CSformer: Combining Channel Independence and Mixing for Robust Multivariate Time Series Forecasting

Haoxin Wang[1], Yipeng Mo[1], Kunlan Xiang[2], Nan Yin[1], Honghe Dai[1]
Bixiong Li[3], Songhai Fan[4], Site Mo[1]*

[1]College of Electrical Engineering, Sichuan University
[2]University of Electronic Science and Technology of China
[3]College of Architecture and Environment, Sichuan University
[4]State Grid Sichuan Electric Power Research Institute
{whx1122, moyipeng}@stu.scu.edu.cn, mosite@126.com

Abstract

In the domain of multivariate time series analysis, the concept of channel independence has been increasingly adopted, demonstrating excellent performance due to its ability to eliminate noise and the influence of irrelevant variables. However, such a concept often simplifies the complex interactions among channels, potentially leading to information loss. To address this challenge, we propose a strategy of channel independence followed by mixing. Based on this strategy, we introduce **CSformer**, a novel framework featuring a two-stage multiheaded self-attention mechanism. This mechanism is designed to extract and integrate both **C**hannel-specific and **S**equence-specific information. Distinctively, CSformer employs parameter sharing to enhance the cooperative effects between these two types of information. Moreover, our framework effectively incorporates sequence and channel adapters, significantly improving the model's ability to identify important information across various dimensions. Extensive experiments on several real-world datasets demonstrate that CSformer achieves state-of-the-art results in terms of overall performance.

Introduction

Time series forecasting is vital in areas such as traffic management (Cirstea et al. 2022; Yin and Shang 2016; Qin et al. 2023), power systems (Stefenon et al. 2023; Wang et al. 2023; Mo et al. 2024), and healthcare (Ahmed, Lin, and Srivastava 2023; Alshanbari et al. 2023; Sen 2022). However, it faces significant challenges due to the complex long-term dependencies and variable interrelations inherent in time series data. These difficulties have propelled multivariate time series forecasting (MTSF) to the forefront of research in these domains.

Recently, many deep learning models have been applied to MTSF, especially Transformer-based models, such as Informer (Li, Hui, and Zhang 2021), Autoformer (Wu et al. 2021), and Fedformer (Zhou et al. 2022). These models have demonstrated improved predictive performance by refining the attention mechanism. However, recent research has raised questions regarding the suitability of Transformer models for MTSF task, proposing a straightforward linear model, DLinear (Zeng et al. 2023), that surpasses traditional Transformer models. Given the Transformer's success in other domains (NLP, CV, etc.) (Devlin et al. 2018; Brown et al. 2020; Dosovitskiy et al. 2021), researchers have shifted their focus from modifying the attention structure within Transformers to altering the input data. iTransformer (Liu et al. 2023) treats sequences from each channel as a single token, embedding them along the sequence dimension and applying attention across the channel dimension. PatchTST (Nie et al. 2023) uses a channel independent approach. It divides the input data into patches before feeding it into the Transformer and then embedding each patch as a token. These approaches have reinstated Transformer methods to a prominent position. We summarize the Transformer-based model based on the channel processing approach. As shown in Figure 1, traditional models like Informer and Autoformer (Li, Hui, and Zhang 2021; Wu et al. 2021) directly embed variables, leading to channel mixing and potential noise introduction, especially from varied sensors and distributions. To eliminate these defects, PatchTST (Nie et al. 2023) adopts a channel-independent approach, risking information loss due to reduced physical intuitiveness. These observations suggest that neither direct channel independence nor mixing is optimal. To address this, hybrid methods like iTransformer (Liu et al. 2023) combine sequence embedding with channel-wise attention, offering a balanced approach. However, embedding methods of iTransformer may

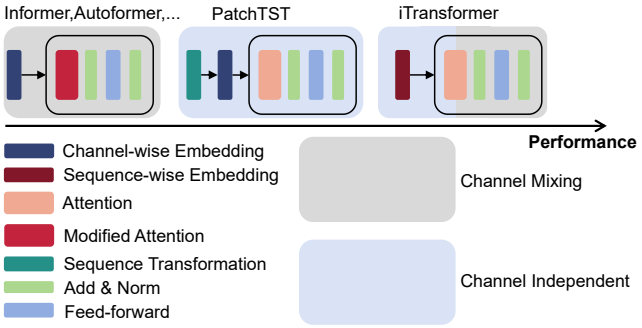


Figure 1: Transformer-based models categorized by channel independence and channel mixing.

*Corresponding author.

lead to erasure of inter-sequence correlations and treat the extraction of sequence information and channel information as separate non-interactive processes.

