0.153	0.256	0.143	0.247	0.134	0.231	0.169	0.271	0.140	0.237	0.171	0.274
	192	0.145	0.242	0.147	0.242	0.154	0.250	0.157	0.253	0.154	0.250	0.168	0.269	0.158	0.260	0.146	0.243	0.180	0.280	0.154	0.251	0.180	0.283
	336	0.160	0.258	0.165	0.260	0.169	0.265	0.170	0.267	0.170	0.268	0.189	0.291	0.168	0.267	0.165	0.264	0.204	0.304	0.169	0.268	0.204	0.305
	720	0.182	0.280	0.199	0.289	0.194	0.288	0.211	0.302	0.212	0.304	0.228	0.320	0.214	0.307	0.237	0.314	0.205	0.304	0.204	0.301	0.221	0.319
	Avg.	<b>0.154</b>	<b>0.252</b>	<u>0.160</u>	<u>0.253</u>	0.163	0.258	0.168	0.261	0.169	0.265	0.184	0.284	0.171	0.270	0.171	0.263	0.190	0.290	0.167	0.264	0.194	0.295
Solar	96	0.159	0.205	0.181	0.247	0.190	0.244	0.218	0.235	0.208	0.255	0.179	0.232	0.170	0.234	0.183	0.208	0.198	0.270	0.199	0.265	0.381	0.398
	192	0.183	0.230	0.200	0.259	0.193	0.257	0.196	0.220	0.229	0.267	0.201	0.259	0.204	0.302	0.208	0.226	0.206	0.276	0.220	0.282	0.395	0.386
	336	0.195	0.243	0.208	0.269	0.203	0.266	0.195	0.220	0.241	0.273	0.190	0.256	0.212	0.293	0.212	0.239	0.208	0.284	0.234	0.295	0.410	0.394
	720	0.193	0.244	0.212	0.275	0.223	0.281	0.208	0.237	0.248	0.277	0.203	0.261	0.215	0.307	0.215	0.256	0.232	0.294	0.243	0.301	0.377	0.376
	Avg.	<b>0.183</b>	<u>0.231</u>	0.200	0.263	0.202	0.262	0.204	<b>0.228</b>	0.232	0.268	<u>0.193</u>	0.252	0.200	0.284	0.205	0.232	0.211	0.281	0.224	0.286	0.391	0.389

Table 7: Full results of long-term forecasting with a 720-step lookback window (Part II). The input length  $L$  is fixed 720 for optimal horizon in the scaling law of TSF Shi et al. (2024). All results are averaged across four different forecasting horizon:  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively.

Models		DMANet (Ours)		TVNet (2025)		RLinear (2023a)		MTS-Mixer (2023c)		MICN (2023)		ModernTCN (2024)		FEDformer (2022)		RAFT (2025)		TSLANet (2024)		GPT4TS (2023)		Time-LLM (2024)	
		Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh1	96	0.287	0.340	0.288	0.343	0.301	0.342	0.314	0.358	0.314	0.360	0.292	0.346	0.326	0.390	0.302	0.349	0.289	0.349	0.292	0.346	0.272	0.334
	192	0.322	0.364	0.326	0.367	0.355	0.363	0.354	0.386	0.359	0.387	0.332	0.368	0.365	0.415	0.329	0.367	0.328	0.370	0.332	0.372	0.310	0.358
	336	0.352	0.381	0.365	0.391	0.370	0.383	0.384	0.405	0.398	0.413	0.365	0.391	0.392	0.425	0.355	0.383	0.355	0.389	0.366	0.394	0.352	0.384
	720	0.403	0.410	0.412	0.413	0.425	0.414	0.427	0.432	0.459	0.464	0.416	0.417	0.446	0.458	0.406	0.413	0.421	0.425	0.417	0.421	0.383	0.411
	Avg.	<u>0.341</u>	<u>0.374</u>	0.348	0.379	0.358	0.376	0.370	0.395	0.383	0.406	0.351	0.381	0.382	0.422	0.348	0.378	0.348	0.383	0.348	0.383	<b>0.329</b>	<b>0.372</b>
ETTm2	96	0.158	0.246	0.161	0.254	0.164	0.253	0.177	0.259	0.167	0.260	0.166	0.256	0.180	0.271	0.164	0.256	0.169	0.259	0.173	0.262	0.161	0.253
	192	0.214	0.287	0.220	0.293	0.219	0.290	0.241	0.303	0.245	0.316	0.222	0.293	0.252	0.318	0.219	0.296	0.224	0.297	0.229	0.301	0.219	0.293
	336	0.264	0.320	0.272	0.316	0.273	0.326	0.297	0.338	0.295	0.350	0.272	0.324	0.324	0.364	0.275	0.336	0.275	0.329	0.286	0.341	0.271	0.329
	720	0.345	0.373	0.349	0.379	0.366	0.385	0.396	0.398	0.389	0.406	0.351	0.381	0.410	0.420	0.359	0.392	0.354	0.380	0.378	0.401	0.352	0.379
	Avg.	<b>0.245</b>	<b>0.307</b>	<u>0.251</u>	<u>0.311</u>	0.256	0.314	0.277	0.325	0.277	0.336	0.253	0.314	0.292	0.343	0.254	0.320	0.256	0.316	0.226	0.326	<u>0.251</u>	0.313
Weather	96	0.141	0.188	0.147	0.198	0.175	0.225	0.156	0.206	0.161	0.226	0.149	0.200	0.238	0.314	0.165	0.222	0.148	0.197	0.162	0.212	0.147	0.201
	192	0.189	0.237	0.194	0.238	0.218	0.260	0.199	0.248	0.220	0.283	0.196	0.245	0.275	0.329	0.211	0.264	0.193	0.241	0.204	0.248	0.189	0.235
	336	0.239	0.275	0.235	0.277	0.265	0.294	0.249	0.291	0.275	0.328	0.238	0.277	0.339	0.377	0.260	0.302	0.245	0.282	0.254	0.286	0.262	0.279
	720	0.303	0.327	0.308	0.331	0.329	0.339	0.336	0.343	0.311	0.356	0.314	0.334	0.389	0.409	0.327	0.355	0.325	0.337	0.326	0.337	0.304	0.316
	Avg.	<b>0.218</b>	<b>0.257</b>	<u>0.221</u>	<u>0.261</u>	0.247	0.279	0.235	0.272	0.242	0.298	0.224	0.264	0.310	0.357	0.241	0.286	0.325	0.337	0.237	0.270	0.225	<b>0.257</b>
Electricity	96	0.130	0.227	0.142	0.223	0.140	0.235	0.141	0.243	0.159	0.267	0.129	0.226	0.186	0.302	0.133	0.232	0.136	0.229	0.139	0.238	0.131	0.224
	192	0.145	0.242	0.165	0.241	0.154	0.248	0.163	0.261	0.168	0.279	0.143	0.239	0.197	0.311	0.149	0.247	0.152	0.244	0.153	0.251	0.160	0.248
	336	0.160	0.258	0.164	0.269	0.171	0.264	0.176	0.277	0.196	0.308	0.161	0.259	0.213	0.328	0.161	0.259	0.168	0.262	0.169	0.266	0.160	0.248
	720	0.182	0.280	0.190	0.284	0.209	0.297	0.212	0.308	0.203	0.312	0.191	0.286	0.233	0.344	0.197	0.297	0.205	0.293	0.206	0.297	0.192	0.298
	Avg.	<b>0.154</b>	<b>0.252</b>	0.165	0.254	0.169	0.261	0.173	0.272	0.182	0.292	<u>0.156</u>	<u>0.253</u>	0.207	0.321	0.160	0.259	0.165	0.257	0.167	0.263	0.158	0.252

Table 8: Full results of long-term forecasting with a 96-step lookback window (Part III). The input sequence length  $L$  is set to 96 for all baselines. All results are averaged across four different forecasting horizon:  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively.

