2024 International Joint Conference on Neural Networks (IJCNN) | 979-8-3503-5931-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IJCNN60899.2024.10651364

MP3Net:Multi-scale Patch Parallel Prediction Networks for Multivariate Time Series Forecasting

Daishun Cui†,Jiwei Qin †,Fang He‡ , Fei Shi †, Qiang Li† ,Dezhi Sun †, Jiachen Xie†
†*School of Computer Science and Technology,Xinjiang University, Urumqi , China*
†*Xinjiang Key Laboratory of Signal Detection and Processing, Xinjiang University, Urumqi , China*
‡*HongYou Software Co., Ltd, Karamay , China*
Emails: {cuidaishun , jwqin_xju}@163.com ,105026827@qq.com ,
sigofei@xju.edu.cn ,{107552201379,dzsun,107552204095}@stu.xju.edu.cn

***Abstract*—Transformers have emerged as a crucial technology in long-term time series forecasting (LTSF). Recent studies have demonstrated that Transformers segment time series into patches, effectively capturing temporal patterns. However, single-scale patches still fail to adequately capture the complex short-term fluctuations and long-term trends in time series. To tackle this issue, we propose a novel framework, the Multi-scale Patch Parallel Prediction Network (MP3Net). Firstly, we propose a multi-scale patch module to extract local features and long-term correlations from the time series to effectively capture and identify data characteristics across different time scales. Secondly, we propose an adaptive fusion module to dynamically balance the weight of local detail features and long- term correlations from each branch. Thirdly, we introduce a hybrid loss function for model training aimed at enhancing the sensitivity and adaptability of the network towards different types of errors, consequently improving overall performance. Experiments conducted on five benchmark datasets demonstrate that MP3Net surpasses existing methods in LTSF and MP3Net presents an effective approach for addressing the challenge of modeling long-term dependencies in time series analysis.**

***Index Terms*—long-term time series forecasting,long-term dependencies,multi-scale Patch, hybrid loss function**

I. Introduction

Long-term time series forecasting (LTSF) is an important task aimed at using extensive historical time series data to forecast future trends over an extended period, which has been widely applied in various fields such as financial market prediction [1], [2], energy management [3], and influenza epidemic early warning [4]. The Transformer model, known for its excellent performance in Computer Vision (CV) [5]–[10] and Natural Language Processing (NLP) [11]–[13], has exhibited remarkable capabilities in various time series tasks. Numerous Transformer-based models, including LogTrans [14], ETSformer [15], Informer [16], and Autoformer [17], have emerged for long-term time series forecasting tasks. However, these studies are predominantly focused on capturing temporal dependencies at the individual data points. Recent research, such as PatchTST [18], Segrnn [19] and PatchMixer [20], has introduced a patch-based approach to forecast long-term time series. In this approach, the time series is divided into multiple patches of defined lengths, with each patch treated as a token. This method decreases the count of input tokens, thereby reducing memory consumption and simplifying the computational complexity involved in the attention map. These studies have effectively demonstrated the advantages of the patch-based approach, yielding excellent results. However, it's worth noting that even with this approach, single-scale patches still face challenges in comprehensively capturing the short-term fluctuations and long-term trends in time series.

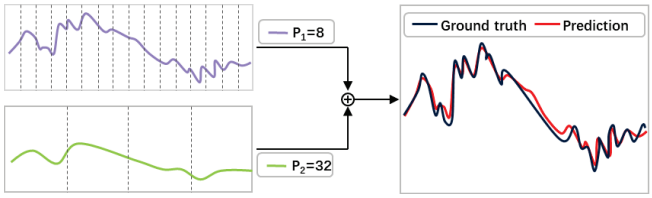

Fig. 1. In the case of the ETTm1 dataset, MP3Net discerns multi-scale temporal patterns through its distinct branches, where P_i denotes the patch size in the i-th branch.