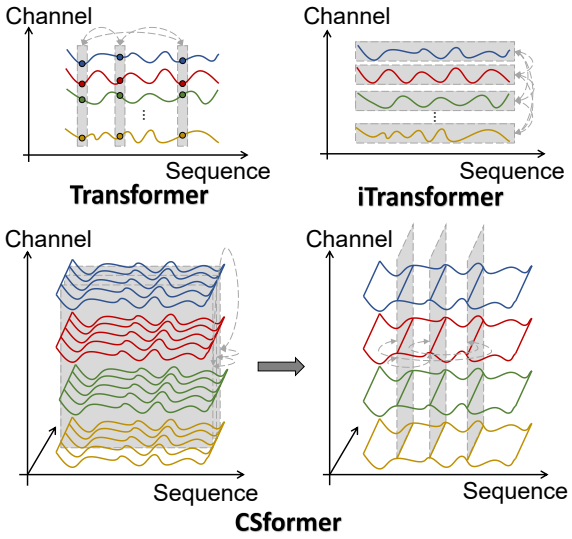


Figure 2: Structural Comparison of Transformer (top left), iTransformer (top right), and proposed CSformer (bottom): In the illustration, we compare the architectures of Transformer (top left) and iTransformer (top right) with the proposed CSformer (bottom). While Transformer and iTransformer employ attention mechanisms separately in the sequence and channel dimensions, CSformer diverges by embedding sequences into a high-dimensional space. Consequently, CSformer performs attention independently in both channel and sequence dimensions.

Drawing on the aforementioned motivation, we believe an effective combination of channel independence and channel mixing is crucial for mining more robust features in multivariate time-series data. In this paper, We propose CSformer, a model adept at extracting and combining sequence and channel information without modifying Transformer's attention mechanism. As depicted in Figure 2, our approach introduces a dimensionality-augmented embedding technique that enhances the sequence dimensionality while preserving the original data's integrity. We then use a unified multi-headed self-attention (MSA) mechanism to apply different attentions on sequence and channel dimensions respectively, employing a shared-parameter approach to realize the interplay of information from different dimensions. After each MSA mechanism, we include an adapter to ensure that different features are extracted by the two-stage self-attention mechanism. The primary contributions of our work are outlined as follows:

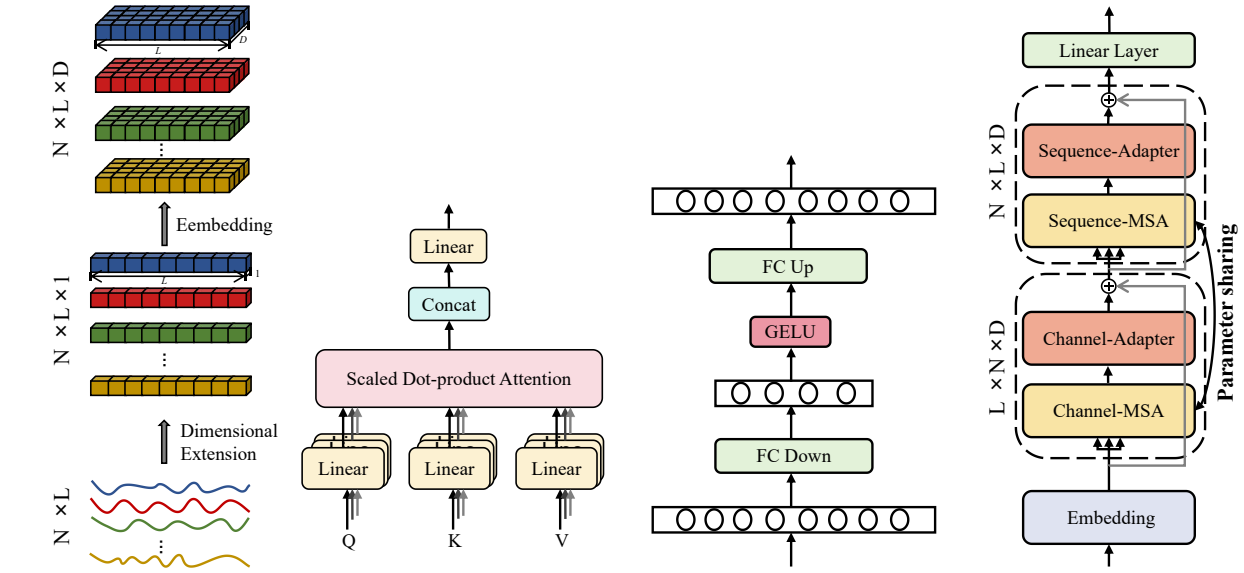
- To enhance the extraction capabilities of channel and sequence information, we propose *CSformer*, a two-stage attention Transformer model. This model efficiently captures the sequential and channel information of multivariate time series while minimizing the increase in model parameters, enabling effective information interaction.
- CSformer achieves state-of-the-art (SOTA) performance on diverse real-world datasets, covering domains such as electricity and weather. Extensive ablation studies further validate the model's design rationality.
- We propose a new training strategy for MTSF: channel independence followed by channel mixing, providing a new insight for future research.

