Unifying Channel Independence and Mixing: Multi-Scale Patch Recursion for Global–Local Representation Synergy in Multivariate Time Series Forecasting

Wenhao Zhang[1,2], Chun Zhang[1,2*], Wei Bai[3,4*], Ning Zhang[1,2], Changxia Gao[1,2], Yuxin Jia[1], Chenhao Shi[1,2], Shaoxiong Pang[1,2]

[1]School of Computer Science & Technology, Beijing Jiaotong University
[2]Engineering Research Center of Network Management Technology for High-Speed Railway of Ministry of Education
[3]The Center of National Railway Intelligent Transportation System Engineering and Technology
[4]China Academy of Railway Sciences Institute of Computing Technologies
{zhveh777, chzhang1, nzhang1, chxiag, yuxinjia, chshi1, sxpang}@bjtu.edu.cn, baiweisxu@163.com

Abstract

Multivariate time series forecasting underpins applications in finance, meteorology, and industrial operations. Yet two persistent hurdles remain: (i) models typically choose between Channel–Independent (CI) and Channel–Mixed (CM) formulations—each with distinct strengths—leading to large performance variance across datasets; and (ii) short-term dynamics and long-term trends are hard to model jointly, making it difficult to capture both transient bursts and periodic patterns. We propose FusionTimePatch (FTP), a purely MLP-driven, lightweight framework composed of three modules: (1) Dual-View Global–Local Fusion (Dual-GLF), which runs CI and CM views in parallel and employs multi-scale patch recursion to adaptively adjust the look-back window, thereby coupling global tendencies with local details; (2) Channel Enhancement (CE), which adaptively identifies and amplifies salient channel signals and diffuses them to others, improving sensitivity to abrupt events and latent drivers; and (3) a Linear Fusion layer, which unifies Dual-GLF and CE outputs to strengthen cross-view interactions and enhance robustness. Extensive experiments on multiple public benchmarks show FTP consistently surpasses state-of-the-art counterparts in both accuracy and efficiency, offering a scalable new paradigm for multichannel forecasting. Code and datasets are publicly available at https://github.com/Zhveh7/FTP.

Introduction

Multivariate time series forecasting (MTSF) is a cornerstone of modern decision making in quantitative finance (Zhu and Shasha 2002), weather prediction (Kadiyala and Kumar 2014), industrial maintenance (Papadimitriou and Yu 2006), and healthcare (Morid, Sheng, and Dunbar 2023). In practice, the sensed data are inherently multichannel: each variable exhibits its own dynamics (Nie et al. 2023) while simultaneously engaging in intricate cross-channel interactions (Liu et al. 2024). Complicating matters further, real-world time series blend short-term fluctuations, long-period cycles (Lai et al. 2018; Eldele et al. 2021), and occasional abrupt events (Al-Musaylh et al. 2018). Delivering accurate and robust forecasts under these heterogeneous, multi-scale (Chen et al. 2024; Wang et al. 2023) conditions remains a formidable challenge.

Current research is dominated by two mutually exclusive paradigms. Channel–Independent (CI) (Nie et al. 2023) models treat every variable in isolation, effectively suppressing cross-channel noise yet overlooking collaborative signals. Channel–Mixed (CM) (Liu et al. 2024; Han, Ye, and Zhan 2024; Li et al. 2023) models aggregate variables to exploit global correlations, but they suffer from interference when channels are highly heterogeneous. Choosing CI or CM therefore yields "specialists" that excel on some datasets but generalize poorly across domains.

Multi-scale temporal structures present an additional obstacle. Recurrent (Elman 1990) and convolutional (Fukushima 1980; LeCun et al. 2002) networks alleviate local pattern extraction but struggle with long-range dependencies. Transformers (Vaswani et al. 2017) capture such dependencies yet rely on point-wise attention, incurring high computational overhead and weak local aggregation. Patch-based (Nie et al. 2023; Tang and Zhang 2025) methods improve short-term modeling by grouping contiguous steps, but their fixed-length patches cannot adapt to variable temporal granularity. Consequently, simultaneously modeling local details and global trends (Ekambaram et al. 2023) across channels and scales remains unsettled.

These bottlenecks suggest that single-view channel modeling and fixed-scale strategies are inherently insufficient. We therefore aim to:

- **Channel dimension:** reconcile CI and CM by unifying their strengths in one framework;
- **Temporal dimension:** dynamically perceive varying scales via adaptive-length patches, fusing local and global signals;
- **Architectural efficiency:** abandon expensive attention in favor of a lightweight, easily extensible MLP backbone.

We introduce **FusionTimePatch (FTP)**, a pure-MLP framework that addresses the above issues through channel-view fusion and Global–Local joint modeling. Its key components are:

*Corresponding authors: Chun Zhang, Wei Bai.

1. **Dual-View Global–Local Fusion (Dual-GLF)**: two parallel pipelines—CI and CM—augmented by multi-scale patch-based recursion that adaptively adjusts the look-back window, thereby jointly capturing global trends and local details.
2. **Channel Enhancement (CE)**: an adaptive mechanism that amplifies salient channel features and propagates them across variables, heightening responsiveness to sudden events and latent drivers.
3. **Linear Fusion Layer**: a unified linear projection that integrates Dual-GLF and CE outputs, strengthening cross-view interactions and boosting robustness and generalization.

FTP is a purely MLP-driven architecture that forgoes elaborate attention operations, thereby drastically reducing computational overhead, and despite its lightweight design, FTP consistently surpasses state-of-the-art methods in accuracy across a suite of real-world benchmark datasets. We believe FTP offers a simple yet powerful solution for multi-channel forecasting and provides fresh insights into the unified design of channel-view fusion and Global–Local modeling.

Related Work

Early RNNs (Hochreiter and Schmidhuber 1997; Cho et al. 2014; Rangapuram et al. 2018; Jia et al. 2023, 2024) and CNNs (Bai, Kolter, and Koltun 2018; Franceschi, Dieuleveut, and Jaggi 2019; Zheng et al. 2014; Chen, Segovia, and Gel 2021) excel at capturing local patterns and short-term dependencies, yet their effectiveness on long sequences and cross-channel interactions (Li et al. 2023) is hampered by gradient decay and limited receptive fields. The Transformer family (Vaswani et al. 2017; Li et al. 2019; Kitaev, Kaiser, and Levskaya 2020; Zhou et al. 2021; Wu et al. 2021; Liu et al. 2022b; Zhou et al. 2022; Cirstea et al. 2022; Zhang and Yan 2023; Liu et al. 2024; Nie et al. 2023; Kalyan, Rajasekharan, and Sangeetha 2021; Wu et al. 2023; Wang et al. 2025), empowered by self-attention, greatly improves long-range modeling and has spawned numerous efficient -former variants; however, its point-wise attention still incurs an $\mathcal{O}(L^2)$ complexity that becomes prohibitive for ultra-long series.

