DROSIA: DECOUPLED REPRESENTATION ON SEQUENTIAL INFORMATION AGGREGATION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting is crucial in various fields, including finance, energy consumption, weather, transportation, and network traffic. It necessitates effective and efficient sequence modeling to encapsulate intricate temporal relationships. However, conventional methods often aggregate sequential information into representations of each time point by considering other points in the sequence, thereby ignoring the intra-individual information and suffering from inefficiency. To address these challenges, we introduce a novel approach, **DROSIA**: **D**ecoupled **R**epresentation **O**n **S**equential **I**nformation **A**ggregation, which only integrates temporal relationships once as an additional representation for each point, achieving sequential information aggregation in a decoupled fashion. Thus balancing between individual and sequential information, along with a reduction in computational complexity. We select several widely used time series forecasting datasets, and previously top-performing models and baselines, for a comprehensive comparison. The experimental results validate the effectiveness and efficiency of DROSIA, which achieves state-of-the-art performance with only linear complexity. When provided with sufficiently long input data, the channel-independent DROSIA even outperforms the current best channel-dependent model, highlighting its proficiency in sequence modeling and capturing long-distance dependencies. Our code will be made open-source in the subsequent version of this paper.

1 INTRODUCTION

A time series is a sequence of data points recorded in chronological order, which reflects the attribute characteristics of an object at various stages of its dynamic development. Time series data spans across numerous fields, including finance, energy consumption, weather, transportation, and network traffic. This type of data typically presents high-dimensional features and long sequences, characterized by intricate nonlinear relationships between time points. These complexities make it challenging to predict future developments accurately based on historical data. Consequently, time series forecasting stands as one of the most significant and challenging domains within data analysis, demanding effective and efficient sequence modeling to capture complex temporal relationships.

In recent years, numerous studies on time series forecasting have shown that deep learning methods significantly outperform traditional approaches, elevating deep learning forecasters to the forefront of research. For example, MLP-based models (Oreshkin et al., 2020; Tolstikhin et al., 2021; Zeng et al., 2023; Li et al., 2023; Zhang et al., 2022; Han et al., 2024) have garnered significant interest for their simplicity, efficiency, and predictive accuracy. CNN-based (Bai et al., 2018; Wang et al., 2022; Gao et al., 2020; Sen et al., 2019; Liu et al., 2022; Wu et al., 2022) and RNN-based (Lai et al., 2018; Voelker et al., 2019; Salinas et al., 2020) models have enhanced forecasting effectiveness by integrating local or global spatio-temporal information from time series data. Subsequently, methods based on attention mechanism have emerged as the dominant approach in sequence modeling, empowering numerous deep learning forecasters (Qin et al., 2017) to further refine their temporal relationship capturing capabilities. Particularly, Transformer-based models (Li et al., 2019; Chen et al., 2021; Zhou et al., 2021; Liu et al., 2021; Zhou et al., 2022; Zhang & Yan, 2022; Nie et al., 2022; Liu et al., 2023; Dai et al., 2024), have showcased unparalleled prowess in sequence modeling.

Existing sequence modeling methods typically aggregate sequential information into representations of each time point by considering other points in the sequence, which overlooks the unique information within individual points and may lacks efficiency. For instance, the self-attention mechanism, attends to all time points to update the current one, leading to a quadratic computational complexity that can become a bottleneck in the training and the inference processes (Dao et al., 2022). Additionally, the distinct information within each point can be compromised during sequence modeling. However, "the 'structure' (sedimented individual meanings) is powerful" (Fine, 1993). Inspired from the concept of Transverse Interaction: Individuals recognize the physical environment as a symbolic other and use this understanding to structure their interaction with a "generalized other" (Weigert, 1991). we propose a sociological perspective on the relationship between time series and individual points, which emphasizes that individual information is of great significance and necessitates a full interaction with the collective to enhance sequence modeling. Current methods, however, may overly sacrifice individual information for the sake of sequential information.

To illustrate our concept and address the limitations of current sequence modeling methods, we have developed a novel approach called DROSIA, which integrates rich temporal relationships as additional representations for each time point, thereby enhancing the expressive power of the data and better balancing the trade-off between sequential information and individual point information. We have conducted comprehensive experiments on several prominent and frequently used multivariate long-term time series forecasting datasets. DROSIA has demonstrated exceptional sequence modeling capabilities, and the results suggest that our proposed model attains state-of-the-art performance in downstream tasks while notably decreasing computational complexity. The contributions of this paper can be summarized as follows:

- We propose a novel sequence modeling method – DROSIA, which aggregates sequential information in a decoupled fashion, effectively balancing it with information of individuals.
- DROSIA exhibits exceptional proficiency in time series forecasting, achieving state-of-the-art performance with linear complexity, especially in experiments involving long sequences and large datasets, highlighting its efficacy in capturing long-distance dependencies.
- Compared with several previous state-of-the-art channel-dependent methods, the channel-independent DROSIA demonstrates superior performance across all datasets, when input length is adequate. Note that DROSIA does not leverage any inter-channel information.

