

# Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting

Peiwan Tang<sup>1,2</sup>, Weitai Zhang<sup>1,2</sup>

<sup>1</sup>IFLYTEK Research, Hefei, China

<sup>2</sup>University of Science and Technology of China, Hefei, China  
tpw@mail.ustc.edu.cn, zwt2021@mail.ustc.edu.cn

## Abstract

Recent studies have attempted to refine the Transformer architecture to demonstrate its effectiveness in Long-Term Time Series Forecasting (LTSF) tasks. Despite surpassing many linear forecasting models with ever-improving performance, we remain skeptical of Transformers as a solution for LTSF. We attribute the effectiveness of these models largely to the adopted Patch mechanism, which enhances sequence locality to an extent yet fails to fully address the loss of temporal information inherent to the permutation-invariant self-attention mechanism. Further investigation suggests that simple linear layers augmented with the Patch mechanism may outperform complex Transformer-based LTSF models. Moreover, diverging from models that use channel independence, our research underscores the importance of cross-variable interactions in enhancing the performance of multivariate time series forecasting. The interaction information between variables is highly valuable but has been misapplied in past studies, leading to suboptimal cross-variable models. Based on these insights, we propose a novel and simple Patch-based MLP (PatchMLP) for LTSF tasks. Specifically, we employ simple moving averages to extract smooth components and noise-containing residuals from time series data, engaging in semantic information interchange through channel mixing and specializing in random noise with channel independence processing. The PatchMLP model consistently achieves state-of-the-art results on several real-world datasets. We hope this surprising finding will spur new research directions in the LTSF field and pave the way for more efficient and concise solutions. Code is available at: <https://github.com/TangPeiwan/PatchMLP>

## Introduction

Long-term Time Series Forecasting (LTSF) is a critical area of research in the fields of statistics and machine learning, aimed at using historical data to predict the future of one or more variables over a certain period (Oreshkin et al. 2019; Gong, Tang, and Liang 2023; Lin et al. 2023). Time series data are organized in chronological order and reveal underlying dynamic patterns, be they cyclical or non-cyclical (Cryer 1986). Such forecasts play a vital role in various sectors, including biomedicine (Liu et al. 2018), economics and

finance (Patton 2013), electricity (Zhou et al. 2021), and transportation (Yin and Shang 2016).

Multivariate Time Series (MTS) consist of multiple variables recorded at the same time point, where each dimension may represent an individual univariate time series or be considered as a signal with multiple channels. With the advancement of deep learning, a number of models have been developed to enhance the performance of MTS forecasting (Borovykh, Bohte, and Oosterlee 2017; Bai, Kolter, and Koltun 2018; Liu et al. 2020). In particular, recent models based on the Transformer (Vaswani et al. 2017) architecture have demonstrated significant potential in capturing long-term dependencies (Radford et al. 2018; Devlin et al. 2018). In recent years, Transformers have emerged as a leading architecture in time series forecasting (Lim et al. 2021; Liu et al. 2022b; Shen et al. 2023), initially applied in the field of Natural Language Processing (NLP) (Radford et al. 2019; Brown et al. 2020) and subsequently expanded to other domains such as Computer Vision (CV) (Liu et al. 2021b; Feichtenhofer et al. 2022) through the adaptation of the **Patch** method, becoming a versatile framework.

Originally, these models utilized a **channel mixing** approach (Li et al. 2023b), which involves projecting vectors from different channels recorded at the same time point into an embedding space and blending the information (Zhou et al. 2021; Tang and Zhang 2023), and many models adopt the idea of time series decomposition (Wu et al. 2021; Zhou et al. 2022; Wang et al. 2023). Nonetheless, recent studies have shown that a **channel independence** model might be more effective (Nie et al. 2022; Han, Ye, and Zhan 2023). Channel independence means that each input token contains information from only one channel, and intuitively, the models treat the MTS as separate univariate series for individual processing. These studies suggest that for LTSF tasks, an emphasis on channel independence may be more effective than channel mixing strategies (Gong, Tang, and Liang 2023).

Moreover, the effectiveness of Transformers in LTSF tasks has been challenged by a reconsideration of the Multi-Layer Perceptron (MLP) (Zeng et al. 2023; Ekambaram et al. 2023; Chen et al. 2023), whose surprisingly simple architecture has outperformed all previous Transformer models. This raises a compelling question: Are Transformers effective for LTSF tasks? In response to this skepticism, recent

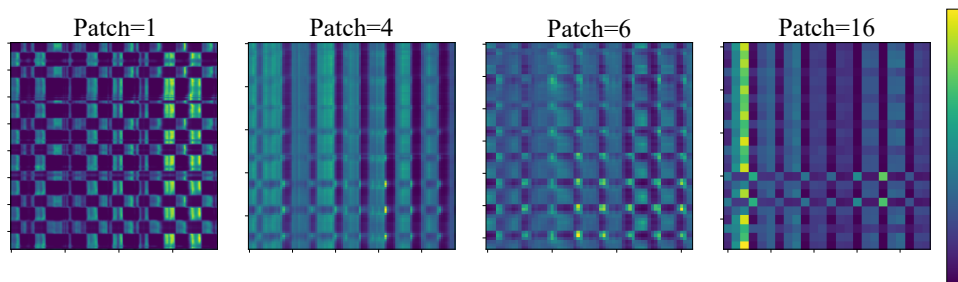


Figure 1: The self-attention scores of a 2-layer Transformer with different Patch size trained on ETTh1. We follow the setup of PathcTST (Nie et al. 2022), retaining only the Encoder while replacing the Decoder with a simple MLP, and using a channel independent approach. A patch size of 1 is equivalent to the original Transformer, indicating that time series data often exhibits a trend of being segmented into patches (Zhang and Yan 2022; Tang and Zhang 2023), and an increase in patch size can mitigate this to some extent.

