

Article

MLPFormer: A Hybrid MLP–Transformer Architecture for Component-Specific Time Series Forecasting

Jing Chen ¹, Yihao Wang ¹, Xiao Chen ^{2,*}  and Mingxin Liu ³¹ College of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang 524088, China² Research Center for Marine Science, Hebei Normal University of Science and Technology, Qinhuangdao 066004, China³ College of Electronic and Information Engineering, Guangdong Ocean University, Zhanjiang 524088, China

* Correspondence: chenxiao0604@hevttc.edu.cn

Abstract

Long-term time series forecasting requires models that can represent smooth global trends and irregular local fluctuations without assigning the same capacity to both. Existing decomposition-based neural forecasters often separate signals but continue to process the resulting components with architecturally homogeneous encoders, which can overfit simple trends and underfit nonlinear residuals. This paper proposes MLPFormer, a hybrid MLP–Transformer framework for component-specific forecasting. The input sequence is decomposed by a moving-average filter into trend and residual terms. A shallow channel-wise MLP encodes the low-frequency trend, providing a low-capacity inductive bias for smooth structures, whereas a Transformer encoder with depthwise separable convolution models high-frequency residual dynamics and cross-variable dependencies. The two paths are fused and decoded by parallel linear and nonlinear prediction heads. Experiments on nine public benchmarks, with results reported as the mean over three random seeds, show average reductions of 11.8% in MSE and 8.1% in MAE relative to the mean of seven competitive baselines spanning Transformer-based, linear/MLP-based, and RNN-based paradigms published between 2023 and 2025. Ablation studies confirm that the gains arise from matching encoder capacity to component complexity rather than from increasing parameter count. MLPFormer provides a practical hybrid artificial intelligence design for forecasting tasks in which trend and residual structures coexist.

Keywords: time series forecasting; hybrid artificial intelligence; sequence decomposition; Transformer; multilayer perceptron; dual-path architecture

MSC: 68T07; 62M10



Academic Editor: Daniele Francesco Santamaria

Received: 7 June 2026

Revised: 4 July 2026

Accepted: 11 July 2026

Published: 13 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Time series forecasting is a fundamental task with broad applications, including financial risk assessment, meteorological early warning, and energy load planning [1–4]. Real-world time series are inherently composite signals, where slowly varying, low-frequency trends coexist with highly stochastic, high-frequency fluctuations [5–7]. Modeling these statistically heterogeneous components within a single artificial intelligence architecture remains a central challenge in the field.

Forecasting research has progressively advanced from classical statistical approaches to deep learning frameworks. Among recent developments, Transformer-based models have demonstrated strong empirical performance through structural innovations:

PatchTST [8] divides sequences into patch tokens under channel independence, while iTransformer [9] inverts the attention dimension to model inter-variable dependencies. Although these high-capacity architectures effectively capture complex nonlinear dynamics, applying them uniformly to composite time series introduces a representational tension. Specifically, the model must avoid overfitting smooth, predictable trend patterns while retaining sufficient expressiveness for irregular high-frequency fluctuations.

A complementary line of lightweight architectures addresses this tension through global capacity constraints, often achieving competitive accuracy with substantially lower computational costs. Representative examples include DLinear’s decomposition-based linear mapping [10] and the MLP-Mixer designs of TSMixer [11] and TimeMixer [12]. However, these constraints are applied uniformly across the network. The same simplified architecture processes all components, resulting in systematic underfitting of residual dynamics that require greater representational power.

The natural resolution to this problem is decomposition-based modeling, which separates the series into sub-components prior to applying dedicated architectures. Models such as Autoformer [13], FEDformer [14], and xPatch [15] demonstrate the value of decomposition as a preprocessing step. Nevertheless, a persistent limitation remains: even after decomposition, most frameworks continue to apply architecturally homogeneous models to all components, such as identical MLP-Mixer blocks at every scale in MSD-Mixer [16] or generic parallel streams in xPatch [15].

Several recent works attempt component-wise specialization, such as N-HiTS [17], which uses multi-rate signal sampling to implicitly target different frequency bands; HP-Mixer [18], which decouples periodic and residual dynamics; IConv [19], which pairs a linear trend encoder with per-channel convolutions for residuals; and DeMa [20] or DecompSSM [21], which employ dual-path state-space models. These designs vary in their decomposition mechanisms and encoder architectures. However, they often lack an explicit link between architectural capacity choices and the statistical properties of each decomposed component.

In this work, neither architectural extreme is applied indiscriminately: a high-capacity Transformer is not assigned to a smooth, globally structured trend, and a lightweight model is not forced to approximate an irregular, nonlinear residual. To resolve this mismatch, we propose MLPFormer, a hybrid dual-path forecasting framework built on the principle that encoder architectures should reflect the statistical character of their target signals. The trend component, characterized by smoothness and global structure, is processed by a two-layer channel-wise MLP that acts as an implicit regularizer for low-frequency patterns. Conversely, the residual component, which is irregular and nonlinearly structured, is processed by a dedicated Transformer encoder augmented with a depthwise separable convolution module for local feature-group enhancement. The outputs from both paths are fused and decoded through parallel linear and MLP prediction heads. As illustrated in Figure 1, MLPFormer achieves competitive or state-of-the-art performance across real-world benchmarks. The main contributions of this work are summarized as follows:

- We characterize a post-decomposition architectural homogeneity problem and analyze it explicitly: applying a uniform encoder to statistically asymmetric components simultaneously over-parameterizes smooth trends and underfits complex residuals. Unlike prior component-specific frameworks that motivate specialization by scale or frequency band, we make the link between encoder *capacity* and the measured statistical complexity of each component explicit, and validate it with a component-level complexity analysis (Appendix H).
- We propose MLPFormer, a hybrid MLP–Transformer architecture in which encoder capacity is explicitly matched to decomposed signal characteristics: a two-layer channel-

wise MLP for trend regularization and a feature-enhanced Transformer for residual modeling.

- Extensive experiments on nine public benchmark datasets, with all MLPFormer results averaged over three random seeds, demonstrate average improvements of 11.8% in MSE and 8.1% in MAE over the mean of seven competitive baselines spanning Transformer-based, linear/MLP-based, and RNN-based architectures published between 2023 and 2025. Comprehensive ablation studies—including decoder-head and fusion-strategy ablations and capacity-matched controls—confirm that these gains are attributable to component-specific encoder heterogeneity rather than a mere increase in overall model capacity.

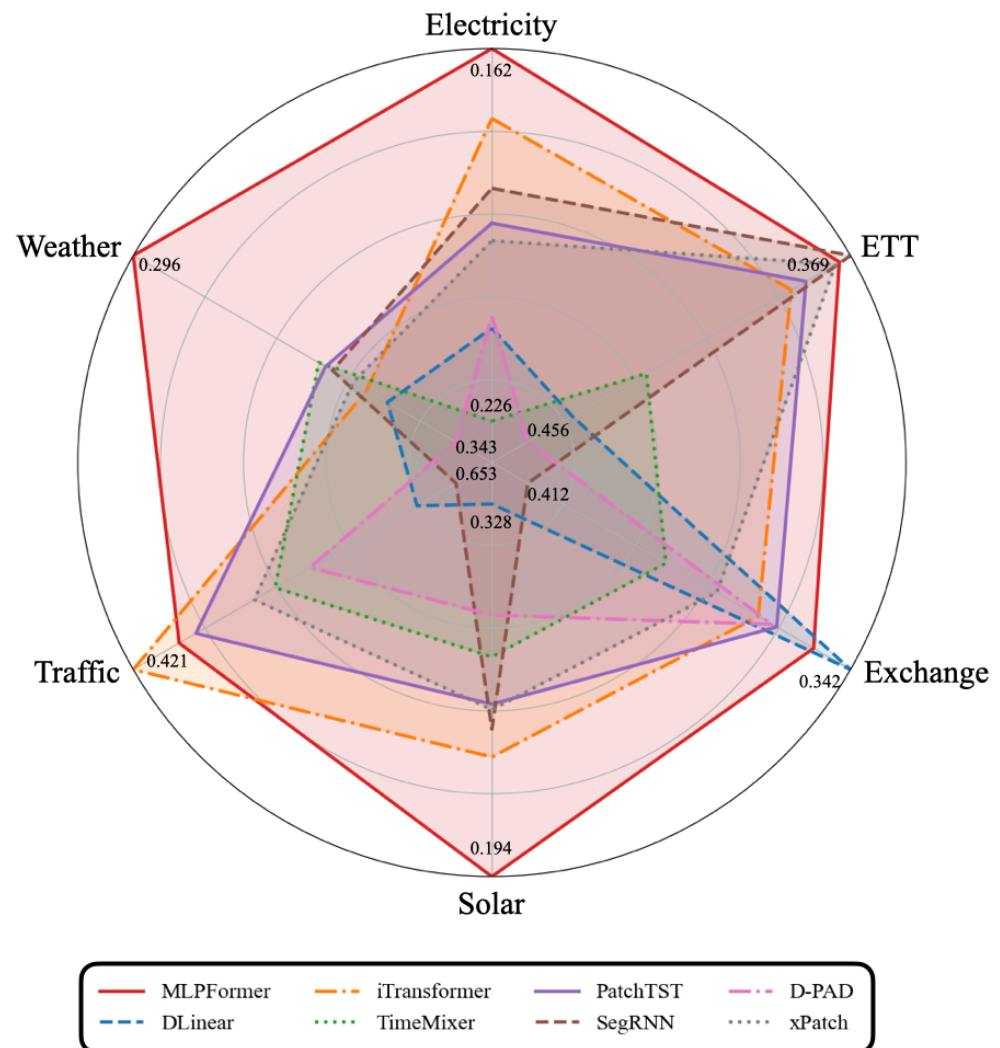


Figure 1. Performance comparison across six benchmark datasets. For each radial axis, a point further from the center indicates lower error (better performance); a larger enclosed area corresponds to superior overall performance.

2. Related Work

2.1. Transformer Architectures for Time Series Forecasting

The Transformer [22] architecture has emerged as a dominant backbone for sequence modeling, with its adaptation to time series forecasting evolving along several trajectories. Early efforts, such as Informer [23] and Autoformer [13], primarily focused on mitigating the quadratic attention complexity associated with long sequences. Subsequent research explored richer representational structures: PatchTST [8] segments sequences into patch

tokens to capture local temporal semantics, iTransformer [9] applies attention across the variable dimension to model inter-series dependencies, and Crossformer [24] introduces a two-stage mechanism for joint temporal and variable modeling. Furthermore, FEDformer [14] integrates frequency-domain analysis via random Fourier feature selection, while D-PAD [25] disentangles multi-frequency patterns through parallel encoding at varying capacities. Despite their strong representational capabilities, high-capacity monolithic Transformers share a fundamental limitation: when applied uniformly to composite time series containing both structurally simple trends and complex residuals, they risk overfitting the former while expending redundant computation on predictable components.

2.2. Lightweight MLP, Linear, and Basis-Expansion Models

Recent studies, notably DLinear [10], have demonstrated that simple decomposition-based linear models can outperform complex Transformers on multiple benchmarks, thereby sparking renewed interest in lightweight architectures. Building on this paradigm, TSMixer [11] proposes an all-MLP structure for feature mixing within and across time steps, whereas TimeMixer [12] employs decomposable multi-scale mixing to exploit periodic structures. In parallel, N-BEATS [26] and N-HiTS [17] utilize learnable basis expansions and multi-rate sampling to implicitly model trend and seasonality, demonstrating robust performance in univariate forecasting. Additionally, TimesNet [27] reshapes 1D series into 2D tensors to facilitate frequency-aware 2D convolutions. These lightweight models achieve a favorable trade-off between efficiency and accuracy, proving particularly adept at capturing global trends and periodic patterns. However, their structural simplicity inherently constrains their expressive power when modeling the stochastic, fine-grained dynamics typical of high-frequency residual components.

2.3. Decomposition-Based and Component-Specific Frameworks

Time series decomposition is a classical technique that has been successfully integrated into numerous deep learning frameworks [10,12–14]. Multi-path architectures have further extended this approach: for instance, MSD-Mixer [16] pairs multi-scale decomposition with an MLP-Mixer structure, and xPatch [15] introduces a dual-stream design based on exponential smoothing decomposition. More recently, several studies have explicitly explored component-wise specialization. IConv [19] pairs a linear trend encoder with per-channel convolutions for residuals, while HPMixer [18] decouples periodic and residual dynamics into specialized cycle-aware and frequency-domain modules. Furthermore, DeMa [20] employs dual-path Mamba models augmented by a spectral decomposition split, DecompSSM [21] facilitates end-to-end trend–seasonal–residual separation via auxiliary reconstruction and orthogonality losses, and AltTS [28] isolates autoregressive from cross-relational components, mitigating gradient entanglement through alternating optimization. Collectively, these contributions reinforce the efficacy of component-specific modeling.

