SDformer: Fusing Series Decomposition for Superior Long-term Time Series Forecasting

Jiayi Li, Zihang Zhang, *Member, IEEE*, Chao Zhang *Member, IEEE*, Jun Tang, and Shangce Gao, *Senior Member, IEEE*

***Abstract*—Long-term time series forecasting is crucial in numerous real-world dynamic systems and has garnered extensive research attention. In the context of time series forecasting, time series decomposition serves as an effective tool for analyzing time series data, enabling the extraction of underlying patterns and trends that enhance predictive accuracy. Despite its potential, time series decomposition has been underutilized in existing models that incorporate decomposition architectures, particularly in feature extraction. To address this gap, we propose the Series Decomposition Encoder (SDE) block, which separates time series data into seasonal and trend components. By leveraging these decomposed elements, the SDE block enhances the model's ability to capture essential temporal features. We substitute the initial layer of the conventional Transformer architecture with SDE, thereby introducing the Series Decomposition Transformer (SDformer). Empirical assessments on nine benchmark datasets substantiate that our proposed SDformer attains state-of-the-art performance in long-term forecasting. The code implementation is available at the provided repository: https://github.com/Rirock/SDformer.**

***Index Terms*—Time series decomposition, Time series forecasting, Long time series forecasting, Transformer**

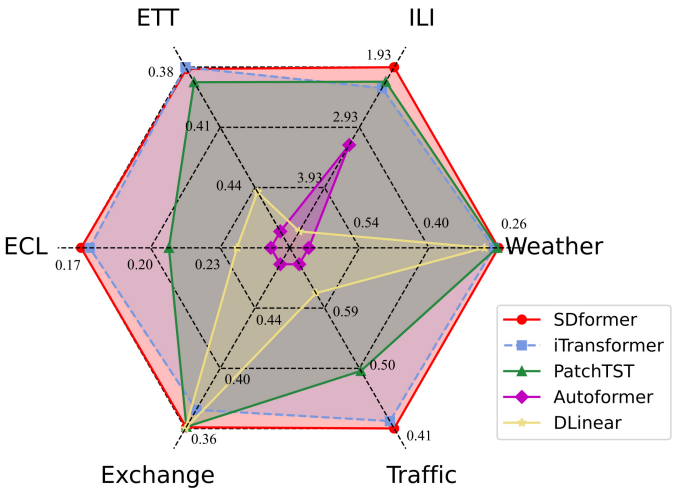

Fig. 1. Performance comparison of SDformer and other models, detailed data is shown in Table II.

I. Introduction

Long-term time series forecasting holds substantial practical significance across a myriad of real-world applications, including traffic management, weather forecasting, and energy consumption. For instance, accurate traffic prediction can help alleviate congestion and improve urban planning, while precise weather forecasts are crucial for agriculture, disaster preparedness, and daily life activities. In energy consumption, effective forecasting can lead to optimized grid operations and energy savings, which are essential for sustainability and cost reduction [1]–[4].

Traditional forecasting methods, such as Autoregressive Integrated Moving Average (ARIMA) and Vector Autoregression (VAR), are commonly used but struggle with complex, non-linear, and high-dimensional data, limiting their applicability for long-term forecasting scenarios [5]. Deep learning approaches, particularly Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), address non-linear temporal dependencies effectively. However, they encounter computational inefficiency and difficulties in modeling long-term dependencies within extensive sequences [6].

Since the significant advancements made by Transformer-based models in long-term time series forecasting [7], the majority of newly proposed deep learning models have adopted an Encoder-Decoder architecture. These models generally integrate mechanisms such as CNNs, linear layers, or time series decomposition to enhance performance [8]–[12]. Recent studies suggest that the Encoder-only structure may offer greater advantages in time series forecasting. For instance, PatchTST divides input data into patches and feeds them into the Encoder for feature extraction, which are then directly utilized for future prediction using fully connected layers [13]. Similarly, iTransformer reverses the dimensions of the data, enabling the attention mechanism to capture multivariate correlations among variables. Empirical evidence from iTransformer also demonstrates superior performance by retaining only the Encoder layers [14].

Time series decomposition is a widely used method for analyzing time series data [15]. It involves breaking down a time series into multiple components such as trend, seasonality, and residual components [16], [17]. By applying different analysis techniques to these components, a better understanding of the structure and features of the time series can be achieved. Time series decomposition is often integrated into various deep

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grants JP25K21298 and JP25K03179, and Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145. (Corresponding author: Shangce Gao)

J. Li, Z. Zhang, C. Zhang and S. Gao are with the Faculty of Engineering, University of Toyama, Toyama-shi, 930-8555, Japan. (E-mail: lijiayi212@gmail.com; zhangzh@eng.u-toyama.ac.jp; zhang@eng.u-toyama.ac.jp; gaosc@eng.u-toyama.ac.jp)

J. Tang is with the Wicresoft Co Ltd, 13810 SE Eastgate Way, Bellevue, WA 98005, USA. (E-mail: juntang@wicresoft.com)

learning models used for time series forecasting to enhance the models' reasoning and generalization capabilities [18].

However, a critical limitation remains as most existing Transformer-based models apply decomposition techniques only within the Decoder, which overlooks the potential of enhancing temporal feature extraction in the Encoder. Recent Transformer-based models, such as Autoformer and FEDformer, integrate decomposition primarily within the Decoder, focusing on seasonal components to improve prediction accuracy [11], [19]. Although this approach enables generated predictions to leverage past information, improving prediction performance, the structure of Autoformer's Encoder resembles that of the Transformer, and it solely focuses on the seasonal part of the time series decomposition, failing to apply time series decomposition to the Encoder to enhance the model's ability to extract time series features. Moreover, DLinear achieves remarkable success in long-term time series forecasting problems by solely employing time series decomposition techniques and simple linear connections, surpassing the performance of many Transformer-based models [9], [20]. Furthermore, there exists a plethora of research employing hybrid models that integrate time series decomposition [21].

Despite the growing popularity of Transformer-based architectures, most existing models underutilize time series decomposition techniques, particularly in the Encoder, thereby limiting their capacity for temporal feature extraction [9]. This underutilization creates a critical research gap, the absence of effective decomposition mechanisms within the Encoder hinders the model's ability to disentangle temporal patterns early in the processing pipeline. For instance, models like Autoformer fail to recognize the significance of the Encoder in time series forecasting, thereby incorporating time series decomposition predominantly within the model's Decoder. Various mainstream approaches to handling time series decomposition information are depicted in Fig. 2. The processing approach of DLinear is illustrated in Fig. 2 - (A) [9], where DLinear directly processes the decomposed parts of the time series separately and aggregates them to obtain the prediction result. The structure of DLinear proves that time series decomposition is effective in extracting temporal features. Fig. 2 - (B) illustrates the processing approach of Autoformer and FEDformer [11], [19], where the decomposed information is passed to the Decoder to enhance the accuracy of predictions, while the Encoder focuses solely on extracting seasonal features from the time series.

The method proposed in this paper is depicted in Fig. 2 - (C). Recognising the central role of the Transformer's Encoder architecture in time series forecasting tasks, we seek to incorporate more temporal information into the structure of the Encoder, enabling the model to better comprehend time series information. We propose to integrate effective time series decomposition techniques into the Encoder, introducing the Series Decomposition Encoder (SDE) block. The SDE block primarily serves two functions: decomposing the input time series into trend and seasonal information, and utilizing this trend and seasonal information to compute multivariate correlations, thereby distilling crucial information from the time series. We suggest that seasonality and trend information

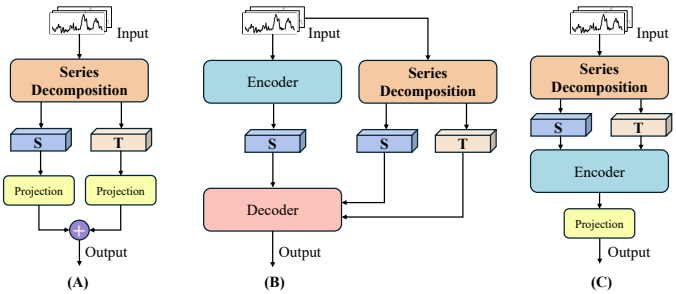

Fig. 2. Common time series decomposition embedding process with our proposed process. (A) is typically represented by DLinear, (B) is typically represented by Autoformer, (C) is our proposed scheme.

can reduce redundancy and improve the feature extraction capability of the model. Direct input of undecomposed data may cause Transformer to learn a mixture of features that are not conducive to long-term forecasting. As shown in Fig. 1, across 9 long-term time series forecasting datasets, SDformer consistently achieves outstanding results, demonstrating state-of-the-art performance in overall performance metrics.

The contributions of this paper are summarized as follows:

1) We propose the SDE block to incorporate time series decomposition information into the Encoder, demonstrating the effectiveness of integrating time series decomposition information into the Encoder structure.
2) Through experiments, we demonstrate that the proposed SDE block can effectively enhance the model's understanding of temporal information, leading to significant improvements in prediction results when integrated into multiple Transformer-based time series forecasting models.
3) We propose SDformer and validate its performance on nine benchmark long-term time series forecasting datasets.