Transformer-based models have adopted patch-based representations and achieved noteworthy performance in LTSF (Nie et al. 2022; Chen et al. 2024).

This study raises three critical inquiries:

- Is the method of channel mixing truly ineffective in MTS forecasting?
- Can simply decomposing the original time series truly better predict the trend and seasonal components?
- Does the impressive performance of the Patch-based Transformer derive from the inherent strength of the Transformer architecture, or is it merely due to the use of patches as the input representation?

In this paper, we specifically discuss the three issues mentioned above and introduce the PatchMLP, a novel and concise Patch-based MLP model tailored for LTSF task. PatchMLP is entirely based on fully connected networks and incorporates the concept of patch. Specifically, we employ a patch methodology to embed the original sequences into representational spaces, then extract the smooth components and noisy residuals of the series using a simple moving average technique for separate processing. We independently process stochastic noise across channels and facilitate semantic information interchange between variables through channel mixing. Upon evaluation on numerous real-world datasets, the PatchMLP model demonstrated state-of-the-art (SOTA) performance. To be precise, our contributions can be summarized in three aspects:

- We analyze the effectiveness of Patches in time series forecasting and propose a Multi-Scale Patch Embedding (MPE) approach. Unlike previous embedding methods that used a single linear layer, MPE is capable of capturing multiscale relationships between input sequences.
- We introduce a novel entirely MLP-based model, named PatchMLP. By utilizing moving averages, it performs the decomposition of latent vector and adopts a different approach to channel mixing for semantic information exchange across variables.
- Our experiments across a wide range of datasets in various fields show that PatchMLP consistently achieves

SOTA performance across multiple forecasting benchmarks. Moreover, we conduct an extensive analysis of Patch-based methods with the aim of charting new directions for future research in time series forecasting.

## Patch for Long-Term Time Series Forecasting

**Why is Patch effective in time series forecasting?** (Lee, Park, and Lee 2023; Zhong et al. 2023) We have studied a patch-based Transformer model adopting channel independent mechanisms, and we set the decoder layer as a simple single-layer MLP for time series forecastine (Nie et al. 2022). As shown in the Figure 2b, we can clearly see that for the same input length, as the size of the patch increases, the overall Mean Squared Error (MSE) of the model exhibits a trend of decreasing and then increasing, or decreasing and then stabilizing. As the input length increases, the patch size required for the model to achieve optimal performance also gradually increases. In the original input data, the effect of Attention is so poor that it cannot even outperform a simple single-layer MLP, which raises the question of whether Attention might not be the optimal choice for time series modeling to some extent, as it cannot handle long-term time series well.

Compared to textual data in the NLP field, original time series data contains many redundant features due to high-frequency sampling, and is easily influenced by noise to some extent during the sampling process (Tang and Zhang 2022). We believe that for data with excessive noise in the original data, the original Transformer’s Attention mechanism is not the optimal choice as it cannot effectively eliminate the noise. Sparse Attention performs better because this sparse mechanism can effectively mitigate the impact of noise to some extent (Li et al. 2019; Wu et al. 2020; Liu et al. 2021a), but it may also attenuate the impact of original features.

Patch, on the other hand, compresses the data, reduces the dimensionality of the input data, and decreases redundant features. Additionally, patch provides a certain degree of smoothing, which can reduce the influence of outliers to some extent, and help filter out fluctuations and random noise, retaining more stable and representative information.

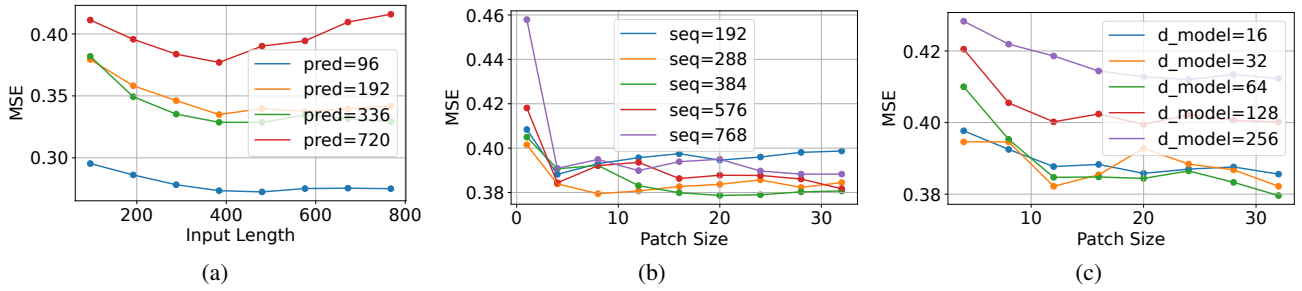


Figure 2: Experimental results of Patch Transformer on the ETTh2 dataset. (a) Maintain all other parameters constant, and present the MSE outcomes for four forecast lengths with only the input length altered. (b) Keeping all other parameters constant and only altering the size of the patch, the MSE results for a forecast length of 720 with five different input lengths. (c) Maintaining all other parameters unchanged and only varying the patch size, the MSE results for five different  $d_{model}$  values with both input and forecast lengths set to 720.