Models	DMANet		iTransformer		Fdfformer		TimeMixer		PatchTST		Crossformer		TimesNet		TIDE		DLinear		FreTS		FEDformer		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
PEMS03	12	<b>0.064</b>	<b>0.167</b>	<u>0.071</u>	<u>0.174</u>	<u>0.068</u>	<u>0.174</u>	0.076	0.188	0.099	0.216	0.090	0.203	0.085	0.192	0.178	0.305	0.122	0.243	0.083	0.194	0.126	0.251
	24	<b>0.086</b>	<b>0.193</b>	<u>0.093</u>	<u>0.201</u>	0.094	0.205	0.113	0.226	0.142	0.259	0.121	0.240	0.118	0.223	0.257	0.371	0.201	0.317	0.127	0.198	0.241	0.275
	48	<u>0.132</u>	<u>0.239</u>	<b>0.125</b>	<b>0.236</b>	0.152	0.262	0.191	0.292	0.211	0.319	0.202	0.317	0.155	0.260	0.379	0.463	0.333	0.425	0.202	0.310	0.227	0.348
	Avg.	<b>0.094</b>	<b>0.200</b>	<u>0.096</u>	<u>0.204</u>	0.105	0.214	0.127	0.235	0.151	0.265	0.138	0.253	0.119	0.271	0.271	0.380	0.219	0.295	0.137	0.234	0.167	0.291
	12	<b>0.069</b>	<b>0.168</b>	<u>0.078</u>	<u>0.183</u>	0.085	0.189	0.092	0.204	0.105	0.224	0.098	0.218	0.087	0.195	0.219	0.340	0.148	0.272	0.097	0.209	0.138	0.262
PEMS04	24	<b>0.082</b>	<b>0.185</b>	<u>0.095</u>	<u>0.205</u>	0.117	0.224	0.128	0.243	0.153	0.257	0.131	0.256	0.103	0.215	0.292	0.398	0.224	0.340	0.144	0.258	0.177	0.293
	48	<b>0.107</b>	<b>0.216</b>	<u>0.120</u>	<u>0.233</u>	0.174	0.276	0.213	0.315	0.229	0.339	0.205	0.326	0.136	0.250	0.409	0.478	0.335	0.437	0.223	0.328	0.270	0.368
	Avg.	<b>0.086</b>	<b>0.190</b>	<u>0.098</u>	<u>0.207</u>	0.125	0.215	0.144	0.254	0.162	0.273	0.145	0.267	0.109	0.220	0.307	0.405	0.236	0.350	0.148	0.265	0.195	0.308
	12	<b>0.057</b>	<b>0.152</b>	0.067	0.165	<u>0.063</u>	<u>0.158</u>	0.073	0.184	0.095	0.207	0.094	0.200	0.082	0.181	0.173	0.304	0.115	0.242	0.078	0.185	0.109	0.225
	PEMS07	24	<b>0.074</b>	<b>0.174</b>	<u>0.088</u>	<u>0.190</u>	0.089	0.192	0.111	0.219	0.150	0.262	0.139	0.247	0.101	0.204	0.271	0.383	0.210	0.329	0.127	0.239	0.125
48		<b>0.109</b>	<b>0.211</b>	<u>0.110</u>	<u>0.215</u>	0.136	0.241	0.237	0.328	0.253	0.340	0.311	0.369	0.134	0.238	0.446	0.495	0.398	0.458	0.220	0.317	0.165	0.288
Avg.		<b>0.080</b>	<b>0.179</b>	<u>0.088</u>	<u>0.190</u>	0.096	0.197	0.140	0.244	0.166	0.270	0.181	0.272	0.106	0.208	0.297	0.394	0.241	0.343	0.142	0.247	0.133	0.282
12		<b>0.066</b>	<b>0.167</b>	<u>0.079</u>	<u>0.182</u>	0.081	0.185	0.091	0.201	0.168	0.232	0.165	0.214	0.112	0.212	0.227	0.343	0.154	0.276	0.096	0.204	0.173	0.273
PEMS08		24	<b>0.085</b>	<b>0.192</b>	0.115	0.219	<u>0.112</u>	<u>0.214</u>	0.137	0.246	0.224	0.281	0.215	0.260	0.141	0.238	0.318	0.409	0.248	0.353	0.152	0.256	0.210
	48	<b>0.121</b>	<b>0.235</b>	0.186	<u>0.235</u>	<u>0.174</u>	<u>0.267</u>	0.265	0.343	0.321	0.354	0.315	0.335	0.198	0.283	0.497	0.510	0.440	0.470	0.247	0.331	0.320	0.394
	Avg.	<b>0.090</b>	<b>0.198</b>	0.127	<u>0.212</u>	<u>0.122</u>	0.222	0.164	0.263	0.238	0.289	0.232	0.270	0.150	0.244	0.347	0.421	0.281	0.366	0.165	0.264	0.234	0.326

Table 9: Univariate long-term forecasting results on ETT datasets. Following PatchTST and Mod-erTCN, input length is fixed as 336 and prediction lengths are  $T \in \{96, 192, 336, 720\}$ . The best and second-best results are highlighted in **bold** and underlined, respectively.

Models	DMANet		ModernTCN		iTransformer		TimeMixer		PatchTST		DLinear		Pyraformer		FEDformer		Autoformer		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh1	96	<b>0.026</b>	<b>0.122</b>	<b>0.026</b>	<b>0.121</b>	0.029	0.127	0.029	0.128	0.029	0.126	0.028	0.123	0.127	0.281	0.033	0.140	0.056	0.183
	192	<b>0.039</b>	<b>0.150</b>	<b>0.040</b>	<b>0.152</b>	0.045	0.162	0.044	0.160	0.043	0.158	0.045	0.156	0.205	0.343	0.058	0.186	0.081	0.216
	336	<b>0.052</b>	<b>0.172</b>	<b>0.053</b>	<b>0.173</b>	0.059	0.189	0.058	0.185	0.056	0.183	0.061	0.182	0.302	0.457	0.084	0.231	0.076	0.218
	720	<b>0.072</b>	<b>0.203</b>	<b>0.073</b>	<b>0.206</b>	0.080	0.218	0.081	0.218	0.080	0.217	0.080	0.210	0.387	0.485	0.102	0.250	0.110	0.267
	Avg.	<b>0.047</b>	<b>0.162</b>	<b>0.048</b>	<b>0.163</b>	0.053	0.174	0.053	0.173	0.052	0.171	0.054	0.168	0.255	0.392	0.069	0.202	0.081	0.221
ETTh2	96	<b>0.063</b>	<b>0.182</b>	<b>0.065</b>	<b>0.183</b>	0.071	0.193	0.068	0.187	0.071	0.192	<b>0.063</b>	<b>0.183</b>	0.074	0.208	0.067	0.198	0.065	0.189
	192	<b>0.093</b>	<b>0.228</b>	0.095	0.232	0.109	0.248	0.101	0.236	0.102	0.237	<b>0.092</b>	<b>0.227</b>	0.116	0.252	0.102	0.245	0.118	0.256
	336	<b>0.117</b>	<b>0.260</b>	<b>0.119</b>	<b>0.261</b>	0.141	0.289	0.133	0.278	0.130	0.274	0.119	0.261	0.143	0.295	0.130	0.279	0.154	0.305
	720	<b>0.167</b>	<b>0.317</b>	<b>0.173</b>	<b>0.323</b>	0.190	0.343	0.183	0.332	0.179	0.328	0.175	0.320	0.197	0.338	0.178	0.325	0.182	0.335
	Avg.	<b>0.110</b>	<b>0.247</b>	0.113	0.250	0.128	0.268	0.121	0.258	0.121	0.258	<b>0.112</b>	<b>0.248</b>	0.133	0.273	0.119	0.262	0.130	0.271
ETTh1	96	<b>0.054</b>	<b>0.176</b>	<b>0.055</b>	<b>0.179</b>	0.059	0.185	0.057	0.181	0.056	0.181	0.056	0.180	0.099	0.277	0.079	0.215	0.071	0.206
	192	<b>0.066</b>	<b>0.200</b>	<b>0.070</b>	<b>0.205</b>	0.073	0.208	0.072	0.204	0.076	0.210	0.071	0.204	0.174	0.346	0.104	0.245	0.114	0.262
	336	<b>0.073</b>	<b>0.215</b>	<b>0.074</b>	<b>0.214</b>	0.084	0.223	0.085	0.227	0.094	0.242	0.098	0.244	0.198	0.370	0.119	0.270	0.107	0.258
	720	<b>0.082</b>	<b>0.227</b>	<b>0.086</b>	<b>0.232</b>	0.089	0.236	0.083	0.227	0.101	0.250	0.189	0.359	0.209	0.348	0.142	0.299	0.126	0.283
	Avg.	<b>0.069</b>	<b>0.205</b>	<b>0.071</b>	<b>0.206</b>	0.076	0.213	0.074	0.210	0.082	0.221	0.104	0.247	0.170	0.335	0.111	0.257	0.105	0.252
ETTh2	96	<b>0.121</b>	<b>0.269</b>	<b>0.124</b>	0.274	0.136	0.287	0.133	0.283	0.130	0.276	0.131	0.279	0.152	0.303	0.128	<b>0.271</b>	0.153	0.306
	192	<b>0.154</b>	<b>0.310</b>	<b>0.164</b>	<b>0.321</b>	0.187	0.342	0.190	0.341	0.181	0.331	0.176	0.329	0.197	0.370	0.185	0.330	0.204	0.351
	336	<b>0.174</b>	<b>0.336</b>	<b>0.171</b>	<b>0.336</b>	0.219	0.374	0.226	0.379	0.226	0.379	0.209	<b>0.367</b>	0.238	0.385	0.231	0.378	0.246	0.389
	720	<b>0.211</b>	<b>0.371</b>	<b>0.228</b>	<b>0.384</b>	0.253	0.403	0.241	0.396	0.253	0.406	0.276	0.426	0.274	0.435	0.278	0.420	0.268	0.409
	Avg.	<b>0.165</b>	<b>0.322</b>	<b>0.172</b>	<b>0.329</b>	0.199	0.352	0.198	0.350	0.198	0.348	0.198	0.350	0.215	0.373	0.206	0.350	0.218	0.364

Table 10: Full results of short-term forecasting on supplementary datasets from domains including Health & Medical (ILI, COVID-19), Web Events (Wiki, Website), Finance (NASDAQ, SP500, DowJones), Market (CarSales), Energy (Power), and Society (Unemp). The best and second-best results are highlighted in **bold** and underlined, respectively.