2 Related Work

Sequential Information Aggregation Methods. Sequence information aggregation, or sequence modeling, is a pivotal technology across various fields, including natural language processing, speech recognition, and time series analysis. RNNs (Elman, 1990) process sequential information through recursive computations. LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014) are two most commonly employed variants, which effectively manage the forgetting and retention of information via gating mechanisms, thereby mitigating the challenges traditional RNNs encounter when learning long-distance dependencies. RCNN (Girshick et al., 2014; Gu et al., 2021) leverages the strengths of both RNNs and CNNs (LeCun et al., 1998), extracting local features through convolutional operations before aggregating information via recursive computations.

Subsequently, the attention mechanism has become the dominant technology for sequence modeling. Traditional models have been bolstered by the integration of attention mechanisms (Qin et al., 2017), and the Transformer (Vaswani et al., 2017), which is built on self-attention, has seen remarkable success across a wide range of tasks. However, the attention mechanism has drawbacks in terms of computational efficiency. Its high computational cost can be a significant barrier for many researchers and engineers, thereby hindering its widespread adoption and dissemination.

Time Series Forecasting Models. In recent years, deep networks have advanced significantly in time series forecasting. RNN-based models (Lai et al., 2018; Voelker et al., 2019; Salinas et al., 2020) are effective in capturing temporal relationships but suffer from computational inefficiency and limited capability in modeling long-distance dependencies. CNN-based models (Bai et al., 2018; Wang et al., 2022; Gao et al., 2020; Sen et al., 2019; Liu et al., 2022; Wu et al., 2022), which perform convolutional operations to hierarchically extract features for temporal patterns, have achieved competitive forecasting performance. MLP-based models (Oreshkin et al., 2020; Tolstikhin et al., 2021;

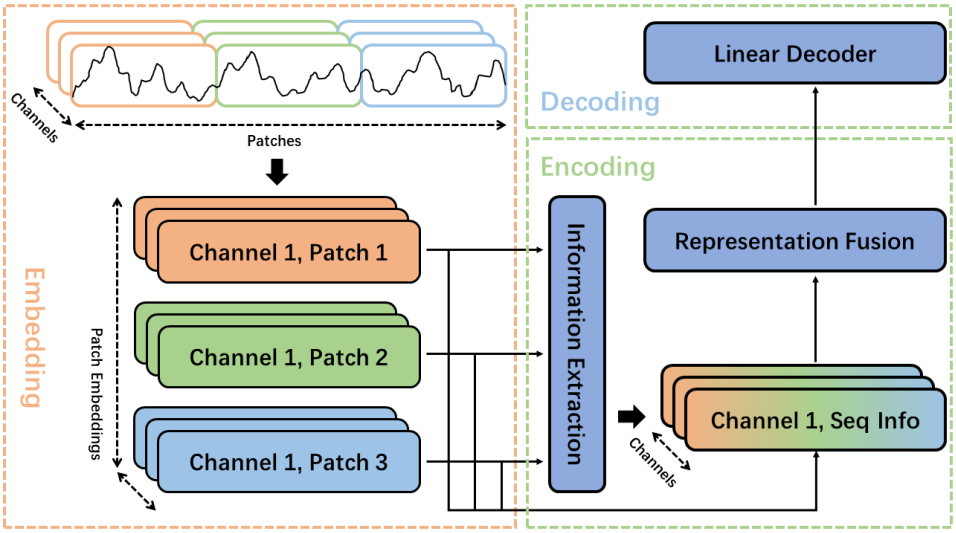


Figure 1: Overall architecture of DROSIA model, includes patch embedding, DROSIA encoding, and linear decoding. Note that DROSIA encoder could be repeatedly used. Information extraction extracts sequential information from all patch embeddings in the same channel, and fuses it with these embeddings in a decoupled manner. We will describe the details in the following of this paper.

Zeng et al., 2023; Li et al., 2023; Zhang et al., 2022; Han et al., 2024) have garnered considerable interest due to their efficient data processing and ability to capture temporal relationships.