PatchTST (Nie et al. 2023) is the first to introduce patching (Dosovitskiy et al. 2020) and a CI view into forecasting, strengthening local modeling with fixed subsequence blocks but lacking cross-channel dependencies and multi-scale representations. iTransformer (Liu et al. 2024) groups whole variables into patches to capture global interactions, yet under-represents local cues; PatchMLP (Tang and Zhang 2025) explores multi-resolution patches but still falls short of true Global–Local synergy.

Purely linear models have recently surged (Zeng et al. 2023; Ekambaram et al. 2023; Lu et al. 2024; Tolstikhin et al. 2021; Touvron et al. 2022; Chen et al. 2021). DLinear (Zeng et al. 2023) leverages trend–residual decomposition to outperform many Transformers on long-sequence tasks. TSMixer (Ekambaram et al. 2023) and SOFTS (Lu et al. 2024), two MLP-based approaches, offer distinctive multi-scale mixing and channel interaction with "fewer parameters, faster inference," yet remain sensitive to long-range dependencies and distribution shift. Frequency-domain methods TimesNet (Wu et al. 2023) and FreDF (Wang et al. 2025) encode periodicity via fast Fourier transforms, but the required domain conversion raises computational cost and has shown limited gains on complex multichannel workloads. Channel-clustering approaches such as Duet (Qiu et al. 2025) rely on manually preset cluster counts and strong distribution assumptions, lacking adaptivity. Meanwhile, large pre-trained models Time-LLM (Jin et al. 2024) demonstrate cross-domain and zero-shot potential, but their billion-scale parameters and costly fine-tuning impede deployment on edge devices.

In summary, (i) CI and CM paradigms remain disjoint, leading to erratic cross-dataset performance; (ii) fixed-scale patches and plain MLPs struggle to capture both transient local events and long-term dependencies; and (iii) large models, though highly accurate, cannot meet lightweight, low-latency demands. To address these gaps, we propose FTP, a pure-MLP framework unifying Dual-GLF (CI + CM dual views with multi-scale recursive patches), CE, and a Linear Fusion layer—achieving efficient multichannel dependency modeling with superior accuracy, stability, and deployability.

Algorithm 1: FusionTimePatch Forward Process

Input: Historical sequence $\mathbf{X}^{(0)} \in \mathbb{R}^{C\times L}$, $e\_layers = E$
Output: Prediction $\mathbf{Y} \in \mathbb{R}^{C\times T}$

for $el = 1$ to E **do**
  $\mathbf{X}_{el}^{\mathrm{I}} \leftarrow \mathrm{GLF\_CI}\big(\mathbf{X}^{(el-1)}\big)$
  $\mathbf{X}_{el}^{\mathrm{M}} \leftarrow \mathrm{GLF\_CM}\big(\mathbf{X}^{(el-1)}\big)$
  $\mathbf{X}_{el}^{\mathrm{e}} \leftarrow \mathrm{CE}\big(\mathbf{X}^{(el-1)}\big)$
  $\mathbf{X}_{el}^{\mathrm{F}} \leftarrow \mathrm{Linear}\big(\mathbf{X}_{el}^{\mathrm{I}} \,\|\, \mathbf{X}_{el}^{\mathrm{M}} \,\|\, \mathbf{X}_{el}^{\mathrm{e}}\big)$
  $\mathbf{X}^{(el)} \leftarrow \mathrm{Linear}\big(\mathbf{X}_{el}^{\mathrm{F}} \,\|\, \mathrm{Emb}\big(\mathbf{X}^{(el-1)}\big)\big)$
end for
$\mathbf{Y} \leftarrow \mathrm{Predict}\big(\mathrm{Linear}(\mathbf{X}^{(E)})\big)$

FusionTimePatch (FTP)

Problem Formulation

In the MTSF task, given a historical observation sequence $X \in \mathbb{R}^{C\times L}$, the goal is to predict the future sequence $Y \in \mathbb{R}^{C\times T}$, where C denotes the number of channels, L is the look-back window length, and T is the prediction horizon.

Framework Overview

The proposed FTP framework consists of three key components: **Dual-GLF**, **CE**, and a **Linear Fusion layer**. The input sequence is processed sequentially through these modules to produce the final prediction. An overview of the framework is illustrated in Figure 1, and the core algorithmic flow is presented in Algorithm 1.