Model	DMANet		TimeMixer		FilterNet		FITS		DLinear		Fredformer		PatchTST		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ILI	24	<b>1.746</b>	<b>0.813</b>	2.110	0.879	2.190	0.870	4.265	1.523	3.158	1.243	2.098	0.894	<u>2.046</u>	<u>0.849</u>
	36	<u>1.718</u>	<b>0.817</b>	2.084	0.890	1.902	<u>0.862</u>	3.718	1.363	3.009	1.200	<b>1.712</b>	0.867	2.344	0.912
	48	<b>1.744</b>	<b>0.826</b>	<u>1.961</u>	<u>0.866</u>	2.051	0.882	3.994	1.422	2.994	1.194	2.054	0.922	2.123	0.883
	60	<b>1.842</b>	<b>0.839</b>	1.926	<u>0.878</u>	2.151	0.925	4.543	1.554	3.172	1.232	<u>1.925</u>	0.913	2.001	0.895
	Avg	<b>1.763</b>	<b>0.824</b>	2.020	<u>0.878</u>	2.073	0.885	4.130	1.465	3.083	1.217	<u>1.947</u>	0.899	2.128	0.885
Covid19	3	<b>1.098</b>	<b>0.489</b>	1.237	0.547	1.195	0.555	2.039	0.790	2.386	0.909	<u>1.165</u>	<u>0.548</u>	1.220	0.573
	6	<u>1.735</u>	<b>0.625</b>	2.003	0.739	1.839	0.711	2.683	0.919	3.220	1.053	<b>1.465</b>	<u>0.685</u>	1.982	0.762
	9	<b>2.167</b>	<b>0.722</b>	2.594	0.860	2.537	0.897	3.147	1.050	3.803	1.160	<u>2.145</u>	<u>0.845</u>	2.633	0.916
	12	<b>2.640</b>	<b>0.843</b>	3.103	<u>0.981</u>	<u>2.782</u>	0.956	3.630	1.156	4.524	1.288	2.833	<u>0.984</u>	3.050	1.030
	Avg	<u>1.910</u>	<b>0.670</b>	2.234	0.782	2.088	0.780	2.875	0.979	3.483	1.102	<b>1.902</b>	<u>0.765</u>	2.221	0.820
NASDAQ	24	<b>0.118</b>	<b>0.214</b>	<u>0.122</u>	<u>0.221</u>	0.130	0.230	0.140	0.244	0.155	0.274	0.128	0.226	0.127	0.224
	36	<b>0.158</b>	<b>0.260</b>	0.183	0.279	0.175	0.273	0.184	0.284	0.196	0.306	<u>0.170</u>	<u>0.268</u>	0.174	0.269
	48	<b>0.200</b>	<b>0.296</b>	<b>0.200</b>	<u>0.298</u>	0.224	0.314	0.234	0.324	0.244	0.344	<u>0.218</u>	0.306	0.225	0.314
	60	<b>0.233</b>	<b>0.323</b>	<u>0.238</u>	<u>0.328</u>	0.259	0.340	0.282	0.357	0.318	0.401	0.262	0.339	0.265	0.339
	Avg	<b>0.177</b>	<b>0.273</b>	<u>0.186</u>	<u>0.281</u>	0.197	0.289	0.210	0.302	0.228	0.331	0.194	0.285	0.198	0.286
Wiki	3	<u>6.116</u>	<u>0.372</u>	6.209	0.392	6.234	0.402	7.470	0.496	6.254	0.438	6.190	0.387	<b>6.112</b>	<b>0.380</b>
	6	<b>6.419</b>	<b>0.388</b>	6.475	0.402	6.460	0.401	8.326	0.544	6.579	0.467	6.696	0.404	<u>6.425</u>	<u>0.395</u>
	9	<b>6.665</b>	<b>0.402</b>	6.702	0.418	6.697	0.416	8.869	0.564	6.776	0.508	6.768	<u>0.411</u>	<u>6.743</u>	0.426
	12	<u>6.824</u>	<u>0.411</u>	6.902	0.426	6.899	0.426	9.394	0.608	6.927	0.513	7.168	0.424	<b>6.814</b>	0.414
	Avg	<b>6.506</b>	<b>0.393</b>	6.572	0.409	6.572	0.411	8.515	0.553	6.634	0.481	6.705	0.406	<u>6.523</u>	<u>0.404</u>
SP500	24	<b>0.153</b>	<b>0.271</b>	<u>0.159</u>	<u>0.288</u>	0.181	0.317	0.193	0.334	0.189	0.330	0.181	0.315	0.164	0.298
	36	<b>0.205</b>	<b>0.315</b>	<u>0.218</u>	<u>0.343</u>	0.224	<u>0.341</u>	0.259	0.389	0.250	0.363	0.239	0.365	0.221	<u>0.341</u>
	48	<b>0.250</b>	<b>0.348</b>	<u>0.264</u>	<u>0.367</u>	0.280	0.384	0.324	0.439	0.291	0.398	0.283	0.394	0.278	0.397
	60	<b>0.293</b>	<b>0.383</b>	0.322	0.416	0.332	0.416	0.391	0.486	0.377	0.475	0.341	0.438	<u>0.321</u>	<u>0.409</u>
	Avg	<b>0.225</b>	<b>0.329</b>	<u>0.241</u>	<u>0.353</u>	0.254	0.365	0.291	0.412	0.277	0.391	0.261	0.378	0.246	0.361
DowJones	24	<b>7.325</b>	<b>0.666</b>	8.327	0.683	8.000	0.683	7.974	0.690	<u>7.590</u>	<u>0.670</u>	7.758	0.672	7.641	<u>0.670</u>
	36	<b>10.422</b>	<b>0.800</b>	11.192	0.813	12.011	0.823	11.907	0.837	<u>10.986</u>	<u>0.803</u>	11.456	0.808	11.210	0.807
	48	<b>13.975</b>	<b>0.917</b>	15.278	0.945	14.814	0.933	15.821	0.969	<u>14.157</u>	<u>0.922</u>	14.696	0.921	14.866	0.935
	60	<b>16.106</b>	<b>1.016</b>	20.997	1.067	18.932	1.054	19.320	1.077	<u>18.018</u>	<u>1.035</u>	18.058	<u>1.032</u>	17.947	1.036
	Avg	<b>11.957</b>	<b>0.850</b>	13.948	0.877	13.439	0.873	13.755	0.893	<u>12.688</u>	<u>0.857</u>	12.992	0.858	12.916	0.862
CarSales	24	<b>0.318</b>	<b>0.314</b>	0.320	0.318	<b>0.318</b>	0.319	0.359	0.347	0.354	0.350	0.319	0.326	0.319	0.319
	36	<u>0.332</u>	<b>0.327</b>	<u>0.332</u>	0.331	<b>0.331</b>	<u>0.330</u>	0.373	0.360	0.368	0.365	0.333	0.335	<u>0.332</u>	<u>0.330</u>
	48	0.346	0.340	0.345	0.343	<b>0.342</b>	<b>0.341</b>	0.385	0.370	0.382	0.379	0.349	0.344	0.347	0.344
	60	<u>0.357</u>	<u>0.352</u>	<b>0.355</b>	<b>0.351</b>	0.352	0.349	0.399	0.385	0.388	0.380	0.359	0.349	<b>0.355</b>	0.348
	Avg	<u>0.338</u>	<b>0.333</b>	<u>0.338</u>	0.336	<b>0.336</b>	<u>0.335</u>	0.379	0.365	0.373	0.368	0.340	0.338	<u>0.338</u>	<u>0.335</u>
Power	24	<b>1.293</b>	<b>0.865</b>	<u>1.341</u>	<u>0.881</u>	1.410	0.916	1.491	0.944	1.390	0.916	1.410	0.913	1.468	0.935
	36	<b>1.334</b>	<b>0.875</b>	<u>1.420</u>	<u>0.914</u>	1.590	0.968	1.621	0.994	1.518	0.957	1.538	0.953	1.593	0.972
	48	<b>1.408</b>	<b>0.917</b>	<u>1.567</u>	<u>0.963</u>	1.680	1.009	1.775	1.052	1.610	0.995	1.652	1.008	1.710	1.020
	60	<b>1.456</b>	<b>0.940</b>	<u>1.609</u>	<u>0.988</u>	1.776	1.053	1.958	1.122	1.679	1.020	1.752	1.049	1.829	1.064
	Avg	<b>1.373</b>	<b>0.899</b>	<u>1.484</u>	<u>0.937</u>	1.614	0.986	1.711	1.028	1.549	0.972	1.588	0.981	1.650	0.998
Website	3	0.083	0.209	0.086	0.215	0.084	0.213	0.191	0.320	0.159	0.288	<u>0.080</u>	<b>0.207</b>	0.089	0.217
	6	<b>0.112</b>	<b>0.238</b>	0.124	0.248	0.116	0.242	0.235	0.356	0.182	0.302	0.116	<u>0.241</u>	0.121	0.246
	9	<u>0.154</u>	<b>0.265</b>	0.159	0.275	0.151	0.269	0.276	0.372	0.220	0.330	0.150	0.268	0.157	0.273
	12	<u>0.199</u>	<b>0.294</b>	0.204	0.306	<u>0.194</u>	0.297	0.409	0.484	0.255	0.355	0.196	0.301	0.200	0.302
	Avg	0.137	<b>0.252</b>	0.143	0.261	0.136	0.255	0.278	0.383	0.204	0.319	<b>0.135</b>	<u>0.254</u>	0.141	0.259
Unemp	3	<b>0.010</b>	<b>0.052</b>	0.015	0.074	<u>0.012</u>	0.062	0.161	0.289	0.072	0.200	0.013	0.068	<u>0.012</u>	<u>0.060</u>
	6	<b>0.036</b>	<b>0.117</b>	0.057	0.154	<u>0.043</u>	0.130	0.229	0.345	0.115	0.255	0.046	0.139	<u>0.043</u>	<u>0.127</u>
	9	<b>0.081</b>	<b>0.180</b>	0.109	0.213	0.107	0.216	0.369	0.443	0.191	0.329	0.095	0.198	<u>0.093</u>	<u>0.190</u>
	12	<b>0.127</b>	<b>0.234</b>	0.195	0.293	0.155	0.255	0.475	0.500	0.240	0.386	<u>0.148</u>	<u>0.250</u>	0.164	0.261
	Avg	<b>0.064</b>	<b>0.146</b>	0.094	0.183	<u>0.079</u>	0.166	0.308	0.394	0.154	0.292	0.075	<u>0.163</u>	0.078	0.160



## D.5 RESULTS FOR HYPERPARAMETER ANALYSIS

In this section, we also explore the effect of the hyperparameters used in our experiments, including the depth-wise convolution kernel size  $K$ , the depth-wise convolution kernel stride size  $s$ , the channel change  $c$  and  $\lambda$  on the loss function.

The channel change  $c$  signifies the alteration in the number of channels during the downsampling process, where values below 1 denote a reduction in channel quantity, whereas values exceeding 1 indicate channel expansion.

For  $\lambda$ , according to FreDF Wang et al. (2025), the loss function is a weighted sum of the time-domain MSE and the frequency-domain MAE.  $\lambda$  represents the proportion of the frequency MAE in the loss function, and  $(1 - \lambda)$  represents the proportion of the time-domain MSE.

We show the experimental results from Table.11 to Table.14.

Table 11: Impact of kernel size. A lower MSE or MAE indicates a better performance.

Models	Metrics	Weather		ETTh2		ETTM2	
		96	336	96	336	96	336
$K = 1$	MSE	0.151	0.274	0.289	0.393	0.169	0.289
	MAE	0.194	0.325	0.334	0.411	0.248	0.326
$K = 3$	MSE	0.148	0.256	0.280	0.393	0.165	0.289
	MAE	0.191	0.282	0.329	0.410	0.244	0.325
$K = 5$	MSE	0.149	0.259	0.286	0.393	0.168	0.296
	MAE	0.192	0.285	0.332	0.410	0.247	0.331
$K = 7$	MSE	0.150	0.258	0.286	0.394	0.167	0.292
	MAE	0.193	0.283	0.331	0.411	0.245	0.327

Table 12: Impact of stride size. A lower MSE or MAE indicates a better performance.