Related Work

Time series forecasting models are broadly classified into statistical and deep learning models. Statistical models, such as ARIMA (Bartholomew 1971) and Exponential Smoothing (Hyndman et al. 2008), are noted for their simplicity and efficacy in capturing temporal dynamics, but mainly focus on univariate forecasting, neglecting additional covariates. Conversely, deep learning models, including RNN-based (Medsker and Jain 2001; Graves and Graves 2012; Dey and Salem 2017; Mo et al. 2023), CNN-based (Bai, Kolter, and Koltun 2018; Wang et al. 2022; Wu et al. 2023), and Transformer-based (Li, Hui, and Zhang 2021; Wu et al. 2021; Zhou et al. 2022), have been prominent in MTSF for their comprehensive capabilities. They aim to improve prediction accuracy by integrating temporal and inter-channel information. Recent studies, however, indicate focusing solely on sequence information may be more effective. Therefore, deep learning models in MTSF can be categorized as channel-independent or channel-mixing based on their handling of inter-channel information.

Channel-independent Models The channel-independent models process each time series channel (variable) independently, disregarding any interdependencies or interactions among different channels. Intuitively, this simplistic approach may not yield optimal results in certain contexts due to its failure to consider the potentially complex interrelationships within time series data. Despite this, significant performance enhancements have been observed with such models. DLinear (Zeng et al. 2023), a proponent of channel-independent architectures, employs a singular linear model along the sequence dimension and surpasses the performance of all contemporary state-of-the-art Transformer-based models. PatchTST (Nie et al. 2023), another exemplar of channel-independent models, initially divides a single time series into segments, treats each segment as a token, and then processes these segments through a transformer model to extract temporal information.

Channel-mixing Models The channel-mixing models are characterized by approaches in which the analysis of one time series channel is influenced by other channels. These models integrate interdependencies among multiple time series variables, facilitating richer and more accurate representations of data patterns and trends. Previous research (Li et al. 2019; Li, Hui, and Zhang 2021; Wu et al. 2021; Zhou et al. 2022) mainly concentrated on reducing the computational complexity of the attention mechanism but addressed inter-channel relationships only through basic embedding operations. This approach may not fully capture mutual information, and at times, it might inadvertently introduce



(a) Dimension-augmented Embedding (b) MSA (c) Adapter (d) CSformer

Figure 3: We present the overall framework of CSformer (d). Initially, the input sequence undergoes a dimensional expansion operation before embedding (a). This dimensional transformation allows the standard MSA (b) to be adapted separately for channels and sequences (c). Note that Channel-MSA and Sequence-MSA share weights but are applied to different input dimensions.

noise. To address this limitation, Crossformer (Zhang and Yan 2023) introduces a two-stage attention layer that leverages both cross-dimensional and cross-temporal dependencies. Similarly, TSMixer (Chen et al. 2023) applies linear layers alternately across time and channel dimensions, thus enabling the extraction of temporal and cross-variate information. In a more recent development, iTransformer (Liu et al. 2023) conducts embedding operations along the time dimension and then utilizes the attention mechanism to extract information between variables, leading to notable performance enhancements.

As highlighted in PatchTST (Nie et al. 2023), improperly handling cross-dependency extraction can lead to the introduction of noise. Our objective with CSformer is to devise a method that effectively utilizes cross-variable information while concurrently adequately extracting temporal data. Additionally, considering that advancements in the attention mechanism have not substantially enhanced efficiency in practical applications (Zeng et al. 2023), this paper maintains adherence to the vanilla attention mechanism architecture.

CSformer

In this section, we present our novel model, the CSformer, which is capable of concurrently learning channel and sequence information. Figure 3 illustrates the overall architecture of CSformer. We first enhance multivariate time series data using dimension-augmented embedding. Then, we apply a two-stage MSA mechanism to extract channel and sequence features, sharing parameters to facilitate interaction between dimensions. Finally, we introduce channel/ sequence adapters for each output to strengthen their capability to learn information across different dimensions.

Preliminaries

In the context of multivariate time series forecasting, the historical sequence of inputs is denoted as $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\} \in \mathbb{R}^{N \times L}$, where N represents the number of variables (channels) and L represents the length of the historical sequence. Let $\mathbf{x}_i \in \mathbb{R}^N$ represent the values of the various variables at the i-th time point. Let $\mathbf{X}^{(k)} \in \mathbb{R}^L$ denote the sequence of the k-th variable. Assuming our prediction horizon is T, the predicted result is denoted as $\hat{\mathbf{X}} = \{\mathbf{x}_{L+1}, \ldots, \mathbf{x}_{L+T}\} \in \mathbb{R}^{N \times T}$. Considering a model $\mathbf{f}_\theta$, where θ represents the model parameters, the overall process of multivariate time series forecasting can be abstracted as $\mathbf{f}_\theta(\mathbf{X}) \to \hat{\mathbf{X}}$.

Reversible Instance Normalization

In practical scenarios, time-series data distributions often shift due to external factors, leading to data distribution bias. To mitigate this, Reversible Instance Normalization (ReVIN) (Kim et al. 2021) has been introduced. As delineated in Equation 1, ReVIN normalizes each input variable's sequence, mapping it to an alternative distribution. This process artificially minimizes distributional disparities among univariate variables in the original data. Subsequently, the original mean, variance, and learned parameters are reinstated in the output predictions.

$$\mathrm{ReVIN}(\mathbf{X}) = \left\{ \gamma_k \frac{\mathbf{X}^{(k)} - \mathrm{Mean}(\mathbf{X}^{(k)})}{\sqrt{\mathrm{Var}(\mathbf{X}^{(k)}) + \varepsilon}} + \quad \beta_k \,\middle|\, k = 1, \cdots, N \right\}, \quad (1)$$

where ε refers to an infinitesimal quantity that prevents the denominator from being 0. $\beta \in \mathbb{R}^N$ is the parameter for performing the normalization and $\gamma \in \mathbb{R}^N$ is the learnable parameter for performing the affine transformation.