Inspired by the capabilities of Transformer-based models (Li et al., 2019; Chen et al., 2021; Zhou et al., 2021; Liu et al., 2021; Zhou et al., 2022; Zhang & Yan, 2022; Nie et al., 2022; Liu et al., 2023; Dai et al., 2024) in capturing long-distance dependencies and complex temporal relationships, they have been extensively applied across various time series tasks. Prior research has largely centered on point-wise modeling. However, due to the computational complexity of Transformer, numerous studies have sought to enhance efficiency. The PatchTST (Nie et al., 2022) has demonstrated the advantages of representing time series through patching, effectively reducing sequence length while boosting forecasting performance. Nevertheless, Transformer-based methods still struggle with efficiency in multivariate long-term prediction scenarios. iTransformer (Liu et al., 2023) approaches the problem by representing each channel as a whole along the time axis and applying the Transformer encoder to these representations, which significantly reduces complexity but at the cost of losing temporal information, leading to suboptimal performance in cases with fewer channels and longer input sequences. TimeXer (Wang et al., 2024) leverages the benefits of both PatchTST and iTransformer, achieving promising results, yet the computational time remains a significant drawback.

3 METHODOLOGY

Time series can be defined as $X = \{x_1, x_2, \ldots, x_t\}, x \in \mathbb{R}^d$, where t represents the current time point, starting from 1, and d denotes the dimensionality of the features at each time point. The objective of time series forecasting is to predict the sequence $Y = \{x_{t+1}, x_{t+2}, \ldots, x_{t+h}\}$, with h being the prediction horizon. We propose a novel method called **D**ecoupled **R**epresentation **O**n **I**nformation **A**ggregation, abbreviated as DROSIA, which comprises three components: patch embedding, DROSIA encoding, and linear decoding. The overall architecture is depicted in Figure 1.

3.1 PATCH EMBEDDING

To enhance prediction accuracy and computational efficiency, we define a sliding window of length k as $T_i = \{x_{i+1}, x_{i+2}, \ldots, x_{i+k}\}$ with a stride of s, to segment the time series into patches. We utilize a fully connected linear layer for patch embedding, which takes each patch as input and

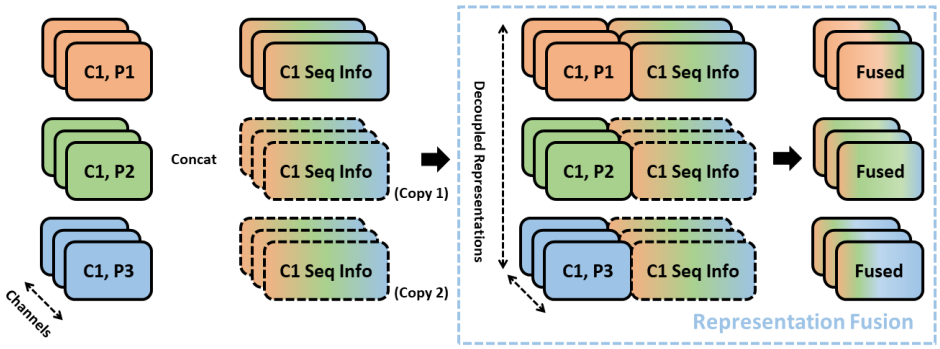


Figure 2: DROSIA encoder concatenates the patch embeddings and extracts sequential information from them. This information is duplicated and combined with the original patch embeddings to create decoupled representations, which are then fused back to the original dimensionality.

produces a single vector as the patch's representation, referred to as S_i for the i-th patch.

$$S_i = Linear(T_i), i = 1, 2, \ldots, n \tag{1}$$

In Equation (1), n represents the total number of input patches. The linear layer treats the multivariate time series as multiple univariate series (in a channel-independent manner) (Han et al., 2023), multiplying the k values within the sliding window by a matrix with dimensions $k \times d$, where d is the dimensionality of patch embedding. This approach aligns with the methodology of PatchTST (Nie et al., 2022). which has been shown the advantages in long-term time series forecasting tasks across various related studies. After extracting patch-wise representations from the time series, PatchTST utilizes the Transformer (Vaswani et al., 2017) to encode these embeddings. In contrast, we employ DROSIA as the encoder. The following subsection will delve into the implementation details of DROSIA and highlight its distinctions from the self-attention mechanism and other methods.

3.2 DROSIA Encoding

The DROSIA encoding module extracts sequential information from the patch embeddings, serving as additional representations of these patches, and then fuses the two back to original dimensionality of embeddings. In multi-layer networks, this process can be repeated, indicating that the fused representation can either be passed through another encoding layer or directly input into the decoder.

$$S^{j+1} = DROSIA(S^j), j = 1, 2, \ldots, l \tag{2}$$

Equation (2) outlines the overall process of the DROSIA encoder, which will be described in detail from Equation (3) to Equation (7). In this context, $DROSIA$ refers to a single encoder layer, l denotes the number of layers. S^1 indicates the input to the first encoder layer, meanwhile the output of the embedding layer. S^j is the input to the j-th layer. We consider sequential information as additional representation of the input, to achieve representation decoupling. The encoder primarily comprises three stages: sequence aggregation, information extraction, and representation fusion.