Models	Metrics	Weather		ETTh2		ETTM2	
		96	336	96	336	96	336
$s = 1$	MSE	0.151	0.259	0.281	0.423	0.170	0.297
	MAE	0.194	0.284	0.331	0.421	0.247	0.331
$s = 2$	MSE	0.148	0.256	0.280	0.393	0.165	0.289
	MAE	0.191	0.282	0.329	0.410	0.244	0.325
$s = 3$	MSE	0.149	0.259	0.274	0.393	0.167	0.290
	MAE	0.192	0.284	0.325	0.410	0.246	0.326
$s = 4$	MSE	0.148	0.257	0.279	0.395	0.167	0.291
	MAE	0.190	0.282	0.326	0.411	0.246	0.327

Table 13: Impact of channel change. A lower MSE or MAE indicates a better performance.

Models	Metrics	Weather		ETTh2		ETTm2	
		96	336	96	336	96	336
$c = 0.25$	MSE	0.149	0.259	0.278	0.396	0.167	0.291
	MAE	0.193	0.284	0.326	0.412	0.245	0.327
$c = 0.5$	MSE	0.148	0.256	0.280	0.393	0.165	0.289
	MAE	0.191	0.282	0.329	0.410	0.244	0.325
$c = 1$	MSE	0.149	0.261	0.284	0.408	0.171	0.293
	MAE	0.191	0.287	0.333	0.415	0.250	0.328
$c = 2$	MSE	0.149	0.257	0.281	0.401	0.169	0.294
	MAE	0.192	0.283	0.332	0.415	0.247	0.327
$c = 4$	MSE	0.152	0.261	0.292	0.398	0.173	0.293
	MAE	0.196	0.287	0.337	0.415	0.250	0.328

Table 14: Impact of  $\lambda$  in loss. A lower MSE or MAE indicates a better performance.

Models	Metrics	Weather		ETTh2		ETTm2	
		96	336	96	336	96	336
$\lambda = 0.1$	MSE	0.149	0.257	0.289	0.392	0.168	0.297
	MAE	0.191	0.283	0.333	0.412	0.246	0.332
$\lambda = 0.3$	MSE	0.149	0.259	0.289	0.394	0.167	0.297
	MAE	0.192	0.284	0.332	0.411	0.246	0.332
$\lambda = 0.5$	MSE	0.149	0.259	0.286	0.394	0.169	0.298
	MAE	0.191	0.284	0.331	0.412	0.246	0.332
$\lambda = 0.7$	MSE	0.149	0.259	0.290	0.396	0.169	0.297
	MAE	0.191	0.283	0.332	0.414	0.247	0.332
$\lambda = 1$	MSE	0.148	0.256	0.280	0.393	0.165	0.289
	MAE	0.191	0.282	0.329	0.410	0.244	0.325

## E MORE DETAILS OF COMPUTATIONAL COSTS

To comprehensively evaluate the efficiency and scalability of DMANet, we conducted controlled experiments on both synthetic and real-world datasets. Our analysis focuses on two key aspects: the computational overhead of our proposed components and the overall model’s performance compared to state-of-the-art methods.

### E.1 EFFICIENCY AND SCALABILITY ANALYSIS ON SYNTHETIC DATA

We first use synthetic data to perform a fine-grained analysis under controlled conditions, isolating the impact of sequence length and channel dimensions. With fixed hyperparameters (look-back window=96, batch size=64, etc.), we measure inference speed (ms) and peak GPU memory (MB) under two scenarios: (1) fixing the number of channels  $C$  while varying the sequence length  $T$ , and (2) fixing  $T$  while varying  $C$ . Each experiment was repeated 500 times for stability. The results are presented in Table.15.

Table 15: Inference Speed (ms) and Memory Usage (MB) Comparison Across Different Models and Configurations. Values for speed are reported as mean  $\pm$  std over 500 runs.

Configuration	DMANet		w/o-ESR		Chebyshev	
	Speed	Memory	Speed	Memory	Speed	Memory
$T = 256$	$1.376 \pm 0.122$	15.71	$1.309 \pm 0.359$	15.71	$1.916 \pm 0.150$	15.71
$T = 512$	$1.372 \pm 0.107$	31.29	$1.325 \pm 0.115$	31.29	$1.942 \pm 0.173$	31.29
$T = 1024$	$1.677 \pm 0.137$	65.38	$1.600 \pm 0.346$	65.38	$2.249 \pm 0.182$	65.38
$T = 2048$	$2.915 \pm 0.151$	146.49	$2.846 \pm 0.256$	146.49	$3.576 \pm 0.019$	146.49
$C = 48$	$1.518 \pm 0.103$	61.36	$1.444 \pm 0.222$	61.36	$1.523 \pm 0.206$	61.36
$C = 96$	$1.982 \pm 0.141$	122.40	$1.882 \pm 0.154$	122.40	$2.640 \pm 0.187$	122.40
$C = 192$	$3.731 \pm 0.047$	251.77	$3.580 \pm 0.072$	251.77	$4.250 \pm 0.182$	251.77
$C = 336$	$7.029 \pm 0.025$	471.27	$6.760 \pm 0.043$	471.27	$7.399 \pm 0.167$	471.27
Configuration	Linear		TransConv		Attention	
	Speed	Memory	Speed	Memory	Speed	Memory
$T = 256$	$1.228 \pm 0.352$	15.89	$1.405 \pm 0.128$	15.92	$3.065 \pm 0.234$	75.04
$T = 512$	$1.201 \pm 0.158$	32.02	$1.605 \pm 0.124$	31.68	$3.412 \pm 0.239$	280.09
$T = 1024$	$1.838 \pm 0.196$	68.39	$2.191 \pm 0.168$	66.70	$8.710 \pm 0.117$	1084.94
$T = 2048$	$5.079 \pm 0.117$	158.52	$3.818 \pm 0.089$	147.73	$29.417 \pm 0.224$	4270.15
$C = 48$	$1.414 \pm 0.225$	61.43	$1.915 \pm 0.161$	61.33	$3.892 \pm 0.186$	298.42
$C = 96$	$1.794 \pm 0.078$	122.31	$2.432 \pm 0.303$	119.46	$5.278 \pm 1.450$	332.12
$C = 192$	$3.374 \pm 0.054$	252.63	$4.418 \pm 0.281$	236.45	$8.956 \pm 0.035$	404.04
$C = 336$	$5.642 \pm 0.150$	471.27	$8.116 \pm 0.019$	414.28	$15.971 \pm 0.102$	518.38

From these results, we draw two key conclusions:

**1. The computational overhead of our dynamic anti-aliasing (ESR Filter) is negligible.** A direct comparison between DMANet and its ablated version (w/o ESR) reveals that the peak memory usage is nearly identical across all configurations. The time overhead introduced by the ESR filter is minimal, with a worst-case relative increase of only 2.4% (at  $T=2048$ ). Furthermore, DMANet is consistently faster than the variant using a classical Chebyshev filter. This empirically proves that our dynamic anti-aliasing mechanism is a computationally lightweight strategy that does not introduce a performance bottleneck.

**2. The efficiency and scalability of frequency-domain interpolation for upsampling.** We further validate our choice of upsampling mechanism by comparing it with common alternatives. The Attention-based method is not viable for long sequences due to the explosive, quadratic growth in its memory and time costs. While a simple Linear layer is fast, it scales poorly when processing very long sequences (e.g., at  $T=2048$ , its speed degrades significantly). Although Transposed Con-

volution is lightweight, our method is faster in most scenarios. In conclusion, our chosen frequency-domain interpolation achieves an excellent balance of cost-effectiveness and scalability across different data shapes.

## E.2 EFFICIENCY COMPARISON WITH STATE-OF-THE-ART MODELS ON REAL-WORLD DATASETS

Next, we benchmark the overall efficiency of DMANet against leading SOTA models on real-world datasets. We fix the input and prediction lengths ( $T = 96, F = 96$ ) to ensure a fair comparison and report on memory usage, multiply-accumulate operations (MACs), and inference speed, alongside predictive accuracy.

Table 16: A comparison of Speed, memory consumption (Memory) and multiply-accumulate operations (MACs) for DMANet and five other models. To ensure a fair comparison, we fix the prediction length  $F = 96$  and the input length  $T = 96$ .

Dataset	ETTm2					Weather				
Metric	Memory	MACs	Speed (s / iter)	MSE	MAE	Memory	MACs	Speed (s / iter)	MSE	MAE
iTransformer	4.45MB	14.16M	0.0117	0.182	0.265	62.63MB	682.50M	0.0178	0.175	0.215
TimeMixer	71.91MB	12.15M	0.0350	0.182	0.267	169.20MB	41.40M	0.0623	0.163	0.209
TimesNet	215.78MB	72.60G	0.1483	0.181	0.261	103.49MB	18.07G	0.0730	0.169	0.220
PatchTST	146.60MB	4.33G	0.0305	0.180	0.264	153.64MB	1.24G	0.0683	0.189	0.230
DLinear	0.42MB	0.30M	0.0033	0.193	0.292	0.97MB	0.30M	0.0035	0.196	0.256
DMANet (Ours)	0.98MB	0.53M	0.0262	0.175	0.253	2.63MB	0.89M	0.0226	0.156	0.201

As shown in Table.16, DMANet demonstrates a state-of-the-art balance between efficiency and performance. Compared to Transformer-based models (iTransformer, PatchTST) and CNN-based models (TimesNet, TimeMixer), DMANet requires significantly less memory and fewer MACs while achieving superior forecasting accuracy. While DLinear is exceptionally fast due to its simple architecture, DMANet provides a substantial accuracy improvement with only a marginal increase in memory and MACs. These results confirm that the lightweight and scalable design choices validated in our synthetic experiments translate directly to a highly competitive and efficient model for real-world applications.