The remainder of this paper is summarized as follows: Section II provides an overview of existing time series forecasting models and related work on time series decomposition. Section III elaborates on the proposed SDE block and the SDformer model. Section IV presents comprehensive experimental results of SDformer in long sequence time forecasting problems, accompanied by discussions and analyses related to the model structure. Finally, Section V concludes the paper and suggests potential future research directions.

II. Related work

A. Traditional time series forecasting models

Time series forecasting has evolved significantly, originating from traditional statistical methods. Classical methods such as ARIMA and VAR have long been foundational for time series forecasting tasks, owing to their interpretability and simplicity [22], [23]. ARIMA models capture linear relationships within the data by combining autoregression, differencing, and moving averages. VAR models extend ARIMA by considering multivariate time series, capturing the interdependencies among multiple time series variables. However, these traditional models generally rely on assumptions of linearity

and stationarity, limiting their effectiveness on complex real-world data exhibiting nonlinear and dynamic patterns. With the advancement of deep learning, RNNs emerged as a powerful alternative due to their ability to model sequential data [24]. However, standard RNNs suffer from limitations such as vanishing gradients and difficulties in capturing long-term dependencies. To address these issues, variants like LSTM and GRU were introduced, showing improved memory retention and modeling capacity for complex temporal structures [25]–[28].

In the process of refining RNN architectures, attention mechanisms have been introduced. Numerous studies have incorporated attention mechanisms into RNNs and applied them to various domains characterised by temporal features [29]–[31]. Attention mechanisms adaptively select the most relevant input features, effectively enhancing RNNs' ability to capture long-term temporal dependencies and process time series data. The Transformer architecture has advanced the attention mechanism and revolutionised the structure of RNNs [32]. Initially successful in natural language processing and later achieving significant breakthroughs in the field of computer vision [33], [34], it is now also applied to time series forecasting. Transformers leverage self-attention mechanisms to capture intricate relationships between different time steps in the input sequence, enabling them to effectively model long-range dependencies [35].

B. Models for long-term time series forecasting

To address the challenges of long-term dependency modeling in time series forecasting, a diverse array of deep learning architectures has been proposed. Among them, Transformer-based models have shown great potential due to their ability to capture global temporal correlations through self-attention mechanisms. Informer introduced a ProbSparse attention mechanism and a generative decoder to enable efficient sequence-level forecasting, demonstrating scalability on long sequences [7]. Autoformer enhanced the Transformer backbone by incorporating series decomposition and auto-correlation mechanisms, thereby improving the modeling of periodic and trend patterns [11]. FEDformer further extended this by integrating Fourier and wavelet transforms into the decomposition framework to disentangle frequency-domain structures [19]. Pyraformer designed a pyramidal attention structure to hierarchically extract temporal dependencies [36]. Crossformer incorporates a two-stage attention mechanism, which considers dependencies across both the time dimension and the variable dimension [37]. Furthermore, numerous models enhance Transformer's attention mechanism to improve its time series forecasting performance [38], [39].

Recent studies have demonstrated that Encoder-only architectures exhibit significant advantages in long-term forecasting tasks by focusing modeling capacity on extracting deep temporal representations. PatchTST discards the decoder and processes time series in patches through a pure Transformer encoder, enabling efficient and scalable long-term forecasting [13]. iTransformer adopts an Encoder-only structure, where proposed an inverse embedding strategy to highlight inter-variable correlations during the attention computation, enhancing multivariate modeling capabilities [14]. TimeXer redesigns embedding layers to reconcile endogenous and exogenous information together for sequence forecasting [40].

Beyond the Transformer architecture, several efficient alternatives have emerged. DLinear introduces a decomposition-based linear projection model that separately forecasts trend and seasonal components using fully connected layers, offering strong performance with minimal overhead [9]. TimesNet adopts multi-periodic convolutional blocks to capture frequency-aware temporal patterns across diverse resolutions [12]. TimeMixer leverages token mixing operations for temporal and feature interactions, eschewing self-attention for a more lightweight implementation [41]. Mamba proposes a selective state-space sequence modeling framework that combines parallelism with long-range memory via continuous-time state-space dynamics, representing a promising alternative to attention-based methods [42]. Furthermore, recent studies have investigated the application of various deep learning paradigms such as dendritic neural networks [43], [44], spiking neural networks [45] and large language models [46] in addressing time series forecasting problems.

C. Application of time series decomposition in models

Time series decomposition involves disassembling a time series into multiple component series with distinct characteristics, facilitating a better understanding of features such as trends and seasonality inherent in the original sequence [47]. Among the various deep learning models used for time series forecasting, time series decomposition is often applied both in the preprocessing stage and within the architecture of the models themselves. For instance, LSTM-MSNet uses multiseasonal decomposition technology to supplement the LSTM learning process [48]. MEMD-LSTM combined with multivariate empirical mode decomposition can capture the inherent characteristics of the complex dynamics of stock price index time series [49]. STLDNM utilizes time series decomposition as a preprocessing technique to aid the model in time series forecasting [50]. N-BEATS introduces basis expansion to fit periodic and trend information in time series data [51]. DLinear directly decomposes the original sequence into two components and predicts each part separately using fully connected layers [9].

With the ascendancy of the Transformer as the predominant model for time series forecasting, Autoformer integrates time series decomposition into the Transformer architecture by employing a simple yet effective decomposition method [11]. It computes the long-term trend of the sequence using moving averages and treats the remainder as the seasonal component. In Autoformer's Decoder architecture, the decomposed long-term trend and seasonal components are utilized to aid in forecasting future sequences. Building upon Autoformer's decomposition strategy, FEDformer incorporates Fourier and wavelet transforms to disentangle seasonal information from the time domain into the frequency domain [19]. Both Autoformer and FEDformer propagate the decomposed information to the Decoder layer.

In summary, we can find that in traditional time series forecasting models, such as LSTM, the time series decomposition is usually used as a feature extraction tool to improve the feature extraction ability of the model. And the models based on the Transformer architecture use the time series decomposition more as a kind of complementary condition for generating predicted values. This paper presents a novel approach that leverages time series decomposition information.

III. SDformer

In the context of long-term time series forecasting problems, the input data is represented as a vector of values, denoted by $\mathbf{X} = \{x_1, ..., x_T\} \in \mathbb{R}^{T\times C}$, where T denotes the number of time steps and C represents the number of variables at each time step. The objective of long-term prediction is to forecast future values of the variables over a specified horizon. The predicted values are denoted by $\hat{\mathbf{X}} = \{x_{T+1}, ..., x_{T+\tau}\} \in \mathbb{R}^{\tau\times C}$, where τ signifies the number of future time steps for which the prediction is made.

A. Structure overview

The overall structure of our proposed SDformer model is illustrated in Fig. 3, comprising three main components: the SDE block, the Transformer block, and the Projection module. This structured representation captures the transformation of the input time series through each stage of the SDformer model, illustrating the flow from initial data input to the final prediction output. The SDE block, which is the focal point of this paper, encompasses three key stages: decomposition, embedding, and attention mechanisms. These stages work in concert to extract and capture the essential temporal features from the input time series data. Subsequently, the Transformer block, analogous to the Encoder layer in a standard Transformer architecture, processes the extracted temporal features, enabling a transition from historical lookback data to future predictions. Finally, the Projection module transforms the processed feature vectors into the desired prediction length, thereby generating the forecasted time series values. This comprehensive structure has been designed to effectively leverage both the temporal characteristics and the powerful sequence modelling capabilities of the Transformer framework, thereby ensuring accurate and robust long-term time series forecasting. The overall structure of the SDformer model can be formally described by the following equations:

$$\begin{aligned} \mathbf{H}_{SDE} &= \text{SDE}(\mathbf{X}) \in \mathbb{R}^{C\times D} \\ \mathbf{H}_{Tra} &= \text{Transformer}(\mathbf{H}_{SDE}) \in \mathbb{R}^{C\times D} \\ \hat{\mathbf{X}} &= \text{Projection}(\mathbf{H}_{Tra}) \in \mathbb{R}^{\tau\times C} \end{aligned} \tag{1}$$

where H denotes the intermediate layer outputs of each module. In these equations, we also describe the shape and dimensions of each intermediate layer output. For instance, $\mathbf{H}_{SDE} \in \mathbb{R}^{C\times D}$ indicates that after the input X is processed by the SDE block, the time dimension T is embedded into a dimension D.

B. Series decomposition encoder

The SDE block is designed to enhance the model's ability to capture both short-term and long-term temporal dependencies in time series data. It comprises three key components: Series Decomposition Attention (SDA), Layer Normalization, and a FeedForward network. Among them, SDA is the core component responsible for decomposing and integrating multiscale patterns. In this work, we further enhance SDA by introducing a variate-wise gate mechanism, enabling selective information fusion between decomposed signals and original input.