In the CV field, patch also works well in ViT (Dosovitskiy et al. 2020), MAE (He et al. 2022). Time series data usually contains patterns at different scales, and patch provides a modeling of short-term time series, enhancing local information in the sequence, allowing the model to better learn and capture local features. Therefore, we believe that the effectiveness of Transformer is not due to the effect of Attention, but rather due to the presence of patch.

So, is a larger patch always better? In Figure 2b, a larger patch does not seem to achieve better results, which may be partly influenced by the size of the hidden layer ( $d_{model}$ ). With a larger patch size, more time points are projected into a fixed-size  $d_{model}$ , which may lead to excessive compression. As shown in Figure 2c, as  $d_{model}$  increases, the performance of the model gradually improves, indicating that larger patches perform better. In the extreme case, a patch that treats the entire input sequence as one patch and projects it into a vector is similar to iTransformer (Liu et al. 2023). However, a larger  $d_{model}$  indicates that there are more parameters for the model to learn, which can easily lead to underfitting and result in decreased model performance.

### PatchMLP

The problem of multivariate time series forecasting is to input the historical observations  $\mathcal{X} = \{x_1, \dots, x_L | x_i \in \mathbb{R}^M\}$ , and the output is to predict corresponding future sequence  $\mathcal{X}' = \{x_{L+1}, \dots, x_{L+T} | x_i \in \mathbb{R}^M\}$ , where  $L$  and  $T$  are the lengths of input and output sequences respectively, and  $M$  is the dimension of variates.

As shown in Figure 3, PatchMLP consists of four network components: Multi-Scale Patch Embedding layer, Feature Decomposition layer, Multi-Layer Perceptron (MLP) layer, Projection layer. The Multi-Scale Patch Embedding layer embeds the multivariate time series into latent space. The Feature Decomposition layer decomposes the latent vector into the smooth components and noisy residuals, then operate separately through MLP layer. Finally, the latent vector are mapped back to the feature space through the Projection layer to obtain the future sequences  $\hat{\mathcal{X}}$ . Next, we will introduce the above modules separately.

### Multi-Scale Patch Embedding

Time series analysis relies on the accurate identification of local information within the sequence and optimization of the model. The use of patches can provide the model with a local view of the time series in the short term, thereby enhancing the representation of local information within the sequence and enabling the model to learn and capture these local features more precisely. However, traditional methods typically employ patches of a single-scale to embed the raw time series, which makes the model inclined to learn single-scale temporal relationships while neglecting the multi-scale nature and complexity of time series data.

In practical applications, the single-scale patch strategy may lead to the model capturing inaccurate or incomplete local features. This is because the data characteristics of different time series often have variation, and a patch of the same scale cannot universally adapt to all types of time series. For example, a time series containing multiple cyclical patterns, a single-scale patch cannot effectively identify and learn the cyclical features present at different frequencies.

To capture local information in more detail and to fully understand the temporal relationships within the time series, we adopt multi-scale patch. This includes shorter patch to capture local high-frequency patterns, as well as longer patch to unearth long-term seasonal, trends and periodic fluctuations. Through the introduction of multi-scale patch, the model can flexibly learn representative features over different lengths of time spans, thus enhancing predictive accuracy and model generalization capabilities.

To decompose the multivariate time series  $\mathcal{X}$  into a univariate series  $x$ , we have developed a suite of patches  $\mathcal{P}$  across various scales to process  $x$ . For a particular scale  $p \in \mathcal{P}$ ,  $x$  is first divided into non-overlapping patches where the length of patch corresponds to  $p$ , thus the patching process yields the sequence of patches  $x_p \in \mathbb{R}^{N \times p}$ , where  $N$  is the number of patches. Subsequently, we employ a single-layer linear layer to embed the patches  $x_p$ , resulting in latent vectors  $x_e \in \mathbb{R}^{N \times d}$ ,  $d$  is the embedding dimension, the  $d$  corresponding to different scales patches can be different. These latent vectors are then unfolded to obtain the final embedding vectors  $X \in \mathbb{R}^{1 \times d_{model}}$ ,  $d_{model}$  is the final embed-