## F MORE DETAILS OF PRE-SAMPLING FILTERING

To comprehensively evaluate the robustness of our model and its generalization ability to different types of signal disturbance, we synthesized noise and superimposed it onto the original clean signals  $x_{\text{clean}}$  to generate noisy signals  $x_{\text{noisy}}$  for model testing. The synthetic noise was generated using a unified framework that supports multiple noise types, with precise control over the intensity of the noise through parameters. Specifically, we implemented the following noise types:

- **Frequency-Domain Noise:** It includes High-frequency noise, Low-frequency noise, and Broadband noise. This type is generated by taking the Fast Fourier Transform (FFT) of the original signal, generating a band-limited or broadband random Gaussian noise spectrum in the frequency domain, and then converting it back to the time domain via Inverse Fast Fourier Transform (IFFT). The frequency band division for high- and low-frequency noise is controlled by the  $r_{\text{cut}}$  parameter, defined as the cutoff proportion in the frequency space.
- **Trend Noise:** Simulates slow-varying, non-periodic disturbances. This noise is generated by creating a low-order (e.g., quadratic) polynomial with random coefficients to simulate the trend component in the time series and adding it to the original signal.
- **Seasonal Noise:** Simulates periodic disturbances. This noise is generated by superimposing one or more sine waves with predefined base frequencies specified by the parameter  $f_{\text{seasonal}}$ , each having a random initial phase.

The noise intensity is precisely controlled by  $\epsilon$ , which defines the desired ratio of noise energy  $E_{\text{noise}}$  to clean signal energy  $E_{\text{clean}}$ , i.e.,  $E_{\text{noise}}/E_{\text{clean}}$ . After generating the noise, which can be denoted as **noise**, the noise energy is calculated and scaled accordingly to ensure that the noise added to the clean signal has a relative energy level consistent with  $\epsilon$ . The final noisy signal  $x_{\text{noisy}}$  is obtained by adding the scaled noise **noise**<sub>scaled</sub> to the original clean signal:  $x_{\text{noisy}} = x_{\text{clean}} + \text{noise}_{\text{scaled}}$ .

Then, we systematically analyze the performance of the model when faced with various signal distortions. In our experiments, concretely, we fixed the  $r_{\text{cut}}$  at 0.3, set  $f_{\text{seasonal}}$  to  $\{1/24, 1/12\}$ , and used  $\epsilon$  values of  $\{0.1, 0.2, 0.5\}$  in different experimental groups. The results are shown in Table.17.

**Comparative Study of Anti-Aliasing Strategies.** To further investigate our proposed Equivalent Sampling Rate mechanism and explore efficient anti-aliasing strategies, we conducted a comparative study on the Weather dataset using a 96-step lookback to predict a 720-step horizon. We benchmarked three distinct anti-aliasing configurations:

- **DMANet (ESR-based):** Our proposed model, which uses the architecture-aware ESR to dynamically determine the cutoff frequency for a sharp filter.
- **DMANet.but (Butterworth):** A variant where the ESR-based filter is replaced by a traditional 4th-order Butterworth low-pass filter, a well-established mathematical filter known for its maximally flat passband.
- **DMANet.mix (Fusion-based):** A hybrid model that first uses ESR to partition the spectrum and then processes the high- and low-frequency bands through separate convolutional layers before fusing them, designed to explore the utility of preserved high-frequency information.
- **DMANet.wo (No Filter):** A baseline variant that removes the anti-aliasing filter entirely, processing the raw input directly through the network to assess the necessity and impact of frequency-domain filtering.

The results under various noise conditions are summarized in Table.17. Overall, most of configurations demonstrate notable robustness, with only graceful performance degradation as noise intensity increases. This highlights the general effectiveness of incorporating a pre-sampling filtering stage to enhance noise resistance.

Our ablation study reveals a insight into the effectiveness of different anti-aliasing strategies. Theoretically, one might expect the Butterworth filter (DMANet.but), with its maximally flat passband, to excel at handling low-frequency and trend noise by preserving the signal fidelity in that band Yin et al. (2024). Conversely, our ESR-based hard-cutoff filter (DMANet) should be superior against high-frequency and seasonal noise due to its removal of aliasing-prone components.

Table 17: Robustness analysis of DMANet variants under different types and intensities of synthetic noise on the Weather dataset. All experiments use a 96-step lookahead to predict a 720-step horizon.

Model Variant	Noise Type	$\epsilon = 1\%$		$\epsilon = 5\%$		$\epsilon = 10\%$	
		MSE	MAE	MSE	MAE	MSE	MAE
DMANet	Seasonal	0.343	0.339	0.342	0.341	0.341	0.343
	Trend	0.344	0.340	0.345	0.345	0.351	0.358
	All (Broadband)	0.345	0.341	0.345	0.342	0.346	0.344
	Low-Frequency	0.345	0.340	0.347	0.342	0.349	0.347
	High-Frequency	0.344	0.340	0.343	0.340	0.343	0.341
DMANet_but	Seasonal	0.346	0.340	0.342	0.340	0.340	0.343
	Trend	0.347	0.342	0.348	0.347	0.354	0.359
	All (Broadband)	0.349	0.342	0.347	0.343	0.346	0.344
	Low-Frequency	0.352	0.344	0.349	0.345	0.350	0.347
	High-Frequency	0.345	0.341	0.343	0.340	0.343	0.343
DMANet_mix	Seasonal	0.353	0.343	0.350	0.345	0.350	0.348
	Trend	0.348	0.342	0.353	0.350	0.356	0.359
	All (Broadband)	0.352	0.344	0.353	0.344	0.357	0.348
	Low-Frequency	0.354	0.345	0.351	0.345	0.353	0.348
	High-Frequency	0.358	0.346	0.353	0.345	0.352	0.347
DMANet_wo	Seasonal	0.348	0.343	0.350	0.346	0.346	0.349
	Trend	0.348	0.343	0.351	0.351	0.357	0.361
	All (Broadband)	0.350	0.344	0.352	0.346	0.347	0.345
	Low-Frequency	0.355	0.346	0.355	0.349	0.353	0.349
	High-Frequency	0.350	0.343	0.350	0.345	0.351	0.344

Interestingly, our empirical results in Table.17 show that while performance is competitive on seasonal and high-frequency noise, DMANet consistently and significantly outperforms DMANet\_but on trend and low-frequency noise. This seemingly counter-intuitive result highlights a critical limitation of applying classical filters naively within a deep learning pipeline. While the Butterworth filter is static and optimally preserves its predefined passband, it is architecture-agnostic. It may still pass frequencies that, while low, are too high for the subsequent strided convolution to process without aliasing. In contrast, our ESR-based approach is architecture-aware. It does not aim to be a perfect mathematical filter in isolation; its sole purpose is to perfectly prepare the signal for the next layer. By dynamically calculating a precise cutoff based on the network’s own parameters, it ensures that no aliasing occurs at any stage, even if this means a slightly more aggressive filtering. This architectural synergy proves to be more practically effective.

Furthermore, the fusion-based DMANet\_mix consistently underperforms the other two variants. This result empirically supports our design rationale for employing a strict cutoff strategy: for a lightweight model, it is more effective to concentrate its limited capacity on core, learnable patterns rather than attempting to fit the complex and often noisy dynamics of high-frequency information. As observed in prior work like FITS Xu et al. (2024), removing a significant portion of high-frequency components largely preserves a time series’ dominant trends. The poor performance of DMANet\_mix indicates that simply preserving and processing this high-frequency content is less effective than principled filtering, likely because this band is dominated by noise that the model cannot distinguish from a true signal.

Collectively, these results validate that our ESR-based approach provides the most robust and adaptive solution. By dynamically and precisely removing only the frequencies that would cause aliasing, it not only focuses the model on the most decisive, learnable patterns but also achieves this with superior adaptability compared to static classical filters, all without the need for manual filter design.

---

## G MORE DETAILS OF OUR METHOD

### G.1 THE RATIONALE FOR THE EMBEDDING FIRST ARCHITECTURE

A critical challenge in multi-scale time series analysis is the fusion of features from different scales without introducing signal distortion. A common approach, which we term multi-scale first, involves downsampling the raw signal and then embedding each scale. However, this seemingly intuitive process hides a significant pitfall: the upsampling step required for feature fusion inevitably causes spectral distortion due to its reliance on a limited reconstruction basis. To circumvent this fundamental issue, our DMANet adopts a principled **embedding first** architecture, ensuring all operations are conducted with high fidelity within a unified feature space.

#### G.1.1 THE PITFALL OF PREMATURE MULTI-SCALE DECOMPOSITION

The multi-scale first approach, seen in models like TimeMixer Wang et al. (2024a), begins by decomposing the raw signal  $X$  into a set of time series  $\{X_m \in \mathbb{R}^{C \times s_m}, s_m < L\}$ . While feasible, the core flaw lies in the subsequent step of unifying these scales for feature fusion. To restore the original length  $L$ , each short sequence  $s_m$  must be upsampled using a linear layer,  $g_m: \mathbb{R}^{s_m} \rightarrow \mathbb{R}^L$ . This process is inherently problematic due to its limited representational capacity:

- **Limited Basis Vectors:** The weight matrix  $W \in \mathbb{R}^{L \times s_m}$  of the upsampling layer provides only  $s_m$  **column vectors**. These vectors form the *entire basis* available to reconstruct the output signal. Consequently, all reconstructed signals are confined to a very small,  $s_m$ -**dimensional subspace** of the target space  $\mathbb{R}^L$ .
- **Deformed Basis Vectors:** To approximate the diverse signals in the training data from this constrained basis, the model is forced to learn complex, **non-smooth, and oscillatory** basis vectors as a poor compromise.
- **Inevitable Spectral Distortion:** When a signal is reconstructed as a linear combination of these deformed basis vectors, it unavoidably inherits their unnatural properties. This leads to severe **spectral distortion**, corrupting the signal’s fidelity and polluting the final prediction.