MLPFormer distinguishes itself from this existing body of literature in three critical respects:

- (i) It employs a fixed moving-average decomposition; empirical validation suggests that its robustness and limited reconstruction artifacts make it well suited for subsequent specialized encoding.
- (ii) The assignment of encoders is grounded in the statistical structure of each component, using the MLP inductive bias for smooth trends and Transformer expressiveness for stochastic residuals, rather than relying strictly on scale or frequency-band heuristics.
- (iii) It enhances the cross-variable Transformer by incorporating a depthwise separable convolution for local feature-group aggregation. This design choice provides a complementary mechanism that explicitly differs from the temporal or spectral augmentations predominantly utilized in related works.

To make these distinctions concrete rather than rhetorical, Table 1 systematically contrasts MLPFormer with representative component-specific frameworks along five axes: the decomposition mechanism, the encoder assigned to the trend component, the encoder assigned to the residual component, the fusion strategy, and whether the design explicitly ties encoder capacity to component complexity. The comparison clarifies that, while architectural heterogeneity and component-specific modeling are individually well explored, the explicit pairing of a deliberately low-capacity trend encoder with a high-capacity residual encoder—justified by a measured complexity gap between the two components—is the specific niche occupied by MLPFormer.

Table 1. Systematic comparison of MLPFormer with representative decomposition-based and component-specific forecasting frameworks. “Capacity matched to complexity?” indicates whether the framework explicitly assigns encoder capacity according to the statistical complexity of each decomposed component, rather than by scale or frequency-band heuristics alone.

Method	Decomposition	Trend Encoder	Residual Encoder	Capacity Matched to Complexity?
DLinear [10]	Moving average	Linear	Linear	No
Autoformer [13]	In-block decomp.	Homogeneous attention blocks		No
MSD-Mixer [16]	Multi-scale	Homogeneous MLP-Mixer blocks		No
xPatch [15]	Exp. smoothing	Parallel homogeneous streams		No
IConv [19]	Linear/decomp.	Linear	Per-channel conv.	Partial (heuristic)
HPMixer [18]	Periodic/residual	Cycle-aware MLP	Freq.-domain MLP	Partial (by band)
DeMa [20]	Spectral split	Mamba	Mamba	No
DecompSSM [21]	Trend/seas./resid.	SSM	SSM	No
MLPFormer (Ours)	Moving average	Shallow channel-wise MLP (low capacity)	Feature-enhanced Transformer (high capacity)	Yes (explicit, measured)

3. MLPFormer

3.1. Preliminaries and Problem Formulation

Let a multivariate time series [29] be denoted as $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$, where B represents the batch size, L denotes the historical lookback window, and N indicates the number of observed variables. The objective of time series forecasting is to learn a parameterized mapping $f(\mathbf{X}; \theta)$ to predict future values $\mathbf{Y} \in \mathbb{R}^{B \times H \times N}$ based on the historical context $\mathbf{X}$, where H signifies the prediction horizon.

3.2. Model Overview

The MLPFormer architecture consists of four primary components: a signal decomposition module, a channel-wise MLP trend encoder, a feature-enhanced Transformer residual encoder, and a dual-head prediction module. The overall architecture is illustrated in Figure 2.

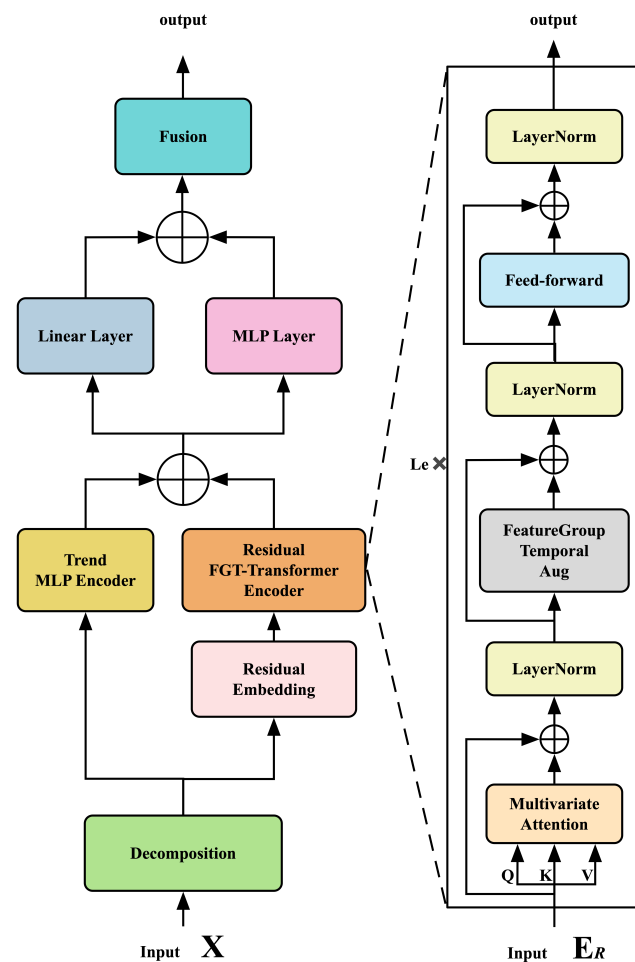


Figure 2. Overall architecture of MLPFormer. The input sequence is decomposed into trend and residual components via a moving-average filter. The trend is processed by a lightweight channel-wise MLP encoder; the residual is embedded and processed by a dedicated Transformer encoder augmented with a depthwise separable convolution module for local feature-group enhancement. Both path outputs are fused additively and decoded via parallel linear and MLP prediction heads.

3.2.1. Data Decomposition

Real-world time series inherently contain mixed-frequency components, typically comprising slowly varying structural trends and rapid, irregular fluctuations. Processing these heterogeneous components through a unified architecture forces the model to balance competing optimization objectives, often degrading both computational efficiency and predictive accuracy. To mitigate this, MLPFormer employs a signal decomposition module to separate the input sequence $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$ into distinct trend and residual components:

$$\mathbf{X} = \mathbf{T} + \mathbf{R}, \quad \mathbf{T}, \mathbf{R} \in \mathbb{R}^{B \times L \times N}, \quad (1)$$

where $\mathbf{T}$ is extracted by a symmetric moving-average filter with kernel size K , and $\mathbf{R} = \mathbf{X} - \mathbf{T}$. Symmetric zero-padding is applied at both sequence boundaries to preserve the original length L . The moving-average filter is computationally efficient, minimizes reconstruction artifacts, and ensures clear frequency separation: $\mathbf{T}$ retains the global low-frequency structure, while $\mathbf{R}$ isolates the high-frequency stochasticity. While the kernel size K is treated as a hyperparameter, sensitivity analysis (Appendix F) confirms that the model's performance remains robust across $K \in \{13, 25, 37, 49\}$.

3.2.2. Trend Path: Channel-Wise MLP Encoder

Given its smooth and globally coherent nature, the trend component $\mathbf{T}$ is highly susceptible to overfitting if processed by a high-capacity model, which may erroneously fit residual noise rather than the underlying structural trajectory. Consequently, a lightweight architecture is more appropriate. To process each variable independently, $\mathbf{T}$ is transposed to $\mathbf{T}_\pi \in \mathbb{R}^{B \times N \times L}$, and the trend representation $\mathbf{T}_e \in \mathbb{R}^{B \times N \times d_{\text{model}}}$ is obtained via a two-layer MLP:

$$\mathbf{T}_e = \mathbf{W}_2^{(T)} \text{GELU}(\mathbf{W}_1^{(T)} \mathbf{T}_\pi + \mathbf{b}_1^{(T)}) + \mathbf{b}_2^{(T)}, \quad (2)$$

where $\mathbf{W}_1^{(T)} \in \mathbb{R}^{d_{\text{model}} \times L}$, $\mathbf{W}_2^{(T)} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, and $\mathbf{b}_1^{(T)}, \mathbf{b}_2^{(T)}$ are learnable parameters. This channel-wise operation encodes the trend of each variable independently, aligning with the empirical observation that global trend dynamics often exhibit weak cross-variable coupling. Furthermore, the shallow network design bounds the representational capacity to low-frequency patterns, acting as an implicit regularizer against high-frequency noise.

3.2.3. Residual Path: Feature-Enhanced Transformer Encoder

Conversely, the residual component $\mathbf{R}$ is characterized by irregular, nonlinear, and highly stochastic dynamics. Because shallow MLPs systematically underfit such complex patterns, MLPFormer employs a dedicated Transformer encoder for the residual path.

The residual signal is transposed to $\mathbf{R}_\pi \in \mathbb{R}^{B \times N \times L}$ and mapped into a continuous feature space via a learned linear projection:

$$\mathbf{E}_R = \mathbf{W}_e \mathbf{R}_\pi + \mathbf{b}_e, \quad \mathbf{E}_R \in \mathbb{R}^{B \times N \times d_{\text{model}}}, \quad (3)$$

where $\mathbf{W}_e \in \mathbb{R}^{L \times d_{\text{model}}}$ projects the L time-step values of each variable into a d_{model} -dimensional feature vector. Following the inverted-attention paradigm of iTransformer [9], each of the N variables is treated as an independent token of dimension d_{model} . Explicit positional encodings are omitted, as temporal information is implicitly captured by the learned projection matrix $\mathbf{W}_e$.

The initial embedding $\mathbf{E}_R$ is subsequently processed by L_e stacked *Feature Group (FG) Encoder Layers*. Within each FG layer, multi-head self-attention (MHA) operates across the N variable tokens to extract global cross-variable dependencies:

$$\mathbf{E}_{R1} = \text{LayerNorm}(\mathbf{E}_R + \text{MHA}(\mathbf{E}_R)), \quad \mathbf{E}_{R1} \in \mathbb{R}^{B \times N \times d_{\text{model}}}. \quad (4)$$

Although cross-variable attention excels at modeling global inter-series relationships, it lacks a mechanism to enforce local structure within the d_{model} feature space of individual variables. To address this limitation, each FG layer incorporates a *feature-group enhancement module*: a 1D depthwise separable convolution [30] applied independently along the feature dimension d_{model} for each variable. This operation promotes local co-activation patterns among adjacent feature dimensions, analogous to the channel-mixing role in hybrid conv-attention architectures [30]. Crucially, this convolution operates on the learned feature space rather than the original temporal axis; it serves as a complementary local aggregation mechanism rather than an enforcer of temporal locality:

$$\mathbf{A} = \text{Dropout}(\text{GELU}(\text{DWConv1D}(\mathbf{E}_{R1}))), \quad \mathbf{A} \in \mathbb{R}^{B \times N \times d_{\text{model}}}. \quad (5)$$

The adoption of depthwise separable convolutions [30] ensures that this enhancement introduces negligible computational overhead. Finally, the augmented representation

is integrated via a residual connection and layer normalization, followed by a standard position-wise feed-forward network (FFN):

$$\mathbf{E}_{R2} = \text{LayerNorm}(\mathbf{E}_{R1} + \mathbf{A}), \quad (6)$$

$$\mathbf{E}_R^{\text{out}} = \text{LayerNorm}(\mathbf{E}_{R2} + \text{FFN}(\mathbf{E}_{R2})), \quad \mathbf{E}_R^{\text{out}} \in \mathbb{R}^{B \times N \times d_{\text{model}}}. \quad (7)$$

3.2.4. Dual-Head Prediction

Following encoding, the distinct trend and residual representations are fused via element-wise addition to form a unified feature map:

$$\mathbf{F} = \mathbf{T}_e + \mathbf{E}_R^{\text{out}}, \quad \mathbf{F} \in \mathbb{R}^{B \times N \times d_{\text{model}}}. \quad (8)$$

To decode $\mathbf{F}$ into the final forecast, MLPFormer employs a parallel dual-head prediction strategy: a linear head applies a projection $\mathbf{W}_{\text{linear}} \in \mathbb{R}^{d_{\text{model}} \times H}$ per variable, and a three-layer nonlinear MLP head applies a nonlinear mapping with the same input–output shape. Both components operate independently along the variable dimension N :

$$\hat{\mathbf{Y}}^T = (\mathbf{F}\mathbf{W}_{\text{linear}} + \mathbf{b}_{\text{linear}}) + \text{MLP}(\mathbf{F}), \quad \hat{\mathbf{Y}}^T \in \mathbb{R}^{B \times N \times H}. \quad (9)$$

The resulting tensor is then transposed to yield the final prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times H \times N}$. This parallel configuration accommodates the heterogeneous information fused within $\mathbf{F}$: the linear head maps the smooth, globally coherent trend structures, while the MLP head facilitates the necessary nonlinear transformations for complex residual corrections. The individual contributions of the linear and MLP heads, together with alternative fusion strategies, are ablated in Section 4.3.1. The complete algorithmic procedure is provided in Algorithm A1 in Appendix D.