Dimension-augmented Embedding

PatchTST (Nie et al. 2023) segments input sequences into temporal blocks, treating each as a Transformer token. TimesNet (Wu et al. 2023) uses a fast Fourier transform (FFT) to discern sequence cycles, reshaping them into 2D vectors for CNN processing. While these methods increase sequence dimensionality, they risk information loss. To address this issue, we introduce a direct sequence embedding approach, which is inspired by word embedding techniques (Mikolov et al. 2013; Yi et al. 2023). To maintain input data integrity, we first apply a dimension-augmentation operation: $\mathbf{X} \in \mathbb{R}^{N\times L} \rightarrow \mathbf{X} \in \mathbb{R}^{N\times L\times 1}$. Then, we multiply the augmented sequence element-wise with a learnable vector $\nu \in \mathbb{R}^{1\times D}$, producing the embedded output $\mathbf{H} \in \mathbb{R}^{N\times L\times D} = \mathbf{X} \times \nu$. This method facilitates dimensionality enhancement and embedding without distorting the original input's intrinsic information, setting the stage for the subsequent two-stage MSA detailed later.

Two-stage MSA

The CSformer consists of M blocks. Due to the dynamic weighting characteristics of the self-attention mechanism, each block employs a two-stage MSA using shared parameters. A brief exposition of this characteristic is as follows: with input data $\mathbf{H} \in \mathbb{R}^{S\times D}$, where S represents N or L, we define $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{S\times D_k}$ as query and key. The attention score $\mathbf{A} \in \mathbb{R}^{S\times S}$ can be derived as $\mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{D_k})$, where $\mathbf{Q} = \mathbf{H}\mathbf{W}_\mathbf{Q}$ and $\mathbf{K} = \mathbf{H}\mathbf{W}_\mathbf{K}$, with $\mathbf{W}_\mathbf{Q}, \mathbf{W}_\mathbf{K} \in \mathbb{R}^{D\times D_k}$ as projection matrices of query and key. By the associative law of matrix multiplication, $\mathbf{A}$ can be written as $\mathrm{Softmax}(\mathbf{H}\mathbf{W}\mathbf{H}^\top/\sqrt{D_k})$, where $\mathbf{W} = \mathbf{W}_\mathbf{Q}{\mathbf{W}_\mathbf{K}}^\top$. Since $\mathbf{W}, D_k$ are fixed values, $\mathbf{A}$ dynamically adapts to changes in input data $\mathbf{H}$. This characteristic implies that with varying dimensional inputs, the attention scores adaptively adjust, even under a shared parameter method. Following the application of MSA, an adapter is incorporated to optimize the discriminative learning of both channel and sequence information. Detailed discussions of the MSA and adapter components are outlined in subsequent sections.

Channel MSA The channel MSA in the first stage is similar to the iTransformer (Liu et al. 2023), both employing MSA along the channel dimension. However, a distinction arises in the treatment of the time series, while the iTransformer considers the entire time series as a single token, herein, we apply channel-wise attention at each time step to discern inter-channel dependencies. Let $\mathbf{H}_c \in \mathbb{R}^{L\times N\times D}$ denote the input subjected to the channel-wise MSA. The output post-attention $\mathbf{Z}_c \in \mathbb{R}^{L\times N\times D}$ is formulated as:

$$\mathbf{Z}_c = \mathrm{MSA}(\mathbf{H}_c). \tag{2}$$

The shared parameters between channel and sequence MSAs in CSformer pose challenges in learning across two dimensions simultaneously. Inspired by fine-tuning technique in NLP's large-scale model adaptation (Houlsby et al. 2019; Hu et al. 2022), we integrate adapter technology (Houlsby et al. 2019). This comprises two fully connected layers interspersed with an activation function. The first layer downscales input features, followed by activation, and the second layer reverts the representation to its original dimensionality. This iterative process enables the model to capture channel representations more effectively. Thus, after channel information extraction, the model output is delineated as follows:

$$\mathbf{A}_c = \mathrm{Adapter}(\mathrm{Norm}(\mathbf{Z}_c)) + \mathbf{H}_c, \tag{3}$$

where $\mathbf{A}_c \in \mathbb{R}^{L\times N\times D}$ denotes the output after channel information extraction, with Norm indicating the normalized MSA output via batch normalization. To mitigate gradient challenges, the adapter's output is additively fused with $\mathbf{H}_c$, enhancing overall framework stability.