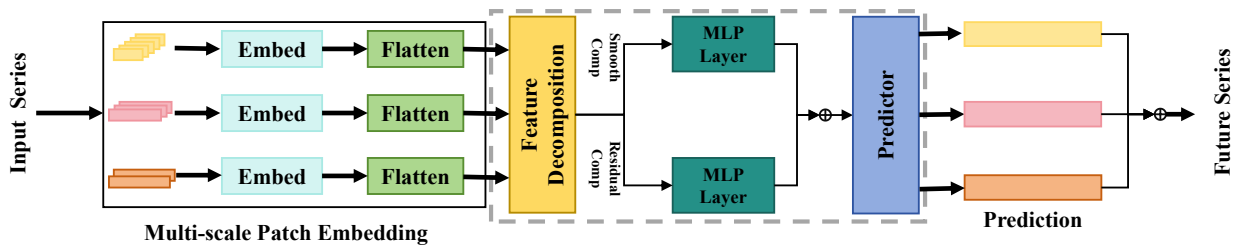


Figure 3: Overall structure of PatchMLP. First, the raw time series of different variables are independently processed through Multi-scale Patch Embedding. Then, Feature Decomposition uses moving averages to decompose the embedded tokens into smooth components and noisy residues. Next, a MLP processes the sequences in two ways: intra-variable and inter-variable. Finally, the Predictor maps the latent vectors back to predictions and aggregates them into future series.

ding dimension input into the model. This multi-scale patch strategy permits the model to capture and learn the intrinsic dynamics of time series at different levels, thereby providing more precise insights when forecasting future trends and patterns.

### Feature Decomposition

Time series often contain complex temporal patterns, including seasonal fluctuations, trends, and other irregular influencing factors. Many studies attempt to decompose time series into these fundamental components through decomposition methods, to achieve better understanding and forecasting. Although decomposition is a powerful tool, traditional methods may encounter some difficulties when dealing with the raw sequences, especially when there are complex and mixed patterns in the sequences. These methods often struggle to accurately separate clear trends and seasonal components, particularly when the data contains substantial noise or nonlinear components.

In response to this issue, we propose a different idea, instead of directly decomposing the original sequence, we decompose the sequence’s embedding vector. Embedding vector are representations formed by mapping time series to a high-dimensional space, and these representations often capture the core information of the original data. By decomposing embedding vector, we can distinguish smoother components and noise-containing residuals, which helps eliminate the interference of random fluctuations on analysis and forecasting. In practice, we use the operation of Average Pooling (*AvgPool*) to smooth the time series, aiming to reduce random fluctuations and noise in the data. Moreover, in order to maintain the length of the time series unchanged while smoothing, we applied a padding operation.

$$\begin{aligned} X_s &= AvgPool(X) \\ X_r &= X - X_s \end{aligned} \quad (1)$$

$X_s$  and  $X_r$ , respectively represent the extracted smooth components and residual components. By operating on embedding vector rather than the original sequence, the model can extract and identify the fundamental components of time series more precisely, enabling a better understanding of the intrinsic structure of time series, and ultimately, improving the accuracy of time series forecasting.

### MLP layer

In the context of MTS forecasting, the MLP layer alternately applies MLPs within Intra-Variable (time domain) and Inter-Variable(feature domain) to enhance predictive performance. The specific architecture, as depicted in Figure 4, can be described in detail as follows:

**Intra-Variable MLP:** This component is focused on the identification of time-correlated patterns within time series. Specifically, it utilizes a network architecture composed of fully connected layers, nonlinear activation functions, and dropout. Parameters are applied in the temporal domain and are shared across inter-variable. Simple linear models are easy to understand and implement, and have been proven to learn complex time patterns and capture more complex time dependencies.

**Inter-Variable MLP:** Complementary to the intra-variable MLP, the inter-variable MLP aims to model the mutual influences among MTS variables. This component also consists of fully connected layers, activation functions, and dropout, applying fully connected layers in the feature domain to achieve parameter sharing within the intra-variable. To enhance the cross-variable interactivity, we utilize a dot product mechanism, integrating the dot product results between the MLP outputs and inputs, which enhances the model’s nonlinear representational capability.

**Residual Connections:** Residual connections (He et al. 2016) are applied after each MLP, enabling the model to learn deeper architectures more efficiently. These connections not only help to mitigate the problems of vanishing gradients in deep networks but also provide the model with a shortcut path that ensures key temporal or feature information is not overlooked.

While the minimalist architecture of PatchMLP may not appear as eye-catching as some of the recently proposed, more complex Transformer models, empirical results indicate that PatchMLP exhibits competitive performance, both in terms of training efficiency and accuracy on standard benchmark tests, relative to SOTA models.

### Loss Function

Our loss function is calculated by the Mean Square Error (MSE) between the model prediction  $\hat{x}$  and the ground truth  $x$ :  $\mathcal{L} = \frac{1}{M} \sum_{M}^{i=1} \|\hat{x}_{L+1:L+T}^i - x_{L+1:L+T}^i\|_2^2$  and the loss is