#### G.1.2 EMBED FIRST: A PRINCIPLED APPROACH IN A UNIFIED FEATURE SPACE

Our DMANet architecture is designed to completely avoid the aforementioned reconstruction problem by first establishing a unified workspace for all operations.

1. **Defining a Unified Workspace:** We begin by projecting the information-complete raw signal  $X \in \mathbb{R}^{C \times L}$  into a new feature basis space using a linear layer. This generates a feature sequence  $X' \in \mathbb{R}^{C \times T}$ , creating a unified and consistent workspace for all subsequent synergistic operations.
2. **High-Fidelity Operations:** Within this consistent feature space, all core operations are performed in a principled manner:
  - **Downsampling:** Our anti-aliasing downsampling module pre-filters features in the frequency domain before reducing resolution, preventing information aliasing and ensuring reliable feature transfer across scales.
  - **Upsampling:** To restore resolution for feature fusion, we employ zero-padding in the frequency domain. This method is equivalent to ideal interpolation and relies on the **Fourier basis (sines and cosines)**—a **fixed, universal, and complete orthogonal basis**. Adhering to the Nyquist-Shannon sampling theorem, this ensures the **smoothest possible reconstruction**, free from the uncontrolled high-frequency artifacts generated by the alternative approach.

By ensuring all features are derived and processed with high-fidelity operations within the same basis space, we maintain inherent consistency and make feature fusion fundamentally more reliable.

### G.1.3 EMPIRICAL VALIDATION

To validate our theoretical analysis, we conducted a comprehensive comparison between our DMANet (Embedding First) and the alternative architecture (Multi-Scale First). We also performed an ablation study by removing the initial embedding module (w/o embed) to verify the effectiveness of operating within a latent space.

The results in Table.18 provide strong empirical support for our design.

Table 18: Comparative analysis and ablation study for the Embedding First architecture. Our full DMANet model is compared against the Multi-Scale First approach and a variant without the initial embedding module.

Model	Metric	ETTh1	ETTm1	Weather	Elect	Wiki	ILI	Unemp	Dowjone
DMANet (Embedding First)	MSE	<b>0.428</b>	<b>0.373</b>	<b>0.236</b>	<b>0.172</b>	<b>6.506</b>	<b>1.763</b>	<b>0.064</b>	<b>11.957</b>
	MAE	<b>0.429</b>	<b>0.385</b>	<b>0.262</b>	<b>0.265</b>	<b>0.393</b>	<b>0.824</b>	<b>0.146</b>	<b>0.850</b>
Multi-Scale First	MSE	0.441	0.385	0.242	0.181	6.555	2.097	0.074	12.420
	MAE	0.435	0.391	0.268	0.273	0.407	0.858	0.167	0.860
w/o embed	MSE	0.436	0.389	0.249	0.188	6.551	2.084	0.073	12.330
	MAE	0.427	0.392	0.274	0.277	0.406	0.879	0.163	0.857

The Multi-Scale First approach consistently underperforms our model. This performance gap is a direct, practical consequence of the spectral distortion introduced by its unprincipled, basis-limited reconstruction step.

Furthermore, we acknowledge that the initial linear mapping carries a potential risk of losing some temporal dependencies. This is a deliberate design choice, and its justification is twofold. First, we incorporate a learnable positional encoding to preserve crucial temporal context. Second, as our ablation study will demonstrate, the benefits of analyzing the series in a latent space—where patterns are more suitable for anti-aliasing and feature extraction—outweigh the alternative of operating directly on the raw signal. The placeholder for the w/o embed results in Table.18 will provide strong evidence for this superiority.

## G.2 PRINCIPLED ANTI-ALIASING VIA DYNAMIC FREQUENCY CUTOFF

For any given downsampling layer  $l$  in DMANet, its anti-aliasing operation is the application of a low-pass filter with a mathematically-derived, strict cutoff frequency. Given the layer’s convolutional parameters—kernel size  $k$ , stride  $s$ , and channel ratio  $c$ —we first calculate its Effective Sampling Ratio ( $ESR^l$ ) using Equation.?? to determine its true signal processing capability. This allows us to establish a new Nyquist frequency,  $f_{\text{Nyquist}}^l$ . As shown in Equation.??, all frequency components above this threshold are strictly zeroed out via a frequency-domain mask. Our method does not partially retain or vaguely attenuate high-frequency components; it employs a principled cutoff scheme where the threshold is dynamically determined for each layer.

### G.2.1 THE RATIONALE: FOCUSING ON LEARNABLE CORE PATTERNS

The core motivation for this strict cutoff strategy is to concentrate the model’s capacity on learnable, core patterns. High-frequency information in time series often contains significant noise or stochastic fluctuations that are difficult to model, and typically exceed the learning capacity of a lightweight model. Attempting to fit these complex dynamics can hinder the model from capturing the more decisive, underlying trends.

As observed in prior work like FITS Xu et al. (2024), removing a significant portion of high-frequency components largely preserves the overall shape and dominant trends of a time series. Our strategy builds on this insight: by proactively simplifying the learning task, we focus the model’s limited capacity on the low-frequency periodic and trend patterns that are most critical for the forecasting task, thereby achieving both efficient and accurate predictions.

### G.2.2 DYNAMIC ADAPTABILITY AND PARAMETER-FREE DESIGN

A key advantage of our method is its dynamic nature. In a complex multi-scale architecture, different layers may employ varying downsampling parameters ( $k, s, c$ ). Our framework automatically



derives a matching, optimal cutoff frequency for each specific layer. This ensures that the anti-aliasing protection remains effective and theoretically grounded across any architectural variation, eliminating the need for tedious, manual parameter tuning required by classical filters or the randomness of heuristic approaches.

Furthermore, this framework possesses theoretical flexibility. By adjusting the convolution parameters, the ESR can be controlled to retain more, or even all, frequency components. For instance, if parameters are set such that  $ESR = 1$  (e.g.,  $s = \min(K, C_{out}/C_{in})$ ), the cutoff frequency matches the original signal’s Nyquist frequency, meaning no valid frequency components are attenuated.

### G.2.3 ADDRESSING THE HIGH-FREQUENCY INFORMATION TRADE-OFF

We acknowledge that this design is built upon a core trade-off: we filter high-frequency components to prevent aliasing at the cost of potentially discarding useful information. This is a deliberate choice motivated by the efficiency and robustness goals for a lightweight model.

We also recognize that high-frequency information can be critical in certain scenarios, such as forecasting sharp spikes or in contexts where high-frequency harmonics are themselves key features. It is precisely for this reason that we deliberately conducted extensive supplementary experiments across a diverse range of domains (including Electricity, Weather, Transportation, Health, Web, Market, Energy, Society, Finance, etc.). The goal was to proactively probe the application boundaries of our method and provide a clear reference for its practical use.

To further investigate this trade-off, we will conduct a controlled experiment comparing our strict cutoff method with an alternative that handles frequencies differently. As shown in Table.19, we will compare our standard DMANet against a variant where, after identifying the cutoff frequency, both the low-frequency and the zeroed-out high-frequency components are independently passed through linear layers and then fused. This will help quantify the practical impact of the information contained in the high-frequency bands.

Table 19: Ablation study on the handling of high-frequency components. We compare our strict cutoff method with a variant that uses linear fusion for high and low frequencies.

Model	Metric	ETTh1	ETTm1	Weather	Elect	Wiki	ILI	Unemp	Dowjone
DMANet (Strict Cutoff)	MSE	<b>0.428</b>	<b>0.373</b>	<b>0.236</b>	<b>0.172</b>	<b>6.506</b>	<b>1.763</b>	<b>0.064</b>	<b>11.957</b>
	MAE	<b>0.429</b>	<b>0.385</b>	<b>0.262</b>	<b>0.265</b>	<b>0.393</b>	<b>0.824</b>	<b>0.146</b>	<b>0.850</b>
DMANet.mix (Fusion-based)	MSE	0.434	0.374	0.237	0.171	6.528	1.986	0.076	12.288
	MAE	0.433	0.385	0.263	0.265	0.398	0.867	0.161	0.858

In summary, DMANet’s anti-aliasing employs a precise, dynamic cutoff strategy tailored to each layer’s actual sampling capability. This design combines theoretical robustness with practical, parameter-free convenience, and its effectiveness is validated through extensive empirical analysis.

## H MORE DETAILS OF DEPENDENCY MODELING

We visualize the temporal dependencies and channel-wise relationships within a batch of the Electricity in Figure.3 and Figure.4 and for Weather in Figure.5 and Figure.6, comparing their states before and after processing by DMANet’s components. To further illustrate the differences between scenarios with and without the anti-aliasing filter, we selected the Electricity dataset to visualize the temporal dependency differences of upsampling before and after applying the anti-aliasing filter in Figure.7, as well as the channel dependency correlation differences of downsampling with and without the anti-aliasing filter in Figure.8.

DMANet tends to leverage more effective dependencies to capture future trends. Comparing the cases with and without the anti-aliasing filter, the figures reveal that the pre-processing anti-aliasing operation, acting as a low-pass filter, smooths or attenuates fine-grained dependencies that are susceptible to aliasing during sampling. This process helps to highlight the main temporal dependency patterns and channel relationships. Furthermore, convolution, leveraging its local receptive field, focuses on local patterns at neighboring time points. Thus, the combination of filtering and convolutional downsampling effectively extracts stable temporal features.