Sequence MSA Transformer (Vaswani et al. 2017) and iTransformer (Liu et al. 2023) models traditionally concentrate on sequence or channel dimensions. Crossformer (Zhang and Yan 2023) introduces a novel approach, applying attention first to the sequence, then the channel dimension. However, its independent two-stage mechanism may result in each MSA only being able to focus on specific aspects of information, reducing the ability to fuse information.

To address this challenge, we present a novel method: the reuse of parameters derived from MSA applied along the channel dimension for sequence modeling. Specifically, the output $\mathbf{A}_c \in \mathbb{R}^{L\times N\times D}$ from the channel MSA undergoes a reshape operation to seamlessly transition into the input $\mathbf{H}_s \in \mathbb{R}^{N\times L\times D}$ for the subsequent sequence MSA. This is designed to establish the interactions between channel and sequence representations and to extract the implicit associations between channels and sequences.

When the input $\mathbf{H}_s$ is fed into the sequence MSA, the output value $\mathbf{Z}_s$ can be formulated as:

$$\mathbf{Z}_s = \mathrm{MSA}(\mathbf{H}_s). \tag{4}$$

The MSA in this study is a shared layer, distinctively applied across different input dimensions. This efficient operation substantially enhances both sequence and channel modeling capabilities while maintaining the number of parameters.

In line with the channel MSA, a subsequent adapter is introduced post-reused MSA layer for sequence feature accommodation. The architecture of this sequence adapter is analogous to that of the channel adapter, described as follows:

$$\mathbf{A}_s = \mathrm{Adapter}(\mathrm{Norm}(\mathbf{Z}_s)) + \mathbf{H}_s, \tag{5}$$

where $\mathbf{A}_s \in \mathbb{R}^{N\times L\times D}$ denotes the output values of the sequence adapter. The integration of this approach is pivotal for imparting distinctiveness to sequence and channel features. The model can attain refined differentiation by utilizing a lightweight adapter, bolstering functionality without substantially increasing complexity.

Models	**CSformer**		iTransformer		PatchTST		Crossformer		SCINet		TiDE		TimesNet		DLinear		FEDformer		Stationary		Autoformer		Informer	
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	**0.385**	**0.400**	0.407	0.410	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	0.481	0.456	0.588	0.517	0.961	0.734
ETTm2	0.282	0.331	0.288	0.332	**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	0.306	0.347	0.327	0.371	1.410	0.810
ETTh1	**0.429**	**0.432**	0.454	0.447	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.747	0.647	0.440	0.460	0.570	0.537	0.496	0.487	1.040	0.795
ETTh2	**0.364**	**0.392**	0.383	0.407	0.387	0.407	0.942	0.684	0.954	0.723	0.611	0.550	0.414	0.427	0.559	0.515	0.437	0.449	0.526	0.516	0.450	0.459	4.431	1.729
Electricity	**0.176**	**0.270**	0.178	**0.270**	0.216	0.304	0.244	0.334	0.268	0.365	0.251	0.344	0.192	0.295	0.212	0.300	0.214	0.327	0.193	0.296	0.227	0.338	0.311	0.397
Solar Energy	**0.230**	0.270	0.233	**0.262**	0.270	0.307	0.641	0.639	0.282	0.375	0.347	0.417	0.301	0.319	0.330	0.401	0.291	0.381	0.261	0.381	0.885	0.711	0.235	0.280
Weather	**0.250**	0.280	0.258	**0.279**	0.259	0.281	0.259	0.315	0.292	0.363	0.271	0.320	0.259	0.287	0.265	0.317	0.309	0.360	0.288	0.314	0.338	0.382	0.634	0.548
1st Count	**6**	**4**	0	3	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1: Multivariate forecasting results. We compare extensive competitive models under different prediction lengths following the setting of iTransformer. The input sequence length is set to 96 for all baselines. Results are averaged from all prediction lengths. The best results are in **bold** and the second best are underlined.

Prediction

After two-stage MSA, we reduce the data's complexity by squeezing the sequence into a lower dimension, forming $\mathbf{Z} \in \mathbb{R}^{N \times (L*D)}$. This step smartly packs the features, keeping the important information. Next, we use a linear layer for the final prediction, getting prediction result $\hat{\mathbf{X}} \in \mathbb{R}^{N \times T}$.

Experiments

In this section, a wide range of experiments was meticulously conducted to evaluate the CSformer model. The experiments were designed with a detailed examination of the rationality of the model's architecture. This thorough evaluation aimed not only to examine the model's performance across various dimensions but also to gain deeper insights into the principles behind its design.

Datasets

To evaluate the performance of CSformer, we employed commonly used datasets for various domains in the multivariate long-term forecasting tasks, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Electricity, and Solar Energy.