1) Series decomposition attention:

a) Time series decomposition: Time series decomposition is a critical component of the SDE block, responsible for decomposing the input time series into two vectors: seasonal and long-term trend components. To facilitate more comprehensive analysis of time series characteristics, the sequence is partitioned into these two components, with the seasonal component representing short-term fluctuations and the trend component capturing long-term trends [11], [16]. Specifically, the moving average method is employed to smooth the original sequence, thereby extracting the long-term trend. The seasonal component is then derived by subtracting the long-term trend from the original sequence. The specific formula is as follows:

$$\begin{aligned} \mathbf{X}_t &= AvgPool(\mathbf{X}, \lambda) \\ \mathbf{X}_s &= \mathbf{X} - \mathbf{X}_t \end{aligned} \tag{2}$$

In our design, the process involves applying a moving average operation, denoted as $AvgPool()$, to the sequence. The window size for this operation, represented as λ, is a parameter that is determined based on the characteristics of the dataset. This value is set individually for each dataset and will be specified in the experimental section. The resulting components, $\mathbf{X}_s$ and $\mathbf{X}_t$, represent the seasonal and long-term trend components of the sequence, respectively. Formally, $\mathbf{X}_s$ and $\mathbf{X}_t$ are defined such that $\mathbf{X}_s, \mathbf{X}_t \in \mathbb{R}^{T\times C}$. This decomposition ensures that the seasonal and trend components are accurately captured, facilitating more effective analysis and prediction of the time series data.

b) Embedding: After decomposition, an embedding operation is performed on the decomposed sequences $\mathbf{X}_s$ and $\mathbf{X}_t$ to transform them into a higher-dimensional space. Embedding is a process that transforms raw input data into a dense vector representation in a higher-dimensional space, making it more suitable for learning complex patterns. In this step, we adopt an embedding approach that treats the entire series as a token. This method, similar to that used in iTransformer, focuses on learning the correlations within multivariate sequences. By considering the entire series as a token, the embedding process captures comprehensive information from both the seasonal and trend components. This enables effective representation learning for the time series data, leveraging the full scope of temporal correlations and patterns present in the decomposed sequences. The embedding process is formulated as follows:

$$\begin{aligned} \mathbf{H_s} &= Embedding(\mathbf{X_s}) \in \mathbb{R}^{C\times D} \\ \mathbf{H_t} &= Embedding(\mathbf{X_t}) \in \mathbb{R}^{C\times D} \\ \mathbf{H_x} &= Embedding(\mathbf{X}) \in \mathbb{R}^{C\times D} \end{aligned} \tag{3}$$

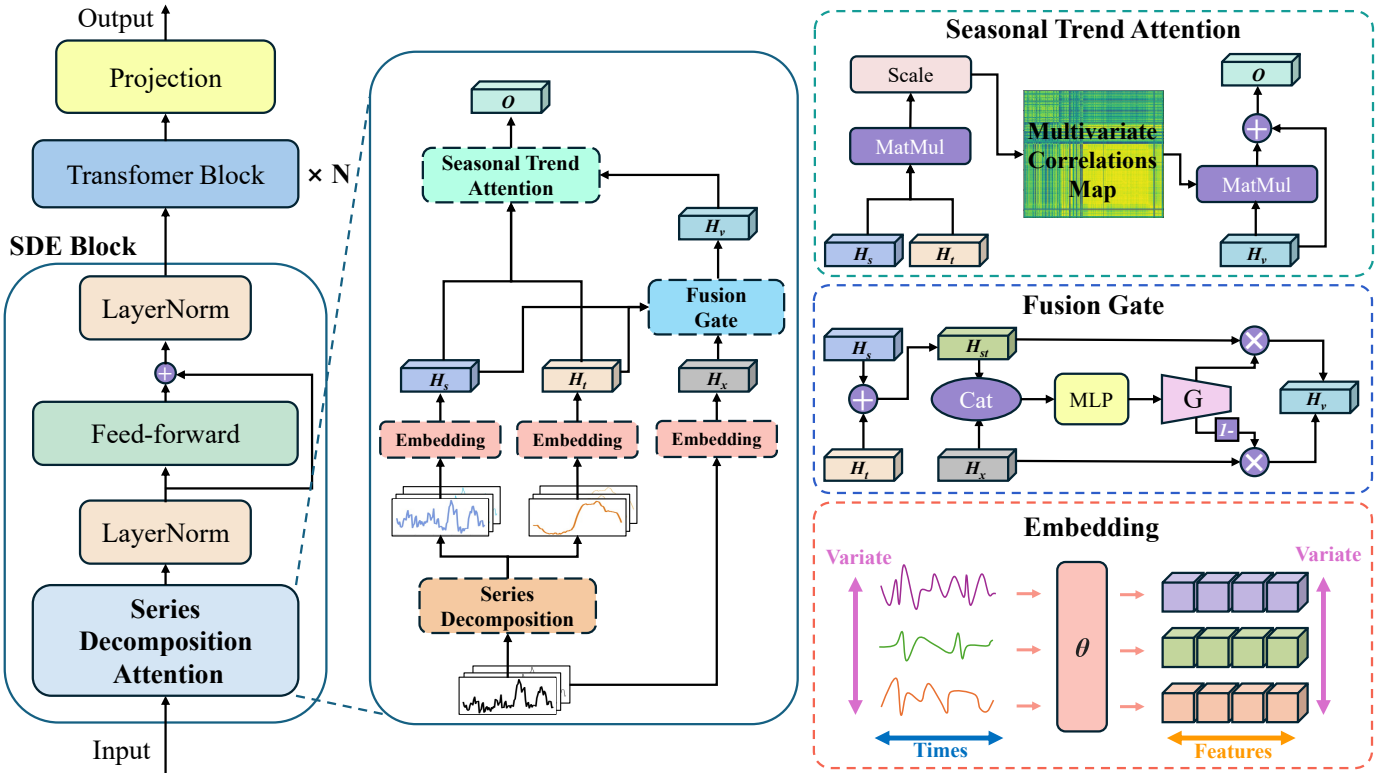

Fig. 3. The overall structure of SDformer consists of three main parts: SDE block, Transformer block and Projection.

c) Fusion gate: To flexibly integrate raw input embeddings with their decomposed counterparts, we propose a fusion gate. This mechanism allows the model to adaptively determine, for each variable, whether to rely more on the seasonal-trend representation or retain original signal characteristics. We first sum the seasonal and trend embeddings to form the decomposed feature:

$$\mathbf{H_{st}} = \mathbf{H_s} + \mathbf{H_t} \in \mathbb{R}^{C\times D} \tag{4}$$

where $\mathbf{H_s}$ captures the structured temporal dynamics, blending both high-frequency (seasonal) and low-frequency (trend) signals. However, in certain cases, the original input embedding $\mathbf{H_x}$ may carry fine-grained details or variable-specific noise that the decomposition process overlooks.

To balance these two representations, we introduce a gate vector $\mathbf{G} \in \mathbb{R}^C$, which controls the contribution of $\mathbf{H_{st}}$ and $\mathbf{H_x}$ for each variable. This vector is obtained by applying a lightweight multilayer perceptron (MLP) with two layers of fully connected network to the concatenation of the two embeddings:

$$\mathbf{G} = \sigma\left(\mathrm{MLP}\left([\mathbf{H_{st}}\|\mathbf{H_x}]\right)\right) \tag{5}$$

where $[\cdot\|\cdot]$ denotes feature-wise concatenation along the last dimension, resulting in a tensor of shape $\mathbb{R}^{C\times 2D}$. The output $\mathbf{G}$ is a vector of shape $\mathbb{R}^C$ after applying the sigmoid function $\sigma(\cdot)$, ensuring values lie in $(0, 1)$.

Finally, the decomposed and original features are linearly blended using the gate, broadcast across the embedding dimension:

$$\mathbf{H}_v = \mathbf{H}_{st} \odot \mathbf{G} + \mathbf{H}_x \odot (1 - \mathbf{G}) \tag{6}$$

where $\odot$ denotes element-wise multiplication, and the gate $\mathbf{G}$ is broadcast along the embedding dimension. This allows the model to selectively preserve original signals while emphasizing decomposed structures for those with clearer trends or seasonality.

d) Seasonal Trend Attention: To model cross-variable temporal dependencies informed by decomposition, we propose a *Seasonal Trend Attention* mechanism. This module uses interactions between seasonal and trend components to construct a multivariate correlations map, which is then used to refine the fused feature representation $\mathbf{H}_v$.

The attention mechanism is formulated as:

$$\mathbf{O} = \mathrm{Softmax}\left(\frac{\mathbf{H_s}\mathbf{H_t}^\top}{\sqrt{d}}\right)\mathbf{H_v} + \mathbf{H_v} \tag{7}$$

By computing the dot product between $\mathbf{H_s}$ and $\mathbf{H_t}^\top$, we obtain a $C \times C$ matrix in which each element reflects the correlation between the seasonal behavior of one variable and the trend dynamics of another. This design allows the model to explore cross-variable dependencies from two complementary temporal perspectives, capturing how short-term patterns in one variable may align with long-term trends in others.