Sequence Aggregation. The output representations from the patch embedding or the previous layer of DROSIA encoder are first concatenated, which we refer to as sequence aggregation.

$$C^j = S_i^j \circ S_{i+1}^j \circ \cdots \circ S_{i+k}^j \tag{3}$$

In Equation (4), the $\circ$ represents the concatenate operation. The high-dimensional representation resulting from this concatenation is rich in temporal information, which must be fully exploited to enhance the model's overall performance in the sequence modeling process.

Information Extraction. The information extraction phase is applied to the high-dimensional representations derived from the sequence aggregation stage. Its objective is to distill more valuable sequential information for subsequent tasks while decreasing the computational complexity. For this purpose, We employ a simple and efficient MLP for the information extraction process.

$$R^j = MLP(C^j) \tag{4}$$

The high-dimensional representations are compressed into a lower-dimensional space to form the sequential information, thereby reducing the number of parameters. Note that we use an MLP just because its simplicity, however, it could be replaced by more sophisticated methods if desired.

Representation Fusion. The extracted sequential information is concatenated with the original patch embeddings or the outputs from the previous encoder, as illustrated in Figure 2. This process resembles residual connection He et al. (2016), but in a decoupled manner, which enhances the information representation capability and improves efficiency, while also facilitating full interaction between the two types of information and optimization for deeper network. Subsequently, the fused representations of patches and sequential information are processed through a normalization layer, where both parts undergo a unified normalization operation. The function is outlined as follows.

$$D = LayerNorm(S^j \circ R^j) \tag{5}$$

$$LayerNorm(H) = \frac{h_i - Mean(H)}{\sqrt{Var(H)}} \tag{6}$$

In Equation (5) and (6), $LayerNorm$ refers to the normalization operation. The H represents the input, while h_i denotes the i-th item of H. $Mean$ and Var are functions to compute the mean and the variance respectively. Normalization (Kim et al., 2021) aids in optimizing training phase and mitigates the adverse effects of non-stationary processes, which are common in time series data.

Unlike conventional sequence modeling approaches, DROSIA extracts sequential information once per encoder layer, and then aggregates it with the original patch embeddings in a decoupled fashion. For instance, self-attention mechanisms attend to all time points and aggregates sequential information through a weighted sum of points' representations, potentially overlooking individual information and lacks efficiency. In contrast, DROSIA considers sequential information as additional representation and decouples the two, thereby preserving the benefits of both sequential and individual information while circumventing issues such as the quadratic computational complexity.

$$S^{j+1} = FFN(D) \tag{7}$$

Ultimately, we utilize a feed-forward network to facilitate complete interaction between the two types of information, and compress the fused representation to the dimensionality of the input data.

3.3 Linear Decoding

Once the data has passed through l layers of DROSIA encoder, the output from the final layer, denoted as s^{l+1}, is then fed into the linear decoding module to yield the final forecasting results.

$$\hat{Y} = Projection(S^{l+1}) \tag{8}$$

In Equation (3), the $Projection$ is performed using a fully connected linear layer. During the training phase, the model's prediction results are compared against the actual subsequent time series data to compute the error. Subsequently, the parameters of DROSIA are updated using the backpropagation algorithm. The error is quantified using the mean squared error (MSE). Our configuration of the decoding module aligns with numerous previous studies, including PatchTST and iTransformer.

4 Experiments

Datasets. We comprehensively assessed the performance of the DROSIA model on eight multivariate long-term time series forecasting datasets: Electricity (ECL), four subsets of ETT (ETTm1, ETTm2, ETTh1, and ETTh2), Traffic, Exchange, and Weather. These datasets are publicly available on GitHub[1]. The data processing and split ratio were consistent with TimesNet(Wu et al., 2022).

Baselines. We selected several previous state-of-the-art models, including Transformer-based models such as iTransformer(Liu et al., 2023), PatchTST(Nie et al., 2022), FEDformer(Zhou et al., 2022), and Autoformer(Chen et al., 2021). CNN-based model, TimesNet(Wu et al., 2022), and MLP-based models, TiDE(Das et al., 2023) and DLinear(Zeng et al., 2023). All models were implemented using the original code or replicated adhering to details described in the respective papers.

[1] https://github.com/thuml/Time-Series-Library

Settings. By default, we configure all Transformer-based models with dropout probability $p = 0.1$ and the number of attention heads $n = 16$. For PatchTST and DROSIA, the patch size is 16 with a stride as 8, in line with previous research. When conducting experiments on Weather, ECL, and Traffic, both DROSIA and Transformer-based models are equipped with 3 encoder layers, and latent dimension $d = 512$. For smaller datasets, such as Exchange and ETT subsets, we employ a smaller model size to mitigate the risk of overfitting: 2 layers and $d = 256$. The dimension ratio of the two types of representations (individual versus sequential) within DROSIA is $1 : 1$ across all scenarios.