Baseline

We have selected a multitude of advanced deep learning models that have achieved SOTA performance in multivariate time series forecasting. Our choice of Transformer-based models includes the Informer (Li, Hui, and Zhang 2021), Autoformer (Wu et al. 2021), Stationary (Liu et al. 2022b), FEDformer (Zhou et al. 2022), Crossformer (Zhang and Yan 2023), PatchTST (Nie et al. 2023), and the leading model, iTransformer (Liu et al. 2023). Notably, Crossformer (Zhang and Yan 2023) shares conceptual similarities with our proposed CSformer. Furthermore, recent advancements in MLP-based models have yielded promising predictive outcomes. In this context, we have specifically selected representative models such as DLinear (Zeng et al. 2023) and TiDE (Das et al. 2023) for our comparative evaluation. Additionally, for a more comprehensive comparison, we also selected TCN-based models, which include SCINet (Liu et al. 2022a) and TimesNet (Wu et al. 2023).

Experimental Setting

We adhered to the experimental configurations employed in iTransformer (Liu et al. 2023). For all datasets, the look-back window length L is set to 96, and the prediction length is designated as $T \in \{96, 192, 336, 720\}$. All experiments were conducted on an NVIDIA A40 GPU, utilizing the PyTorch (Paszke et al. 2019) framework. The mean squared error (MSE) serves as the chosen loss function, with both MSE and mean absolute error (MAE) employed as evaluation metrics to gauge the efficacy of the prediction results.

Main Results

As shown in Table 1, CSformer achieves the best overall performance across all datasets, ranking first in 6 MSE and 4 MAE metrics. This dominance underscores its superiority. Notably, compared to the latest SOTA model, iTransformer (Liu et al. 2023), CSformer further explores Transformer potential, achieving additional gains. It also significantly outperforms Crossformer (Zhang and Yan 2023), likely due to the latter's lack of interaction in cross-sequence and cross-channel extraction. Moreover, CSformer surpasses the channel-independent PatchTST (Nie et al. 2023), emphasizing the importance of effectively capturing inter-variable dependencies.

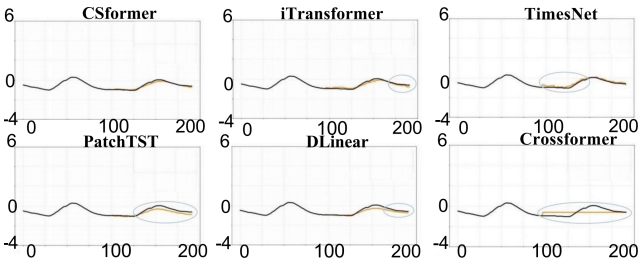


Figure 4: Visualization of input-96-predict-96 results on the ETTm2 dataset.

Figure 4 shows that CSformer outperforms other models in prediction accuracy, stability, detail capture, and trend tracking, likely due to its superior integration of features across sequences and channels.

Ablation Study

Two-stage MSA To evaluate the importance of each MSA component in the two-stage MSA framework, we removed one MSA (including its adapter) for comparative analysis. As evidenced by Table 2, CSformer demonstrates superior overall performance. Notably, retaining only the sequence MSA also yields satisfactory results, attributable to the resulting channel independence.

Datasets	ETTh1		Weather		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
w/o Channel MSA	0.431	0.433	**0.248**	0.278	0.197	0.282
w/o Sequence MSA	0.443	**0.432**	0.259	**0.272**	0.200	0.286
CSformer	**0.429**	**0.432**	0.250	0.280	**0.176**	**0.270**

Table 2: Ablation of two-stage MSA on three real-world datasets with MSE and MAE metrics.

Comparing parameter sharing between channel MSA and sequence MSA with their non-shared counterparts reveals that shared parameters generally yield better prediction outcomes (see Table 3). This improvement is attributed to the interaction between channel and sequence information. Although the performance of the non-shared method does not degrade significantly compared to the performance of the parameter-sharing method, it significantly increases the number of parameters in the model, which may lead to parameter redundancy. Therefore, parameter sharing is a better choice.

Datasets	ETTh1		Weather		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
w/o Parameter Sharing	0.440	0.438	0.256	0.282	0.189	0.280
Parameter Sharing	**0.429**	**0.432**	**0.250**	**0.280**	**0.176**	**0.270**

Table 3: Ablation of parameter sharing on three real-world datasets with MSE and MAE metrics.

Finally, we thoroughly tested the influence of varying the order of application between sequence MSA and channel MSA on the predictive performance of our model.

As shown in Table 4, the experimental results suggests a superior efficacy when channel MSA precedes sequence MSA (C $\rightarrow$ S). This ordered approach facilitates the initial establishment of inter-channel interactions and connections, thereby providing a comprehensive base of information. Such information is invaluable in laying down a relevant understanding of channel dynamics. This initial comprehension of inter-channel relationships significantly enriches the subsequent analysis of temporal evolution of these variables in sequence MSA. This insight highlights the critical nature of the order of operations in MSA applications, especially in the context of enhancing predictive accuracy in complex datasets.

Datasets	ETTh1		Weather		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
S $\rightarrow$ C	0.443	0.442	0.250	0.280	0.178	0.274
C $\rightarrow$ S	**0.429**	**0.432**	**0.250**	**0.280**	**0.176**	**0.270**

Table 4: Ablation of two-stage MSA's order on three real-world datasets with MSE and MAE metrics.