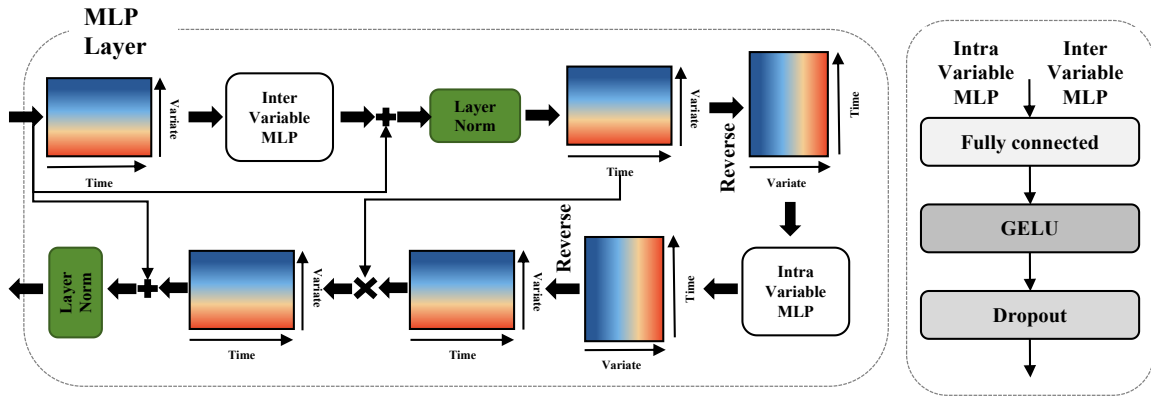


Figure 4: Overall structure of MLP layer. The embedded vectors first interact with the temporal information within the variable through the Intra-Variable MLP. Then interact with the feature domain information between variables through the Intra-Variable MLP. Subsequently, they are multiplied by the input of the Inter-Variable MLP using a dot-product approach. Finally, they are added to the initial input of the MLP Layer using skip connections.

propagated back from the Projection output across the entire model.

## Experiments

We comprehensively evaluated the proposed PatchMLP in various time series forecasting applications, verifying the generality of the proposed framework, and have further delved into the investigation of the effectiveness of individual components of PatchMLP when applied in LSTF tasks.

**Datasets:** We evaluated the performance of our proposed PatchMLP on 8 commonly used LSTF benchmark datasets: Solar Energy (Lai et al. 2018), Weather, Traffic, Electricity (ECL) (Wu et al. 2021), and 4 ETT datasets (ETT<sub>h1</sub>, ETT<sub>h2</sub>, ETT<sub>m1</sub>, ETT<sub>m2</sub>) (Zhou et al. 2021).

**Baselines and Metrics:** We selected nine widely acknowledged SOTA forecasting models for our benchmarking analysis, which comprises the Transformer-based models: iTransformer (Liu et al. 2023), PatchTST (Nie et al. 2022), Crossformer (Zhang and Yan 2022), and FEDformer (Zhou et al. 2022); the CNN-based models: TimeNet (Wu et al. 2022), SCINet (Liu et al. 2022a); and the significant MLP-based models: Timemixer (Wang et al. 2024), DLinear (Zeng et al. 2023), TiDE (Das et al. 2023), and RLinear (Li et al. 2023a). To evaluate the performance of these models, we used widely used evaluation metrics: MSE and Mean Absolute Error (MAE).

### Multivariate Long-Term Forecasting

Table 1 presents the results of MTS forecasting, with lower MSE/MAE values indicating greater predictive accuracy. Overall, the model we propose has achieved a surprisingly efficacious outcome, realizing optimal performance across all datasets. Out of a total of 16 benchmarks, we achieved 100% state-of-the-art (SOTA) results, surpassing all Transformer architectures. This suggests that the Attention mechanism may not necessarily be the optimal choice for LSTF tasks, as even simple linear models are capable of achieving impressive results. Further analysis indicates that chan-

nel independence models, represented by PatchTST, often failed to perform to their potential, thus implying the indispensable role of inter-variable interactions in MTS analysis. Notably, despite explicitly capturing multivariate correlations, the performance of Crossformer did not meet expectations, underscoring that inappropriate utilization of inter-variable correlations can negatively impact model efficacy. Hence, we conclude that skillful leverage of interactions between variables is vital and, in certain scenarios, simple linear models may suffice to deliver outstanding performance compared to the more complex Transformer architectures.

### Model Analysis

**Ablation Study** As shown in the Table 2, we investigate the impact of each module of PatchMLP on performance. ⑨ follows the conventional decomposition approach, which involves decomposing the time series first and then embedding it before separately handling the prediction of the trend and seasonal components. ⑤ eliminates the decomposition module, directly inputs the embedded original sequence into the model for prediction. In both ② and ⑥, the MPE is removed, and the embedding method is altered to a single linear layer. ③ and ⑦, on the other hand, eliminate the dot product in the MLP among variables and replace it with a simple residual connection, while ④ and ⑧ completely remove the MLP among variables, canceling the interaction between them. We only change the corresponding modules and keep all other settings unchanged, with the input length still set to 96.