Table 1: Overall experimental outcomes for long-term time series forecasting, using four prediction horizons: $H \in \{96, 192, 336, 720\}$ across all datasets, and the length of the input $L = 96$, which are consistent with iTransformer (Liu et al., 2023). The results are averaged from these four horizons.

Model	DROSIA		iTransformer		PatchTST		TiDE		TimesNet		DLinear		FEDformer		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.441**	**0.435**	0.464	0.455	0.443	0.438	0.491	0.469	0.496	0.494	0.462	0.458	0.464	0.468	0.499	0.484
ETTh2	**0.376**	**0.401**	0.384	0.407	0.378	0.402	0.401	0.419	0.422	0.429	0.513	0.519	0.432	0.447	0.450	0.458
ETTm1	**0.383**	**0.396**	0.408	0.412	0.388	0.400	0.424	0.416	0.432	0.430	0.404	0.408	0.446	0.452	0.579	0.510
ETTm2	**0.277**	**0.322**	0.291	0.334	0.280	0.326	0.292	0.334	0.304	0.339	0.354	0.402	0.298	0.345	0.318	0.362
Exchange	0.351	**0.398**	0.364	0.407	0.362	0.408	0.364	0.409	0.405	0.445	**0.339**	0.414	0.507	0.496	0.504	0.501
Weather	**0.255**	**0.277**	0.260	0.280	0.256	0.278	0.273	0.322	0.262	0.287	0.265	0.316	0.302	0.333	0.322	0.357
ECL	0.190	0.278	**0.185**	**0.274**	0.196	0.283	0.257	0.344	0.192	0.294	0.215	0.304	0.216	0.328	0.228	0.335
Traffic	0.479	**0.312**	**0.467**	0.314	0.486	0.322	0.759	0.473	0.629	0.343	0.643	0.400	0.621	0.379	0.637	0.383

To reduce the impact of randomness, each experiment is conducted three times to calculate the average result. The outcomes from different prediction horizons are then further averaged for each dataset. The mean squared error (MSE) and mean absolute error (MAE) serve as the evaluation metrics. All experiments are performed on a single NVIDIA 4090 GPU with 24GB of memory.

Table 2: Experiments on ECL and Traffic for a fair comparison between DROSIA and iTransformer, involving various lengths of input time series $L \in \{96, 192, 336, 512\}$, and different output horizons: $H \in \{96, 192, 336, 720\}$. Results in **bolded red** indicate the winner in each scenario.

Length		512				336				192				96			
Model		DROSIA		iTransformer		DROSIA		iTransformer		DROSIA		iTransformer		DROSIA		iTransformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.131**	**0.229**	0.135	0.234	**0.134**	**0.232**	0.136	0.234	**0.141**	**0.237**	**0.141**	**0.237**	0.166	0.253	**0.158**	**0.249**
	192	**0.150**	**0.246**	0.154	0.253	**0.151**	**0.247**	0.155	0.251	**0.158**	**0.253**	0.159	0.254	0.176	0.264	**0.170**	**0.260**
	336	**0.167**	**0.266**	0.170	0.269	**0.170**	**0.266**	0.173	0.269	**0.176**	**0.270**	0.177	0.273	0.193	0.282	**0.187**	**0.278**
	720	**0.203**	**0.295**	0.206	0.300	**0.208**	**0.298**	0.210	0.301	**0.215**	**0.304**	0.216	0.306	0.232	0.316	**0.224**	**0.310**
Traffic	96	**0.371**	**0.263**	0.395	0.289	**0.381**	**0.268**	0.401	0.291	**0.401**	**0.276**	0.421	0.298	0.454	**0.299**	**0.434**	**0.299**
	192	**0.389**	**0.271**	0.415	0.301	**0.402**	**0.277**	0.423	0.306	**0.422**	**0.285**	0.443	0.310	0.466	**0.305**	**0.454**	0.306
	336	**0.400**	**0.276**	0.430	0.311	**0.419**	**0.285**	0.441	0.314	**0.438**	**0.293**	0.459	0.316	0.483	**0.313**	**0.472**	0.315
	720	**0.436**	**0.296**	0.472	0.335	**0.446**	**0.300**	0.476	0.336	**0.466**	**0.309**	0.489	0.337	0.515	**0.332**	**0.507**	0.337

4.1 Experimental Results

The overall experimental results are presented in Table 1. The **bolded** values denote the top performance on each dataset, while the underlined indicate the second-highest. As observed, DROSIA achieves superior or competitive results compared with the previous state-of-the-art models and outperforms each baseline in the majority of scenarios. However, on datasets with a large number of variates, such as ECL (321 channels) and Traffic (862 channels), the channel-independent DROSIA does not outperform the channel-dependent iTransformer model. This comparison is not entirely equitable to DROSIA, as the number of channels significantly exceeds the length of the input data.