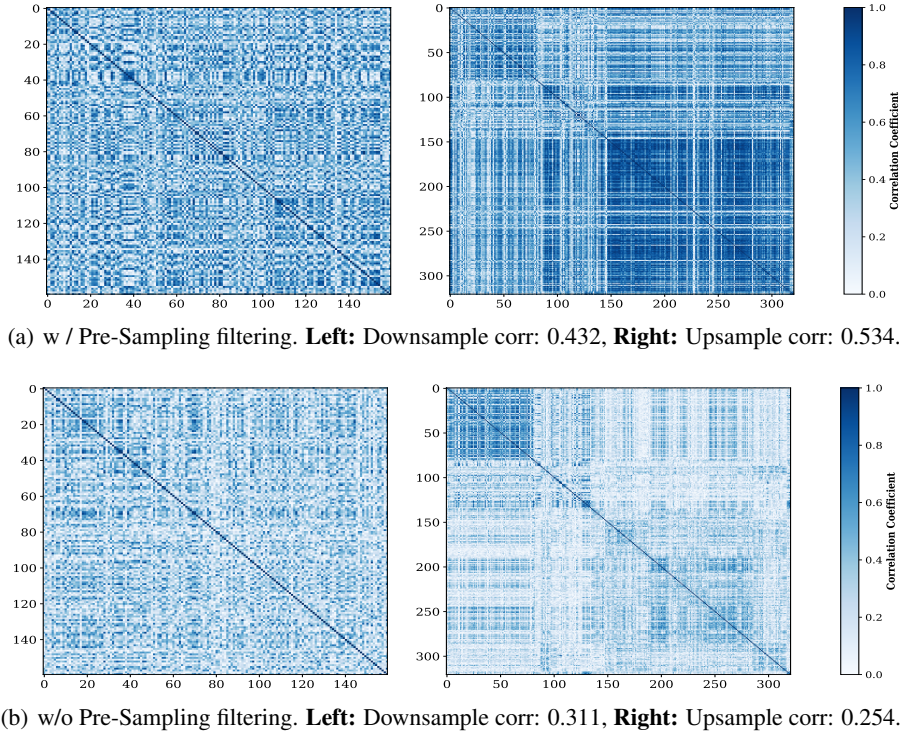


Figure 3: Visualization for channel dependency modeling on Electricity in the first layer of the second multiscale encoder block.

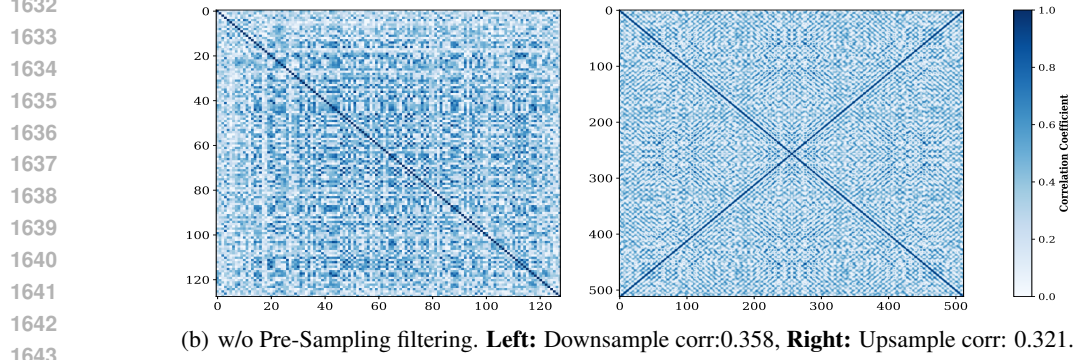
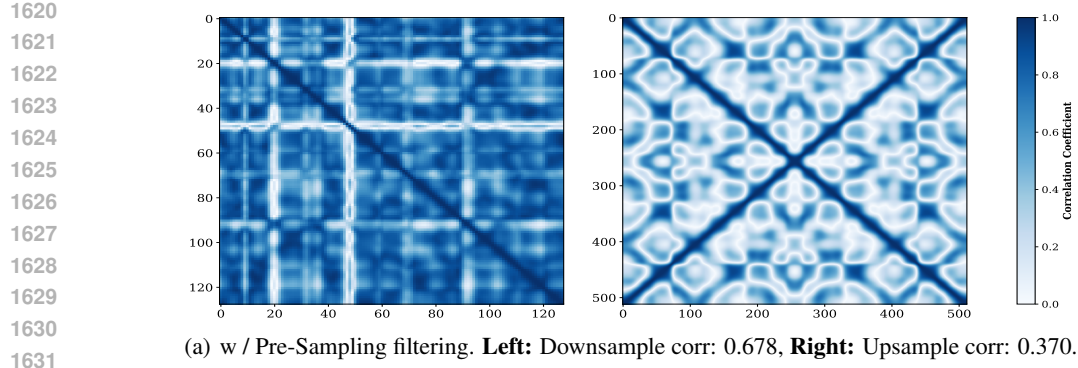


Figure 4: Visualization for temporal dependency modeling on Electricity in the first layer of the second multiscale encoder block.

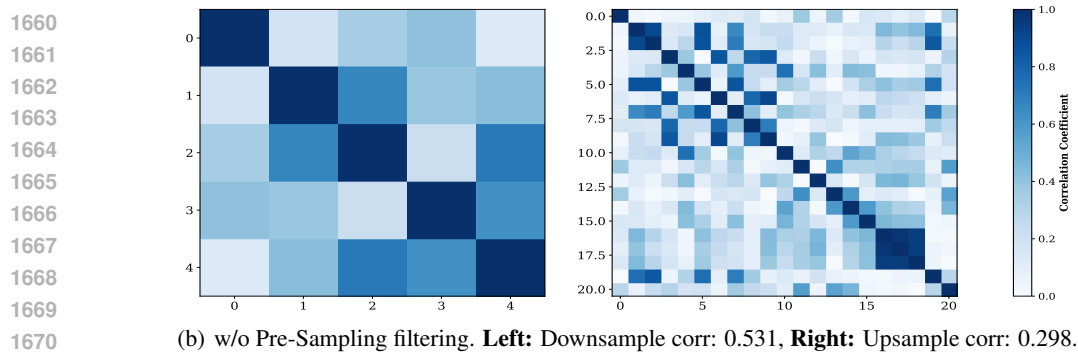
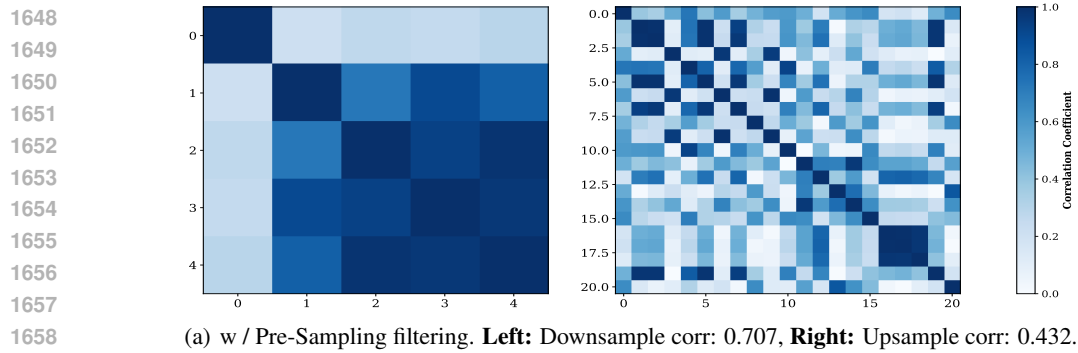


Figure 5: Visualization for channel dependency modeling on Weather in the first layer of the first multiscale encoder block.

1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727

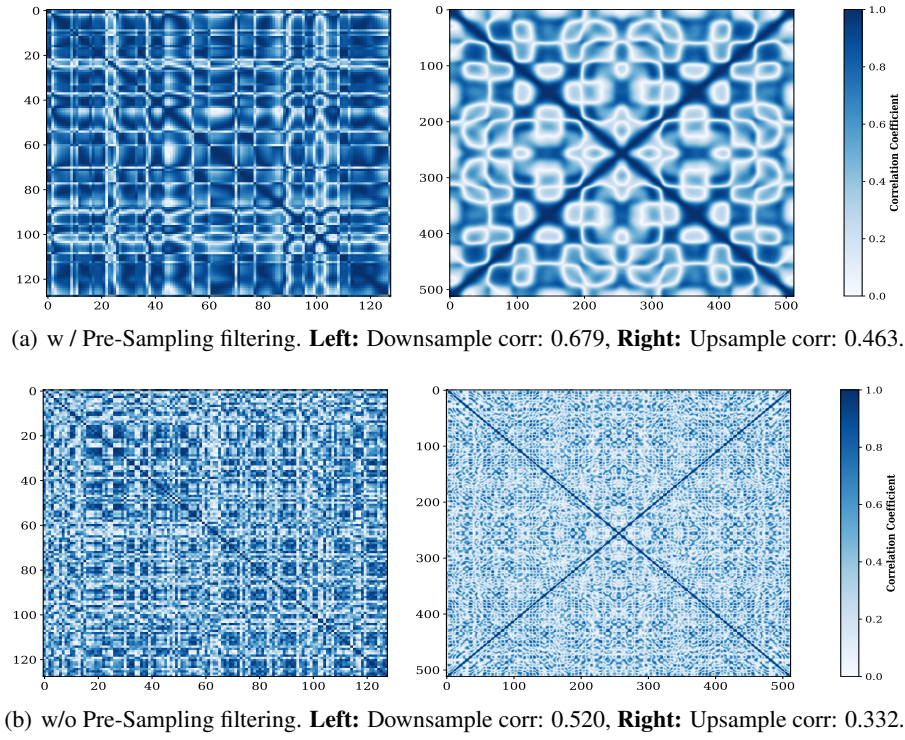


Figure 6: Visualization for temporal dependency modeling on Weather in the first layer of the first multiscale encoder block.