It can be observed that conventional decomposition methods do not perform well, which is attributable to the excessively complex temporal relationships inherent in the original time series that cannot be effectively deconstructed through simple decomposition techniques. In contrast, latent vector decomposition can adeptly circumvent this issue. Furthermore, the observed performance deterioration upon the removal of the MPE indicates that MPE plays a significant role in learning the various temporal relationships. Additionally, it is noteworthy that the dot product

Methods	PatchMLP	TimeMixer	iTransformer	RLinear	PatchTST	Crossformer	TiDE	TimesNet	DLinear	SCINet	FEDformer
Metric	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE
ETTm1	<b>0.374 0.382</b>	0.381 0.395	0.407 0.410	0.414 0.407	0.387 0.400	0.513 0.496	0.419 0.419	0.400 0.406	0.403 0.407	0.485 0.481	0.448 0.452
ETTm2	<b>0.269 0.311</b>	0.275 0.323	0.288 0.332	0.286 0.327	0.281 0.326	0.757 0.610	0.358 0.404	0.291 0.333	0.350 0.401	0.571 0.537	0.305 0.349
ETTh1	<b>0.438 0.429</b>	0.447 0.440	0.454 0.447	0.446 0.434	0.469 0.454	0.529 0.552	0.541 0.507	0.458 0.450	0.456 0.452	0.747 0.647	0.440 0.460
ETTh2	<b>0.349 0.378</b>	0.364 0.395	0.383 0.407	0.374 0.398	0.387 0.407	0.942 0.684	0.611 0.550	0.414 0.427	0.559 0.515	0.954 0.723	0.437 0.449
ECL	<b>0.171 0.265</b>	0.182 0.272	0.178 0.270	0.219 0.298	0.216 0.304	0.244 0.334	0.251 0.344	0.192 0.295	0.212 0.300	0.268 0.365	0.214 0.327
Traffic	<b>0.417 0.273</b>	0.484 0.297	0.428 0.282	0.626 0.378	0.555 0.362	0.550 0.304	0.760 0.473	0.620 0.336	0.625 0.383	0.804 0.509	0.610 0.376
Weather	<b>0.231 0.256</b>	0.240 0.271	0.258 0.279	0.272 0.291	0.259 0.281	0.259 0.315	0.271 0.320	0.259 0.287	0.265 0.317	0.292 0.363	0.309 0.360
Solar-Energy	<b>0.211 0.261</b>	0.216 0.280	0.233 0.262	0.369 0.270	0.307 0.641	0.6398 0.347	0.417 0.301	0.319 0.330	0.401 0.282	0.375 0.281	0.291 0.381
1 <sup>st</sup> Count	<b>8 8</b>	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

Table 1: Results of the Long-Term Time Series Forecasting task. For all baselines, we adhere to the setting of the iTransformer with an input sequence length of 96. We conducted comparisons among an array of competitive models during various forecast horizons. All the results are averaged from 4 different prediction lengths, that is {96, 192, 336, 720}. A lower MSE or MAE indicates a better prediction, we denote the optimal results with boldface for clarity.

Case	Decompose	MPE	Dot Product	Intra Variable	ECL MSE MAE	Traffic MSE MAE	Solar-Energy MSE MAE	Weather MSE MAE	ETTm1 MSE MAE
①	✓	✓	✓	✓	<b>0.171 0.265</b>	<b>0.417 0.273</b>	<b>0.211 0.277</b>	<b>0.231 0.256</b>	<b>0.374 0.382</b>
②	✓	✗	✓	✓	0.183 0.279	0.431 0.282	0.223 0.288	0.241 0.270	0.383 0.395
③	✓	✓	✗	✓	0.177 0.271	0.426 0.280	0.219 0.283	0.237 0.262	0.383 0.387
④	✓	✓	✓	✗	0.179 0.276	0.426 0.284	0.218 0.287	0.242 0.266	0.382 0.393
⑤	✗	✓	✓	✓	0.186 0.275	0.429 0.287	0.223 0.288	0.243 0.266	0.384 0.390
⑥	✗	✗	✓	✓	0.198 0.284	0.442 0.298	0.233 0.301	0.255 0.274	0.396 0.402
⑦	✗	✓	✗	✓	0.193 0.280	0.435 0.295	0.230 0.293	0.252 0.275	0.390 0.396
⑧	✗	✓	✓	✗	0.194 0.284	0.439 0.296	0.234 0.301	0.253 0.274	0.395 0.398
⑨	✓	✓	✓	✓	0.180 0.275	0.429 0.281	0.219 0.287	0.240 0.265	0.382 0.390

Table 2: Ablations of PatchMLP. We replace the components of PatchMLP one by one and explore the performance of different MLPs. All the results are averaged from 4 different prediction lengths. A check mark ✓ and a wrong mark ✗ indicate with and without certain components respectively.

method outperforms simply because it enhances the inter-activeness between variables, while mere addition does not confer this advantage. Correspondingly, models that lack variable interactiveness experience a predictable decline in performance, which reiterates the critical importance of appropriately leveraging the interrelationships between variables.

**Increasing look-back Window** In principle, a longer input sequence might enhance model performance, primarily due to the provision of a richer historical context that assists the model in more effectively learning and identifying long-term patterns within time series. Generally, a robust LSTF model, equipped with substantial capability to extract temporal relationships, is expected to yield improved outcomes with extended input lengths. However, longer inputs necessitate the model’s heightened ability to capture long-term dependencies, as a deficit in this capacity could readily precipitate a decline in model performance.

As illustrated in Figure 5, there is a gradual augmentation in the performance of all models with increasing input length, but as the input extends to considerable lengths (768), some models begin to exhibit diminished performance, which may also be attributed to the amplified noise accompanying the expanded input. In contrast, our model

and DLinear consistently demonstrate steady improvements in performance as the input lengthens, exemplifying the superiority of linear models in this context.