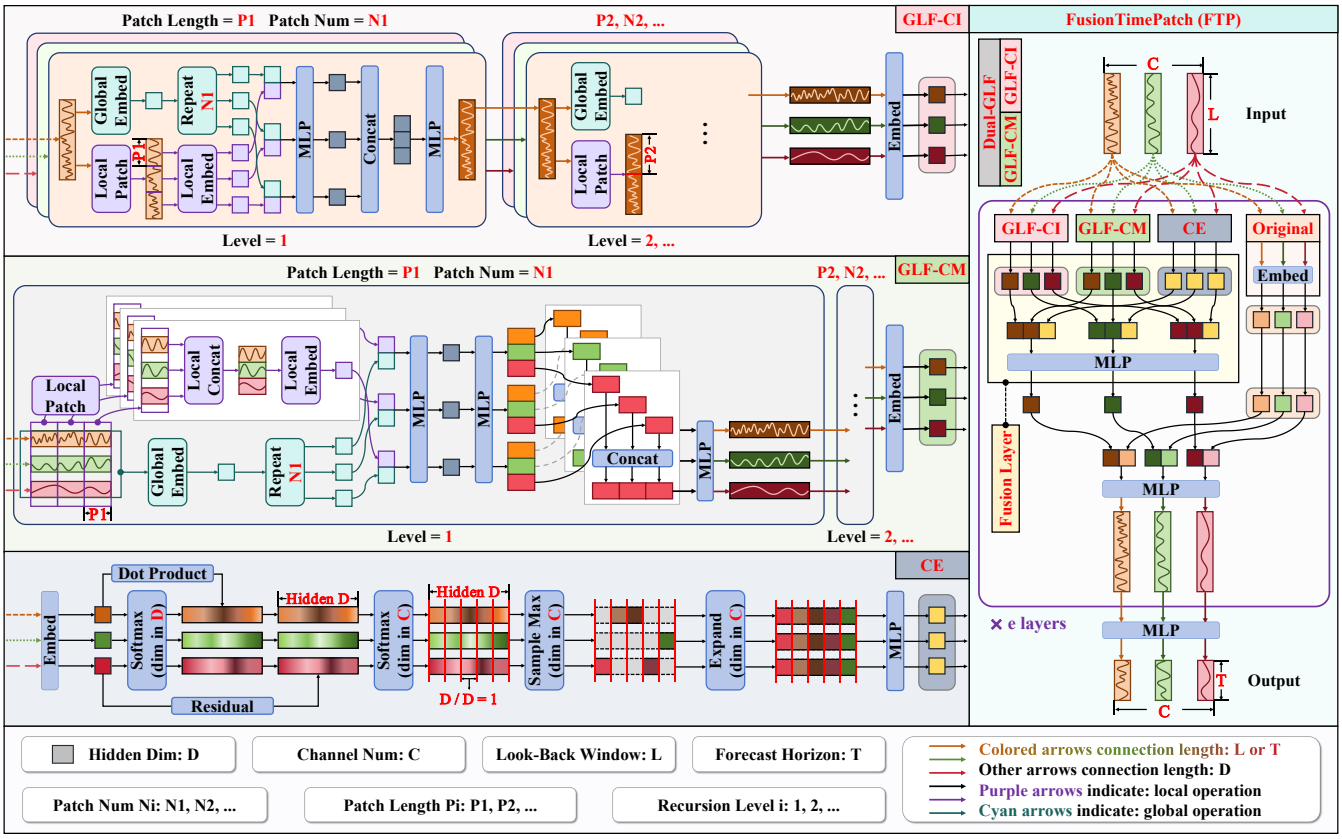

Figure 1: The overall architecture of FusionTimePatch (FTP). FTP consists of three core components: Dual-GLF introduces CI and CM perspectives in parallel, leveraging multi-scale patch recursion to capture both local and global temporal patterns; the CE module enhances salient channel features and diffuses them across channels, improving sensitivity to anomalies and underlying drivers; the Fusion layer linearly merges the outputs of Dual-GLF and CE, concatenates the result with the original encoder input, propagates the combined representation through the encoder stack, and finally applies an MLP head to produce the forecast. This design enables the synergistic modeling of unified CI/CM perspectives and temporal structures.

Dual-View Global–Local Fusion (Dual-GLF)

As illustrated in Figure 1, Dual-GLF captures short-term dynamics and long-term trends in parallel under two complementary views: a Channel–Independent branch (GLF-CI) and a Channel–Mixed branch (GLF-CM), organized in many hierarchical recursive levels. At recursion level i, we adopt a multi-scale, adaptive patching scheme with span $P_i = i \times$ unit, where unit is the base patch length. To ensure full coverage of the look-back window L, we pad a stride of length S at the end; the number of patches at level i is $N_i = \left\lfloor \frac{L-P_i}{S} \right\rfloor + 2$. For local information, we embed each patch of length P_i to obtain $X_l \in \mathbb{R}^D$; for global information, we embed the whole window L to obtain $X_g \in \mathbb{R}^D$. We then concatenate X_g and X_l along the hidden dimension and project back to length L. The output is passed to the next level with an updated patch span, continuing the recursive process.

GLF-CI keeps channels independent and performs scale-specific embeddings per channel: $\mathbb{R}^{C\times L} \to \mathbb{R}^{C\times N_i\times P_i} \to \mathbb{R}^{C\times D}$. GLF-CM explicitly models cross-channel interactions at multiple scales by stacking channels within each patch: $\mathbb{R}^{C\times L} \to \mathbb{R}^{[C\times P_i]\times N_i} \to \mathbb{R}^{C\times D}$. The two views are finally fused by a linear mapper, which mitigates interference while preserving complementary cues. This design bridges local short-term fluctuations and global long-horizon trends, leveraging the strengths of both CI and CM perspectives.

Channel Enhancement (CE)

As illustrated in Figure 1, CE first embeds each channel along the temporal dimension to obtain high-dimensional features $X_e \in \mathbb{R}^{C\times D}$. We then apply a hidden-dimension Softmax to assess the importance of each latent component and reweight the features multiplicatively, amplifying salient dimensions while suppressing minor ones, yielding an updated $X_e \in \mathbb{R}^{C\times D}$. Next, a channel-wise Softmax produces inter-channel importance weights. According to these weights, we perform probabilistic sampling to select dominant channels (higher weight $\to$ higher sampling probability), broadcast their features to all channels, and apply an MLP to obtain the enhanced representation $X_e \in \mathbb{R}^{C\times D}$.

CE thus highlights critical anomalies and abrupt changes while preserving cross-channel dependencies.

Linear Fusion and Prediction (Fusion)

As illustrated in Figure 1, the Fusion layer first projects the CI and CM outputs from Dual-GLF, $X_I, X_M \in \mathbb{R}^{C\times L}$, to a common hidden size D. The projected features are concatenated with the CE output $X_e \in \mathbb{R}^{C\times D}$, $[X_I, X_M, X_e] \in \mathbb{R}^{C\times 3D}$, and compressed by a linear layer to obtain the fused representation $X_F \in \mathbb{R}^{C\times D}$. We then concatenate X_F with the embedded input of the current layer (Concat Original) $X_o \in \mathbb{R}^{C\times D}$ to form $[X_F, X_o] \in \mathbb{R}^{C\times 2D}$, map it back to the sequence space $X \in \mathbb{R}^{C\times L}$ via an MLP, and propagate through E encoder layers. Finally, an MLP head takes $\mathrm{Emb}(X)$ and produces the forecast $Y \in \mathbb{R}^{C\times T}$.

Computational Complexity Analysis

We analyze the theoretical complexity of FTP to assess its scalability for long-sequence forecasting. Let B be the batch size, D the hidden dimension, L the look-back length, C the number of channels, A the number of adaptive levels, and E the depth of the Linear-Fusion stack (e_layers).

Each adaptive level in GLF-CI and GLF-CM involves patching and MLP projection with complexity $\mathcal{O}(A\,C\,L)$. The CE module contributes $\mathcal{O}(C\,L)$ due to two Softmax operations and a linear layer. As these modules run in parallel per layer, the per-layer cost is $\mathcal{O}((2A+1)\,C\,L)$, and the total forward cost over E layers becomes: $T_{\text{FTP}} = \mathcal{O}(B\,D\,E\,A\,C\,L) \approx \mathcal{O}(A\,C\,L)$, with B, D, E fixed.