4. Experiments

The performance of MLPFormer is rigorously evaluated through comparative experiments, comprehensive ablation studies, and computational efficiency analyses across nine publicly available benchmark datasets.

4.1. Experimental Setup

4.1.1. Datasets and Metrics

Experiments are conducted on nine widely adopted benchmarks: ETT (comprising four sub-datasets: ETTh1, ETTh2, ETTm1, ETTm2), Weather, Traffic, Electricity (ECL), Exchange, and Solar [13,31]. Each dataset is chronologically partitioned into training (70%), validation (10%), and testing (20%) sets. Detailed dataset statistics are provided in Table A1 in Appendix A. All models are subjected to an identical preprocessing pipeline: each variable is normalized to zero mean and unit variance using statistics computed exclusively from the training split to prevent data leakage.

The historical lookback window is fixed at $L = 96$ across all models, adhering to the controlled evaluation protocol established by prior research on these benchmarks [8–10,12]. This standardized setting isolates the temporal feature extraction capability of the architectures and facilitates direct comparisons with reported literature; results under varied lookback lengths are further analyzed in Section 4.3.4. Forecasting accuracy is evaluated using mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE), with lower values denoting superior predictive performance. Unless otherwise specified, all reported metrics are averaged over four prediction horizons $S \in \{96, 192, 336, 720\}$.

4.1.2. Baseline Models

To establish a comprehensive baseline, MLPFormer is evaluated against seven state-of-the-art models spanning three distinct architectural paradigms and published between 2023 and 2025: Transformer-based (iTransformer [9], PatchTST [8], Crossformer [24], D-PAD [25], xPatch [15]), linear/MLP-based (DLinear [10], TimeMixer [12]), and RNN-based (SegRNN [32]). This pool covers the principal architectural families that have defined recent progress in long-term forecasting, including the most recent lightweight linear/MLP-based (TimeMixer, 2024) and patch-based Transformer (xPatch, 2025) designs. All models are trained under a uniform protocol, with hyperparameters systematically optimized via grid search on the validation sets. Comprehensive implementation details, hardware specifications, and training configurations are detailed in Appendices B and C.

4.2. Main Results

Table 2 summarizes the forecasting errors averaged over the four prediction horizons ($S \in \{96, 192, 336, 720\}$) with a fixed lookback window of $L = 96$. Complete per-horizon results are available in Table A4 in Appendix E.

Table 2. Forecasting results averaged over prediction horizons $S \in \{96, 192, 336, 720\}$ with fixed lookback $L = 96$. **Bold:** best result; underline: second best. ETT(Avg) denotes the mean over all four ETT sub-datasets. All values are 4-horizon averages; per-horizon breakdowns are provided in Table A4 in Appendix E.

Models	MLPFormer (Ours)			DLinear (2023)			iTransformer (2024)			TimeMixer (2024)			PatchTST (2023)			SegRNN (2024)			D-PAD (2024)			xPatch (2025)		
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
Electricity	0.162	0.257	0.401	0.210	0.296	0.457	<u>0.174</u>	<u>0.265</u>	<u>0.416</u>	0.226	0.328	0.475	0.192	0.281	0.438	0.186	0.279	0.430	0.208	0.292	0.456	0.195	0.282	0.441
ETT(Avg)	0.374	<u>0.392</u>	0.607	0.444	0.445	0.658	0.385	0.401	0.616	0.424	0.423	0.646	0.381	0.401	0.613	0.369	0.389	0.603	0.456	0.440	0.667	<u>0.373</u>	<u>0.392</u>	<u>0.606</u>
Exchange	0.367	0.407	0.556	0.342	0.413	0.545	0.362	<u>0.406</u>	0.554	0.382	0.414	0.566	<u>0.358</u>	0.403	<u>0.551</u>	0.412	0.424	0.581	0.359	0.403	<u>0.551</u>	0.371	0.410	0.560
Solar	0.201	0.244	0.449	0.328	0.399	0.572	<u>0.237</u>	<u>0.263</u>	<u>0.486</u>	0.273	0.308	0.522	0.256	0.294	0.505	0.247	0.294	0.496	0.288	0.307	0.535	0.254	0.283	0.503
Traffic	<u>0.455</u>	<u>0.285</u>	<u>0.674</u>	0.625	0.384	0.790	0.421	0.280	0.648	0.523	0.359	0.723	0.466	0.294	0.682	0.653	0.322	0.808	0.548	0.357	0.740	0.508	0.318	0.712
Weather	0.316	0.357	0.557	0.333	0.374	0.574	0.330	0.339	0.570	<u>0.323</u>	<u>0.338</u>	<u>0.564</u>	0.324	0.336	0.566	0.325	0.359	0.566	0.343	0.349	0.582	0.328	0.339	0.569
1st Count	8			2			3			0			2			3			1			0		

MLPFormer achieves the best or second-best predictive performance across five of the six dataset groups, securing 8 first-place metric finishes (all MLPFormer metrics are reported as the mean over three random seeds $\{2025, 2026, 2027\}$, with small standard deviations tabulated in Appendix I). The performance gains are particularly pronounced on datasets characterized by strong trend–volatility coupling. On the Solar dataset—which exhibits distinct daily and seasonal energy cycles superimposed with weather-driven stochastic fluctuations—MLPFormer (MSE = 0.201) outperforms the most competitive Transformer baseline, iTransformer (0.237), by 15.2%, and DLinear (0.328) by 38.7%. Similarly, on the Electricity dataset, where trend-dominated daily cycles coexist with unpredictable load spikes, MLPFormer (MSE = 0.162) consistently surpasses both iTransformer (0.174) and PatchTST (0.192).

Crucially, the empirical results also delineate the operational boundaries of the proposed dual-path design. On the ETT datasets, SegRNN (MSE = 0.369) marginally outperforms MLPFormer (0.374). This aligns with the well-documented advantage of recurrent architectures on low-dimensional, short-cycle sequence data where continuous long-range temporal memory is highly beneficial. On Exchange—a dataset resembling a random-walk process with minimal periodic structure—DLinear achieves a lower average MSE (0.342 vs. 0.367); this indicates that the specific trend–residual coupling targeted by MLPFormer is less critical for series dominated by stochastic drift. Furthermore, on Traffic, which features 862 highly correlated sensor variables, iTransformer (0.421) outperforms MLPFormer (0.455).

This suggests that the current additive path fusion mechanism may not exploit dense, high-dimensional cross-sensor dependencies as effectively as an unconstrained variable-dimension attention mechanism. Overall, these edge cases confirm that component-specific heterogeneity yields the most substantial benefits in environments with pronounced and separable trend–residual structures.

These boundary cases admit a consistent explanation rooted in the assumptions of the architecture. MLPFormer’s central premise is that the series decomposes into a smooth, learnable trend and a structured (rather than purely random) residual, and that these two parts benefit from encoders of different capacity. Two situations weaken this premise. First, for *random-walk-like* series such as Exchange, the increments are close to white noise: the moving-average “trend” is largely a lagged copy of the most recent value, and the residual carries little predictable structure beyond it. Under these conditions the theoretically optimal predictor is close to the last observation (a persistence/linear model), so the additional capacity of the Transformer residual path has little signal to exploit and can only add estimation variance—explaining why the deliberately simple DLinear is marginally better. Second, for datasets with *dense cross-variable dependencies* such as Traffic, most of the predictive information lives in the instantaneous correlations among the 862 sensors rather than in the within-series trend/residual split. MLPFormer routes only the residual path through cross-variable attention and processes the trend channel-independently, after which the two paths are combined by a fixed additive fusion; this design intentionally trades some cross-variable modeling capacity for trend/residual specialization. When cross-sensor coupling dominates, an architecture that applies attention across variables over the entire signal (iTransformer) can capture those dependencies more directly. The fusion-strategy ablation in Section 4.3.2 probes whether a more flexible (gated or concatenation-based) fusion recovers part of this gap on Traffic, and the component-complexity analysis in Appendix H quantifies, per dataset, how separable the trend and residual actually are, providing a statistical basis for predicting where the dual-path design is expected to help.

4.3. Ablation Studies

4.3.1. Component Analysis

To isolate and quantify the contribution of each architectural design choice, we construct three ablation variants: (1) *w/o Decomposition*—the raw sequence is fed directly into both encoders without prior signal separation; (2) *w/o Transformer Path*—the Transformer encoder is replaced by an MLP of equivalent channel width, resulting in a homogeneous dual-MLP architecture; (3) *w/o MLP Path*—the MLP trend encoder is replaced by a Transformer encoder, yielding a homogeneous dual-Transformer architecture. Consistent with Table 2 and Table 3, all reported ablation metrics reflect 4-horizon averages.

Table 3. Ablation results averaged over four prediction horizons ($S \in \{96, 192, 336, 720\}$) on three benchmark datasets. To isolate the effect of each architectural component under identical conditions, all variants in this table are trained with a single fixed random seed; the Full-model row may therefore differ marginally from the three-seed means reported in Table 2. **Bold** indicates the best result in each metric column.

Ablation Variant	Electricity			Weather			Solar		
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
MLPFormer (Full)	0.162	0.257	0.402	0.296	0.340	0.540	0.195	0.246	0.441
w/o Decomposition	0.171	0.264	0.412	0.320	0.359	0.561	0.200	0.255	0.447
w/o Transformer Path	0.181	0.270	0.425	0.307	0.345	0.550	0.241	0.287	0.491
w/o MLP Path	0.168	0.263	0.409	0.319	0.363	0.560	0.203	0.255	0.450

Removing the decomposition step (w/o Decomposition) consistently degrades performance across all datasets, verifying that prior signal separation is a fundamental prerequisite for effective component-specific modeling. The most severe performance drop occurs upon removing the Transformer path (w/o Transformer Path), evidenced by a 23.6% MSE surge on the Solar dataset. This establishes that high-capacity modeling is essential for capturing complex residual dynamics.

Critically, the dual-Transformer variant (w/o MLP Path) also underperforms the full MLPFormer, even though replacing the shallow trend MLP with a second Transformer *increases* the parameter count. This performance degradation demonstrates that deploying a powerful, high-capacity encoder on inherently smooth trends leads to overfitting—specifically, fitting the residual noise inadvertently retained within the trend component. This outcome provides concrete empirical evidence of the architectural mismatch that motivated this work: extra capacity applied to the wrong component actively hurts, rather than merely wasting parameters. To rigorously separate architecture from capacity, we further conduct a capacity-matched study in Appendix K, in which MLPFormer is shrunk below the iTransformer baseline and iTransformer is scaled well beyond MLPFormer’s size; the dual-path advantage persists at matched and lower parameter budgets and is not recovered by enlarging the homogeneous baseline. Consequently, the observed performance gains are attributable to strategic architectural heterogeneity rather than merely scaling up model capacity.

Table 4 further details the performance contributions of the individual paths when utilized in isolation at a single prediction horizon ($S = 96$).

Table 4. Contribution of individual paths versus the fused model at prediction horizon $S = 96$, with lookback $L = 96$. These are *single-horizon* results and are not directly comparable to the 4-horizon averages in Table 2. **Bold** indicates the best result in each metric column.

Dataset	Trend-Path Only		Residual-Path Only		MLPFormer (Fused)	
	MSE	MAE	MSE	MAE	MSE	MAE
Electricity (ECL)	0.160	0.252	0.136	0.233	0.134	0.230
Weather	0.230	0.285	0.426	0.440	0.204	0.266

The predictive dominance of each path is intrinsically tied to the statistical properties of the dataset. For the relatively smooth Weather dataset, the trend path alone ($\text{MSE} = 0.230$) substantially outperforms the residual path ($\text{MSE} = 0.426$), indicating that the global structural backbone carries the primary predictive signal. Conversely, for the highly volatile Electricity dataset, the residual path alone ($\text{MSE} = 0.136$) achieves performance comparable to the fused model (0.134), reflecting the overriding influence of high-frequency load dynamics. Ultimately, the fused representation yields the lowest error in both scenarios, validating that the two paths extract complementary, non-redundant predictive features.