To ensure stable training, the result is scaled by the square root of the embedding dimension d, following the common practice in scaled dot-product attention. A row-wise Softmax is then applied to transform this correlation matrix into a normalized attention distribution, referred to as a multivariate correlations map. This map determines how each variable selectively aggregates information from other variables based on their decomposition-derived relationships.

The resulting attention weights are used to reweight the fused representation $\mathbf{H_v}$, which contains both original and decomposition-enhanced features. The product yields an updated representation where each variable is enhanced by relevant information from others. Finally, a residual connection adds $\mathbf{H_v}$ back to the output, preserving the original signal and improving the robustness and convergence of the learning process.

Overall, this attention formulation enables the model to incorporate decomposition-aware variable interactions in a principled and efficient manner, enriching the temporal modeling capacity without introducing significant additional parameters.

2) LayerNorm: Similar to the classical Transformer architecture, layer normalization is employed to stabilize training, ensuring consistent input distribution across layers. The formula for layer normalization is as follows:

$$\begin{aligned} \text{LayerNorm}(\mathbf{H}_v) &= \left\{ \frac{\mathbf{h}_v^c - \text{Mean}\left(\mathbf{h}_v^c\right)}{\sqrt{\text{Var}\left(\mathbf{h}_v^c\right)}} \middle| \, c = 1, \ldots, C \right\} \\ \mathbf{H}_v &= \mathbf{h}_v^1, ..., \mathbf{h}_v^C \in \mathbb{R}^{C \times D} \end{aligned} \tag{8}$$

where Mean and Var represent the mean and variance of $\mathbf{h}_v^c$ respectively. This normalization step ensures that the input to each layer has a consistent distribution, thereby improving the efficiency and effectiveness of the training process.

3) Feed-forward: At the end of the SDE block, we also retain the classic feed-forward structure, encoding the temporal dimension of the sequence using two fully connected layers. This design ensures that the temporal features are effectively captured and transformed, maintaining the integrity of the sequence data throughout the processing stages.

C. Transformer block

The Transformer block in SDformer plays a pivotal role in modeling long-range temporal dependencies. Architecturally, it follows the standard Transformer encoder design, which consists of multi-head self-attention layers followed by position-wise feedforward networks. This structure enables the model to capture complex temporal interactions across different time steps and variable dimensions. By leveraging the attention mechanism, the Transformer block allows each time step to selectively attend to others, thereby enriching the temporal feature representation learned from the SDE block. Additionally, residual connections and layer normalization are applied to stabilize training and facilitate gradient flow. The output of this block serves as a high-level representation of the temporal dynamics, which is then passed to the projection layer for final forecasting. As shown in Fig. 3, SDformer employs N Transformer blocks to enhance the model's ability to process temporal information. The hyperparameter N is set individually for each dataset and is specified in the experimental section.

D. Projection module

The Projection module transforms the temporal representations produced by the Transformer block into the final forecasted sequence. Specifically, the output tensor $\mathbf{H}_{Tra} \in \mathbb{R}^{C \times D}$, which encodes the embedded temporal features for each variable, is passed through a fully connected layer that maps the embedding dimension D to the prediction length τ. This operation results in an output of shape $\mathbb{R}^{C \times \tau}$, which is subsequently transposed to obtain the final prediction $\hat{\mathbf{X}} \in \mathbb{R}^{\tau \times C}$. This simple design ensures that the model produces multivariate multi-step forecasts efficiently, while fully utilizing the temporal features learned by the previous modules.

E. Effectiveness of SDformer's improvement

The effectiveness of SDformer stems from its introduction of the SDE block, which enhances the Transformer's ability to extract temporal features by explicitly decomposing time series data. This section analyses why SDformer's improvements contribute to superior forecasting performance.

1) Feature decoupling and redundancy reduction: Traditional Transformer-based models process raw time series data without explicit decomposition, requiring the attention mechanism to simultaneously learn trend, seasonality, and noise within a single representation space. This often results in mixed feature representations, limiting the model's ability to distinguish between different temporal components. By incorporating the SDE module, SDformer decomposes the time series into long-term trend and seasonal components, enabling the model to process each component separately. This structured decomposition reduces redundancy in feature representations, allowing for a more efficient and interpretable feature extraction process.

2) Enhanced temporal modeling capability: Attention mechanisms in Transformer architectures heavily rely on the structure of the input data. When applied to raw time series data, the model may attend to irrelevant patterns, leading to suboptimal forecasting performance. SDformer mitigates this issue by explicitly injecting trend and seasonal information into the Encoder. By leveraging decomposed components, the model refines the attention mechanism's focus, capturing essential temporal dependencies more effectively and preventing unnecessary feature interactions. This targeted feature extraction enhances the model's ability to discern meaningful temporal patterns.

3) Improved generalization ability: Transformer-based time series forecasting models often suffer from limited generalization across datasets with varying data distributions and noise characteristics. SDformer alleviates this challenge by utilizing explicit decomposition, which stabilizes feature extraction and reduces sensitivity to noise. By learning distinct long-term and short-term temporal variations separately, SDformer improves its adaptability to different forecasting scenarios, leading to more robust performance across diverse datasets.

IV. Experiment

The experiments were conducted on a computer equipped with an Intel i9-12900K CPU and an NVIDIA RTX 3090 GPU, running the Windows 11 operating system. SDformer and the comparative models were implemented using Python 3.12 and PyTorch 2.1. To ensure the accuracy of the experiments, all comparative models were implemented based on their original descriptions.

TABLE I
DETAILED DESCRIPTION OF THE DATASET AND HYPERPARAMETER SETTINGS.

Dataset	Data description					Hyperparameter settings		
	C	T	Prediction Length (τ)	Frequency	Dataset Size	N	D	λ
ETTh1	7	96	{96, 192, 336, 720}	Hourly	{8545, 2881, 2881}	2	256	25
ETTh2	7	96	{96, 192, 336, 720}	Hourly	{8545, 2881, 2881}	2	256	25
ETTm1	7	96	{96, 192, 336, 720}	15 mins	{34465, 11521, 11521}	2	128	25
ETTm2	7	96	{96, 192, 336, 720}	15 mins	{34465, 11521, 11521}	2	128	25
ECL	321	96	{96, 192, 336, 720}	Hourly	{18317, 2633, 5261}	3	512	25
Exchage	8	96	{96, 192, 336, 720}	Daily	{5120, 665, 1422}	2	128	25
Traffic	862	96	{96, 192, 336, 720}	Hourly	{12185, 1757, 3509}	4	512	25
Weather	21	96	{96, 192, 336, 720}	10 mins	{36792, 5271, 10540}	3	512	25
ILI	7	36	{24, 36, 48, 60}	Weekly	{617, 74, 170}	2	2048	9

TABLE II
DETAILED RESULTS IN NINE BENCHMARK DATASETS.