Unlike Transformer-based models with $\mathcal{O}(C\,L^2)$ complexity, FTP retains the linear efficiency $\mathcal{O}(A\,C\,L)$ of MLP architectures while enabling cross-channel interaction and multi-scale modeling. Though multi-scale recursion adds an A-fold factor, it significantly enhances representational power, making FTP both efficient and scalable for long-horizon, high-dimensional forecasting.

Experiments

We conduct comprehensive evaluations of the proposed FTP framework across a wide range of time series forecasting tasks to verify its overall effectiveness, and further investigate the individual contributions of each component.

Datasets and Preprocessing. We evaluate FTP on 12 public multivariate time-series datasets (Zhou et al. 2021; Lu et al. 2024; Wang et al. 2025; Tang and Zhang 2025): four transformer oil temperature benchmarks (ETTh1/ETTh2, ETTm1/ETTm2), and eight traffic–weather–energy datasets (Electricity, Traffic, Weather, Solar, PEMS03/04/07/08). All samples are normalized with Reversible Instance Normalization (RevIN) (Kim et al. 2021): inputs are centered to zero mean and scaled to unit variance; after prediction, outputs are inverse-transformed using the original per-series statistics to mitigate the impact of local volatility.

Metrics and Optimizer. We report Mean Squared Error (MSE) and Mean Absolute Error (MAE). Parameters are optimized with Adam (Kingma and Ba 2014); the training loss is MSE.

Baselines. We compare against representative models from three families.

- Linear-MLP: PatchMLP (Tang and Zhang 2025), SOFTS (Lu et al. 2024), TSMixer (Ekambaram et al. 2023), TiDE (Das et al. 2023), DLinear (Zeng et al. 2023).
- Transformer: FreDF (Wang et al. 2025), iTransformer (Liu et al. 2024), PatchTST (Nie et al. 2023), Crossformer (Zhang and Yan 2023).
- CNN: SCINet (Liu et al. 2022a), TimesNet (Wu et al. 2023).

Experimental Setup. Prediction horizons T = {12, 24, 48, 96} for PEMS datasets, and T = {96, 192, 336, 720} for the others. The look-back window is fixed as L = 96. Train/validation/test splits follow the protocol released by SOFTS, ensuring fair comparability across methods.

Multivariate Time Series Forecasting

The main results are summarized in Table 1. Across all 12 benchmark datasets, FTP consistently achieves the best or near-best performance, outperforming existing state-of-the-art methods in its category by a clear margin. Notably, FTP obtains the largest number of top-1 results among all compared methods. For instance, on the PEMS08 dataset (with 170 variables), FTP reduces the average MSE from 0.138 (achieved by the previous best model) to 0.121, yielding a relative improvement of 12.31%. Similar gains are observed on other datasets, particularly when compared to patch-based MLP models such as PatchMLP, highlighting the superiority of FTP under the same modeling paradigm. These results demonstrate that FTP is not only effective across diverse forecasting tasks, but also robust to the scale of channel dimensionality, performing well on both high-dimensional and low-dimensional time series.

Model Efficiency

On Electricity dataset (L = 96, T = 96, batch = 16), Figure 2 plots MSE (y) vs. Training Time/ms per iter (x) with bubble size = Memory (MB). FTP uses 276 MB, 8.5 ms/iter and attains the lowest MSE (smaller is better), delivering high accuracy at low cost. In contrast, PatchTST requires ≈4713 MB, 137.3 ms/iter, and the peer PatchMLP costs 476 MB, 21.2 ms/iter, both with higher errors. FTP sits at the bottom-left with the smallest bubble, evidencing substantially lower memory and latency while achieving superior accuracy.

Ablation Study

To quantify the effectiveness and contribution of each component in FTP, we conduct a series of ablation studies. Specifically, we independently remove six key modules: Dual-GLF, GLF-CI, GLF-CM, CE, Concat Original, and Fusion, while keeping all other settings unchanged. Each experiment was repeated five times. Table 2 reports the average MSE/MAE across four prediction horizons.

As shown in Table 2, removing GLF-CM yields the largest error increases on PEMS04, Weather, and Electricity, underscoring the importance of cross-channel Global–Local complementarity in multivariate settings.

Models	FTP (ours)		FreDF (2025)		PatchMLP (2025)		SOFTS (2024)		iTransformer (2024)		PatchTST (2023)		TSMixer (2023)		Crossformer (2023)		TiDE (2023)		TimesNet (2023)		DLinear (2023)		SCINet (2022)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.386**	**0.398**	0.392	0.399	0.388	**0.398**	0.393	0.403	0.407	0.410	0.396	0.406	0.398	0.407	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.485	0.481
ETTm2	**0.272**	**0.317**	0.278	0.319	0.282	0.327	0.287	0.330	0.288	0.332	0.287	0.330	0.289	0.333	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.571	0.537
ETTh1	**0.432**	**0.434**	0.437	0.435	0.450	0.441	0.449	0.442	0.454	0.447	0.453	0.446	0.463	0.452	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.747	0.647
ETTh2	**0.359**	**0.391**	0.371	0.396	0.383	0.409	0.373	0.400	0.383	0.407	0.385	0.410	0.401	0.417	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.954	0.723
ECL	**0.162**	**0.257**	0.170	0.259	0.187	0.284	0.174	0.264	0.178	0.270	0.189	0.276	0.186	0.287	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.268	0.365
Traffic	0.447	0.278	0.421	0.279	0.478	0.326	**0.409**	**0.267**	0.428	0.282	0.454	0.286	0.522	0.357	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509
Weather	**0.237**	**0.265**	0.254	0.274	0.243	0.272	0.255	0.278	0.258	0.278	0.256	0.279	0.256	0.279	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.292	0.363
Solar	**0.222**	0.259	0.235	**0.253**	0.247	0.281	0.229	0.256	0.233	0.262	0.236	0.266	0.260	0.297	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401	0.282	0.375
PEMS03	**0.102**	**0.206**	0.160	0.260	0.132	0.240	0.104	0.210	0.113	0.221	0.137	0.240	0.119	0.233	0.169	0.281	0.326	0.419	0.147	0.248	0.278	0.375	0.114	0.224
PEMS04	0.093	**0.198**	0.216	0.302	0.136	0.249	0.102	0.208	0.111	0.221	0.145	0.249	0.103	0.215	0.209	0.314	0.353	0.437	0.129	0.241	0.295	0.388	**0.092**	0.202
PEMS07	**0.087**	**0.180**	0.168	0.261	0.128	0.233	**0.087**	0.184	0.101	0.204	0.144	0.233	0.112	0.217	0.235	0.315	0.380	0.440	0.124	0.225	0.329	0.395	0.119	0.234
PEMS08	**0.121**	**0.205**	0.217	0.288	0.179	0.268	0.138	0.219	0.150	0.226	0.200	0.275	0.165	0.261	0.268	0.307	0.441	0.464	0.193	0.271	0.379	0.416	0.158	0.244
1st Count	**10**	**10**	0	1	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 1: Multivariate time series forecasting results (averaged over 4 prediction horizons $T \in \{96, 192, 336, 720\}$, with a fixed look-back window of $L = 96$). FTP ranks first on 10 out of 12 datasets, significantly outperforming all baselines, including the most competitive patch-based MLP model, PatchMLP. All best results are highlighted in **bold**, and second-best ones are underlined, providing a clear comparison of model effectiveness. This highlights the strong and consistent performance of FTP.