Figure 3 provides a qualitative validation of this complementarity. The trend path robustly tracks the global trajectory while deliberately smoothing local volatility; simultaneously, the residual path accurately captures fine-grained, high-frequency oscillations while lacking a global anchor. Their fusion successfully reconstructs both the macroscopic coherence and microscopic accuracy of the original signal.

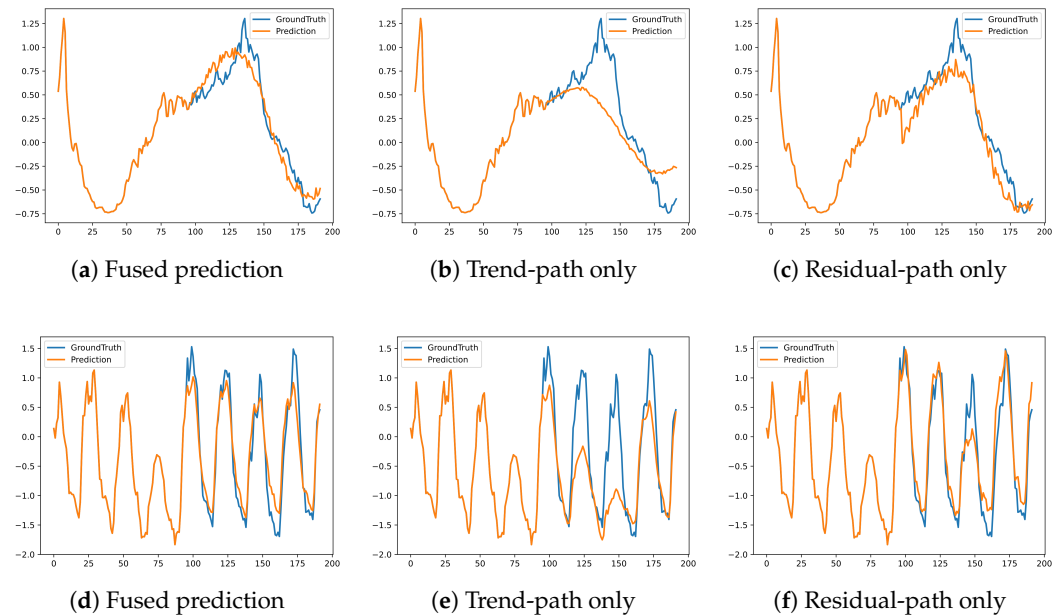


Figure 3. Dual-path predictions on Weather (top row) and Electricity (bottom row) at $S = 96$. The trend path captures the dominant low-frequency backbone; the residual path models localized high-frequency fluctuations; their fusion closely tracks the ground truth across both frequency ranges.

4.3.2. Decoder-Head and Fusion-Strategy Ablations

To isolate the contribution of the dual-head decoder and the path-fusion operator, we conduct two additional ablations. For the decoder, we compare the full dual head against linear-only and MLP-only variants. For the fusion operator, we compare the default additive fusion against a learned gated fusion ($\mathbf{F} = \mathbf{g} \odot \mathbf{E}_R^{\text{out}} + (1 - \mathbf{g}) \odot \mathbf{T}_e$, with $\mathbf{g}$ produced by a sigmoid projection of the concatenated features) and a concatenation-then-projection fusion. All variants are trained under the identical protocol and averaged over the four horizons. Results are reported in Table 5.

Table 5. Decoder-head and fusion-strategy ablations, averaged over the four prediction horizons ($S \in \{96, 192, 336, 720\}$) at $L = 96$. The default configuration is a dual (linear + MLP) head with additive fusion. All variants use a single fixed seed for a controlled comparison, so the Full-model row may differ marginally from the three-seed means in Table 2. **Bold** indicates the best result in each metric column.

Variant	Electricity			Weather			Solar			Traffic		
	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
Dual head + additive (Full)	0.161	0.256	0.401	0.312	0.363	0.553	0.202	0.240	0.450	0.452	0.285	0.672
Linear head only	0.163	0.259	0.403	0.292	0.341	0.536	0.201	0.246	0.448	0.459	0.287	0.677
MLP head only	0.161	0.257	0.401	0.317	0.361	0.558	0.198	0.247	0.445	0.495	0.306	0.703
Gated fusion	0.160	0.255	0.399	0.310	0.353	0.552	0.195	0.245	0.441	0.458	0.287	0.677
Concat-projection fusion	0.160	0.255	0.399	0.311	0.353	0.551	0.193	0.242	0.439	0.442	0.280	0.665

Table 5 yields three observations. First, the dual head is a robust default: it matches or improves upon both single-head variants on Electricity, Solar, and Traffic, and the isolated heads are only preferable in narrow cases (the linear head is best on the smooth, near-linear Weather series, while the MLP head is marginally stronger on Solar). Because no single head dominates all datasets whereas the dual head is never far from the best, combining the two provides the most reliable behaviour without dataset-specific tuning. Second, the fusion operator has a small but consistent effect on the trend-dominated benchmarks: gated and concatenation-projection fusion recover a further 0.5–4.5% MSE on

Electricity and Solar over additive fusion, confirming that a lightweight learned combiner can help when the two paths carry complementary signals. Third, and most relevant to the cross-variable concern (R3#2), the more expressive fusions do narrow the Traffic gap—concatenation-projection fusion lowers Traffic MSE from 0.452 to 0.442—but they do not close it relative to iTransformer (0.421). This indicates that the residual gap on Traffic is not merely a fusion-capacity artifact; it stems from the architecture routing only the residual path through cross-variable attention, so datasets whose predictability is dominated by instantaneous cross-sensor coupling remain better served by full variable-dimension attention. We therefore retain additive fusion with the dual head as the default for its simplicity and stability, while noting learned fusion as a low-cost option when squeezing additional accuracy from strongly separable datasets.

4.3.3. Feature-Group Enhancement and Decomposition Strategy

To validate the customized FG encoder layer, we evaluate it against a standard Transformer encoder operating on the identical residual path. The inclusion of the depthwise separable convolution module consistently yields MSE reductions of 1.0–2.6% across multiple datasets, while concurrently reducing training time by approximately 14% (comprehensive results are documented in Table A2 in Appendix B). Notably, this performance enhancement transfers reliably when the module is integrated into a standalone iTransformer framework, proving that its utility is not strictly confined to the dual-path design. Operating along the d_{model} feature dimension rather than the temporal axis, the depthwise separable convolution facilitates local feature-group aggregation, effectively complementing global cross-variable attention. This consistent empirical gain, coupled with computational speedups derived from the lightweight convolution design, highlights the utility of this module as a generalized enhancement for variate-wise Transformer architectures.

Regarding the decomposition methodology, empirical comparisons demonstrate that the moving-average decomposition (evaluated across kernel sizes $K \in \{13, 25, 37, 49\}$) consistently matches or surpasses multi-level wavelet decomposition (1–4 levels) on both the ECL and Weather benchmarks. Although wavelet decomposition theoretically offers superior multi-resolution frequency separation, this structural advantage does not translate into downstream forecasting improvements. This discrepancy is largely attributable to boundary reconstruction artifacts inherent in wavelets, which compromise the statistical purity of the decomposed components. The sustained robustness of the moving-average approach across varying kernel sizes further indicates that MLPFormer’s predictive capability is resilient to precise decomposition boundary shifts. An extended comparative analysis between moving-average and wavelet techniques is available in Appendix F.

4.3.4. Sensitivity Analysis

Figure 4 illustrates MLPFormer’s stability as the historical input length expands from 96 to 720 steps. While several baseline models exhibit marked sensitivity and degradation as the lookback window increases, MLPFormer maintains a consistent error profile. This resilience stems from the complementary dynamics of the dual-path architecture: the channel-wise MLP extracts stable, globally coherent trend structures irrespective of window length, allowing the Transformer residual encoder to selectively attend to localized context without being overwhelmed by extended irrelevant histories. The sustained competitiveness of MLPFormer at extended sequence lengths also confirms that the standardized $L = 96$ evaluation protocol does not artificially inflate the model’s relative standing. Additional sensitivity analyses concerning encoder depth and hidden dimensions are provided in Appendix G.

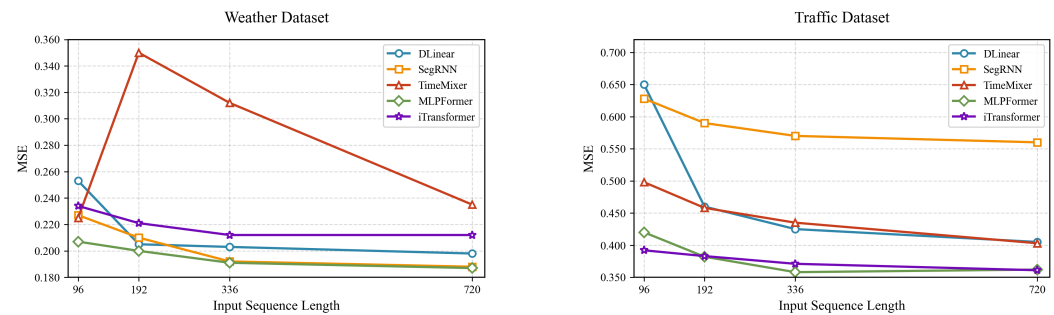


Figure 4. MSE as a function of input sequence length (96–720, with fixed prediction horizon $S = 96$) on Weather (left) and Traffic (right). MLPFormer maintains stable performance as the lookback window increases, indicating that the fixed $L = 96$ evaluation protocol does not systematically disadvantage the proposed model relative to baselines.

4.4. Computational Efficiency

Table 6 details the computational profile of the models. Throughout this work, our computational-efficiency claims refer specifically to wall-clock *training* time per epoch (in seconds), measured on identical hardware and batch configurations; theoretical MACs and parameter counts are reported alongside for completeness but are not the basis of the efficiency claim. While MLPFormer incurs a higher theoretical compute cost (MACs) than iTransformer (e.g., 29.6 vs. 21.4 GMac on Weather), the actual wall-clock training times remain comparable (22 vs. 24 s/epoch). This phenomenon becomes more pronounced on the high-dimensional Traffic dataset ($N = 862$, Batch Size = 8), where MLPFormer requires significantly more MACs (304.6 vs. 89.3 GMac) but exhibits nearly equivalent epoch times (62 vs. 58 s). This discrepancy highlights a critical characteristic of modern GPU execution: when processing small batch sizes with large variable counts, model operations are predominantly memory-bandwidth-bound rather than compute-bound. In these regimes, theoretical MACs become a poor proxy for wall-clock latency, as execution speed is constrained by memory access rates rather than floating-point throughput. The dense operations within the MLP path, despite their theoretical computational density, execute highly efficiently as batched matrix multiplications, thereby maximizing GPU utilization without substantially inflating runtime.

Table 6. Computational efficiency at input length 96, prediction horizon 192. Time: wall-clock training time in seconds per epoch; MACs: theoretical multiply–accumulate operations in GMac; Params: millions of parameters; Mem: peak GPU memory in MB. “MLPFormer (trend path only)” removes the Transformer residual path, leaving the shallow MLP and linear head; its parameter count (~ 6.5 M) coincides with the iTransformer baseline, serving as a capacity-controlled reference. (This is distinct from the homogeneous dual-Transformer studied in Appendix K, which replaces the MLP with a second Transformer and is therefore larger.)

Model	Arch. Params				Weather ($N = 21$, BS = 128)				Traffic ($N = 862$, BS = 8)			
	d	d_{ff}	h	Layers	Time	MACs	Params	Mem	Time	MACs	Params	Mem
MLPFormer (Ours)	512	2048	8	2	22	29.6	11.05	585	62	304.6	11.06	7.6k
MLPFormer (trend path only)	512	2048	8	2	20	17.3	6.46	387.4	56	89.0	6.46	3.2k
iTransformer	512	2048	8	2	24	21.4	6.45	891.9	58	89.3	6.45	3.2k
TimeMixer	512	2048	–	2	64	0.11	0.14	43.6	36	1.9	0.21	506.5
SegRNN	512	–	–	2	20	25.8	1.63	696.8	39	265.4	1.85	6.5k

Finally, the w/o MLP ablation variant reinforces the core thesis of this work. By removing the MLP path, this variant halves the theoretical MACs and aligns the parameter count (~ 6.5 M) with the iTransformer baseline. However, despite matched capacity and computational overhead, this structurally homogeneous variant underperforms the full,

heterogeneous MLPFormer on trend-dominated datasets. This firmly confirms that the performance gains of MLPFormer are fundamentally driven by its component-specific architectural design, rather than a mere escalation of model parameters.