### Hyperparameter Sensitivity

We evaluated the hypersensitivity of the PatchMLP concerning the following parameters: learning rate (lr), the number of blocks (layers) N in the MLP, and the hidden dimension D of the embedding. The results are illustrated in Figure 6. We observed that the performance of PatchMLP is not particularly sensitive to these hyperparameters, as evidenced by the relatively inconspicuous variations in performance. It is advisable that both the number of blocks and the hidden dimension size not be excessively large, which may be attributable to the increase in the total number of parameters that require learning with larger values of these hyperparameters. However, the hidden dimension size should not be excessively small either, to avoid a disproportionate compression of features into an insufficiently capacious latent vector.

### Conclusions

This paper provides an in-depth exploration of existing solutions for Long-Term Time Series Forecasting (LTSF) tasks

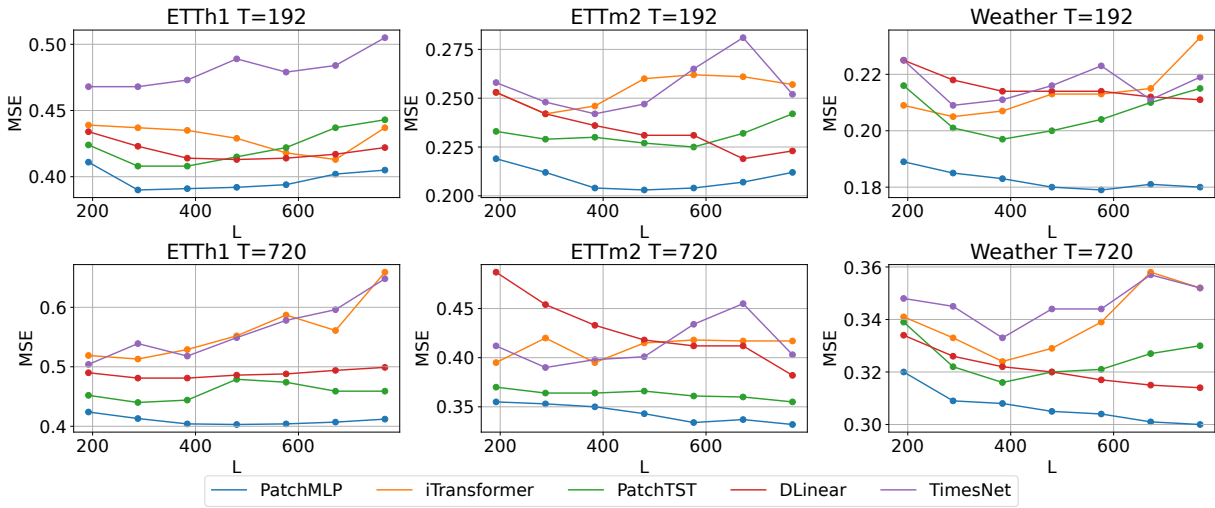


Figure 5: Forecasting performance (MSE) with varying look-back windows on 3 datasets: ETTh1, ETTm2, and Weather. The look-back windows are selected to be  $L = \{192, 288, 384, 480, 576, 672, 768\}$ , and the prediction horizons are  $T = \{192, 720\}$ .

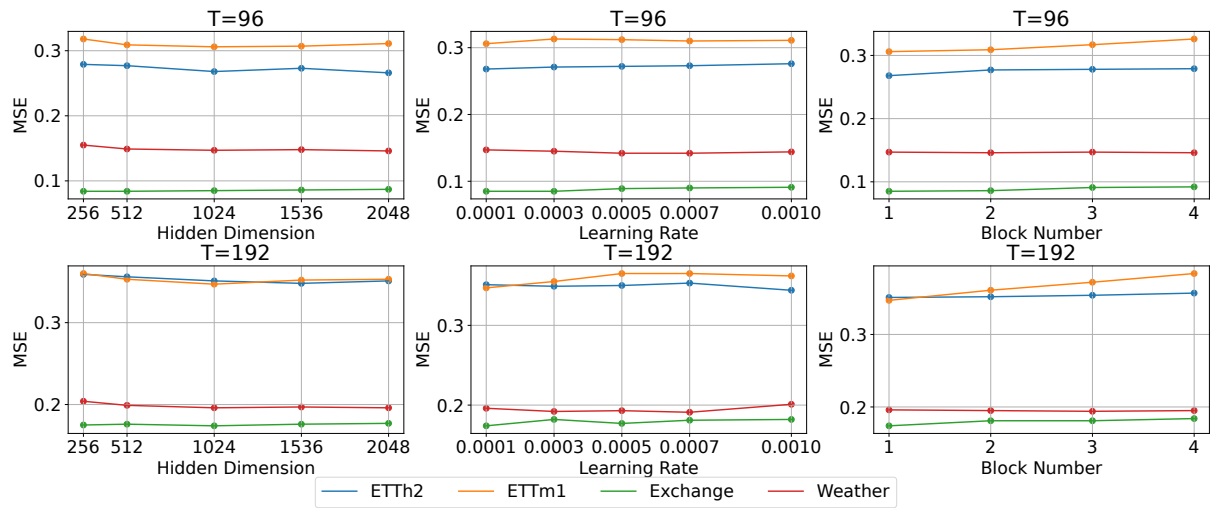


Figure 6: Hyperparameter sensitivity with respect to the learning rate, the number of MLP blocks, and the hidden dimension of Embedding tokens. The results are recorded with the lookback window length  $L = 96$  and the forecast window length  $T = \{96, 192\}$