Method		SDformer		iTransformer		PatchTST		Autoformer		DLinear		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.388	0.406	**0.387**	**0.405**	0.439	0.434	0.457	0.464	0.397	0.412	0.940	0.767
	192	0.444	0.440	**0.441**	**0.436**	0.486	0.461	0.483	0.474	0.446	0.441	1.016	0.793
	336	0.490	**0.462**	0.491	0.462	0.522	0.482	0.528	0.503	**0.489**	0.467	1.018	0.773
	720	**0.498**	**0.485**	0.509	0.494	0.530	0.509	0.537	0.518	0.513	0.510	1.182	0.858
	Avg	**0.455**	**0.448**	0.457	0.449	0.494	0.472	0.501	0.490	0.461	0.457	1.039	0.798
ETTh2	96	0.302	0.352	0.301	0.350	**0.295**	**0.344**	0.370	0.406	0.340	0.394	3.057	1.381
	192	0.385	0.402	0.380	**0.399**	**0.375**	0.399	0.444	0.445	0.482	0.479	6.262	2.090
	336	0.431	0.436	0.424	0.432	**0.420**	**0.429**	0.488	0.484	0.591	0.541	5.879	2.049
	720	0.449	0.457	**0.430**	**0.447**	0.431	0.451	0.484	0.490	0.839	0.661	4.219	1.713
	Avg	0.392	0.412	0.384	0.407	**0.380**	**0.406**	0.447	0.456	0.563	0.519	4.854	1.808
ETTm1	96	0.345	0.377	**0.342**	0.377	0.344	**0.369**	0.555	0.496	0.346	0.374	0.619	0.553
	192	**0.380**	0.396	0.383	0.396	0.384	**0.387**	0.588	0.520	0.382	0.391	0.706	0.604
	336	0.420	0.421	0.418	0.418	0.417	**0.408**	0.626	0.540	**0.415**	0.415	1.014	0.743
	720	0.486	0.457	0.487	0.457	0.475	**0.440**	0.583	0.527	**0.473**	0.451	0.957	0.731
	Avg	0.408	0.413	0.408	0.412	0.405	**0.401**	0.588	0.521	**0.404**	0.408	0.824	0.658
ETTm2	96	**0.185**	**0.269**	0.186	0.272	0.188	0.272	0.250	0.320	0.193	0.293	0.422	0.487
	192	0.254	0.316	0.254	0.314	**0.251**	**0.310**	0.283	0.341	0.284	0.361	0.821	0.705
	336	0.313	0.352	0.316	0.351	**0.311**	**0.348**	0.332	0.368	0.382	0.429	1.398	0.902
	720	**0.407**	0.405	0.414	0.407	0.411	**0.402**	0.425	0.419	0.558	0.524	4.270	1.532
	Avg	**0.290**	0.335	0.292	0.336	0.290	**0.333**	0.323	0.362	0.354	0.402	1.728	0.907
ECL	96	**0.148**	**0.240**	0.149	0.240	0.182	0.271	0.205	0.322	0.210	0.302	0.332	0.416
	192	**0.163**	**0.255**	0.166	0.257	0.187	0.276	0.225	0.333	0.210	0.305	0.350	0.432
	336	0.180	0.273	**0.178**	**0.271**	0.203	0.292	0.253	0.355	0.223	0.319	0.352	0.434
	720	**0.218**	**0.307**	0.228	0.312	0.245	0.325	0.260	0.361	0.258	0.350	0.500	0.527
	Avg	**0.177**	**0.269**	0.180	0.270	0.204	0.291	0.236	0.343	0.225	0.319	0.383	0.452
Exchange	96	0.089	0.211	**0.086**	0.208	0.088	**0.205**	0.144	0.276	0.090	0.220	1.029	0.851
	192	0.182	0.304	0.188	0.311	**0.176**	**0.299**	0.283	0.386	0.179	0.316	1.360	0.985
	336	0.333	0.419	0.326	0.414	**0.301**	**0.397**	0.507	0.532	0.338	0.443	1.396	1.000
	720	0.845	0.695	0.912	0.727	0.887	0.708	1.096	0.813	**0.840**	**0.689**	2.906	1.420
	Avg	0.362	0.407	0.378	0.415	0.363	**0.402**	0.507	0.502	**0.361**	0.417	1.673	1.064
Traffic	96	**0.389**	**0.265**	0.401	0.274	0.486	0.319	0.669	0.408	0.649	0.397	1.297	0.697
	192	**0.402**	**0.272**	0.413	0.277	0.487	0.316	0.683	0.435	0.599	0.371	1.103	0.615
	336	**0.416**	**0.279**	0.425	0.283	0.502	0.321	0.655	0.409	0.605	0.374	1.327	0.718
	720	**0.445**	**0.293**	0.458	0.300	0.537	0.338	0.672	0.416	0.646	0.395	0.985	0.553
	Avg	**0.413**	**0.278**	0.424	0.284	0.503	0.323	0.670	0.417	0.625	0.384	1.178	0.646
Weather	96	**0.173**	**0.212**	0.175	0.214	0.175	0.215	0.304	0.362	0.196	0.255	0.507	0.500
	192	0.222	0.257	0.226	0.258	**0.221**	**0.257**	0.302	0.358	0.237	0.295	0.368	0.406
	336	**0.280**	**0.297**	0.281	0.299	0.281	0.298	0.358	0.393	0.282	0.331	0.449	0.464
	720	0.357	0.349	0.359	0.351	0.358	**0.349**	0.445	0.443	**0.345**	0.382	1.126	0.762
	Avg	**0.258**	**0.279**	0.260	0.280	0.259	0.280	0.352	0.389	0.265	0.316	0.613	0.533
ILI	24	**1.929**	**0.852**	2.342	0.940	2.168	0.892	3.229	1.235	4.793	1.658	5.219	1.567
	36	**1.925**	**0.875**	2.172	0.945	2.326	0.931	3.316	1.260	4.453	1.559	5.023	1.565
	48	**2.010**	**0.889**	2.238	0.958	2.143	0.895	3.096	1.194	4.121	1.476	4.961	1.539
	60	**1.939**	**0.896**	2.299	0.985	2.033	0.911	2.811	1.132	4.246	1.485	5.152	1.549
	Avg	**1.951**	**0.878**	2.263	0.957	2.167	0.907	3.113	1.205	4.403	1.544	5.089	1.555
1st Count		**23**	**21**	6	5	9	18	0	0	7	1	0	0

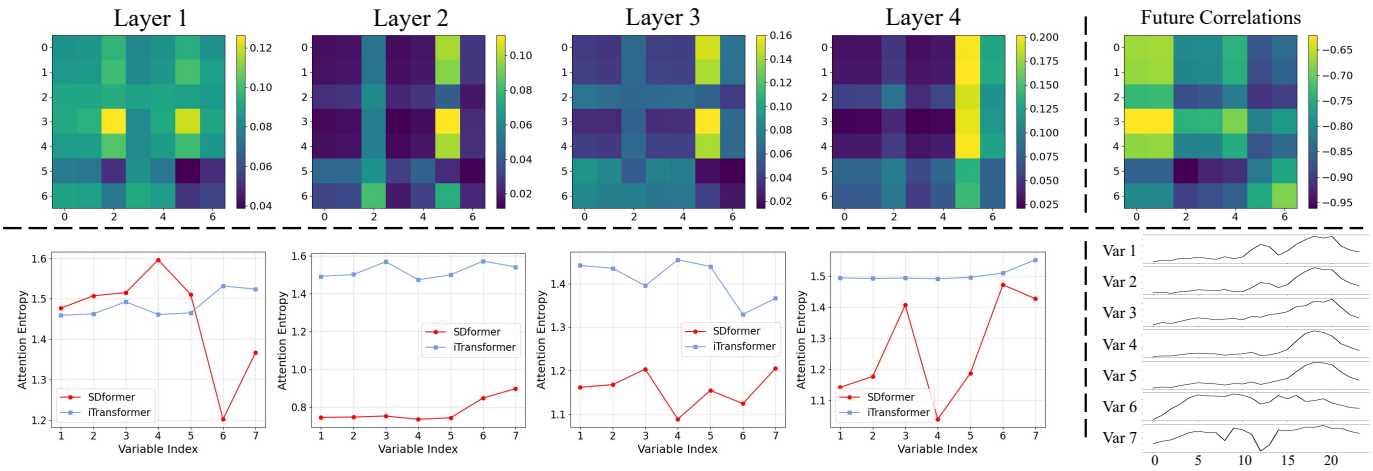

Fig. 4. Visualization of attention structure and entropy in SDformer on the ILI dataset.

TABLE III
THE EFFECT OF ADDING SDE TO EXISTING TRANSFORMER-BASED STRUCTURAL MODELS.

Method		SDE+iTransformer		iTransformer		SDE+PatchTST		PatchTST		SDE+Autoformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.147**	**0.239**	0.149	0.240	**0.168**	**0.262**	0.182	0.271	**0.186**	**0.302**	0.205	0.322
	192	**0.164**	**0.257**	0.166	0.257	**0.181**	**0.273**	0.187	0.276	**0.223**	**0.332**	0.225	0.333
	336	0.181	0.273	**0.178**	**0.271**	**0.197**	**0.288**	0.203	0.292	**0.223**	**0.330**	0.253	0.355
	720	**0.215**	**0.305**	0.228	0.312	**0.237**	**0.321**	0.245	0.325	**0.256**	**0.360**	0.260	0.361
	Avg	**0.177**	**0.269**	0.180	0.270	**0.196**	**0.286**	0.204	0.291	**0.222**	**0.331**	0.236	0.343
Traffic	96	**0.391**	**0.268**	0.401	0.274	**0.429**	**0.291**	0.486	0.319	**0.637**	**0.401**	0.669	0.408
	192	**0.401**	**0.272**	0.413	0.277	**0.447**	**0.299**	0.487	0.316	**0.614**	**0.386**	0.683	0.435
	336	**0.417**	**0.280**	0.425	0.283	**0.461**	**0.304**	0.502	0.321	**0.631**	**0.392**	0.655	0.409
	720	**0.446**	**0.294**	0.458	0.300	**0.494**	**0.323**	0.537	0.338	**0.669**	0.419	0.672	**0.416**
	Avg	**0.414**	**0.279**	0.424	0.284	**0.458**	**0.304**	0.503	0.323	**0.638**	**0.400**	0.670	0.417
Weather	96	0.175	0.214	**0.175**	**0.214**	0.178	0.216	**0.175**	**0.215**	**0.251**	**0.333**	0.304	0.362
	192	**0.222**	**0.256**	0.226	0.258	0.225	**0.257**	**0.221**	0.257	**0.276**	**0.351**	0.302	0.358
	336	**0.281**	**0.299**	0.281	0.299	**0.281**	**0.297**	0.281	0.298	**0.342**	0.396	0.358	**0.393**
	720	**0.357**	**0.350**	0.359	0.351	**0.358**	**0.348**	0.358	0.349	0.447	0.469	**0.445**	**0.443**
	Avg	**0.259**	**0.280**	0.260	0.280	0.260	**0.279**	**0.259**	0.280	**0.329**	**0.387**	0.352	0.389