5. Conclusions

This paper addresses the enduring challenge of jointly modeling heterogeneous components in time series, specifically the coexistence of stable trends and high-frequency fluctuations, by introducing MLPFormer, a component-specific hybrid forecasting architecture. Departing from homogeneous paradigms, MLPFormer employs a dual-path framework: a lightweight MLP extracts structural trends while regularizing high-frequency noise, and a feature-enhanced Transformer models complex residual dynamics. Extensive empirical evaluations across nine benchmark datasets, with all reported MLPFormer metrics averaged over three random seeds, demonstrate average reductions of 11.8% in MSE and 8.1% in MAE relative to the mean of seven competitive baselines. Comprehensive ablation studies confirm that these predictive gains stem from architectural heterogeneity matched to signal characteristics, rather than from indiscriminate increases in overall model capacity.

While MLPFormer exhibits robust forecasting capabilities, its current instantiation highlights several promising avenues for future research. First, although the moving-average decomposition is empirically robust and computationally efficient, integrating more sophisticated adaptive signal separation techniques could further enhance component purity prior to encoding. Second, our ablation of fusion operators (Section 4.3.2) shows that learned gated and concatenation-based fusion can complement the default additive scheme, particularly on datasets with dense cross-variable structure such as Traffic; a fully input-adaptive, attention-weighted fusion that dynamically rebalances the two paths per series remains a natural extension. Finally, extending this component-specific hybrid design philosophy to adjacent domains, such as time series anomaly detection and classification, represents a useful trajectory toward more interpretable and structurally efficient sequence modeling frameworks.

Author Contributions: Conceptualization, J.C. and X.C.; methodology, J.C., Y.W. and X.C.; software, J.C. and Y.W.; validation, J.C., Y.W. and M.L.; formal analysis, J.C. and Y.W.; investigation, J.C. and Y.W.; resources, X.C. and M.L.; data curation, J.C. and Y.W.; writing—original draft preparation, J.C. and Y.W.; writing—review and editing, X.C. and M.L.; visualization, J.C. and Y.W.; supervision, X.C. and M.L.; project administration, X.C.; funding acquisition, X.C. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Nos. 62172352, 62171143 and 42306218), the Natural Science Foundation of Guangdong Province (No. 2026A1515011448), the Key Research Project of Guangdong Province's Colleges and Universities in the Key Field of New Generation Electronic Information (Semiconductor) (No. 2025ZDZX1007), the Guangdong Provincial Department of Education Ocean Ranch Equipment Information and Intelligent Innovation Team Project (No. 2023KCXTD016), the Guangdong Ocean University Research Fund Project (No. 060302102304), and the Natural Science Foundation of Hebei Province (No. F2023407003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The benchmark datasets analyzed in this study are publicly available datasets commonly used in long-term time series forecasting. Dataset descriptions, hyperparameter configurations, pseudocode, and complete per-horizon results are provided in the Appendices. The source code used during the current study is publicly available at <https://github.com/Henryyihao/MLPFormer> (accessed on 7 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
ECL	Electricity Consuming Load
ETT	Electricity Transformer Temperature
FG	Feature Group
FFN	Feed-forward network
MAE	Mean absolute error
MLP	Multilayer perceptron
MHA	Multi-head self-attention
MSE	Mean squared error
RMSE	Root mean squared error

Appendix A. Dataset Details

Table A1 summarizes the nine benchmark datasets utilized in our experiments. The lookback window is uniformly fixed at $L = 96$ across all models and experimental configurations. The prediction horizons are defined as $S \in \{96, 192, 336, 720\}$. In accordance with standard protocols established in prior research, all datasets are chronologically partitioned into training (70%), validation (10%), and testing (20%) sets.

Table A1. Benchmark dataset descriptions. “Dim (N)” is the number of variables. “Dataset Size” lists the number of time steps in each split. “Frequency” is the sampling interval.

Dataset	Dim (N)	Prediction Lengths (S)	Lookback (L)	Frequency	Dataset Size (Train/Val/Test)	Information Source
ETTh1	7	{96, 192, 336, 720}	96	Hourly	(8545, 2881, 2881)	Electricity Transformer Temp.
ETTh2	7	{96, 192, 336, 720}	96	Hourly	(8545, 2881, 2881)	Electricity Transformer Temp.
ETTm1	7	{96, 192, 336, 720}	96	15 min	(34,465, 11,521, 11,521)	Electricity Transformer Temp.
ETTm2	7	{96, 192, 336, 720}	96	15 min	(34,465, 11,521, 11,521)	Electricity Transformer Temp.
Weather	21	{96, 192, 336, 720}	96	10 min	(36,792, 5271, 10,540)	Meteorological Data
Traffic	862	{96, 192, 336, 720}	96	Hourly	(12,185, 1757, 3509)	Transportation
Electricity (ECL)	321	{96, 192, 336, 720}	96	Hourly	(18,317, 2633, 5261)	Electricity
Exchange	8	{96, 192, 336, 720}	96	Daily	(5120, 665, 1422)	Exchange Rates
Solar-Energy	137	{96, 192, 336, 720}	96	10 min	(36,601, 5161, 10,417)	Energy

Note: ETTh1, ETTh2, ETTm1, and ETTm2 results are averaged and reported as “ETT(Avg)” in the main text.

Appendix B. Implementation Details

All models were implemented using PyTorch 2.8 and trained on a single NVIDIA RTX 4090 GPU (24 GB VRAM). Network parameters were optimized using the Adam optimizer, with gradient clipping applied at a maximum norm of 1.0. An early stopping mechanism with a patience of 30 epochs, monitored via validation MSE, was employed to prevent overfitting. The maximum number of training epochs was capped at 300. To ensure consistent scale, input features were normalized to zero mean and unit variance per variable, utilizing statistics derived exclusively from the training split. For MLPFormer, the moving-average kernel size was uniformly set to 25 across all datasets; a detailed sensitivity analysis of this parameter is provided in Section F.

Baseline fairness and tuning protocol.

To ensure a fair comparison, every baseline was run within the same unified codebase, data pipeline, and train/validation/test splits as MLPFormer, rather than copied from heterogeneous source papers. All models share the identical lookback window ($L = 96$),

the four prediction horizons, the same normalization, optimizer, gradient clipping, early-stopping criterion, and evaluation metrics. For each baseline we performed a grid search over its principal hyperparameters (model dimension, number of encoder layers, learning rate, and batch size) on the validation split, and selected the configuration with the lowest validation MSE; where the original publication recommended dataset-specific settings, those served as the center of the search grid. The radar-style summary in Figure 1 and the qualitative panels are produced from exactly these retuned baselines under this shared protocol, so no model is advantaged by inconsistent preprocessing or evaluation. To further quantify run-to-run variability under this protocol, we report the mean and standard deviation of MLPFormer and its strongest competitors over multiple random seeds in Appendix I.

Table A2 details the per-dataset training time and forecasting performance of the customized Feature Group (FG) Encoder Layer compared to a standard Transformer encoder, illustrating the consistent accuracy and computational efficiency gains afforded by the 1D depthwise convolution augmentation.

Table A2. Effect of the customized FGT encoder layer versus a standard Transformer encoder, averaged over four prediction horizons. Each row pair is a single-seed controlled comparison, so absolute values may differ marginally from the three-seed means in Table 2; the relative improvement within each pair is the quantity of interest. Time: seconds per training epoch. **Bold** indicates the better value within each comparison group (lower is better).

Dataset	Encoder Variant	MLPFormer (w/o Customized)			iTransformer		
		MSE	MAE	Time (s/ep)	MSE	MAE	Time (s/ep)
Electricity	Standard encoder	0.164	0.258	43	0.174	0.265	39
	FGT encoder (customized)	0.162	0.257	36	0.166	0.260	31
	Relative improvement	1.0%	0.3%	16.3%	4.5%	1.9%	20.5%
Weather	Standard encoder	0.304	0.342	21	0.330	0.339	24
	FGT encoder (customized)	0.296	0.340	19	0.321	0.334	21
	Relative improvement	2.6%	0.6%	9.5%	2.6%	1.4%	12.5%
Solar	Standard encoder	0.196	0.247	44	0.237	0.263	45
	FGT encoder (customized)	0.194	0.249	40	0.232	0.262	38
	Relative improvement	1.0%	−0.8%	9.1%	2.0%	0.2%	15.6%

Appendix C. Hyperparameter Settings

Table A3 details the specific hyperparameter configurations utilized for MLPFormer across all reported experiments. Initial settings were adapted from the well-validated configurations of the iTransformer baseline. Subsequently, the batch size for each dataset was empirically adjusted to optimize validation performance while accommodating GPU memory constraints.

Table A3. Hyperparameter settings for MLPFormer across datasets.

Hyperparameter	Electricity	ETT	Exchange	Solar	Traffic	Weather
Model Dimension (d_{model})	512	128	128	512	512	512
Encoder Layers (L_e)	2	2	2	2	2	2
Attention Heads	8	8	8	8	8	8
Feed-Forward Dim (d_{ff})	2048	128	128	2048	2048	2048
Moving Avg Kernel (K)	25	25	25	25	25	25
Dropout	0.1	0.1	0.1	0.1	0.1	0.1
Learning Rate	5×10^{-4}	10^{-4}	10^{-4}	5×10^{-4}	10^{-4}	10^{-3}
Batch Size	16	32	16	32	8	128
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Activation	GELU	GELU	GELU	GELU	GELU	GELU

Appendix D. Pseudocode of MLPFormer

Algorithm A1 presents a PyTorch-style pseudocode that outlines the complete forward pass of the MLPFormer model, corresponding to the architectural design detailed in Section 3.

Algorithm A1 MLPFormer forward pass

Require: Input $\mathbf{X} \in \mathbb{R}^{B \times L \times N}$; encoder layers L_e ; prediction horizon H .
Ensure: Prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times H \times N}$.

- 1: $\mathbf{T}, \mathbf{R} \leftarrow \text{MovingAvgDecomp}(\mathbf{X})$ ▷ Equation (1) in main text
- 2: **// Trend path (lightweight MLP)**
- 3: $\mathbf{T}_e \leftarrow \text{TrendMLP}(\mathbf{T}^\top)$ ▷ $\mathbf{T}^\top \in \mathbb{R}^{B \times N \times L}$, Equation (2)
- 4: **// Residual path (FG Transformer)**
- 5: $\mathbf{H} \leftarrow \text{LinearProjection}(\mathbf{R}^\top)$ ▷ Initialize with $\mathbf{E}_R$, Equation (3)
- 6: **for** $l = 1$ **to** L_e **do**
- 7: $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{H} + \text{MHA}(\mathbf{H}))$ ▷ Equation (4)
- 8: $\mathbf{A} \leftarrow \text{Dropout}(\text{GELU}(\text{DWConv1D}(\mathbf{H})))$ ▷ Equation (5)
- 9: $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{H} + \mathbf{A})$ ▷ Equation (6)
- 10: $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{H} + \text{FFN}(\mathbf{H}))$ ▷ Equation (7)
- 11: **end for**
- 12: $\mathbf{E}_R^{\text{out}} \leftarrow \mathbf{H}$ ▷ Final residual representation
- 13: **// Fusion and dual-head decoding**
- 14: $\mathbf{F} \leftarrow \mathbf{T}_e + \mathbf{E}_R^{\text{out}}$ ▷ Equation (8)
- 15: $\hat{\mathbf{Y}}^\top \leftarrow \text{LinearHead}(\mathbf{F}) + \text{MLPHead}(\mathbf{F})$ ▷ Equation (9)
- 16: **return** $\hat{\mathbf{Y}}$ ▷ Transpose to $\mathbb{R}^{B \times H \times N}$

Appendix E. Full Forecasting Results

Table A4 provides the comprehensive forecasting results for all evaluated models across the four designated prediction horizons on each benchmark dataset. For clarity, the reported ETT results represent the arithmetic mean of the individual performances on its four constituent sub-datasets (ETTth1, ETTth2, ETTm1, and ETTm2) at each corresponding horizon.