To address this challenge, our study proposes an innovative multi-scale patch parallel prediction network(MP3Net). The network features a multi-branch architecture where each branch processes two-dimensional patches of time series at distinct scales, thus capturing dynamic changes across different temporal scales. As shown in an example from the ETTm1 dataset (Fig 1), patches of different sizes—smaller patches for short-term fluctuations and larger patches for long-term trends and periodic changes—are assigned to different branches, enabling parallel learning of multi-scale features. In contrast to traditional subsampling methods like Preformer [21] and Scaleformer [22], which rely on enhancing self-attention efficiency through mechanisms like segment-wise correlation in place of the typical point-wise dot product, MP3Net dynamically constructs multi-resolution features by

This work was supported by the Key Research and Development Program of Xinjiang Uygur Autonomous Region: Research and Development of Key Technologies for Calculation, Measurement, and Digitized Management of Carbon Emission and Carbon Sink Indicators in the Energy Sector (NO.2022B01010), Outstanding Doctoral Student Innovation Project of Xinjiang University (NO.XJU2024BS085).(Corresponding author: Jiwei Qin).

adjusting the size of patches. Furthermore, we propose an adaptive fusion module that intelligently integrates the most salient features from each scale based on their relevance to the forecasting task, further optimizing the model performance. To improve the robustness of the model to outlier data, we introduce a hybrid loss function, which merges the benefits of Mean Absolute Error (MAE) and Mean Squared Error (MSE). This strategy enhances the sensitivity and adaptability of the network towards different types of errors.

Our work contributions are summarized as follows:

- In order to simultaneously extract local features and long-term correlations from the time series, our study propose the Multi-Scale Patch Parallel Prediction Network (MP3Net), which employs the Multi-scale patch module and adaptive fusion module to integrate the most salient features from each scale.
- We introduce a hybrid loss function that combines MAE and MSE for model training, enhancing the sensitivity and adaptability of the network towards different types of errors.
- In evaluating MP3Net on five real-world datasets, our experiments demonstrate that MP3Net surpasses the latest state-of-the-art methods in performance.

II. RELATED WORK

A. long-term time series forecasting model

Following advancements in Transformers for CV and NLP, transformers have shown superior performance in capturing long-term dependencies. Subsequently, numerous transformer-based models have emerged for LTSF. To tackle the issue of excessive computational complexity, LogTrans [14] introduces a sparse, locally-aware attention mechanism that eliminates the need to compute attention weights between every pair of data points in the sequence. ETSformer [15] combines traditional exponential smoothing models with the Transformer architecture, reducing model parameters and simplifying computational processes. Informer [16] introduces ProbSparse self-attention, predicting attention score distributions for each time point and computing scores for only a subset with the highest probabilities.Autoformer [17] introduces an adaptive seasonal-trend decomposition mechanism, which aggregates point-wise representations to the subsequence level, thereby breaking the bottleneck of information utilization. Segformer [23] subdivides time series into multiple parts with significant dependencies through multi-component decomposition block and collaboration block and introduces a novel attention mechanism, SegAttention, to perform sequence segmentation and segment-level information aggregation effectively. PatchTST [18]introduces a patch program, which divides time series into multiple patches of specific lengths, using each as a token to reduce input tokens, memory usage, and attention map complexity.Inspired by these advancements, we propose a multi-scale patch modeling strategy, assigning different-sized patches to various branches for parallel multi-scale feature learning.

B. Multi-scale Feature modeling

TFT [24] introduces a static covariate encoder and a temporal self-attention decoder to capture the temporal patterns, but it does not fully consider the dynamic interaction between different time scales thereby constraining its efficacy in capturing multi-scale temporal dependencies. Pyraformer [25] innovatively employs a multi-scale building block based on pyramidal attention to create multi-resolution C-trees capable of time dependencies, but the model encounters challenges in handling the balance between high-frequency and low-frequency information. Preformer [21] utilizes a Multi-Scale Segment-wise Correlation (MSSC) mechanism, featuring segmented correlation and a multi-scale structure to aggregate dependencies across time scales, but its capacity to capture features at extreme time scales remains somewhat limited. Scaleformer [22], an iterative multi-scale refining Transformer, aims to enhance time series forecasting performance via a multi-scale framework, suitable for advanced models like FEDformer [26] and Autoformer [17], but its real-time dynamic scaling capabilities could be further enhanced. MICN [27] employs a multi-scale branch structure, modeling distinct patterns separately through downsampled and equidistant convolution, effectively utilizing latent information in time series, but it may be insufficient in the comprehensive extraction of global and local features. SCINet [28] employs the SCI-Block, utilizing varied convolutional kernels for feature extraction, which is complemented by an interactive learning mechanism, facilitating the exchange of features to enable interaction between subsequences of information. However,SCINet may not be efficient enough to deal with multi-scale feature fusion. In our work, we adopt a novel strategy different from downsampling, directly adjusting the size of patches to construct multi-scale feature representations, and use our proposed adaptive fusion module to effectively integrate the most informative features from all scales, thereby optimizing model performance.