A. Data description and experimental settings

To evaluate the proposed SDformer, we conducted extensive experiments on nine popular real-world datasets: ETT (four subsets), ECL, Exchange, Traffic, Weather, and ILI. The ETT dataset comprises two hourly-level datasets (ETTh) and two 15-minute-level datasets (ETTm), capturing load features of seven oil and power transformers from July 2016 to July 2018. The ECL, also known as the electricity dataset, contains hourly electricity consumption data for 321 customers collected from 2012 to 2014. The Exchange dataset includes daily exchange rate panel data for eight countries from 1990 to 2016. The Traffic dataset consists of hourly data recorded by 862 sensors on San Francisco freeways from 2015 to 2016. The Weather dataset encompasses 21 weather indicators, such as air temperature and humidity, recorded every 10 minutes in 2020. Lastly, the ILI dataset describes the ratio of influenza-like illness patients to the total number of patients, with weekly records spanning from 2002 to 2021.

The data set division method is the same as TimesNet [12]. A detailed summary of the dataset splits and input dimensions is provided in Table I. Specifically, the prediction lengths are set to $\{96, 192, 336, 720\}$, indicating that the model is evaluated under varying forecasting horizons. The Dataset Size column lists the number of samples in the training, validation, and test sets, respectively. For each dataset, we trained the models for 10 epochs using the Adam optimizer with a fixed learning rate of 1e-3. Other dataset-specific hyperparameters are summarized in Table I. During training, early stopping and consistent random seeds were used to mitigate overfitting and ensure the stability of experimental results.

B. Forcasting results

We selected five effective models for long-term time series forecasting as benchmarks: iTransformer [14], PatchTST [13], Autoformer [11], DLinear [9], and Informer [7]. Among them, iTransformer and PatchTST are currently the most advanced models for long-term time series forecasting. Autoformer and DLinear incorporate time series decomposition methods within the Transformer-based and linear forecasting architectures, respectively. Informer is a representative model in long-term time series forecasting, surpassing traditional models and significantly influencing subsequent developments in the field. Table II presents the comprehensive prediction results across multiple prediction lengths $(96, 192, 336, 720, Avg)$, where lower MSE and MAE values indicate better predictive

performance. The best results are highlighted in bold, and the second-best results are underlined.

The results in Table II show that SDformer consistently outperforms the other models across most datasets, showcasing its superiority in time series forecasting. Notably, SDformer exhibits a remarkable performance improvement on the Traffic dataset, where it significantly surpasses existing models. This highlights the effectiveness of the proposed time series decomposition and encoding approach, particularly in the context of high-dimensional, multivariate data. The ability to efficiently decompose and encode complex temporal dependencies contributes to the model's superior predictive accuracy in such challenging scenarios. In contrast, DLinear maintains a notable advantage on certain low-dimensional datasets. This is primarily due to its simple time series decomposition strategy and linear encoding mechanism, which prove effective in scenarios with less complex temporal relationships. However, SDformer still demonstrates strong performance on these datasets, outperforming models like iTransformer, which indicates that SDformer's approach is robust across different types of datasets, not just high-dimensional ones. It is worth noting that on some datasets such as ETTh2 and Exchange, the improvements brought by the SDE module are relatively marginal. This phenomenon can be attributed to the inherent characteristics of these datasets. ETTh2 and Exchange exhibit relatively weak periodicity and lower inter-variable correlations compared to datasets like Traffic or ECL. As a result, the decomposition of time series into seasonal and trend components may not yield significantly distinguishable patterns for further modeling. In such scenarios, the advantage of introducing an explicit decomposition-based encoding, as done in SDformer, becomes less pronounced. Nevertheless, SDformer still maintains competitive performance, which demonstrates its generalization ability even when the effectiveness of decomposition is reduced.

Moreover, Table II also shows the frequency with which each model achieves the best results at various prediction lengths. SDformer consistently achieves the highest number of optimal results, underlining its stability and effectiveness across multiple forecasting horizons. This result suggests that SDformer is not only effective in short-term forecasting but also maintains strong performance in long-term forecasting, which is a critical aspect for real-world applications.

C. Visualization of SDformer's Attention map

We visualize and analyze the attention behavior of SDformer using the ILI dataset to demonstrate the effectiveness of the proposed SDE. As shown in Fig. 4, the top row presents the layer-wise attention maps learned by SDformer alongside the Pearson correlation matrix between the lookback and future time steps. Remarkably, the structural patterns observed in the attention maps closely resemble the statistical correlations, indicating that the model learns to focus on temporally meaningful positions that are highly predictive of future values. This resemblance confirms that SDE effectively decomposes the input into components that enhance the attention mechanism's ability to align with true temporal dependencies, thereby improving the interpretability and reliability of the learned attention weights.

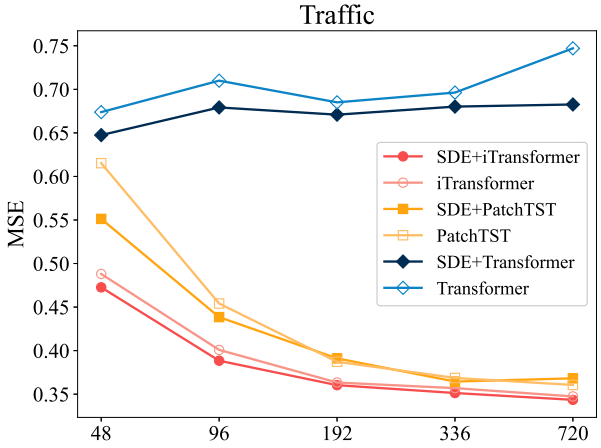

Fig. 5. The effect of increasing the lookback length in models with added SDE block.

The bottom row of Fig. 4 offers a deeper inspection into the attention entropy across layers for each variable and presents raw value plots of seven representative variables from the ILI dataset. Entropy here quantifies the degree of concentration in the attention distribution, lower entropy reflects a sharper focus on specific temporal patterns. Compared to iTransformer, SDformer exhibits more distinct and fluctuating entropy curves across layers, indicating its ability to dynamically adjust focus depending on the content and structure of the input. This fluctuation is not merely noise, but reflects the model's ability to respond to variable-specific characteristics. Each layer in SDformer captures different levels of abstraction and therefore may attend to different subsets of variables; such shifting focus leads to natural variations in entropy. For instance, variables 6 and 7 in the raw value plot exhibit behaviors that are noticeably distinct from others, and SDformer responds with significantly lower attention entropy for these variables, signaling stronger and more selective focus. In contrast, iTransformer fails to capture this inter-variable heterogeneity, resulting in uniformly distributed and less informative attention. These observations collectively highlight the advantage of SDE in enhancing attention adaptability and ensuring that critical variable-level distinctions are preserved and utilized in the learning process.

D. Effects of SDE on Transformer-based models

To assess the generalizability of our proposed SDE block, we conducted a comprehensive comparative analysis of the performance differences between models with and without the integration of SDE. Specifically, we incorporated the SDE block into three popular Transformer-based time series forecasting models: iTransformer, PatchTST, and Autoformer. The integration of the SDE block was achieved by adding it directly before the original model's embedding layer. This allows the SDE block to encode the input time series data before it is processed by the core model structure. Crucially,

TABLE IV
THE IMPACT OF REMOVING DIFFERENT BLOCKS ON SDFORMER.