Ablation Operation		w/o Dual-GLF		w/o GLF-CI		w/o GLF-CM		w/o CE		w/o Concat Original		w/o Fusion		Full FTP	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	Avg	0.379	0.403	0.379	0.403	0.382	0.404	0.385	0.405	0.407	0.424	0.380	0.403	**0.359**	**0.391**
	Range	-5.60%	-2.99%	-5.64%	-2.94%	-6.27%	-3.26%	-7.17%	-3.58%	-13.30%	-8.38%	-5.76%	-3.13%		
Weather	Avg	0.257	0.279	0.242	0.270	0.262	0.285	0.246	0.275	0.275	0.299	0.261	0.282	**0.237**	**0.266**
	Range	-8.59%	-4.79%	-2.00%	-1.41%	-10.55%	-7.14%	-3.80%	-3.29%	-15.93%	-12.31%	-10.26%	-5.86%		
Solar	Avg	0.241	0.275	0.240	0.277	0.236	0.270	0.260	0.293	0.290	0.309	0.253	0.284	**0.223**	**0.260**
	Range	-7.87%	-5.61%	-7.40%	-6.44%	-5.72%	-3.85%	-16.59%	-12.60%	-30.16%	-18.75%	-13.64%	-9.15%		
Electricity	Avg	0.178	0.268	0.166	0.261	0.177	0.267	0.179	0.270	0.247	0.332	0.191	0.275	**0.162**	**0.258**
	Range	-9.66%	-3.80%	-2.47%	-1.07%	-9.26%	-3.59%	-10.65%	-4.65%	-52.16%	-28.68%	-17.75%	-6.63%		
PEMS04	Avg	0.160	0.261	0.102	0.208	0.158	0.258	0.122	0.228	0.127	0.240	0.250	0.321	**0.094**	**0.199**
	Range	-70.27%	-31.25%	-8.24%	-4.27%	-67.55%	-29.77%	-30.05%	-14.45%	-34.84%	-20.48%	-166.17%	-61.35%		

Table 2: Ablation of FTP. We systematically remove individual components from FTP to assess their impact on performance. All results are averaged over four prediction horizons. "w/o" denotes the absence of a specific module.

Eliminating GLF-CI likewise degrades performance—most notably on PEMS04 and Solar—confirming the necessity of intra-channel dynamics and trend modeling. The CE module is especially critical for high-dimensional datasets such as PEMS04, Solar, and Electricity; the pronounced error growth highlights its ability to amplify key variables. Disabling Concat Original (residual injection of the raw input) consistently harms accuracy, stressing the stabilizing role of low-level features. Omitting Fusion inflicts the most severe overall decline—dramatic on short-horizon PEMS04—demonstrating that the tri-stream linear fusion is pivotal for precision. In sum, the six components perform complementary roles across datasets and horizons; their synergy is central to FTP's state-of-the-art performance.

Model Effectiveness Study

To further substantiate the practical gains of FTP's key mechanisms, we conduct additional evaluations. The look-back L and horizon T are 96. Each experiment is repeated five times and the mean is reported. Except where CE itself is examined, CE is disabled to avoid confounding effects.

Effectiveness of Dual-GLF. Pure Channel-Independent models excel at capturing intra-channel dynamics but overlook cross-channel synergy, whereas Channel-Mixed strategies expose latent dependencies at the risk of "over-fusion." FTP linearly fuses Global–Local features from both views, yielding complementary and robust representations. Figure 3 shows that using only GLF-CI or GLF-CM leads to dataset-dependent behavior. After linear fusion (Dual-GLF), the error is reduced on most datasets, mitigating underfitting (CI) and over-fusion (CM) simultaneously.

Effectiveness of Global–Local Fusion. Global–Local fusion enables joint modeling of long-term trends and short-term dynamics. The Global branch captures enduring temporal patterns, whereas the Local branch focuses on transient fluctuations and anomalies; the GLF module then in-

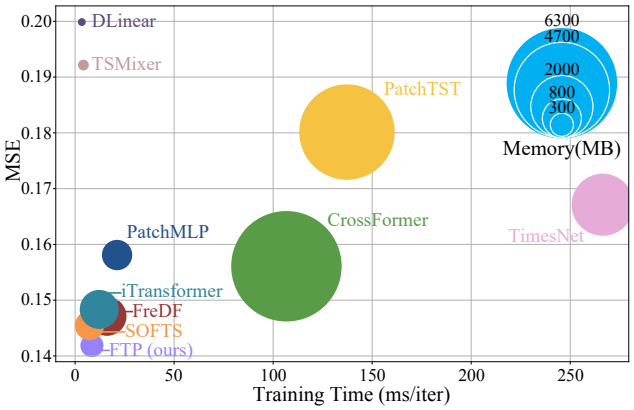

Figure 2: Efficiency–accuracy trade-off on Electricity (L = 96, T = 96, batch = 16). FTP sits at the bottom-left with a smaller bubble, indicating its strong accuracy and low cost.

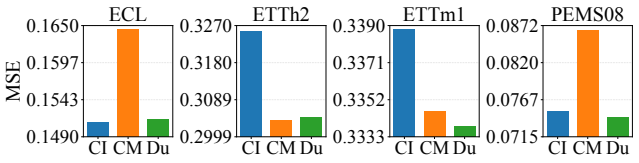

Figure 3: Impact of Dual-GLF variants on forecasting accuracy. MSE for the Channel–Independent branch (GLF-CI, CI), Channel–Mixed branch (GLF-CM, CM), and their fusion (Dual-GLF, Du). CI and CM alone give mixed results—each improves some datasets and harms others—whereas Dual-GLF unifies both views and generally lowers error across datasets.

tegrates the two. To ensure that both branches rely solely on raw time-series signals—avoiding deeper representations produced by recursive stacking—we set the $level$ = 1 (i.e., no recursion). We compare three variants: Global only, Local only, and Global–Local Fusion (GLF). As shown in Figure 4, GLF achieves the lowest error on ETTh2, ETTm2, and PEMS07, and ranks second on Weather, confirming the importance of coordinated Global–Local modeling.