III. METHODOLOGY

In this study, our objective is to tackle the task defined as follows: Provided with a multivariate time series data with a historical look-back window of length L , denoted as: $x = (x_1, ..., x_L)$, where each x_t at time step t represents a vector consisting of M variables, we aim to forecast T time steps in the future, resulting in the prediction sequence $(\hat{x}_{L+1}, ..., \hat{x}_{L+T})$.

The overall architecture of MP3Net is illustrated in Fig 2. We employ a multi-scale patch module to convert normalized time series into two-dimensional patches at various scales and route them to different branches for processing. Each branch is independently designated to process data of the specified patch size , thereby capturing dynamic changes on different timescales. The core of each branch is a Transformer backbone designed to learn the interrelationships between patches. Based on it, we use an adaptive fusion strategy to integrate the denormalized prediction of all branches to produce a comprehensive final prediction result.

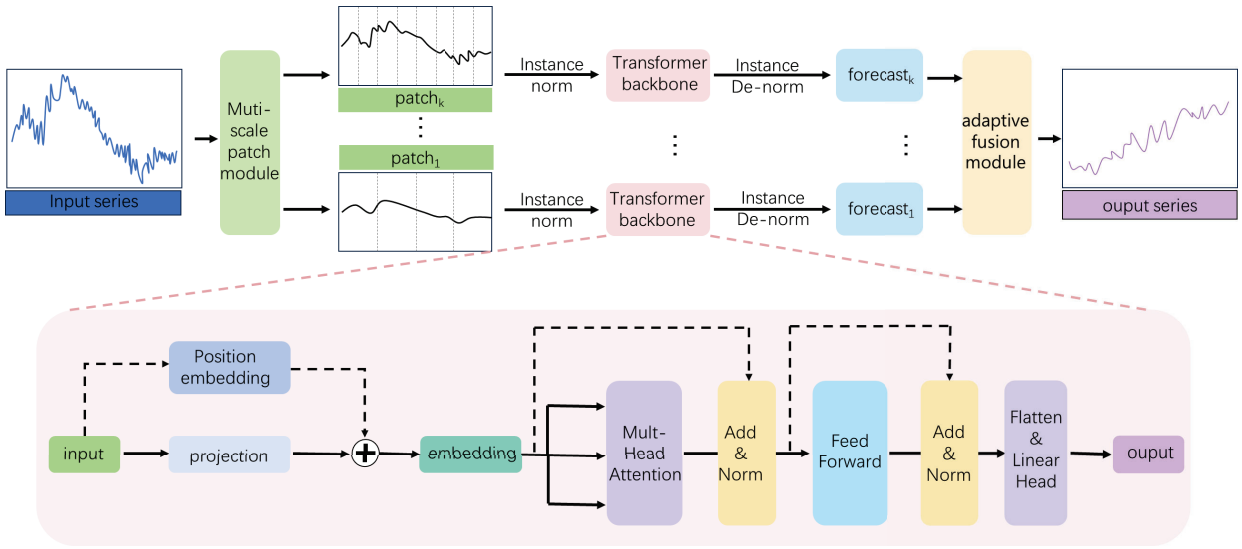

Fig. 2. Multi-Scale Patch Parallel Prediction Network (MP3Net) Architecture.

A. Multi-scale patch module

The patch design employs a series of points as tokens, which significantly reduces the memory costs of long sequences. This reduction enables the model to process longer historical sequences, thereby enhancing its predictive performance. In order to more adequately capture local features and long-term trends in time series, we propose the multi-scale patch module to divide the time series into a set of two-dimensional patches at multiple scales.

The multi-scale patch module processes the input multivariate time series x by dividing it into k non-overlapping patches of varying scales. Let $P = \{patch_1, patch_2, \ldots, patch_k\}$ represent the lengths of these patches. The stride length, defined as the non-overlapping area between two consecutive patches, is denoted by $S = \{S_1, S_2, \ldots, S_k\}$. Consequently, this module produces a set of patches $x_p = \{x_{p1}, x_{p2}, \ldots, x_{pk}\}$. Each patch x_{pi} belongs to $\mathbb{R}^{patch_i \times N_i}$, where $N_i = \left\lfloor \frac{L - Patch_i}{S_i} \right\rfloor + 2$ indicates the number of patches for the i_{th} scale.

We dedicate each branch to processing patches of different lengths in this design. Branches handling larger patches focus more on long-term trends and periodic changes, while branches dealing with smaller patches are inclined to learn small fluctuations in the short term. This approach enables the network to learn information across multiple time scales simultaneously.