within the framework of the Transformer architecture and introduces an innovative approach: Patch-based Multi-Layer Perceptron (PatchMLP). Through empirical analysis, we demonstrate the efficacy of the patch mechanism in time series forecasting and address the current limitations of Transformer models by designing a novel, simplified architecture. The PatchMLP utilizes a simple moving average to separate the smooth components from the noisy residuals in time series, and employs a unique channel mixing strategy to enhance the interchange of semantic information across variables. Additionally, we present the Multi-Scale Patch Embedding method to enable the model to more effectively learn the diverse associations among input sequences. Through extensive experimentation on multiple real-world

datasets, PatchMLP shows superior performance compared to existing technologies. This research not only validates the potential of reducing complexity and simplifying network structures to improve performance on LSTF tasks but also emphasizes the critical role of cross-variable interactions in enhancing the accuracy of multivariate time series forecasting. We anticipate that the success of PatchMLP will not only advance further research in the field of LSTF but also motivate future efforts toward developing models that prioritize efficiency, simplicity, and interpretability. Ultimately, we hope that this research will inspire the creation of more innovative time series forecasting methods focused on addressing specific problems rather than the pursuit of model complexity.

## References

- 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*.
- Borovykh, A.; Bohte, S.; and Oosterlee, C. W. 2017. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.
- 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.-A.; Li, C.-L.; Yoder, N.; Arik, S. O.; and Pfister, T. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*.
- Cryer, J. D. 1986. *Time series analysis*, volume 286. Duxbury Press Boston.
- Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissborn, 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. *arXiv preprint arXiv:2306.09364*.
- Feichtenhofer, C.; Li, Y.; He, K.; et al. 2022. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35: 35946–35958.
- Gong, Z.; Tang, Y.; and Liang, J. 2023. PatchMixer: A Patch-Mixing Architecture for Long-Term Time Series Forecasting. *arXiv preprint arXiv:2310.00655*.
- Han, L.; Ye, H.-J.; and Zhan, D.-C. 2023. The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *arXiv preprint arXiv:2304.05206*.
- He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- 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.
- Lee, S.; Park, T.; and Lee, K. 2023. Learning to Embed Time Series Patches Independently. In *The Twelfth International Conference on Learning Representations*.
- 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.; Qi, S.; Li, Y.; and Xu, Z. 2023a. Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping. *arXiv preprint arXiv:2305.10721*.
- Li, Z.; Rao, Z.; Pan, L.; and Xu, Z. 2023b. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*.
- Lim, B.; Arik, S. Ö.; Loeff, N.; and Pfister, T. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.
- Lin, S.; Lin, W.; Wu, W.; Wang, S.; and Wang, Y. 2023. Pet-former: Long-term time series forecasting via placeholder-enhanced transformer. *arXiv preprint arXiv:2308.04791*.
- Liu, L.; Shen, J.; Zhang, M.; Wang, Z.; and Tang, J. 2018. Learning the joint representation of heterogeneous temporal events for clinical endpoint prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- 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. 2021a. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.
- Liu, Y.; Gong, C.; Yang, L.; and Chen, Y. 2020. DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications*, 143: 113082.
- Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.
- Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary transformers: Rethinking the stationarity in time series forecasting. *arXiv preprint arXiv:2205.14415*.
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.
- Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

- Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.
- Patton, A. 2013. Copula methods for forecasting multivariate time series. *Handbook of economic forecasting*, 2: 899–960.
- Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.
- Shen, L.; Wei, Y.; Wang, Y.; and Qiu, H. 2023. Take an Irregular Route: Enhance the Decoder of Time-Series Forecasting Transformer. *IEEE Internet of Things Journal*.
- Tang, P.; and Zhang, X. 2022. MTSMAE: Masked Autoencoders for Multivariate Time-Series Forecasting. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, 982–989. IEEE.
- Tang, P.; and Zhang, X. 2023. Infomaxformer: Maximum Entropy Transformer for Long Time-Series Forecasting Problem. *arXiv preprint arXiv:2301.01772*.
- 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, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2023. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.
- Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and Zhou, J. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.
- Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.
- 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.
- Wu, S.; Xiao, X.; Ding, Q.; Zhao, P.; Wei, Y.; and Huang, J. 2020. Adversarial sparse transformer for time series forecasting. *Advances in neural information processing systems*, 33: 17105–17115.
- Yin, Y.; and Shang, P. 2016. Multivariate multiscale sample entropy of traffic time series. *Nonlinear Dynamics*, 86: 479–488.
- 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. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.
- Zhong, S.; Song, S.; Li, G.; Zhuo, W.; Liu, Y.; and Chan, S.-H. G. 2023. A multi-scale decomposition mlp-mixer for time series analysis. *arXiv preprint arXiv:2310.11959*.
- 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.