Table A4. Full long-term forecasting results. Input length $L = 96$ for all models. “Avg” denotes the mean over all four prediction horizons. **Bold:** best; underline: second best.

Models	MLPFormer (Ours)			DLinear (2023)			iTransformer (2024)			TimeMixer (2024)			PatchTST (2023)			SegRNN (2024)			D-PAD (2024)			xPatch (2025)			
	Metric	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE	MSE	MAE	RMSE
Electricity	96	0.135	0.231	0.367	0.195	0.278	0.441	0.147	0.238	0.384	0.186	0.293	0.431	0.168	0.261	0.410	0.158	0.252	0.398	0.184	0.270	0.429	0.177	0.266	0.421
	192	0.152	0.245	0.389	0.194	0.281	0.440	0.163	0.254	0.404	0.226	0.334	0.476	0.176	0.268	0.420	0.171	0.264	0.414	0.192	0.278	0.439	0.180	0.268	0.424
	336	0.166	0.262	0.407	0.207	0.296	0.455	0.176	0.268	0.419	0.228	0.332	0.478	0.191	0.280	0.437	0.188	0.282	0.434	0.208	0.294	0.456	0.193	0.281	0.440
	720	0.194	0.289	0.440	0.242	0.330	0.492	0.210	0.299	0.458	0.266	0.355	0.515	0.234	0.316	0.484	0.225	0.317	0.474	0.249	0.326	0.499	0.231	0.313	0.481
	Avg	0.162	0.257	0.401	0.210	0.296	0.457	0.174	0.265	0.416	0.226	0.328	0.475	0.192	0.281	0.438	0.186	0.279	0.430	0.208	0.292	0.456	0.195	0.282	0.441
ETT (Avg)	96	0.293	0.342	0.536	0.320	0.368	0.561	0.305	0.351	0.547	0.314	0.355	0.555	0.300	0.347	0.542	0.290	0.338	0.533	0.385	0.395	0.607	0.293	0.342	0.536
	192	0.355	0.378	0.592	0.394	0.414	0.624	0.364	0.386	0.601	0.409	0.413	0.639	0.358	0.384	0.595	0.348	0.373	0.586	0.442	0.428	0.656	0.355	0.377	0.592
	336	0.399	0.408	0.630	0.472	0.464	0.685	0.410	0.415	0.639	0.444	0.436	0.666	0.405	0.416	0.634	0.396	0.404	0.627	0.478	0.453	0.685	0.399	0.408	0.629
	720	0.449	0.442	0.669	0.589	0.533	0.762	0.459	0.452	0.677	0.530	0.486	0.724	0.462	0.456	0.679	0.445	0.440	0.666	0.520	0.483	0.718	0.443	0.439	0.665
	Avg	0.374	0.392	0.607	0.444	0.445	0.658	0.385	0.401	0.616	0.424	0.423	0.646	0.381	0.401	0.613	0.369	0.389	0.603	0.456	0.440	0.667	0.373	0.392	0.606
Exchange	96	0.083	0.201	0.288	0.093	0.224	0.304	0.086	0.206	0.294	0.085	0.205	0.292	0.086	0.206	0.294	0.084	0.201	0.291	0.086	0.206	0.293	0.087	0.205	0.294
	192	0.179	0.301	0.423	0.187	0.327	0.432	0.180	0.303	0.425	0.183	0.305	0.428	0.175	0.298	0.419	0.184	0.302	0.429	0.177	0.300	0.421	0.183	0.304	0.428
	336	0.338	0.421	0.582	0.322	0.433	0.568	0.337	0.421	0.580	0.353	0.429	0.594	0.330	0.417	0.575	0.342	0.424	0.585	0.327	0.415	0.572	0.339	0.422	0.583
	720	0.869	0.704	0.932	0.767	0.670	0.876	0.844	0.694	0.919	0.905	0.717	0.951	0.841	0.691	0.917	1.035	0.769	1.018	0.844	0.693	0.919	0.875	0.707	0.936
	Avg	0.367	0.407	0.556	0.342	0.413	0.545	0.362	0.406	0.554	0.382	0.414	0.566	0.358	0.403	0.551	0.412	0.424	0.581	0.359	0.403	0.551	0.371	0.410	0.560
Solar	96	0.185	0.227	0.430	0.288	0.376	0.537	0.206	0.237	0.454	0.238	0.290	0.488	0.228	0.284	0.477	0.220	0.277	0.476	0.242	0.280	0.492	0.224	0.267	0.473
	192	0.204	0.242	0.452	0.318	0.396	0.564	0.240	0.263	0.490	0.275	0.313	0.525	0.255	0.297	0.505	0.243	0.293	0.493	0.280	0.303	0.529	0.252	0.284	0.502
	336	0.210	0.254	0.458	0.351	0.414	0.592	0.251	0.274	0.501	0.286	0.314	0.535	0.272	0.306	0.522	0.259	0.300	0.509	0.313	0.321	0.560	0.270	0.293	0.520
	720	0.206	0.253	0.454	0.355	0.412	0.596	0.251	0.276	0.501	0.295	0.316	0.543	0.275	0.303	0.525	0.256	0.295	0.506	0.315	0.323	0.561	0.267	0.289	0.517
	Avg	0.201	0.244	0.449	0.328	0.399	0.572	0.237	0.263	0.486	0.273	0.308	0.522	0.256	0.294	0.505	0.247	0.294	0.496	0.288	0.307	0.535	0.254	0.283	0.503
Traffic	96	0.428	0.275	0.654	0.649	0.396	0.805	0.392	0.266	0.626	0.498	0.344	0.705	0.444	0.284	0.667	0.627	0.311	0.792	0.541	0.355	0.735	0.492	0.319	0.702
	192	0.438	0.277	0.662	0.598	0.371	0.774	0.414	0.276	0.643	0.510	0.352	0.714	0.453	0.288	0.673	0.636	0.313	0.797	0.531	0.349	0.728	0.492	0.310	0.702
	336	0.460	0.285	0.679	0.605	0.374	0.778	0.419	0.279	0.647	0.528	0.361	0.727	0.469	0.294	0.684	0.657	0.322	0.810	0.542	0.352	0.736	0.504	0.311	0.710
	720	0.493	0.305	0.702	0.646	0.395	0.804	0.458	0.298	0.677	0.557	0.379	0.746	0.497	0.309	0.705	0.691	0.340	0.831	0.580	0.372	0.759	0.544	0.333	0.735
	Avg	0.455	0.285	0.674	0.625	0.384	0.790	0.421	0.280	0.648	0.523	0.359	0.723	0.466	0.294	0.682	0.653	0.322	0.808	0.548	0.357	0.740	0.508	0.318	0.712
Weather	96	0.205	0.263	0.453	0.254	0.308	0.504	0.234	0.271	0.484	0.227	0.270	0.476	0.228	0.265	0.478	0.226	0.281	0.476	0.254	0.287	0.504	0.239	0.275	0.489
	192	0.278	0.328	0.527	0.301	0.349	0.549	0.291	0.3132	0.540	0.283	0.314	0.532	0.288	0.3134	0.537	0.283	0.331	0.532	0.308	0.327	0.555	0.293	0.316	0.541
	336	0.348	0.391	0.590	0.353	0.394	0.594	0.357	0.357	0.594	0.349	0.358	0.590	0.348	0.355	0.590	0.347	0.379	0.589	0.363	0.365	0.603	0.349	0.356	0.591
	720	0.432	0.447	0.658	0.422	0.445	0.650	0.440	0.413	0.663	0.435	0.412	0.659	0.433	0.408	0.658	0.444	0.443	0.666	0.446	0.416	0.667	0.431	0.408	0.652
	Avg	0.316	0.357	0.557	0.333	0.374	0.574	0.330	0.339	0.570	0.323	0.338	0.564	0.324	0.336	0.566	0.325	0.359	0.566	0.343	0.349	0.582	0.328	0.339	0.569
1st Count		40			9			16			0			7			15			2			4		

Appendix F. Analysis of Decomposition Strategies

The choice of decomposition methodology is a pivotal architectural decision; it dictates the statistical properties of the resulting trend and residual components, thereby directly influencing the efficacy of the subsequent specialized modeling paths. To systematically evaluate the impact of various decomposition strategies and empirically justify the selection of the moving-average filter, we compare two representative approaches: moving average (functioning as a direct low-pass filter) and wavelet decomposition (offering multi-resolution frequency analysis). Comparative experiments were conducted on the ECL and Weather datasets. The moving-average approach was evaluated across kernel sizes of 13, 25, 37, and 49, while wavelet decomposition was tested across 1 to 4 decomposition levels. The corresponding forecasting performances are illustrated in Figure A1.

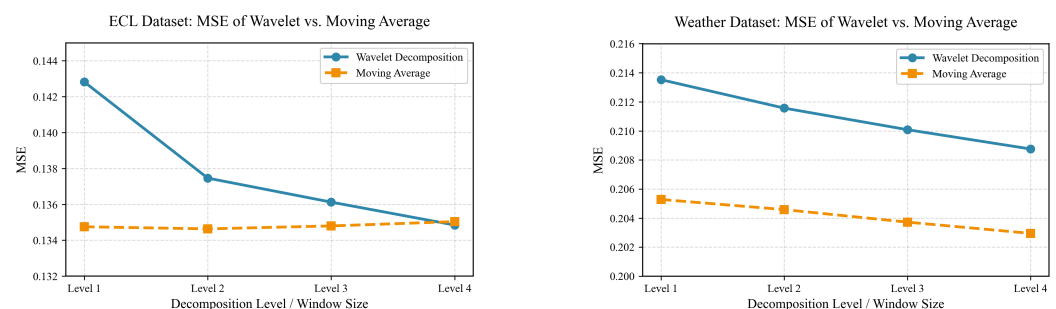


Figure A1. Performance comparison on the Weather and Electricity datasets under different decomposition methods and parameter settings for a fixed prediction horizon of 96.

Several key conclusions emerge from this comparative analysis. First, across both the highly volatile ECL dataset and the relatively smooth Weather dataset, the moving-average filter consistently yields lower prediction errors (MSEs) than wavelet decomposition at all tested granularities. For instance, on the ECL dataset, the optimal moving-average configuration (kernel size 25, MSE = 0.1346) marginally outperforms the best wavelet configuration (4 levels, MSE = 0.1348); this performance gap becomes substantially more pronounced on the Weather dataset. Second, while the efficacy of wavelet decomposition improves monotonically with additional decomposition levels—suggesting that finer frequency resolution offers some predictive utility—the moving-average method demonstrates remarkable stability and robustness across varying kernel sizes. Specifically, a kernel size of 25 achieves near-optimal performance, with alternative sizes inducing only negligible metric fluctuations. These empirical findings strongly substantiate the adoption of the moving-average filter as the default decomposition mechanism.

Although wavelet decomposition theoretically provides superior frequency-domain expressiveness, practical implementation often introduces boundary reconstruction artifacts within the trend component, or inappropriately leaves high-frequency remnants that should be relegated to the residual path. Such leakages inherently disrupt the efficacy of downstream component-specific encoding. Conversely, the signal separation achieved by the moving-average filter is remarkably clean and stable. This operational clarity yields clearly delineated, “easy-to-learn” trend and residual signals, enabling the MLP and Transformer encoders to optimally focus on the distinct structural patterns they are architecturally designed to capture. Furthermore, the observed insensitivity of the moving-average filter to specific kernel sizes reinforces a core thesis of this work: the predictive superiority of MLPFormer is not contingent upon hyper-optimized decomposition boundaries, but is fundamentally driven by the subsequent component-specific modeling mechanisms. Consequently, within the context of the proposed dual-path architecture, the moving-average filter proves to be a highly efficient, robust, and practically optimal

decomposition strategy, striking an ideal balance between rigorous signal separation and minimized hyperparameter complexity.

Appendix G. Additional Parameter Sensitivity Analysis

To evaluate the sensitivity of MLPFormer to core architectural parameters beyond the input sequence length analyzed in the main text, we systematically investigated the impact of the number of residual encoder layers and the hidden feature dimension (d_{model}) on forecasting performance. The empirical results across four representative benchmark datasets are visualized in Figure A2.