B. Transformer backbone

In our architecture,each branch is encoded by the vanilla Transformer backbone. Following the multi-scale patch module, we obtain multiple sets of parallel multi-scale patch data x_p. To mitigate the distribution shift effect between training and test data, we normalize the time series instance x_{pi} from i-th branch with zero mean and unit standard deviation [29] and add back the mean and deviation when outputting predictions.

$$x_{pi} = Norm(x_{pi}) \tag{1}$$

Considering the Transformer architecture's intrinsic indifference to sequence order, we embed positional information to effectively capture temporal dependencies within the input sequence. We map these patches to the Transformer latent space of dimension D via a trainable linear projection $W_p \in \mathbb{R}^{D \times patch_i}$,while employing a learnable additional positional encoding $W_{pos} \in \mathbb{R}^{D \times N_i}$ to monitor the temporal ordering of patches.

$$x_{pi}^{emb} = w_{pi} x_{pi} + W_{pos} \tag{2}$$

The embedded patches x_{pi}^{emb} serve as input to the Transformer encoder. The encoder processes this input via the multi-head scaled dot product attention mechanism, formally expressed as follows:

$$Q_h^i, V_h^i, K_h^i = Projection\left(x_{pi}^{emb}\right)_h \tag{3}$$

$$H_h^i = Attention\left(Q_h^i, K_h^i, V_h^i\right) = Softmax\left(\frac{Q_h^i (K_h^i)^T}{\sqrt{d}}\right) V_h^i \tag{4}$$

$$H^i = Linear\left(Concat\left(H_1^i, H_2^i, ..., H_H^i\right)\right) \tag{5}$$

Where H denotes the number of heads in the multi-head attention mechanism with $h \in \{1, 2, \ldots, H\}$, and Q_h^i, K_h^i, and V_h^i represent the queries, keys, and values for the input patch embeddings of the i-th branch at the h-th head.

Layer normalization and residual connection operations are performed on the output of the multi-head attention mechanism, and the results are fed into the feedforward network.Formally, the process can be written as

$$H_{enc}^i = LN\left(Forward\left(LN\left(H^i + x_{pi}^{emb}\right)\right) + H^i\right) \tag{6}$$

TABLE I
MULTIVARIATE LONG-TERM FORECASTING RESULTS ON FIVE BENCHMARK DATASETS.BOLD AND UNDERLINED DENOTE THE BEST AND SECOND-BEST RESULTS RESPECTIVELY.

Models		MP3Net		PatchTST		DLinear		MICN		FEDformer		Autoformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.361**	**0.388**	0.375	0.399	0.382	0.406	0.421	0.431	0.376	0.415	0.435	0.446	0.941	0.769
	192	**0.400**	**0.413**	0.414	0.421	0.405	0.414	0.474	0.487	0.423	0.446	0.456	0.457	1.007	0.786
	336	**0.430**	**0.432**	0.432	0.437	0.449	0.451	0.569	0.551	0.444	0.462	0.486	0.487	1.038	0.784
	720	**0.427**	**0.453**	0.450	0.466	0.502	0.511	0.770	0.672	0.469	0.492	0.515	0.517	1.144	0.857
ETTh2	96	**0.273**	**0.332**	0.274	0.336	0.289	0.352	0.299	0.364	0.332	0.374	0.332	0.368	1.549	0.952
	192	**0.339**	**0.375**	0.340	0.379	0.370	0.408	0.441	0.454	0.407	0.446	0.426	0.434	3.792	1.542
	336	**0.326**	**0.378**	0.331	0.380	0.442	0.457	0.654	0.567	0.400	0.447	0.477	0.479	4.215	1.642
	720	**0.373**	**0.415**	0.377	0.420	0.727	0.604	0.956	0.716	0.412	0.469	0.453	0.490	3.656	1.619
ETTm1	96	**0.285**	**0.335**	0.289	0.341	0.300	0.344	0.316	0.362	0.326	0.390	0.510	0.492	0.626	0.560
	192	**0.324**	**0.364**	0.330	0.369	0.337	0.366	0.363	0.390	0.365	0.415	0.514	0.495	0.725	0.619
	336	**0.360**	**0.382**	0.367	0.394	0.370	0.386	0.408	0.426	0.392	0.425	0.510	0.492	1.005	0.741
	720	**0.412**	**0.413**	0.421	0.425	0.429	0.424	0.481	0.476	0.446	0.458	0.527	0.493	1.133	0.845
ETTm2	96	**0.161**	**0.249**	0.164	0.253	0.171	0.262	0.179	0.275	0.180	0.271	0.205	0.293	0.355	0.462
	192	**0.218**	**0288**	0.220	0.292	0.227	0.305	0.307	0.376	0.252	0.318	0.278	0.336	0.595	0.586
	336	**0.272**	**0.324**	0.278	0.329	0.291	0.352	0.325	0.388	0.324	0.364	0.343	0.379	1.270	0.871
	720	**0.353**	**0.376**	0.367	0.385	0.404	0.427	0.502	0.490	0.410	0.420	0.414	0.419	3.001	1.267
ILI	24	**1.345**	**0.706**	1.522	0.814	2.269	1.053	2.684	1.112	2.624	1.095	2.906	1.182	4.657	1.449
	36	1.642	**0.830**	**1.430**	0.834	2.259	1.061	2.667	1.068	2.516	1.021	2.585	1.038	4.650	1.463
	48	**1.563**	**0.835**	1.673	0.854	2.254	1.061	2.558	1.052	2.505	1.041	3.024	1.145	5.004	1.542
	60	1.803	0.898	**1.529**	**0.862**	2.425	1.125	2.747	1.110	2.742	1.122	2.761	1.114	5.071	1.543