Effectiveness of Multi-scale Recursive Levels. We vary Recursion Level from 1 to 14. Figure 5 reveals a clear positive correlation between depth and accuracy. At $level$ = 1, the model sees only a single-resolution patch—equivalent to conventional fixed-scale patching. With $level$ = 2, a second-scale local view is added. For $level$ = 3, each layer recursively merges preceding multi-scale local features with the current global context, progressively enlarging the receptive field. The resulting multi-resolution coverage strengthens characterization of periodic patterns and transients, confirming the benefit of adaptive multi-scale recursion.

Effectiveness of CE. In multivariate series, channel importance changes over time; a few dominant channels often drive key events. Ignoring this leads to diluted anomalies and larger errors. We compare the full model with a w/o CE variant. Figure 6 shows noticeable error drops when CE is enabled across most datasets, demonstrating its ability to

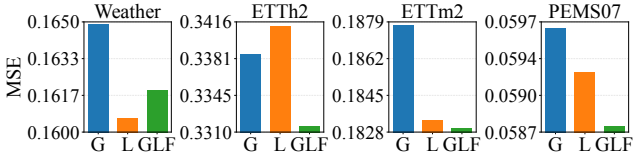

Figure 4: Global–Local fusion study. MSE of three variants—Global only (G), Local only (L), and their integration (GLF)—under $level$ = 1 (no recursion). Global and Local branches excel on different datasets, while GLF combines both and attains the lowest error on ETTh2, ETTm2, and PEMS07, and second-best on Weather, underscoring the value of coordinated Global–Local modeling.

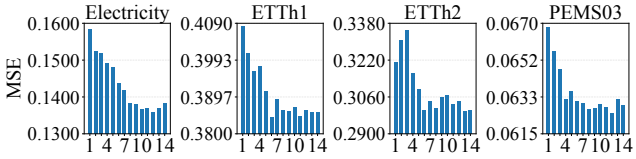

Figure 5: Impact of adaptive recursion level. MSE versus recursive level 1-14. Error drops steadily as the level increases, showing that deeper multi-scale recursion enlarges the receptive field and consistently improves forecasting accuracy.

dynamically amplify dominant channels, broadcast critical events, and improve robustness.

Multi-Scale vs. Single-Scale Patch Study. To assess the impact of multi-scale patching on long-horizon forecasting, we evaluate three FTP variants on the four ETT benchmarks: FTP-MS8 (multi-scale spans $\{6, 12, 18, ..., 48\}$, $level$ = 8, $unit$ = 6), FTP-SS8 (single scale = 27—the arithmetic mean of the multi-scale spans—with $level$ = 8), and FTP-SS1 (single scale = 27, $level$ = 1). As shown in Figure 7, averaged over five runs, FTP-MS8 achieves the lowest MSE in most settings and records the largest gains at the longest horizon (T = 1024). These results confirm that hierarchical multi-scale recursion—by enlarging the receptive field and promoting cross-scale information flow—substantially enhances long-range forecasting accuracy. Even without widening the view (single scale), a deep recursive variant (FTP-SS8) outperforms the conventional single-patch baseline (FTP-SS1); however, at the longest horizon (T = 1024) it still falls short of the multi-scale, multi-level design of FTP-MS8.

Impact of Look-Back Window Length. We evaluate FTP on the four ETT datasets with $L \in \{96, 336\}$ and horizons $T \in \{96, 192, 336, 720, 1024\}$, averaging five runs per setting. As shown in Figure 8, ETTh2 and ETTm1: A longer window (L = 336) consistently lowers MSE across all horizons, showing that richer context sharpens periodic and cross-channel cues. ETTh1 and ETTm2: Overall parity is observed, but at ultra-long horizons (T = 720,1024) the shorter window (L = 96) performs slightly better, indicating that FTP can distill salient signals from limited history and avoid noise accumulation. Hence, a longer window is not universally advantageous. FTP leverages extensive context

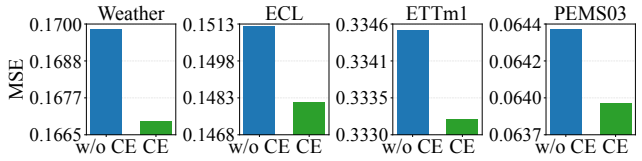

Figure 6: Impact of Channel Enhancement (CE). Bars compare MSE for the full model and its w/o-CE variant across datasets. Adding CE consistently lowers error, confirming that dynamically amplifying dominant channels reduces forecasting loss and enhances robustness.

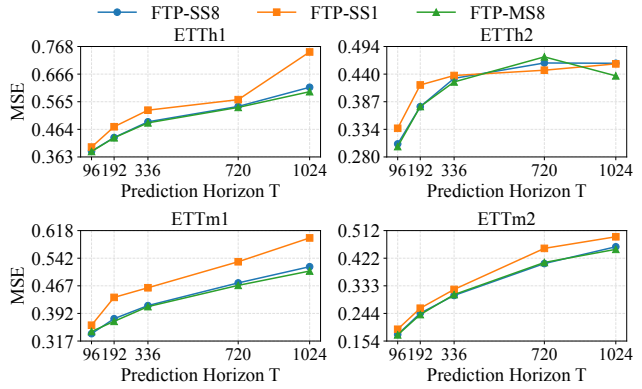

Figure 7: Multi-scale patch recursion vs. fixed-scale baselines. Average MSE on the four ETT datasets for FTP-MS8 (multi-scale, $level$ = 8), FTP-SS8 (single scale, $level$ = 8), and FTP-SS1 (single scale, $level$ = 1). The multi-scale variant ranks first in most settings and achieves the largest gain at the longest horizon (T = 1024), showing that hierarchical multi-scale recursion—by widening the receptive field and propagating cross-scale cues—significantly outperforms conventional fixed-patch designs for long-range forecasting.

when available yet still extracts key features from short sequences, preserving—or even improving—long-range accuracy. This highlights the model's high information efficiency and strong generalization.

Hyper-Parameter Sensitivity

To gauge FTP's robustness to key hyper-parameters, we perturb one setting at a time while keeping all others fixed. Unlike the common "96 → 96" setup, we adopt a more demanding "336 → 96" scenario: the extended look-back window amplifies performance differences, making the impact of each hyper-parameter easier to quantify.