Consequently, we adjust the input lengths for ECL and Traffic datasets to facilitate fairer comparisons between DROSIA and iTransformer. As shown in Table 2, when the input data length is sufficiently long ($L \geq 192$), DROSIA consistently outperforms iTransformer on datasets with a large number of variates, without utilizing any inter-channel information. This outcome underscores the powerful capability of DROSIA in time series modeling and capturing long-distance dependencies.

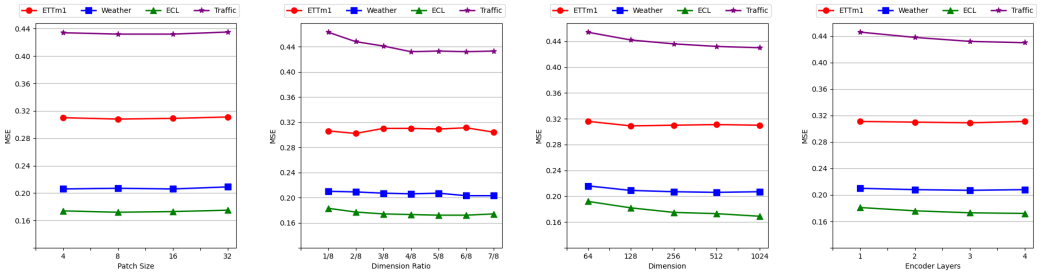
As shown in Table 1, DROSIA significantly outperforms the MLP-based methods, TiDE and DLinear, across most scenarios. For the Exchange dataset, which comprises only 8 channels and is subject to a high degree of data randomness, DROSIA still achieves the best performance in MAE and ranks second in MSE. In comparison to Transformer-based and CNN-based models, DROSIA consistently exceeds the performance of FEDformer, Autoformer, TimesNet, and PatchTST, and demonstrates superior behavior to iTransformer in datasets with a small number of variates.

Table 3: Efficiency comparisons between DROSIA and various typical time series forecasters with the computational complexity, which is consistent with (Han et al., 2024). DROSIA is the only method that is linear to the input length L, prediction horizon H, and number of channels C.

	DROSIA	iTransformer	PatchTST	Transformer
Complexity	$O(CL + CH)$	$O(CL + C^2 + CH)$	$O(CL^2 + CH)$	$O(CL + L^2 + HL + CH)$

4.2 Ablation Study

Efficiency Analysis. We assessed the efficiency of DROSIA against various typical forecasters. DROSIA mainly comprises patch embedding, information extraction, representation fusion, and linear decoding modules. Assuming an input length L, a number of channels C, a patch size p with a stride s, model dimension is d with ratio of two types of information $1:1$ ($d/2$ for each), and prediction horizon is H. The computational complexities are $O(CpdL/2s)$, $O(Cd^2L/4s)$, $O(Cd^2L/2s)$, and $O(CHd/2)$ respectively. By ignoring all constants, we derive the overall computational complexity of DROSIA as $O(CL + CH)$, which is linear to L, C, and H. The complexity of other models was also computed in this way, as presented in Table 3. DROSIA stands out as the only method with linear complexity to L and C of time series data, demonstrating its high efficiency for time series forecasting tasks, particularly in scenarios of large variate sizes and long input lengths.



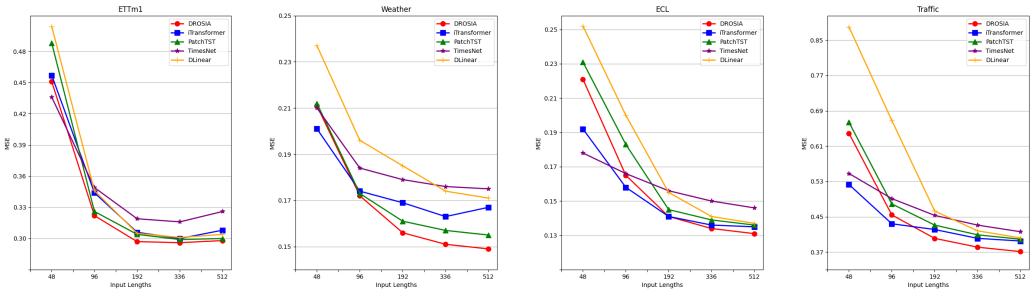
(a) (b) (c) (d)

Figure 3: Hyperparameter sensitivity analysis of DROSIA. Four datasets with different variate size are adopted: ETTm1, Weather, ECL and Traffic, with the patch sizes: $p \in \{4, 8, 16, 32\}$, dimension ratio of information within each time patch: $r \in \{1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8\}$, model dimension: $d \in \{64, 128, 256, 512, 1024\}$, and number of encoder layers: $n \in \{1, 2, 3, 4\}$.