Where LN denotes Layer normalization helping maintain stability between network layers.

In order to obtain the prediction result of each of our patch branches, we flatten the encoded information obtained from the feedforward network and feed it into a Linear Head(LH). Formally, the process can be written as

$$forecast_i = LH(Flatten(H^i_{enc})) \tag{7}$$

Considering that the final prediction results need to be evaluated on the same scale as the actual application, we perform the inverse normalization of the prediction results. Formally, the process can be written as

$$forecast_i = Denorm(forecast_i) \tag{8}$$

C. Adaptive fusion module

In order to preserve the learned long-term trends and short-term subtle fluctuations at multiple scales, this study introduces an adaptive fusion module.

$$forecast = \sum_{i=1}^{k} \alpha_i \times forecast_i \tag{9}$$

where α_i is a learnable parameter,which denotes the weight of the branch of patch scale $patch_i$.

This method enables the model to automatically adjust the weight of patches of different scales according to their importance in the prediction process.This flexible integration method enhances the adaptability and understanding depth of the model to the multi-scale characteristics of time series data.

D. hybrid loss function

In the field of existing time series forecasting, most models, such as PatchTST [18], Dlinear [30], and Informer [16], mainly adopt the MSE as their loss function. However, MSE shows high sensitivity to occasional outliers in the data. Aiming to improve the robustness of the model to outlier data, some researchers, such as Segrnn [19] , explore the use of MAE as the loss function. Given this, we propose a hybrid loss function that combines the squared loss and the absolute loss.

$$L_{hybrid} = \frac{1}{T}\sum_{i=1}^{T} \|\hat{x}_{L+i} - x_{L+i}\|_2^2 + \frac{1}{T}\sum_{i=1}^{T} \|\hat{x}_{L+i} - x_{L+i}\| \tag{10}$$

We fuse these two loss methods in a 1:1 ratio, aiming to balance the accuracy of the prediction and the robust response to outliers. In this way, while achieving superior overall prediction accuracy, our model also shows greater efficiency and robustness in handling occasional outliers in time series data.

IV. EXPERIMENTS

A. Datasets and experimental setups

We evaluated the performance of our proposed MP3Net on five publicly available benchmark datasets for LTSF: The statistics of those datasets are summarized in Table II. We

divide each dataset into training, validation, and testing subsets, with a ratio of 0.6:0.2:0.2 for the four ETT datasets and 0.7:0.1:0.2 for the ILI.

TABLE II
STATISTICS OF POPULAR DATASETS FOR BENCHMARK.

Datasets	ETTh1	ETTh2	ETTm1	ETTm2	ILI
Features	7	7	7	7	7
Timesteps	17420	17420	69680	69680	966

To benchmark our study, we select state-of-the-art (SOTA) and representative LTSF models as baselines. This includes Transformer-based models like PatchTST [18] (2023), FEDformer [26] (2022), Autoformer [17] (2021), and Informer [16] (2021), the CNN-based model MICN [27] (2023), and the significant MLP-based model DLinear [30] (2023). We use the widely recognized evaluation metrics MSE and MAE to assess the performance of these models.