Adapter Then, we analyze the impact of dual adapters on model performance. Table 5 reports results after removing adapters from the architecture, revealing suboptimal performance. This decline stems from the inherent dissimilarity between sequence and variable information, whose interaction is crucial, especially for datasets with many variables like Electricity. The absence of an adapter significantly reduces predictive accuracy, highlighting its role in integrating diverse data characteristics for improved performance.

Datasets	ETTh1		Weather		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
w/o all Adapter	0.434	0.438	0.252	0.282	0.192	0.284
w/o Channel Adapter	0.436	0.436	**0.250**	**0.280**	0.201	0.291
w/o Sequence Adapter	0.429	0.434	**0.250**	0.281	0.192	0.284
CSformer	**0.429**	**0.432**	**0.250**	**0.280**	**0.176**	**0.270**

Table 5: Ablation of adapter on three real-world datasets with MSE and MAE metrics.

Model Analysis

Correlation Analysis We show the results of the correlation visualization in the Weather dataset in Figure 5. Initially, the blocks exhibit exploratory, uniform attention

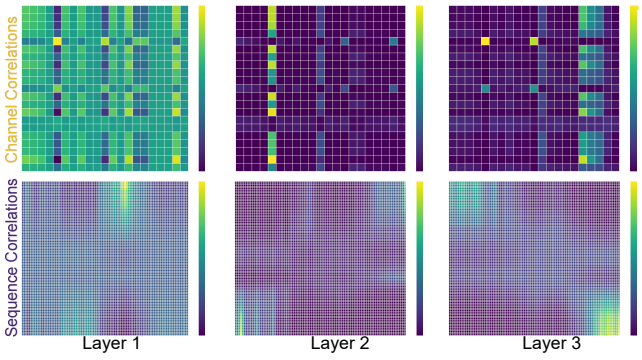


Figure 5: A case visualization of two-stage attention.

distribution. However, as the model deepens, attention becomes more focused on specific channels and sequence dimensions, indicating a growing learning of inter-channel and temporal dependencies. Particularly in the final blocks, heightened attention concentration reveals the model's ability to identify key features and moments. This highlights the cross-dimensional information integration facilitated by shared parameters, crucial for understanding the intrinsic representation mechanisms of deep learning models in multivariate time series analysis.

Patch Embedding vs Dimension-augmented Embedding Dimension-augmented embedding enhances the dimensional representation of sequence data without compromising original information. This approach not only preserves data integrity but also offers processing flexibility, including the option of channel independence or mixing. Such augmentation, a vital preprocessing step, ensures data compatibility with advanced models while enriching its representation to uncover complex patterns and dependencies.

To validate the efficacy of Dimension-augmented Embedding (DE), we replaced the Patch Embedding (PE) in PatchTST (Nie et al. 2023) with Dimension-augmented Embedding and conducted comparisons across three datasets. Figure 6 compares the performance. It can be seen that our proposed embedding method is generalized and effective.

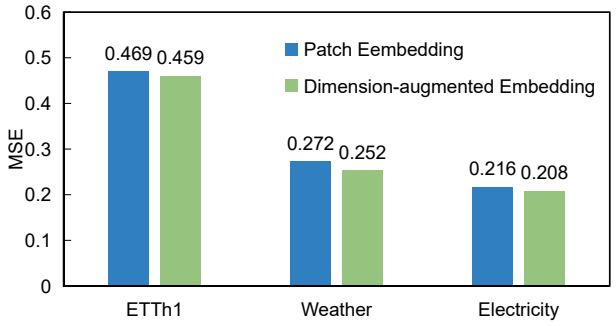


Figure 6: Performance of Patch Embedding and Dimension-augmented Embedding.

Generalization The Two-stage MSA in CSformer enhances the integration of sequence and channel information, enabling the model to learn inherent correlations and improve generalization. To assess this, we trained the model on one dataset and tested it on another, using both same-frequency (e.g., ETTh2 → ETTh1) and different-frequency (e.g., ETTh1 → ETTm1) datasets.

The experimental results, as shown in Table 6, demonstrate that CSformer outperforms other models in various scenarios. We observed that iTransformer has the poorest generalization performance, which can be attributed to its embedding method in the sequence dimension cannot learn robust representations. When cross-domain data distribution changes, the parameters learned in the source domain become inapplicable. The poor performance of PatchTST can be attributed to its use of a channel-independent method, which fails to integrate hidden correlation between channels and sequences. Therefore, the CSformer model exhibits outstanding generalization capabilities.

Dataset	ETTh2 → ETTh1				ETTh1 → ETTm1			
Horizon	96	192	336	720	96	192	336	720
PatchTST	0.491	0.529	0.555	0.627	0.761	0.788	0.777	0.790
iTransformer	0.880	0.920	0.924	0.919	0.923	0.938	0.938	0.945
CSformer	**0.444**	**0.484**	**0.512**	**0.555**	**0.726**	**0.776**	**0.759**	**0.789**

Table 6: Comparison of generalization capabilities between CSformer and other Transformer-based models. "Dataset A → Dataset B" indicates training and validation on the training and validation sets of Dataset A, followed by testing on the test set of Dataset B.