Hyperparameter Sensitivity Analysis. We selected four datasets with varying numbers of channels: ETTm1 (7 channels), Weather (21 channels), ECL (321 channels), and Traffic (862 channels), and conducted a sensitivity analysis on several key hyperparameters of DROSIA, which include the patch size p, the dimension ratio of patch embeddings r, the model dimension: d, and the number of encoder layers n. To ensure fairness and avoid bias due to an excessively large patch size, we set the input length to 192. For all scenarios, we used Mean Squared Error (MSE) as the evaluation metric. All other settings were aligned with the default experimental configurations. The results of this analysis are depicted in Figure 3.

The analysis reveals that variations in patch size have a negligible impact on the overall performance of DROSIA across all datasets. For the dimension ratio, model dimension, and number of encoder layers, datasets with a large number of variates, such as ECL and Traffic, benefit from increased values of r, d and n to achieve improved prediction performance. Conversely, for smaller scale datasets like ETTm1 and Weather, DROSIA does not derive significant advantages from larger values of these hyperparameters, in some cases, the performance even deteriorates.

Influence of Input Length. We selected four datasets with varying variate sizes: ETTm1, Weather, ECL, and Traffic, to conduct a detailed analysis on the impact of input length. For comparison, we adopted four baselines, which include Transformer-based models iTransformer and PatchTST, CNN-based model TimesNet, and MLP-based model DLinear. It should be noted that iTransformer and TimesNet are channel-dependent models, whereas the others are channel-independent.



(a) (b) (c) (d)

Figure 4: Ablation study of prediction performance of DROSIA, iTransformer, PatchTST, DLinear, and TimesNet. Four datasets with different variate sizes are adopted: ETTm1, ECL and Traffic, with varying input lengths: $L \in \{48, 96, 192, 336, 512\}$, and the prediction horizon $H = 96$.

As depicted in Figure 4, the DROSIA model demonstrates its superior effectiveness across all scenarios when compared to channel-independent models like PatchTST and DLinear. It achieves the best performance on all datasets with longer input time series data lengths ($L \geq 192$), even outperforming channel-dependent models such as TimesNet and iTransformer. For datasets with larger variate sizes like ECL and Traffic, TimesNet and iTransformer exhibit superior performance when the input length is set to 48. However, their advantage diminishes and is eventually overtaken as the input length increases. This trend suggests the value of inter-channel information in time series forecasting and highlights a limitation of channel-dependent models in capturing long-distance dependencies. The question of how to better balance sequential and inter-channel information warrants further investigation. Moreover, the performance of DROSIA is consistent and progressively improves with increasing input length, ultimately achieving state-of-the-art forecasting accuracy. This trend already attests to the model's robust capability in sequence modeling.

Table 4: Experiments on the decoupled representations, which encompass three cases: "P+S" indicates the inclusion of both patch and sequential representations, while "P" or "S" signifies only one respectively. The "P+S" configuration of PatchTST means the patch representations and sequential information extracted via Self-Attention are concatenated, whereas "S" refers to the model's original settings. To mitigate the randomness in the results, we utilized two large datasets (ECL and Traffic).

Model		DROSIA				PatchTST				P	
		P+S		S		P+S		S			
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.166**	**0.253**	0.184	0.277	**0.177**	**0.267**	0.183	0.274	0.174	0.282
	192	**0.176**	**0.264**	0.194	0.287	**0.187**	**0.273**	0.192	0.279	0.198	0.287
	336	**0.193**	**0.282**	0.211	0.302	**0.201**	**0.292**	0.205	0.296	0.213	0.301
	720	**0.232**	**0.316**	0.254	0.335	**0.241**	**0.317**	0.246	0.324	0.255	0.334
Traffic	96	**0.454**	**0.299**	0.485	0.324	**0.474**	**0.306**	0.479	0.312	0.576	0.370
	192	**0.466**	**0.305**	0.502	0.335	**0.481**	**0.309**	0.485	0.315	0.550	0.354
	336	**0.483**	**0.313**	0.522	0.346	**0.492**	**0.314**	0.496	0.321	0.564	0.358
	720	**0.515**	**0.332**	0.558	0.363	**0.523**	**0.329**	0.525	0.339	0.603	0.377

Effectiveness of DROSIA. We investigate the role that decoupled representations play in the overall performance of DROSIA and the efficacy in time series forecasting. To mitigate the randomness in experimental outcomes, we selected two datasets with the largest variate sizes: ECL and Traffic, for comparison. PatchTST, which employs the patch embedding and Transformer encoder to integrate sequential information, was chosen as the benchmark. As indicated in Table 4, DROSIA achieves the lowest MSEs and MAEs across all prediction horizons. When using the DROSIA approach (labeled as "P+S"), the performance of both models surpass that of the setting where only sequential information is utilized across all scenarios. This finding validates the effectiveness of DROSIA in aggregating sequential information and significantly enhances the model's prediction accuracy.