For training, the batch size is selected from $\{16, 32, 64, 128\}$ via grid search and the patch branch number is set to k=2. For ETTh1, ETTh2, ETTm1 and ETTm2, the patch size is set to $\{8, 16, 32, 48, 64, 96\}$, and the lookback window size is set to 336.For the ILI dataset, the patch size is set to $\{3, 6, 12, 24\}$, and the lookback window size is set to 104. MP3Net is implemented using PyTorch and runs on a single NVIDIA GeForce RTX 3060 GPU.

B. Main results

In Table I, we present the primary experimental results for MP3Net and all baseline models on MSE and MAE for five datasets. We highlight the best and second-best results for each case (dataset, horizontal line, and metrics) in bold and underlined, respectively. MP3Net outperforms the SOTA baseline method, achieving 37 best results and 3 second best results in 40 cases. Compared with the best existing method PatchTST, the accuracy of MP3Net is slightly improved. Compared with the MLP-based sota baseline DLinear, the MSE of MP3Net is reduced by 13.17% and the MAE is reduced by 9.01%. MP3Net achieves significant improvements of 15.14% and 9.58% on average in terms of MSE and MAE compared to the CNN-based sota baseline MICN.

C. Multi-scale patch module ablation

We conduct ablation experiments with Multi-scale patch module with the same configuration as the main experiment on four datasets: ETTh1, ETTh2, ETTm1, ETTm2, to verify the effectiveness of our proposed multi-scale branch patch module. To this end, we construct two variants of MP3Net: one is MP3Net(p=32), which focuses only on branches with large-size patches in order to focus on capturing long-term trends and periodic patterns of sequences; The other is MP3Net(p=8), which involves only branches of small size patches and focuses on capturing local features of sequences.

The experimental results from Table III show that removing either branch leads to a decrease in the overall performance

TABLE III
ABLATION STUDY ON MULTI-SCALE PATCH MODULE.

Models		MP3Net		MP3Net(p=32)		MP3Net(p=8)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.361**	**0.388**	0.374	0.397	0.366	0.391
	192	**0.400**	**0.413**	0.415	0.422	0.405	0.415
	336	**0.430**	**0.432**	0.434	0.435	0.431	0.433
	720	**0.427**	**0.445**	0.454	0.470	0.439	0.457
ETTh2	96	**0.273**	**0.332**	0.278	0.337	0.274	0.333
	192	**0.339**	**0.375**	0.343	0.380	0.341	0.376
	336	**0.326**	**0.378**	0.335	0.386	0.328	0.379
	720	**0.373**	**0.415**	0.381	0.421	0.375	0.416
ETTm1	96	**0.285**	**0.335**	0.296	0.343	0.293	0.340
	192	**0.324**	**0.354**	0.333	0.365	0.330	0.365
	336	**0.360**	**0.382**	0.370	0.387	0.366	0.384
	720	**0.412**	**0.413**	0.423	0.417	0.418	0.415
ETTm2	96	**0.161**	**0.249**	0.163	0.250	0.163	**0.249**
	192	**0.218**	**0.288**	**0.218**	**0.288**	0.219	0.289
	336	**0.272**	**0.324**	0.273	0.325	0.273	0.325
	720	**0.353**	**0.376**	0.362	0.383	0.363	0.380

of the model. It is noteworthy to mention that the branch prediction effect of small patch size is usually better than that of large patch size. One potential reason is that the number of large patch sizes is less, making it more prone to overfitting in the training process. This result highlights the complementarity of patches of different sizes in capturing different features and also verifies the importance of our proposed Multi-scale patch module in improving the performance of the model.

D. Hybrid Loss Function ablation

In order to verify the contribution of our proposed hybrid loss function to improve the performance of the model, we used three different loss functions: hybrid loss function, MSE, and MAE, to train the MP3Net model. We compare the training performance of these different loss functions on multiple datasets and list the experimental results in Table IV in detail.

The results clearly show that the hybrid loss function achieves the best prediction performance across most datasets and prediction lengths, outperforming the training methods using only a single loss function. It is worth noting that the performance of the model trained by only MSE is not ideal. This traditional method cannot achieve the performance of the model trained by only MAE or Hybrid Loss. Moreover, although using MAE alone as the loss function can achieve better MAE metrics in some cases, it usually sacrifices the performance of MSE. These experimental results fully illustrate the unique advantages of our proposed hybrid loss function in integrating model robustness to outliers and sensitivity to significant error penalty, effectively balancing the strengths and weaknesses of MSE and MAE, and significantly improving the accuracy and reliability of time series forecast- ing.