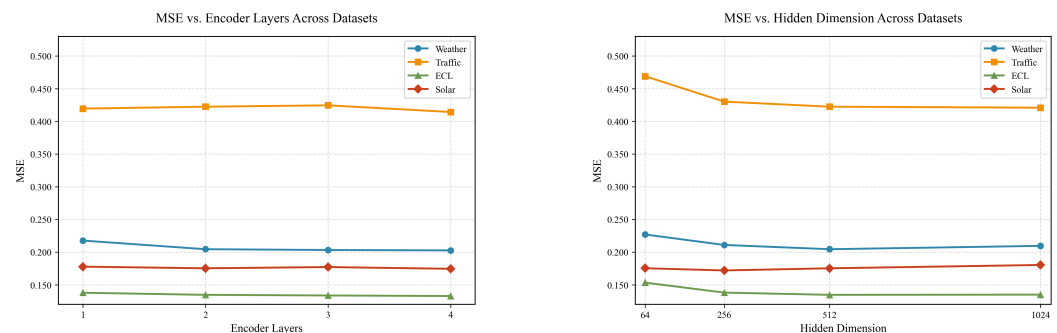


Figure A2. Comparison of forecasting errors under different numbers of residual encoder layers and hidden dimensions on four representative datasets, where the input sequence length and prediction horizon are fixed at 96.

Regarding the depth of the residual encoder (Figure A2, left panel), the performance impact is highly task-dependent. On the ECL and Weather datasets, predictive accuracy improves monotonically as the number of layers is scaled from 1 to 4. Conversely, on the Solar and Traffic datasets, the primary performance gains are realized when scaling from 1 to 2 layers, with deeper configurations yielding diminishing returns. This divergence is indicative of the intrinsic statistical properties of the residual signals within these datasets. The residuals in ECL and Weather exhibit complex, hierarchical temporal dependencies that benefit from the broader receptive fields provided by deeper encoders. In contrast, the residuals in Solar and Traffic are predominantly characterized by localized, high-frequency fluctuations, which are adequately captured by a shallower two-layer architecture. Consequently, a two-layer encoder is established as a robust default configuration offering broad generalization capability, while the architecture retains the flexibility to be scaled deeper for datasets exhibiting highly complex residual dynamics.

Regarding the hidden feature dimension (Figure A2, right panel), forecasting performance exhibits steady improvement as the dimensionality increases from 64 to 512. However, scaling the dimension further to 1024 produces diminishing returns across most datasets, and even induces a slight performance degradation on the Solar benchmark. This trajectory highlights a classic capacity–generalization trade-off: while a moderate expansion of the hidden dimension enhances the network’s ability to model fine-grained temporal features, exceeding the intrinsic complexity of the data (empirically observed here at dimensions beyond 512) exacerbates the risk of overfitting. Therefore, configuring the hidden dimension between 256 and 512 establishes an optimal equilibrium between representational expressiveness and robust generalization across diverse forecasting scenarios.

Appendix H. Component-Level Complexity Analysis

A core premise of MLPFormer is that the moving-average trend is statistically simpler (smoother, lower-frequency, more autocorrelated) than the residual, which justifies assigning a low-capacity MLP to the trend and a high-capacity Transformer to the residual.

To substantiate this premise quantitatively, we decompose every channel of each benchmark with the same moving-average filter ($K = 25$) used by the model and compute a panel of complexity descriptors on the trend and residual components: the normalized spectral centroid, the high-frequency energy ratio (fraction of spectral power above one quarter of the Nyquist frequency), the spectral entropy, the lag-1 autocorrelation, the order-3 permutation entropy, the Hjorth mobility, and the standard deviation of the first difference. Descriptors are averaged over channels (subsampling channels for the widest datasets) on the training split. The procedure requires no model training and operates purely on the decomposed components; the complete output for all nine benchmarks, including the additional Hjorth-complexity descriptor, is reported below.

Table A5. Component-level complexity of the moving-average trend versus the residual across representative benchmarks (training split, kernel $K = 25$, values averaged over channels). For each descriptor the more complex of the two components is shown in bold. The residual consistently exhibits higher spectral centroid, high-frequency energy ratio, permutation entropy, Hjorth mobility, and first-difference variability, and lower lag-1 autocorrelation than the trend.

Descriptor	Electricity		Weather		Solar		Traffic	
	Trend	Resid.	Trend	Resid.	Trend	Resid.	Trend	Resid.
Spectral centroid	0.003	0.070	0.004	0.092	0.008	0.062	0.010	0.082
HF energy ratio	1.9×10^{-4}	0.034	7.4×10^{-5}	0.094	2.2×10^{-6}	0.029	1.2×10^{-3}	0.046
Spectral entropy	0.332	0.313	0.420	0.825	0.248	0.729	0.449	0.477
Lag-1 autocorrelation	0.998	0.859	0.999	0.739	0.999	0.871	0.992	0.809
Permutation entropy	0.807	0.897	0.490	0.822	0.413	0.619	0.735	0.788
Hjorth mobility	0.054	0.512	0.037	0.668	0.053	0.501	0.114	0.607
Std. of 1st difference	0.023	0.419	0.032	0.206	0.048	0.108	0.034	0.579

The measurements confirm the premise across all benchmarks. The trend is overwhelmingly low-frequency and smooth: its lag-1 autocorrelation is near unity on every dataset (0.992–0.999), its spectral centroid and high-frequency energy ratio are one to four orders of magnitude smaller than those of the residual (for example, the high-frequency energy ratio is roughly $180\times$ larger in the residual on Electricity and over $10^4\times$ larger on Solar), and its first-difference standard deviation is a fraction of the residual's. Conversely, the residual is consistently higher-frequency and less regular, with markedly higher Hjorth mobility (a $5\text{--}18\times$ increase) and permutation entropy. The only near-tie is the spectral entropy on Electricity (trend 0.332 vs. residual 0.313), where the strongly periodic daily-cycle structure keeps the trend spectrum comparatively broad; on every other dataset the residual spectrum is substantially flatter (e.g., Weather 0.420 vs. 0.825, Solar 0.248 vs. 0.729). Taken together, these descriptors give direct, model-free evidence that the two components occupy distinct complexity regimes, justifying the use of a low-capacity MLP for the smooth trend and a high-capacity Transformer for the irregular residual.

The magnitude of this complexity gap also tracks where the dual-path design is most beneficial. Solar and Electricity—on which MLPFormer achieves its largest margins over the baselines—exhibit the cleanest separation, with the highest residual-to-trend high-frequency energy ratios (1.3×10^4 and 1.8×10^2). Traffic shows the weakest separation: its trend is the least smooth of all datasets (lowest trend autocorrelation, 0.992, and the highest trend high-frequency energy ratio), so the residual-to-trend high-frequency energy ratio collapses to about 39, the smallest in the panel. This statistically explains why the component-specific decomposition yields the smallest advantage on Traffic, where instantaneous cross-sensor dependencies—rather than a clean trend/residual split—carry most of the predictive signal, complementing the discussion in Sections 4.2 and 4.3.2.

Appendix I. Multi-Seed Stability

To assess run-to-run variability, MLPFormer is retrained with three random seeds $\{2025, 2026, 2027\}$ under the identical protocol across all nine benchmarks; all other settings match the main experiments. Table A6 reports the resulting mean and standard deviation, averaged over the four prediction horizons. The standard deviations are uniformly small—at most 0.006 in MSE and 0.003 in RMSE, and below 0.001 on several datasets—confirming that the reported improvements reflect stable behaviour rather than a favourable initialization. To verify that this stability is not unique to our model, we additionally retrain the strongest competing baselines on their most competitive datasets (iTransformer on Electricity, Solar, and Traffic; PatchTST on Weather); their seed-to-seed deviations are of the same small magnitude (Table A7), so the ranking between MLPFormer and its rivals is preserved under reseeding.

Table A6. MLPFormer run-to-run stability: mean $\pm$ standard deviation over three random seeds $\{2025, 2026, 2027\}$, averaged over the four prediction horizons ($S \in \{96, 192, 336, 720\}$) at $L = 96$.

Dataset	MSE	MAE	RMSE
Electricity	0.162 ± 0.001	0.257 ± 0.001	0.401 ± 0.001
Weather	0.316 ± 0.006	0.357 ± 0.005	0.557 ± 0.006
Solar	0.201 ± 0.001	0.244 ± 0.003	0.448 ± 0.001
Traffic	0.455 ± 0.004	0.286 ± 0.001	0.674 ± 0.003
Exchange	0.367 ± 0.003	0.407 ± 0.002	0.556 ± 0.002
ETTh1	0.448 ± 0.003	0.440 ± 0.002	0.669 ± 0.002
ETTh2	0.386 ± 0.002	0.410 ± 0.002	0.620 ± 0.002
ETTm1	0.385 ± 0.001	0.397 ± 0.000	0.619 ± 0.001
ETTm2	0.276 ± 0.000	0.322 ± 0.000	0.519 ± 0.000

Table A7. Seed-to-seed stability of the strongest baselines on their most competitive datasets: mean $\pm$ standard deviation over three random seeds $\{2025, 2026, 2027\}$, averaged over the four prediction horizons at $L = 96$.

Model	Dataset	MSE	MAE	RMSE
iTransformer	Electricity	0.175 ± 0.002	0.266 ± 0.002	0.417 ± 0.002
	Solar	0.203 ± 0.002	0.251 ± 0.000	0.451 ± 0.002
	Traffic	0.420 ± 0.001	0.280 ± 0.000	0.648 ± 0.000
PatchTST	Weather	0.325 ± 0.000	0.335 ± 0.001	0.566 ± 0.000

Appendix J. Optimal-Lookback Comparison

The main results fix the lookback window at $L = 96$ for a controlled comparison, following the convention adopted by the iTransformer, PatchTST, DLinear, and TimeMixer benchmarks, where $L = 96$ is the standard input length for the long-term forecasting protocol. Because several baselines (notably PatchTST and iTransformer) are reported to benefit from longer histories, we additionally evaluate MLPFormer and iTransformer at $L \in \{96, 336, 512\}$ on four representative datasets, keeping all other settings fixed. Table A8 shows that MLPFormer remains competitive—and on the trend-dominated datasets superior—when iTransformer is granted its preferred longer lookbacks, confirming that the $L = 96$ protocol does not unfairly disadvantage the baselines and that our conclusions are not an artifact of the fixed window.

Table A8. Forecasting error (averaged over the four horizons $S \in \{96, 192, 336, 720\}$) as a function of the lookback window $L \in \{96, 336, 512\}$. Longer lookbacks are the regime in which iTransformer is reported to perform best. **Bold** marks each model’s best lookback per dataset. Entries are single-seed runs; the $L = 96$ column is therefore a representative run consistent with, but not identical to, the three-seed means in Table 2.

Dataset	L	MLPFormer (Ours)		iTransformer	
		MSE	MAE	MSE	MAE
Electricity	96	0.161	0.256	0.173	0.265
	336	0.162	0.257	0.166	0.260
	512	0.158	0.254	0.161	0.256
Weather	96	0.312	0.363	0.321	0.358
	336	0.301	0.344	0.347	0.379
	512	0.295	0.347	0.348	0.385
Solar	96	0.202	0.240	0.202	0.251
	336	0.193	0.245	0.214	0.270
	512	0.194	0.247	0.204	0.268
Traffic	96	0.452	0.285	0.421	0.280
	336	0.410	0.276	0.389	0.270
	512	0.413	0.281	0.382	0.269

Two conclusions follow. First, extending the lookback does not overturn the ranking on the datasets where the dual-path design is most beneficial: on Electricity, Weather, and Solar, MLPFormer at $L = 96$ already outperforms iTransformer at its *best* lookback (e.g., on Weather, MLPFormer at $L = 96$ attains an MSE of 0.312 versus iTransformer’s best of 0.321 at $L = 96$, and the gap widens to 0.295 versus 0.348 at $L = 512$, since iTransformer degrades as the window grows). This confirms that the fixed $L = 96$ protocol does not artificially favour MLPFormer. Second, on Traffic—the dataset governed by dense cross-sensor dependencies—iTransformer improves with longer histories and retains its advantage, consistent with the analysis in Section 4.3.2: when predictability is dominated by instantaneous cross-variable coupling, additional temporal context primarily benefits the full variable-attention baseline. MLPFormer nonetheless also improves with longer lookbacks on Traffic (MSE $0.452 \rightarrow 0.410$), showing that it makes effective use of extended context rather than saturating. We report the MLPFormer–iTransformer comparison because iTransformer is the strongest and most directly comparable variable-attention baseline; PatchTST was excluded from the long-lookback sweep because its patching mechanism makes memory and training cost grow rapidly with the input length, exceeding the 24 GB budget of our single RTX 4090 at $L \in \{336, 512\}$ on the high-dimensional benchmarks. Since PatchTST is already reported at the standard $L = 96$ in the main comparison (Table 2), and iTransformer is the more memory-efficient long-context Transformer, the sweep above is sufficient to establish that the $L = 96$ protocol is fair.