Training Strategy In our earlier discussion, we highlighted the benefits of a training strategy that initially processes channels independently (CI) and then mixes them. This method prevents noise introduced by premature channel mixing (CM) and overcomes the limitations of prolonged independence impeding inter-channel information exchange. To investigate this hypothesis, we conducted an experiment with PatchTST (Nie et al. 2023), modifying the dimension of its attention mechanism. Specifically, we shifted the focus of the attention mechanism from the sequence dimension to the channel dimension to facilitate channel mixing. Table 7 indicates that a reasonable balance of channel independence and mixing enhances predictive performance in multivariate time series forecasting, significantly boosting modeling capabilities. This also provides a new perspective for future research: *how to effectively combine channel independence and mixing?*

Datasets	ETTh1		Weather		Electricity	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
CI	0.469	0.454	0.259	0.281	0.216	0.304
CM	**0.442**	**0.431**	**0.248**	**0.278**	**0.179**	**0.273**
Promotion	5.8%	5.1%	4.2%	1.1%	17.1%	10.2%

Table 7: A comprehensive performance comparison between channel independence and channel mixing is presented. The results represent the average values across all four prediction lengths.

Conclusion and Future Work

In this paper, we summarized the channel-independent and mixing models and show that channel-independent followed by channel mixing is a superior strategy. Based on this, we introduced CSformer, which is an architecture for a two-stage attention mechanism. It balances channel independence and channel mixing to robustly model multivariate time series. In future work, we will investigate more rational ways of combining channel independence and channel mixing and reducing their computational demand.

Acknowledgments

This work is supported by Science and Technology Project of State Grid Corporation of China (Research on Transmission Line Damage Detection and Safety Risk Assessment Technology for Earthquakes and Secondary Disasters, No. 521999230017).

References

Ahmed, U.; Lin, J. C.-W.; and Srivastava, G. 2023. Multivariate time-series sensor vital sign forecasting of cardiovascular and chronic respiratory diseases. *Sustainable Computing: Informatics and Systems*, 38: 100868.

Alshanbari, H. M.; Iftikhar, H.; Khan, F.; Rind, M.; Ahmad, Z.; and El-Bagoury, A. A.-A. H. 2023. On the Implementation of the Artificial Neural Network Approach for Forecasting Different Healthcare Events. *Diagnostics*, 13(7): 1310.

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*.

Bartholomew, D. J. 1971. Time Series Analysis Forecasting and Control.

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. *NeurIPS*.

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*.

Cirstea, R.-G.; Yang, B.; Guo, C.; Kieu, T.; and Pan, S. 2022. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2900–2913. IEEE.

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*.

Dey, R.; and Salem, F. M. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 1597–1600. IEEE.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Graves, A.; and Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hyndman, R.; Koehler, A. B.; Ord, J. K.; and Snyder, R. D. 2008. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media.

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. *ICLR*.

Li, J.; Hui, X.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. *arXiv: 2012.07436*.

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.

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. *NeurIPS*.

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. *NeurIPS*.

Medsker, L. R.; and Jain, L. 2001. Recurrent neural networks. *Design and Applications*, 5(64-67): 2.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mo, S.; Wang, H.; Li, B.; Fan, S.; Wu, Y.; and Liu, X. 2023. TimeSQL: Improving Multivariate Time Series Forecasting with Multi-Scale Patching and Smooth Quadratic Loss. *arXiv preprint arXiv:2311.11285*.

Mo, S.; Wang, H.; Li, B.; Xue, Z.; Fan, S.; and Liu, X. 2024. Powerformer: A temporal-based transformer model for wind power forecasting. *Energy Reports*, 11: 736–744.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *ICLR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NeurIPS*.

Qin, Y.; Luo, H.; Zhao, F.; Fang, Y.; Tao, X.; and Wang, C. 2023. Spatio-temporal hierarchical MLP network for traffic forecasting. *Information Sciences*, 632: 543–554.

Sen, J. 2022. A forecasting framework for the Indian healthcare sector index. *International Journal of Business Forecasting and Marketing Intelligence*, 7(4): 311–350.

Stefenon, S. F.; Seman, L. O.; Aquino, L. S.; and dos Santos Coelho, L. 2023. Wavelet-Seq2Seq-LSTM with attention for time series forecasting of level of dams in hydroelectric power plants. *Energy*, 274: 127350.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *NeurIPS*.

Wang, D.; Gan, J.; Mao, J.; Chen, F.; and Yu, L. 2023. Forecasting power demand in China with a CNN-LSTM model including multimodal information. *Energy*, 263: 126012.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2022. 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. *ICLR*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *NeurIPS*.

Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; Lian, D.; An, N.; Cao, L.; and Niu, Z. 2023. Frequency-domain MLPs are More Effective Learners in Time Series Forecasting. *arXiv preprint arXiv:2311.06184*.

Yin, Y.; and Shang, P. 2016. Forecasting traffic time series with multivariate predicting method. *Applied Mathematics and Computation*, 291: 266–278.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? *AAAI*.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. *ICLR*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *ICML*.