E. Experiments with multi-scale patch combinations

In order to verify the advantages of the proposed multi-scale patch idea for single scale, we configured the model into

TABLE IV
ABLATION STUDY ON HYBRID LOSS FUNCTION.

Models		hybrid loss		only MSE		only MAE	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.361**	**0.388**	0.370	0.398	0.367	0.390
	192	**0.400**	**0.413**	0.406	0.418	0.410	0.415
	336	0.430	**0.432**	**0.428**	0.434	0.431	0.433
	720	**0.427**	**0.445**	0.431	0.453	0.435	0.453
ETTh2	96	**0.273**	**0.332**	0.275	0.335	0.275	0.333
	192	**0.339**	**0.375**	0.339	0.377	0.346	0.375
	336	**0.326**	**0.378**	0.329	0.382	0.334	0.379
	720	**0.373**	**0.415**	0.374	0.418	0.375	0.416
ETTm1	96	**0.285**	0.335	0.288	0.341	0.286	**0.334**
	192	**0.324**	**0.354**	0.329	0.366	0.326	0.356
	336	**0.360**	0.382	0.366	0.389	0.378	**0.381**
	720	**0.412**	**0.413**	0.416	0.416	0.421	0.414
ETTm2	96	**0.161**	**0.249**	0.163	0.253	0.162	0.251
	192	**0.218**	**0.288**	0.221	0.292	0.218	0.290
	336	**0.272**	**0.324**	0.276	0.328	0.275	0.325
	720	**0.353**	**0.376**	0.367	0.382	0.367	0.380

single-branch, double-branch, and three-branch structures, and performed different combinations of multi-scale patches. These combinations were experimentally tested on the ETTh1 with a forecasting horizon of 96. The experimental results are shown in Table V.

TABLE V
EXPERIMENTS WITH MULTI-SCALE PATCH COMBINATIONS.

k=1	p=8	p=16	p=32	p=48	p=64
	0.366	0.372	0.374	0.400	0.398
	p=96	-	-	-	-
	0.406	-	-	-	-
k=2	p=(8,16)	p=(8,32)	p=(8,48)	p=(8,64)	p=(8,96)
	0.366	0.361	0.367	0.368	0.369
	p=(16,32)	p=(16,48)	p=(16,64)	p=(16,96)	p=(32,48)
	0.372	0.375	0.379	0.376	0.385
	p=(32,64)	p=(32,96)	p=(48,64)	p=(48,96)	p=(64,96)
	0.383	0.385	0.392	0.398	0.391
k=3	p=(8,16,32)	p=(8,16,48)	p=(8,16,64)	p=(8,16,96)	p=(8,32,48)
	0.366	0.367	0.365	0.365	0.366
	p=(8,32,64)	p=(8,32,96)	p=(8,48,64)	p=(8,48,96)	p=(8,64,96)
	0.367	0.367	0.367	0.368	0.367
	p=(16,32,48)	p=(16,32,64)	p=(16,32,96)	p=(16,48,64)	p=(16,48,96)
	0.373	0.375	0.373	0.376	0.375
	p=(16,64,96)	p=(32,48,64)	p=(32,48,96)	p=(32,64,96)	p=(48,64,96)
	0.375	0.381	0.381	0.381	0.389

In the single-branch combination experiment (k=1), we observe that the model generally achieves better results when the single branch uses smaller patches (such as p=8). This is because small-scale patches can capture more subtle local features in the time series and thus reveal detailed information about the series itself. However, when a single branch uses a larger patch, such as p=64, the model tends to overfit due to the small number of patches, leading to decreased prediction performance.

Combining the two-branch structure (k=2), such as p=(8,32), can better balance capturing local features and long-term trends. Smaller patches capture local features of the series, while larger patches help the model identify long-term trends and patterns in the time series. This combination, therefore, leads to balanced results at all scales. For the other combinations, we note that some combinations may be prone to overfitting because the granularity of patches is too fine to adapt to the characteristics of the original sequence, or because the patch scale is too large and the number of patches is too small, so that the model cannot capture enough contextual information.

The experimental results verify the effectiveness of the multi-scale patch structure in the time series forecasting task. By appropriately combining patches of different scales, the model can learn the features in the time series more comprehensively, neither missing crucial local information nor ignoring the overall trend and pattern.

F. Experiment on the number of patches at multiple scales

After conducting the combination experiment, we found that the model can learn the features in the time series more comprehensively by properly combining patches of different scales, but it is questionable to adopt patches of several scales. In this section, experiments were conducted on one to six branches one by one, and the results are shown in Fig 3.