Table 5: Experiments for five different sequential information extraction methods: MLP (ours), Self-Attention, CNN, RNN, and Max Pooling of DROSIA, with the prediction horizon $H = 96$, and the length of input time series data $L = 96$ for all of the 8 datasets. The **bolded** values denote the best performance, and underlined values denote the second-best, which are the same as Table 1.

Model	MLP		Self-Attention		CNN		RNN		Max Pooling	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.376	0.397	0.381	0.399	**0.375**	0.394	0.377	**0.393**	0.384	0.396
ETTh2	**0.291**	0.341	0.294	0.341	0.292	0.341	0.295	0.342	**0.291**	**0.340**
ETTm1	**0.319**	0.360	0.320	**0.359**	0.330	0.367	0.334	0.369	0.328	0.365
ETTm2	0.175	0.258	**0.174**	0.257	**0.174**	**0.256**	0.176	0.260	0.175	0.258
Exchange	**0.082**	0.200	0.083	**0.199**	0.083	0.201	0.083	0.200	0.084	0.202
Weather	0.173	0.214	**0.171**	**0.213**	0.172	0.214	0.173	0.214	0.175	0.215
ECL	**0.166**	**0.253**	0.177	0.267	0.178	0.272	0.181	0.272	0.179	0.266
Traffic	**0.454**	**0.299**	0.474	0.306	0.476	0.322	0.467	0.304	0.479	0.318

Different Information Extraction Methods. As discussed in Section 3.2, we utilize an MLP for sequential information extraction primarily due to its simplicity, however, it could be substituted with any methods. To evaluate the impact of various information extractors on the overall effectiveness of DROSIA, we compare five methods: MLP, Self-Attention, CNN, RNN, and Max Pooling. CNN refers to a single convolutional layer, and RNN denotes the vanilla version in this context. We employ all eight datasets for this comparison, with both the input length and the prediction horizon of the time series data set to 96. The outcomes of these experiments are presented in Table 5.

The results indicate that each sequential information extraction method could excel on different datasets with smaller variate sizes, such as the four subsets of ETT, Exchange, and Weather. Overall, the Self-Attention method slightly outperforms the others in these cases. However, for datasets with larger variate sizes, such as ECL and Traffic, the MLP proves to be more effective. These findings suggest that for smaller datasets, there is minimal distinction between various extractors, which underscores the universal effectiveness of DROSIA and the inherent data variability across these datasets. In contrast, more complex datasets necessitate more advanced sequential information extraction methods to achieve optimal performance.

5 Conclusion and Future Work

This paper introduces a novel approach, DROSIA, which incorporates rich temporal relationships as additional representations within each time patch. This method achieves sequential information aggregation in a decoupled fashion, effectively balancing sequential and individual information with linear complexity for sequence modeling. Through comprehensive experimentation, we show that DROSIA attains state-of-the-art performance, particularly in scenarios involving long sequences and large scale data. Compared with previous top-performing channel-dependent models like iTransformer, the channel-independent DROSIA exhibits superior performance across all datasets when the input sequence length is adequate. Notably, DROSIA does not rely on inter-channel information, highlighting its efficacy in sequence modeling and capturing long-distance dependencies. We contend that DROSIA is broadly applicable to a variety of scenarios.

In the ablation study, we have thoroughly demonstrated the efficacy of DROSIA through a multitude of meticulously designed experiments. However, we also observed that when the input length of time series is inadequate and the dataset has a large variate size, the prediction accuracy of DROSIA may fall short of channel-dependent methods. This underscores the significance of inter-channel information. Consequently, our future research will concentrate on integrating inter-channel information without excessively compromising the information within each channel, while also considering the model's overall efficiency to achieve a better balance. Through additional experiments (not detailed in this paper), we have verified that inter-channel information significantly diverges from sequential information, necessitating distinct integration strategies. Simply applying DROSIA to inter-channel information aggregation may not be feasible. Overall, this paper presents a successful method for enhanced intra-channel modeling and identifies a challenging research direction in time series forecasting: how to efficiently model both intra- and inter-channel information simultaneously.

REFERENCES

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12270–12280, 2021.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. Periodicity decoupling framework for long-term series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

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

Gary Alan Fine. The sad demise, mysterious disappearance, and glorious triumph of symbolic interactionism. *Annual review of sociology*, 19(1):61–87, 1993.

Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545*, 2020.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *arXiv preprint arXiv:2304.05206*, 2023.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Yunzhong Qiu, Haoran Zhang, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *arXiv preprint arXiv:2402.19072*, 2024.

Andrew J Weigert. Transverse interaction: A pragmatic perspective on environment as other. *Symbolic Interaction*, 14(3):353–363, 1991.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, pp. 11121–11128, 2023.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2022.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.