Appendix K. Capacity-Matched Controls

Reviewer feedback correctly notes that the full MLPFormer (~ 11 M parameters) is roughly twice the size of the homogeneous *w/o* MLP ablation and the iTransformer baseline (~ 6.45 M). To disentangle architecture from capacity, we bracket the comparison from both sides: (i) we shrink MLPFormer *down* to 6.32 M by narrowing its model and feed-forward dimensions ($d_{\text{model}} = 384$, $d_{\text{ff}} = 1536$), matching the iTransformer baseline, and

(ii) we scale iTransformer *up* well past 11 M by widening it and adding layers. Trainable parameter counts are computed directly from the instantiated models and are reported in Table A9; because the inverted-attention parameters of iTransformer are dominated by the feed-forward blocks (which are independent of the variable count N), matching a precise target is coarse, and our scaled iTransformer overshoots to ~ 28.6 M on Electricity/Solar and ~ 50.7 M on Weather. This does not weaken the argument: it means the baseline is granted *more* capacity than the full MLPFormer, not less.

Table A9. Capacity-matched comparison, averaged over the four prediction horizons ($S \in \{96, 192, 336, 720\}$) at $L = 96$. “Params (M)” is the number of trainable parameters computed directly from the instantiated model (identical across datasets, as the parameter count is dominated by N -independent feed-forward blocks). [†] The scaled-up iTransformer uses $d_{\text{model}}/d_{\text{ff}} = 768/3072$ with four layers on Electricity and Solar (~ 28.6 M) and $1024/4096$ on Weather (~ 50.7 M); it therefore has *more* capacity than the full MLPFormer. All entries are single-seed runs under a fixed configuration to keep the capacity comparison controlled, so the full-model column may differ marginally from the three-seed means in Table 2. **Bold** marks the best result per metric column.

Model (Capacity)	Params (M)	Electricity		Weather		Solar	
		MSE	MAE	MSE	MAE	MSE	MAE
MLPFormer (full)	11.05	0.161	0.256	0.312	0.363	0.202	0.240
MLPFormer (shrunk)	6.32	0.163	0.258	0.309	0.356	0.197	0.244
iTransformer (baseline)	6.45	0.173	0.265	0.321	0.358	0.202	0.251
iTransformer (scaled up) [†]	28.6/50.7	0.170	0.263	0.670	0.635	0.215	0.261
w/o MLP (homogeneous, $2 \times$ Transformer)	17.36	0.165	0.260	0.332	0.370	0.200	0.242

The results isolate architecture from capacity in three ways. First, *shrinking* MLPFormer to 6.32 M—below the 6.45 M iTransformer baseline—barely changes its accuracy (Electricity MSE $0.161 \rightarrow 0.163$; Solar even improves slightly to 0.197) and it still clearly outperforms the equally sized iTransformer on all three datasets. The dual-path advantage therefore survives at matched, and even lower, parameter budgets. Second, *growing* iTransformer well beyond MLPFormer’s size does not close the gap: at ~ 28.6 M it improves only marginally on Electricity ($0.173 \rightarrow 0.170$) and *worsens* on Solar, and at ~ 50.7 M on Weather it collapses to 0.670 MSE—a textbook case of over-parameterized attention overfitting a smooth, low-dimensional signal. Extra homogeneous capacity is thus not a substitute for the component-specific decomposition. Third, the homogeneous dual-Transformer (*w/o* MLP, 17.36 M) also fails to match the full MLPFormer despite having more parameters, confirming that applying high-capacity attention to the smooth trend is actively harmful rather than merely wasteful. Taken together, the improvements attributable to MLPFormer stem from routing each component to an encoder matched to its complexity, not from raw parameter count.

Appendix L. Visualization of Prediction Results

Figures A3 and A4 qualitatively compare the 96-step predictive trajectories of all evaluated models against the ground truth for the Weather and Electricity datasets, respectively. The visualizations demonstrate that MLPFormer more accurately tracks both the low-frequency structural trends and the localized high-frequency fluctuations compared to the baseline models. This superior tracking capability visually corroborates the effectiveness of the proposed dual-path architecture in capturing the inherently composite nature of real-world time series.

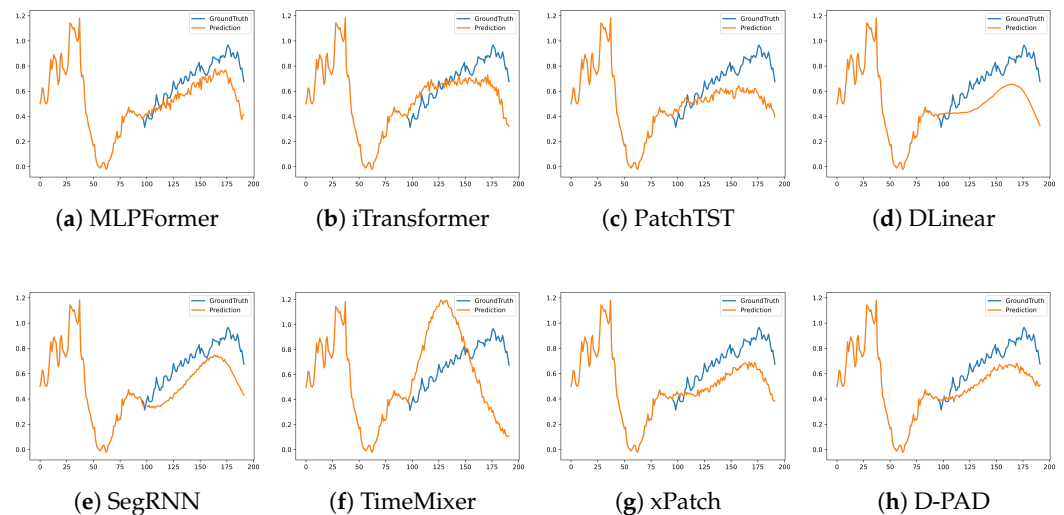


Figure A3. Predicted vs. ground-truth time series for input-96-predict-96 on the Weather dataset.

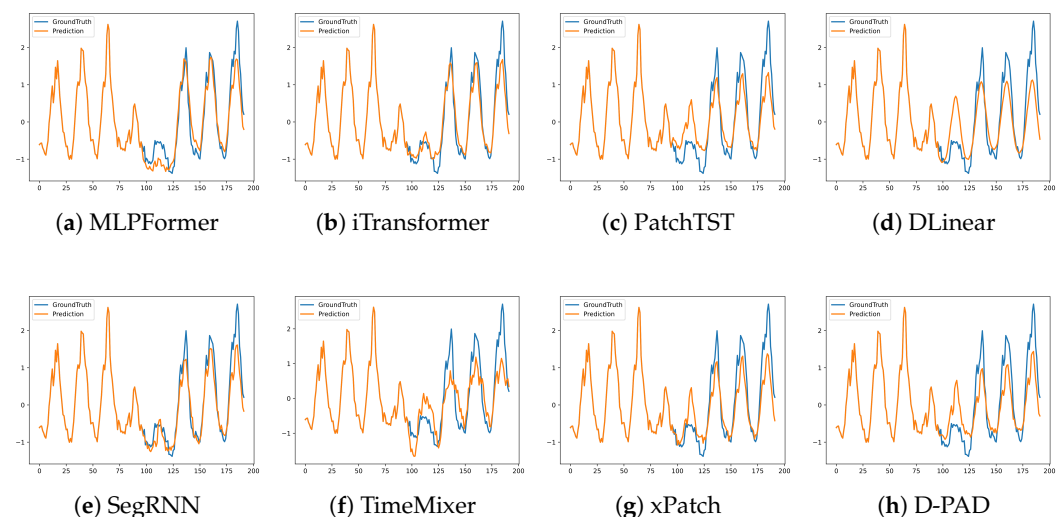


Figure A4. Predicted vs. ground-truth time series for input-96-predict-96 on the Electricity (ECL) dataset.

References

1. Wen, Q.; Yang, L.; Zhou, T.; Sun, L. Robust time series analysis and applications: An industrial perspective. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 4836–4837.
2. Lin, H.; Gao, Z.; Xu, Y.; Wu, L.; Li, L.; Li, S.Z. Conditional local convolution for spatio-temporal meteorological forecasting. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 7470–7478.
3. Baek, M.; Seo, Y. Hybrid forecasting of university electricity demand using time series and deep learning. *Energy Build.* **2025**, *347*, 116400.
4. Chaffard, O.; Cavazza, M.; Prendinger, H. Enhancing Large Language Models for Bitcoin Time Series Forecasting. *Knowl.-Based Syst.* **2025**, *330*, 114449.
5. Zhang, L.; Wilson, R.; Sumner, M.; Wu, Y. Advanced multimodal fusion method for very short-term solar irradiance forecasting using sky images and meteorological data: A gate and transformer mechanism approach. *Renew. Energy* **2023**, *216*, 118952.
6. Wu, H.; Liang, Y.; Zuo, J. Human-inspired spatiotemporal feature extraction and fusion network for weather forecasting. *Expert Syst. Appl.* **2022**, *207*, 118089.
7. Zhao, H.; Wu, L.; Li, Z.; Zhang, X.; Liu, Q.; Chen, E. Predicting the dynamics in Internet finance based on deep neural network structure. *J. Comput. Res. Dev.* **2019**, *56*, 1621–1631.
8. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv* **2022**, arXiv:2211.14730.

9. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. iTransformer: Inverted transformers are effective for time series forecasting. *arXiv* **2023**, arXiv:2310.06625.
10. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 11121–11128.
11. Chen, S.A.; Li, C.L.; Yoder, N.; Arik, S.O.; Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv* **2023**, arXiv:2303.06053.
12. Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J.Y.; Zhou, J. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv* **2024**, arXiv:2405.14616.
13. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
14. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2022; pp. 27268–27286.
15. Stitsyuk, A.; Choi, J. xPatch: Dual-Stream Time Series Forecasting with Exponential Seasonal-Trend Decomposition. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 20601–20609.
16. Zhong, S.; Song, S.; Zhuo, W.; Li, G.; Liu, Y.; Chan, S.H.G. A multi-scale decomposition mlp-mixer for time series analysis. *arXiv* **2023**, arXiv:2310.11959.
17. Challu, C.; Olivares, K.G.; Oreshkin, B.N.; Garza Ramirez, F.; Mergenthaler Canseco, M.; Dubrawski, A. NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 6989–6997. <https://doi.org/10.1609/aaai.v37i6.25854>.
18. Choi, J.M.; Yalavarthi, V.K.; Schmidt-Thieme, L. HPMixer: Hierarchical Patching for Multivariate Time Series Forecasting. *arXiv* **2026**, arXiv:2602.16468.
19. Lee, G.; Park, H.; Kim, M.; Kim, D.; Bae, H. IConv: Focusing on Local Variation with Channel Independent Convolution for Multivariate Time Series Forecasting. *arXiv* **2025**, arXiv:2509.20783.
20. An, R.; Qu, H.; Fan, W.; Shang, X.; Li, Q. DeMa: Dual-Path Delay-Aware Mamba for Efficient Multivariate Time Series Analysis. *arXiv* **2026**, arXiv:2601.05527.
21. Nagashima, S.; Suzuki, S.; Koyama, S.; Hirano, S. A Decomposition-based State Space Model for Multivariate Time-Series Forecasting. *arXiv* **2026**, arXiv:2602.05389.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
23. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115.
24. Zhang, Y.; Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of the The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 1–5 May 2023.
25. Yuan, X.; Chen, L. D-PAD: Deep-Shallow Multi-Frequency Patterns Disentangling for Time Series Forecasting. *arXiv* **2024**, arXiv:2403.17814.
26. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2019**, arXiv:1905.10437.
27. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv* **2022**, arXiv:2210.02186.
28. Yuan, Z.; Liu, Z.; Marina, M.K. AltTS: A Dual-Path Framework with Alternating Optimization for Multivariate Time Series Forecasting. *arXiv* **2026**, arXiv:2602.11533.
29. Hamilton, J. *Time Series Analysis*; Princeton University Press: Princeton, NJ, USA, 1994.
30. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1251–1258.
31. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 95–104.
32. Lin, S.; Lin, W.; Wu, W.; Zhao, F.; Mo, R.; Zhang, H. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv* **2023**, arXiv:2308.11200.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.