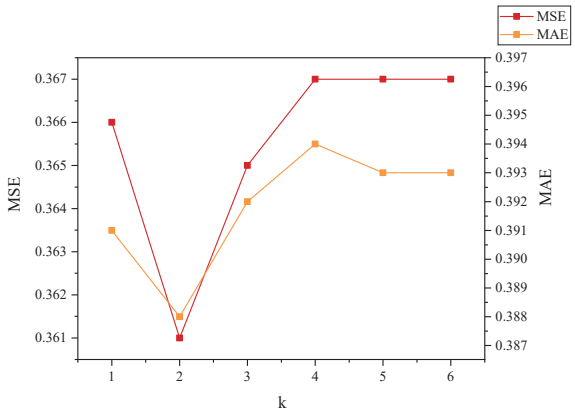

Fig. 3. Experiment on the number of patches at multiple scales.

We can see that using only two different scale patch combinations can achieve better results; when there are too many branches, the additional scale information will not provide adequate information but will be regarded as noise to affect the final prediction results.

V. CONCLUSIONS AND PROSPECT

This study introduces the Multi-scale Patch Parallel Prediction Network (MP3Net), an innovative approach to LTSF. The network's multi-scale patch module effectively captures diverse time scales, enhancing the extraction of local features and long-term correlations. MP3Net leverages a hybrid loss function combining MAE and MSE to improve sensitivity and adaptability to different error types. While MP3Net has shown superiority on benchmark datasets, future work will focus on developing autonomous methods for learning optimal

feature size combinations to elevate prediction accuracy in LTSF further.

REFERENCES

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

[2] M. Wang, F. Chen, J. Guo, and W. Jia, "Improving stock trend prediction with multi-granularity denoising contrastive learning," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–10.

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

[4] Y. Matsubara, Y. Sakurai, W. G. Van Panhuis, and C. Faloutsos, "Funnel: automatic mining of spatially coevolving epidemics," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 105–114.

[5] Y. Ma, M. Li, and J. Chang, "A hierarchical vision transformer using overlapping patch and self-supervised learning," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–7.

[6] N. Li, Y. Chen, and D. Zhao, "Dense attention: A densely connected attention mechanism for vision transformer," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[7] L. Jin and E. Y. Kim, "Gvit: Locality enhanced vision transformer using spectral graph convolutional network," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–6.

[8] J. Liu, Y. Li, G. Cao, Y. Liu, and W. Cao, "Feature pyramid vision transformer for medmnist classification decathlon," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[9] J. Yao, T. Wu, and X. Zhang, "Improving depth gradient continuity in transformers: A comparative study on monocular depth estimation with cnn," *arXiv preprint arXiv:2308.08333*, 2023.

[10] J. Yao and J. Zhang, "Depthssc: Depth-spatial alignment and dynamic voxel resolution for monocular 3d semantic scene completion," *arXiv preprint arXiv:2311.17084*, 2023.

[11] P. Li, P. Zhong, J. Zhang, and K. Mao, "Convolutional transformer with sentiment-aware attention for sentiment analysis," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

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

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] X. Nie, X. Zhou, Z. Li, L. Wang, X. Lin, and T. Tong, "Logtrans: Providing efficient local-global fusion with transformer and cnn parallel network for biomedical image segmentation," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2022, pp. 769–776.

[15] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," *arXiv preprint arXiv:2202.01381*, 2022.

[16] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[17] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[18] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[19] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, "Segrnn: Segment recurrent neural network for long-term time series forecasting," *arXiv preprint arXiv:2308.11200*, 2023.

[20] Z. Gong, Y. Tang, and J. Liang, "Patchmixer: A patch-mixing architecture for long-term time series forecasting," *arXiv preprint arXiv:2310.00655*, 2023.

[21] D. Du, B. Su, and Z. Wei, "Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[22] A. Shabani, A. Abdi, L. Meng, and T. Sylvain, "Scaleformer: iterative multi-scale refining transformers for time series forecasting," *arXiv preprint arXiv:2206.04038*, 2022.

[23] J. Chen, J. Fan, Z. Liu, J. Xiang, and J. Wu, "Segformer: Segment-based transformer with decomposition for long-term series forecasting," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[24] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[25] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International conference on learning representations*, 2021.

[26] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27 268–27 286.

[27] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *The Eleventh International Conference on Learning Representations*, 2022.

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

[29] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.

[30] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.