Experimental setup. We systematically vary five key FTP hyper-parameters: Embedding dimension $d\_model \in \{128, \mathbf{256}, 384, 512, 768, 1024\}$; Encoder layer $e\_layers \in \{1, \mathbf{2}, 3, 4\}$; Unit patch length $unit \in \{1, 2, \mathbf{3} \ldots, 9\}$; Stride $stride \in \{2, 4, 6, \mathbf{8} \ldots, 16\}$; Recursive level $level \in \{2, 4, 6, 8, \mathbf{10}, \ldots, 18\}$. The default parameter value is highlighted in boldface.

Hyper-Parameter Sensitivity Results Analysis. From

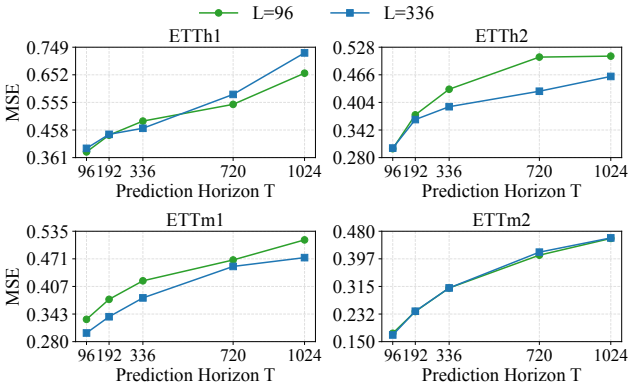

Figure 8: Effect of look-back length (L = 96 vs. L = 336). Average MSE over five runs on the four ETT datasets. A long window improves ETTh2 and ETTm1 at every horizon, but offers no advantage—and is slightly worse at $T \geq 720$ —on ETTh1 and ETTm2. Thus, richer context helps only where beneficial, while FTP can still distill key cues from short histories, underscoring its information efficiency and adaptability.

Figure 9, increasing $level$, e_layers, or d_model consistently lowers error, showing that greater representational capacity and deeper hierarchical recursion help capture complex temporal dependencies. Variations in $unit$ and $stride$ have only minor effects, indicating that FTP is relatively insensitive to input granularity and patch coverage. Overall, FTP maintains stable and controllable performance across a broad hyper-parameter range, confirming its reliability for real-world deployment and transfer scenarios.

Prediction Results Visualization

We visualize FTP's forecasts in Figure 10 (PEMS04, Electricity). Ground-truth trajectories are plotted in blue, FTP predictions in orange. The close overlap of the two curves demonstrates FTP's ability to track both short-term fluctuations and longer trends.

Conclusions

We present FusionTimePatch (FTP), a purely MLP-driven framework that unifies Channel–Independent and Channel–Mixed views, fuses Global–Local cues via recursive multi-scale patches, enhances the extraction of dominant information in the critical time steps of multiple channels, and integrates information with lightweight linear layers. On 12 public benchmarks FTP attains the best MSE and MAE on 10/12 datasets while exhibiting markedly lower memory use and time complexity than state-of-the-art methods. Ablation studies confirm the complementary roles of Dual-GLF, CE, and Linear Fusion in the overall gain. FTP offers a new paradigm that unifies CI–CM perspectives, Global–Local synergy, and patch-based recursion within an efficient MLP backbone—delivering accuracy, robustness, and scalability in one package. All code and scripts are publicly available on GitHub to facilitate research and applications.

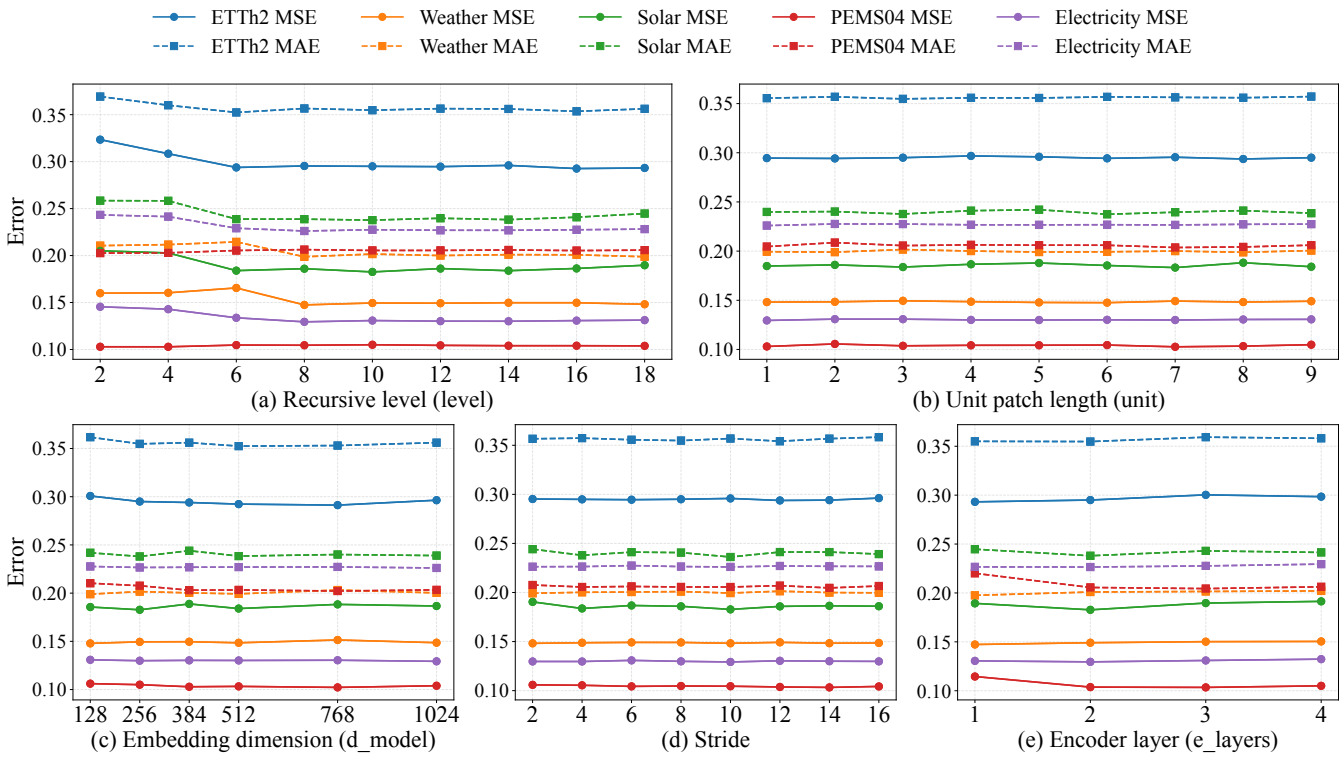

Figure 9: Hyper-parameter sensitivity of FTP. Error trends versus (a) Recursive level ($level$), (b) Unit patch length ($unit$), (c) Embedding dimension (d_model), (d) Stride, and (e) Encoder layer (e_layers). Increasing $level$, e_layers, or d_model steadily lowers MSE, whereas unit and stride have only minor effects, illustrating FTP's robustness across a wide hyper-parameter range.