Method		SDformer		wo-Trans		wo-FFN		wo-TransFFN		$H_v = H_x$		$H_v = H_{st}$	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	0.148	0.240	0.166	0.252	0.185	0.276	0.210	0.290	**0.147**	**0.239**	0.148	0.240
	192	0.163	0.255	0.176	0.262	0.179	0.269	0.201	0.285	0.164	0.257	**0.163**	**0.254**
	336	0.180	0.273	0.192	0.279	0.191	0.281	0.214	0.298	0.177	0.270	**0.176**	**0.269**
	720	0.218	0.307	0.229	0.310	0.226	0.312	0.254	0.330	**0.214**	**0.304**	0.219	0.307
	Avg	0.177	0.269	0.191	0.276	0.195	0.285	0.220	0.301	**0.176**	0.267	0.177	**0.267**
Exchange	96	0.089	0.211	**0.086**	**0.205**	0.139	0.270	0.129	0.261	0.093	0.214	0.089	0.209
	192	0.182	0.304	0.182	**0.303**	0.236	0.354	0.234	0.350	0.198	0.321	**0.181**	0.304
	336	0.333	0.419	0.338	0.422	0.409	0.469	0.395	0.458	**0.328**	**0.416**	0.337	0.423
	720	**0.845**	**0.695**	0.857	0.698	0.957	0.745	0.986	0.754	0.854	0.699	0.872	0.707
	Avg	**0.362**	0.407	0.366	**0.407**	0.435	0.459	0.436	0.456	0.368	0.412	0.370	0.411
Traffic	96	0.389	0.265	0.432	0.293	0.453	0.330	0.524	0.367	0.392	0.270	**0.386**	**0.265**
	192	**0.402**	**0.272**	0.446	0.295	0.442	0.306	0.499	0.344	0.409	0.275	0.407	0.275
	336	**0.416**	**0.279**	0.461	0.302	0.452	0.308	0.507	0.344	0.417	0.281	0.416	0.280
	720	**0.445**	**0.293**	0.492	0.320	0.587	0.407	0.533	0.358	0.451	0.297	0.447	0.294
	Avg	**0.413**	**0.278**	0.458	0.303	0.484	0.338	0.516	0.354	0.417	0.281	0.414	0.279
Weather	96	**0.173**	**0.212**	0.188	0.225	0.233	0.261	0.235	0.262	0.174	0.214	0.174	0.213
	192	0.222	0.257	0.234	0.264	0.279	0.292	0.282	0.296	0.222	**0.256**	**0.222**	0.256
	336	0.280	**0.297**	0.289	0.304	0.325	0.323	0.329	0.326	**0.279**	0.298	0.281	0.298
	720	**0.357**	**0.349**	0.364	0.352	0.394	0.365	0.398	0.369	0.359	0.351	0.357	0.349
	Avg	**0.258**	**0.279**	0.269	0.286	0.307	0.310	0.311	0.313	0.259	0.280	0.259	0.279
1st Count		**9**	**9**	1	3	0	0	0	0	5	4	5	4

TABLE V
COMPARISON OF THE NUMBER OF SDE BLOCKS AND WHETHER OR NOT TRANSFORMER BLOCK IS USED.

SDE / Transformer		1 / √		2 / √		1 / ×		2 / ×		3 / ×		4 / ×	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	0.147	0.239	**0.146**	**0.238**	0.166	0.252	0.151	0.242	0.150	0.241	0.148	0.240
	192	0.164	0.257	0.164	0.257	0.176	0.262	0.166	0.255	0.164	0.254	**0.163**	**0.254**
	336	0.181	0.273	0.182	0.276	0.192	0.279	0.184	0.275	0.183	0.274	**0.179**	**0.271**
	720	0.215	**0.305**	**0.215**	0.306	0.229	0.310	0.225	0.310	0.220	0.305	0.222	0.308
	Avg	0.177	0.269	**0.177**	0.269	0.191	0.276	0.182	0.270	0.179	0.269	0.178	**0.268**
Traffic	96	**0.389**	**0.265**	0.474	0.337	0.432	0.293	0.404	0.273	0.401	0.272	0.404	0.274
	192	0.402	0.272	**0.397**	**0.270**	0.446	0.295	0.423	0.280	0.422	0.280	0.424	0.283
	336	0.416	0.279	**0.415**	**0.279**	0.461	0.302	0.438	0.287	0.434	0.288	0.437	0.289
	720	0.445	0.293	**0.443**	**0.292**	0.492	0.320	0.470	0.305	0.466	0.305	0.468	0.305
	Avg	**0.413**	**0.278**	0.432	0.295	0.458	0.303	0.434	0.286	0.431	0.286	0.433	0.288
Weather	96	0.173	0.212	**0.170**	**0.210**	0.188	0.225	0.178	0.217	0.174	0.213	0.174	0.213
	192	0.222	0.257	**0.221**	**0.255**	0.234	0.264	0.227	0.259	0.222	0.255	0.222	0.255
	336	0.280	0.297	**0.278**	**0.296**	0.289	0.304	0.285	0.300	0.281	0.298	0.278	0.297
	720	**0.357**	**0.349**	0.359	0.351	0.364	0.352	0.360	0.349	0.358	0.349	0.357	0.349
	Avg	0.258	0.279	**0.257**	**0.278**	0.269	0.286	0.263	0.281	0.259	0.279	0.258	0.279
1st Count		3	4	**10**	**8**	0	0	0	0	0	0	2	3

the core architectures of PatchTST and Autoformer remained unchanged, ensuring that only the temporal encoding was modified without altering the underlying model dynamics.

For this experiment, we applied the SDE block to three diverse datasets: ECL, Traffic, and Weather. Each dataset was evaluated with a lookback sequence length of 96, while the prediction length varied among $\{96, 192, 336, 720\}$. The results, which are summarized in Table III, clearly demonstrate that the integration of the SDE block led to significant performance improvements across all models and datasets. This reinforces the effectiveness of the SDE block in enhancing temporal encoding and improving forecasting accuracy.

E. Increasing lookback length

In the field of time series forecasting, certain Transformer-based models do not exhibit performance improvements with increased lookback lengths [9]. On the other hand, models such as PatchTST and iTransformer demonstrate enhanced predictive performance with longer lookback lengths. To assess whether our proposed SDE block is affected by lookback length during the temporal encoding process, we conducted experiments using the Traffic dataset. These experiments evaluated the impact of increased lookback lengths on the predictive performance of iTransformer, PatchTST, and standard Transformer models after integrating the SDE block. The lookback length T was varied among $\{48, 96, 192, 336, 720\}$, with a fixed prediction length of 96. The results are illustrated in Fig. 5. As shown in Fig. 5, the integration of the SDE block leads to performance improvements across these models, indicating that the SDE block effectively encodes temporal data even with extended lookback lengths.

F. Ablation study

To investigate the contribution of each core component in SDformer, we conduct a comprehensive ablation study, with the results presented in Table IV. The full model is compared

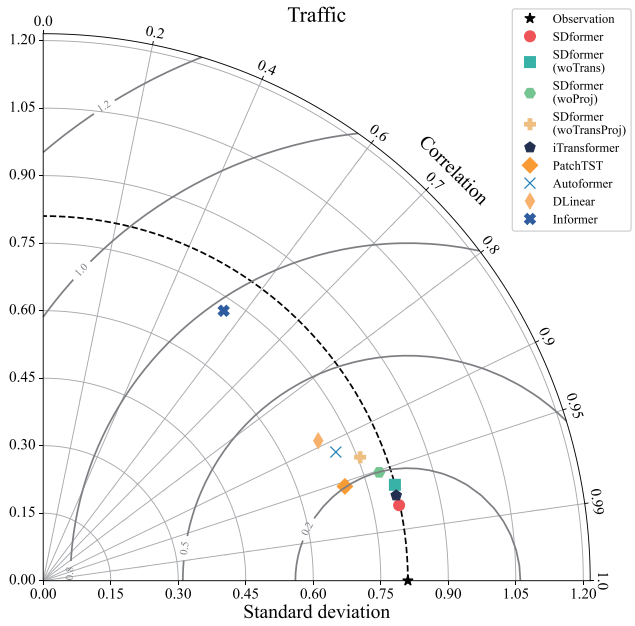

Fig. 6. Taylor diagram of different models.

against five modified versions: one with the Transformer block removed (denoted as wo-Trans), one with the Projection module removed (wo-FFN), and another with both the Transformer and Projection modules removed (wo-TransFFN), thus retaining only the SDE block. In addition, we explore two simplified variants of the fusion gate strategy by directly substituting the fused representation H_v with the raw input embedding H_x and with the sum of seasonal and trend components H_{st}, respectively. These two configurations are denoted as $H_v = H_x$ and $H_v = H_{st}$ in the table.

The results indicate that the original SDformer consistently achieves the best performance across all four benchmark datasets demonstrating the overall effectiveness of the model design. Interestingly, the wo-Trans variant, which removes the Transformer block but retains the SDE and Projection modules, still performs competitively. This suggests that the SDE block alone is capable of extracting meaningful temporal patterns, validating its design as a core building block for time-series modeling. Performance deteriorates more noticeably when the Projection module is removed, as shown by the results of the wo-FFN variant. This observation aligns with our expectations, since the Projection module serves as a critical step that transforms high-level features from the encoder into output representations suitable for prediction. When both the Transformer and Projection modules are removed (wo-TransFFN), performance drops further, confirming that while the SDE provides a strong backbone, it benefits significantly from deeper representation learning and feature projection. Nevertheless, this simplified version still outperforms baseline models such as Autoformer, highlighting the intrinsic capability of the SDE to capture temporal dynamics. Finally, the two simplified fusion gate strategies $H_v = H_x$ and $H_v = H_{st}$ also result in performance degradation compared to the gated fusion mechanism used in SDformer. These results emphasize the importance of the learnable fusion gate, which adaptively integrates both the input and decomposed features based on contextual relevance. By enabling a dynamic weighting of raw and decomposed information, the Fusion Gate enhances the model's ability to capture both global and local temporal dependencies, which is crucial for long-term forecasting accuracy.