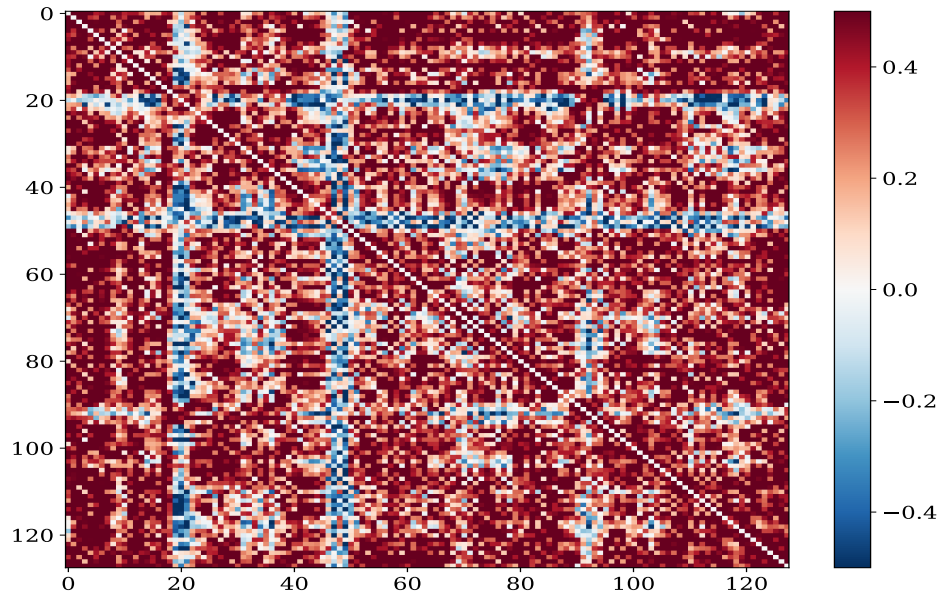


Figure 7: Temporal dependency differences in up-sampling with or without the application of an anti-alias filter on Electricity. Red indicates increased dependency after use anti-alias filter.



1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781

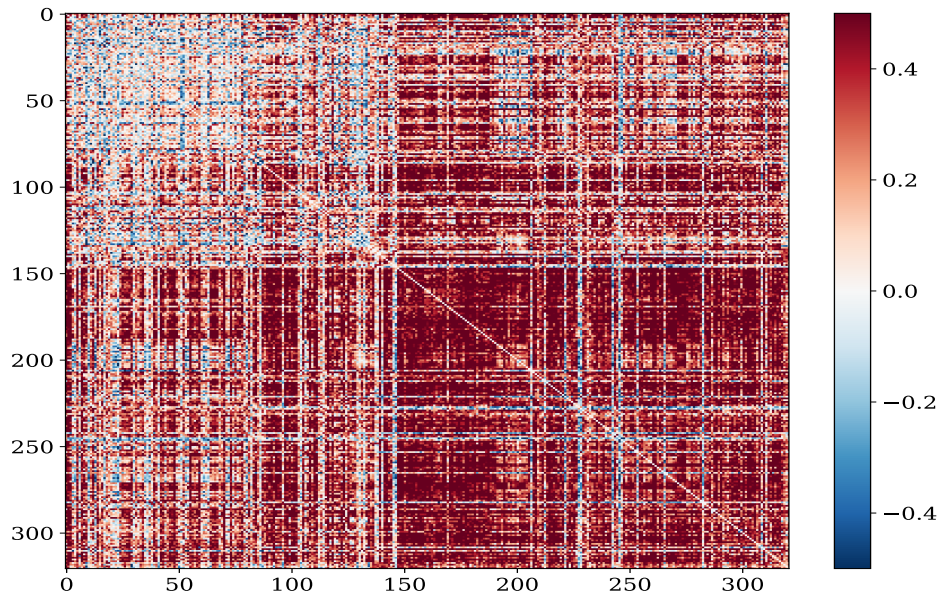


Figure 8: Channel dependency differences in down-sampling with or without the application of an anti-alias filter on Electricity. Red indicates increased dependency after use anti-alias filter.

---

1782 I THE USE OF LARGE LANGUAGE MODELS  
1783

1784 Large Language Models were employed as general-purpose assistive tools throughout the research  
1785 process. Specifically, LLMs were used to polish the language and improve the readability of this  
1786 manuscript, including refining grammar, improving clarity, and restructuring sentences for better  
1787 readability. The authors take full responsibility for the content of this paper.  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835

---

## J REFERENCES

### REFERENCES

- Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*, 2024.
- Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. *International Conference on Learning Representations*, 2024.
- Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.
- Luo Donghao and Wang Xue. ModernTCN: A modern pure convolution structure for general time series analysis. *International Conference on Learning Representations*, 2024.
- Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. In *International Conference on Machine Learning*, 2024.
- Julia Grabinski, Steffen Jung, Janis Keuper, and Margret Keuper. Frequencylowcut pooling–plug & play against catastrophic overfitting. In *European Conference on Computer Vision*, 2022. URL <https://arxiv.org/abs/2204.00491>.
- Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, 2024.
- Sungwon Han, Seungeon Lee, Meeyoung Cha, Serkan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=GUDnecJdJU>.
- Songtao Huang, Zhen Zhao, Can Li, and LEI BAI. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=wTLc79YNbh>.
- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Unb5CVPtac>.
- Chenghan Li, Mingchen Li, and Ruisheng Diao. TVNet: A novel time series analysis method based on dynamic convolution and 3d-variation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=MZDdTzN6Cy>.
- Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping, 2023a. URL <https://arxiv.org/abs/2305.10721>.
- Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023b.
- Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*, 2023c.
- Qinglong Liu, Cong Xu, Wenhao Jiang, Kaixuan Wang, Lin Ma, and Haifeng Li. Timestacker: A novel framework with multilevel observation for capturing nonstationary patterns in time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=5RYSqSKz9b>.
- Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *ICLR*, 2022a. URL <https://openreview.net/forum?id=0EXmFzUn5I>.

---

1890 Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring  
1891 the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:  
1892 9881–9893, 2022b.

1893

1894 Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long.  
1895 iTransformer: Inverted transformers are effective for time series forecasting. *International Con-*  
1896 *ference on Learning Representations*, 2024.

1897

1898 Xiaowen Ma, Zhen-Liang Ni, Shuai Xiao, and Xinghao Chen. Timepro: Efficient multivariate  
1899 long-term time series forecasting with variable- and time-aware hyper-state. In *Forty-second*  
1900 *International Conference on Machine Learning*, 2025. URL [https://openreview.net/](https://openreview.net/forum?id=s69Ei2VrIW)  
1901 [forum?id=s69Ei2VrIW](https://openreview.net/forum?id=s69Ei2VrIW).

1902 Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth  
1903 64 words: Long-term forecasting with transformers. In *The Eleventh International Confer-*  
1904 *ence on Learning Representations*, 2023. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=Jbdc0vTOcol)  
1905 [Jbdc0vTOcol](https://openreview.net/forum?id=Jbdc0vTOcol).

1906

1907 Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Insti-*  
1908 *tute of Electrical Engineers*, 47(2):617–644, 1928.

1909

1910 Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer:  
1911 Frequency debiased transformer for time series forecasting. In *KDD*, pp. 2400–2410, 2024. URL  
1912 <https://doi.org/10.1145/3637528.3671928>.

1913

1914 Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering  
1915 enhanced multivariate time series forecasting. *arXiv preprint arXiv:2412.10859*, 2024.

1916

1917 Jingzhe Shi, Qinwei Ma, Huan Ma, and Lei Li. Scaling law for time series forecasting. *Advances in*  
1918 *Neural Information Processing Systems*, 2024.

1919

1920 Hao Wang, Lichen Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao  
1921 Liu, Haoxuan Li, and Dacheng Tao. FreDF: Learning to forecast in the frequency domain. In  
1922 *The Thirteenth International Conference on Learning Representations*, 2025. URL [https://](https://openreview.net/forum?id=4A9IdSalul)  
1923 [openreview.net/forum?id=4A9IdSalul](https://openreview.net/forum?id=4A9IdSalul).

1924

1925 Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-  
1926 scale local and global context modeling for long-term series forecasting. In *The Eleventh Interna-*  
1927 *tional Conference on Learning Representations*, 2023. URL [https://openreview.net/](https://openreview.net/forum?id=zt53IDURlU)  
1928 [forum?id=zt53IDURlU](https://openreview.net/forum?id=zt53IDURlU).

1929

1930 Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang,  
1931 and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In  
1932 *The Twelfth International Conference on Learning Representations*, 2024a. URL [https://](https://openreview.net/forum?id=7oLshfEIC2)  
1933 [openreview.net/forum?id=7oLshfEIC2](https://openreview.net/forum?id=7oLshfEIC2).

1934

1935 Yuxuan Wang, Haixu Wu, Jiayang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jian-  
1936 min Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting  
1937 with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Pro-*  
1938 *cessing Systems*, 2024b. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=INAEUQ04lT)  
1939 [INAEUQ04lT](https://openreview.net/forum?id=INAEUQ04lT).

1940

1941 Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition trans-  
1942 formers with auto-correlation for long-term series forecasting. *CoRR*, abs/2106.13008, 2021.  
1943 URL <https://arxiv.org/abs/2106.13008>.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet:  
Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: modeling time series with 10k parameters. In *ICLR*,  
2024.



---

1944 Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filtarnet: Harnessing  
1945 frequency filters for time series forecasting. In *The Thirty-eighth Annual Conference on Neural*  
1946 *Information Processing Systems*, 2024a. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=ugL2D9idAD)  
1947 [ugL2D9idAD](https://openreview.net/forum?id=ugL2D9idAD).

1948 Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Long-  
1949 bing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series  
1950 forecasting. *Advances in Neural Information Processing Systems*, 36, 2024b.

1951 Tianyi Yin, Jingwei Wang, Yunlong Ma, Han Wang, Chenze Wang, Yukai Zhao, Min Liu, Weiming  
1952 Shen, and Yufeng Chen. Apollo-forecast: Overcoming aliasing and inference speed challenges in  
1953 language models for time series forecasting, 2024. URL [https://arxiv.org/abs/2412.](https://arxiv.org/abs/2412.12226)  
1954 [12226](https://arxiv.org/abs/2412.12226).

1955 Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. Freeformer:  
1956 Frequency enhanced transformer for multivariate time series forecasting. *arXiv preprint*  
1957 *arXiv:2501.13989*, 2025.

1958 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series  
1959 forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp.  
1960 11121–11128, 2023.

1961 Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency  
1962 for multivariate time series forecasting. In *ICLR*, 2022.

1963 Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency  
1964 enhanced decomposed transformer for long-term series forecasting. In *International Conference*  
1965 *on Machine Learning*, pp. 27268–27286. PMLR, 2022.

1966 Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis  
1967 by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

1968 Yifan Zhou, Zeqi Xiao, Shuai Yang, and Xingang Pan. Alias-free latent diffusion models: Improving  
1969 fractional shift equivariance of diffusion latent space. In *CVPR*, 2025.

1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997