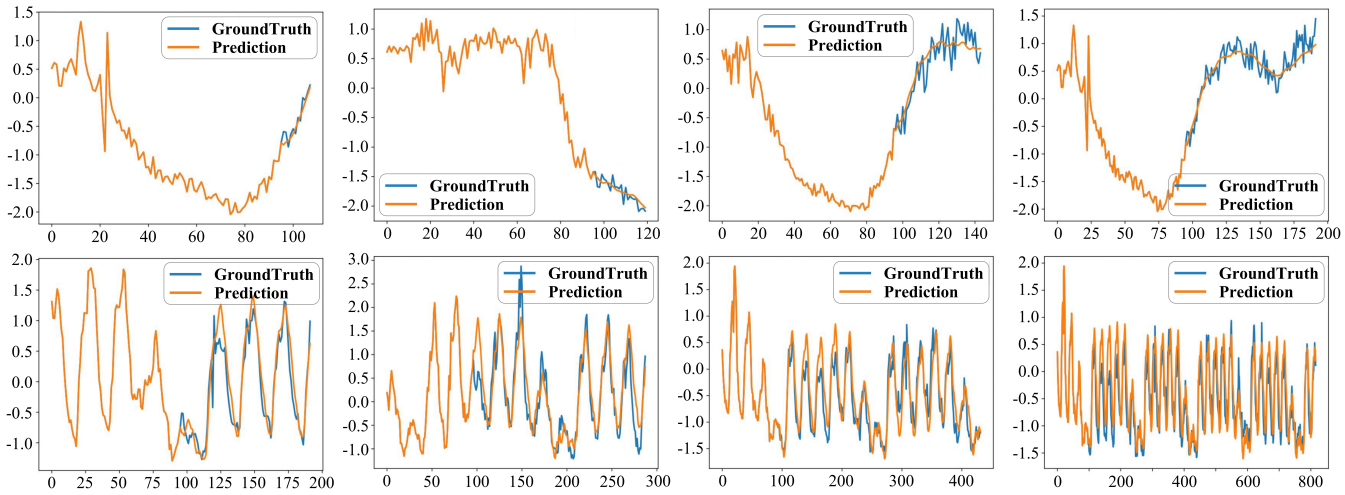

Figure 10: Forecast visualizations for FTP. Row 1: PEMS04 at horizons $T = \{12, 24, 48, 96\}$. Row 2: Electricity at horizons $T = \{96, 192, 336, 720\}$. Look-back length $L = 96$. The blue curve represents the ground truth, and the orange curve represents the prediction. It can be observed that the predicted values closely align with the ground truth, demonstrating the model's excellent forecasting performance.

Acknowledgments

This work was supported by the China Academy of Railway Sciences Corporation Limited (CARS) under Project No. RITS2023KF10.

References

Al-Musaylh, M. S.; Deo, R. C.; Adamowski, J. F.; and Li, Y. 2018. Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand

data in Queensland, Australia. *Advanced Engineering Informatics*, 35: 1–16.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Chen, P.; Zhang, Y.; Cheng, Y.; Shu, Y.; Wang, Y.; Wen, Q.; Yang, B.; and Guo, C. 2024. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. *arXiv preprint arXiv:2402.05956*.

Chen, S.; Xie, E.; Ge, C.; Chen, R.; Liang, D.; and Luo, P. 2021. Cyclemlp: A mlp-like architecture for dense prediction. arXiv 2021. *arXiv preprint arXiv:2107.10224*.

Chen, Y.; Segovia, I.; and Gel, Y. R. 2021. Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In *International Conference on Machine Learning*, 1684–1694. PMLR.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cirstea, R.-G.; Guo, C.; Yang, B.; Kieu, T.; Dong, X.; and Pan, S. 2022. Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting–Full Version. *arXiv preprint arXiv:2204.13767*.

Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Trans. Mach. Learn. Res.*, 2023.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ekambaram, V.; Jati, A.; Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 459–469.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*.

Elman, J. L. 1990. Finding structure in time. *Cognitive science*, 14(2): 179–211.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32.

Fukushima, K. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4): 193–202.

Han, L.; Ye, H.-J.; and Zhan, D.-C. 2024. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Jia, Y.; Lin, Y.; Hao, X.; Lin, Y.; Guo, S.; and Wan, H. 2023. WITRAN: Water-wave Information Transmission and Recurrent Acceleration Network for Long-range Time Series Forecasting. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jia, Y.; Lin, Y.; Yu, J.; Wang, S.; Liu, T.; and Wan, H. 2024. PGN: The RNN's New Successor is Effective for Long-Range Time Series Forecasting. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.; Liang, Y.; Li, Y.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kadiyala, A.; and Kumar, A. 2014. Multivariate time series models for prediction of air quality inside a public transportation bus using available software. *Environmental Progress & Sustainable Energy*, 33(2): 337–341.

Kalyan, K. S.; Rajasekharan, A.; and Sangeetha, S. 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 2002. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

Li, Z.; Rao, Z.; Pan, L.; and Xu, Z. 2023. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022a. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2022b. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lu, H.; Chen, X.-Y.; Ye, H.-J.; and Zhan, D.-C. 2024. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175.

Morid, M. A.; Sheng, O. R. L.; and Dunbar, J. 2023. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1): 1–29.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Papadimitriou, S.; and Yu, P. 2006. Optimal multi-scale patterns in time series streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 647–658.

Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2025. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1185–1196.

Rangapuram, S. S.; Seeger, M. W.; Gasthaus, J.; Stella, L.; Wang, Y.; and Januschowski, T. 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31.

Tang, P.; and Zhang, W. 2025. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12640–12648.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. 2022. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE transactions on pattern analysis and machine intelligence*, 45(4): 5314–5321.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Pan, L.; Shen, Y.; Chen, Z.; Yang, D.; Yang, Y.; Zhang, S.; Liu, X.; Li, H.; and Tao, D. 2025. FreDF: Learning to Forecast in the Frequency Domain. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; and Zhao, J. L. 2014. Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management*, 298–310. Springer.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 27268–27286. PMLR.

Zhu, Y.; and Shasha, D. 2002. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, 358–369. Elsevier.