To better illustrate the impact of the proposed SDE module on model performance, we visualize the results of SDformer and its variants, along with other models, on the Traffic dataset using a Taylor diagram, as shown in Fig. 6. The Taylor diagram is a widely used statistical visualization tool that simultaneously conveys three key metrics of predictive performance: the standard deviation, the correlation coefficient, and the centered root-mean-square error (implicitly through the spatial distance from the Observation point). Specifically, the Observation point marks the standard deviation of the ground truth series. Each model's prediction is represented as a point in the polar coordinate system, where the radial distance from the origin indicates the standard deviation of the predicted values, and the angular position reflects the correlation coefficient between predictions and observations. A smaller angle (closer to 0°) indicates a higher correlation between the model predictions and the observed data. As depicted in the figure, even when using only the SDE module, our model outperforms most existing models. This demonstrates that the SDE module effectively captures complex patterns in time series data, thereby enhancing both the accuracy and stability of the model.

G. *Effect of multiple SDE blocks*

To further explore the structural design of SDformer, we conduct an ablation study analyzing the effect of stacking multiple SDE blocks and the presence of the Transformer block. The results are summarized in Table V. Specifically, we vary the number of SDE blocks from 1 to 4 and compare two settings: with the Transformer block (denoted as $\surd$) and without it (denoted as $\times$).

From the results, we can observe that stacking SDE blocks generally improves forecasting performance when transitioning from one to two blocks. Similar improvements are observed in the Traffic and Weather datasets. This supports the effectiveness of the SDE block design in extracting meaningful temporal components when applied in a cascaded fashion. However, the performance gains diminish as more SDE blocks are added beyond two. We attribute this to a critical limitation: repeated decomposition and embedding operations tend to convert the temporal input into increasingly implicit representations, making it difficult for subsequent SDE modules to extract meaningful seasonal and trend components. In contrast, the Transformer block is better suited to process these learned implicit features through its powerful self-attention mechanism. This hypothesis is further supported by comparing the single-layer SDE+Transformer setting ($1/\surd$) with the purely 4-layer SDE configuration ($4/\times$). The single-layer combination consistently achieves better or comparable results across all datasets and metrics, confirming the complementary nature of SDE and Transformer modules.

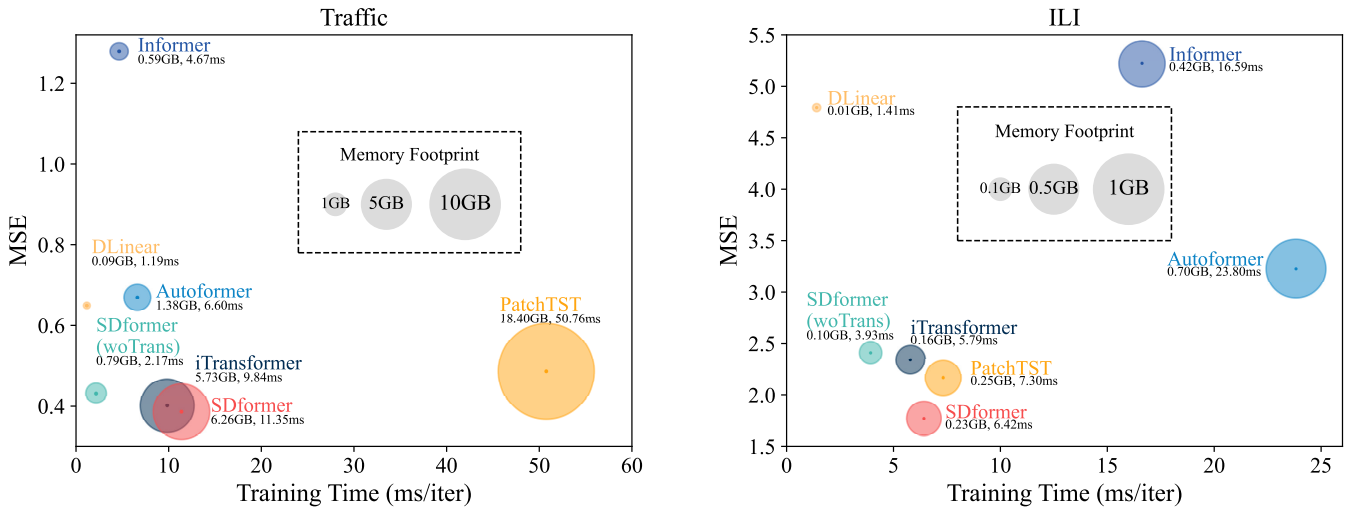

Fig. 7. Efficiency comparison of SDformer and other models.

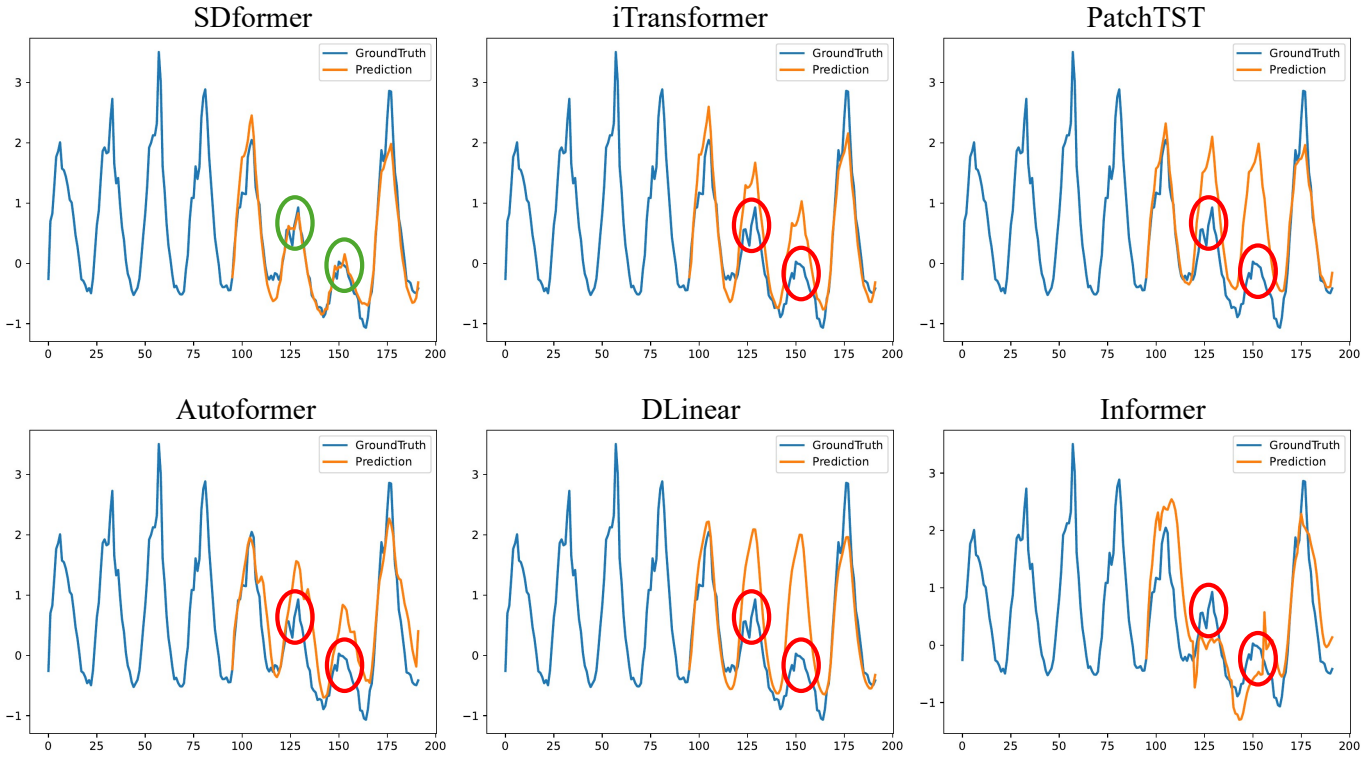

Fig. 8. Visualization of the results of setting input-96-predict-96 on the ECL dataset.

It is also worth noting that the 2-layer SDE with Transformer (2/√) consistently outperforms the single-layer setting, indicating that modest depth still contributes positively to performance. Nevertheless, the marginal improvement is limited, suggesting that further depth increases may yield diminishing returns. Considering the trade-off between performance and computational cost, we adopt the single-layer SDE + Transformer structure as the default architecture of SDformer throughout this paper.

H. Model efficiency comparison

To evaluate the differences in computational efficiency between various models, we conducted experiments on the highest-dimensional Traffic dataset and the lowest-dimensional ILI dataset. This was done to assess the prediction performance, training speed, and memory usage of SDformer, SDformer without the Transformer block, and other comparative models. The results are summarized in Fig. 7. As shown in Fig. 7 (left), for the high-dimensional Traffic dataset, SDformer demonstrated the best prediction performance, with training speed and memory usage comparable to iTransformer, and

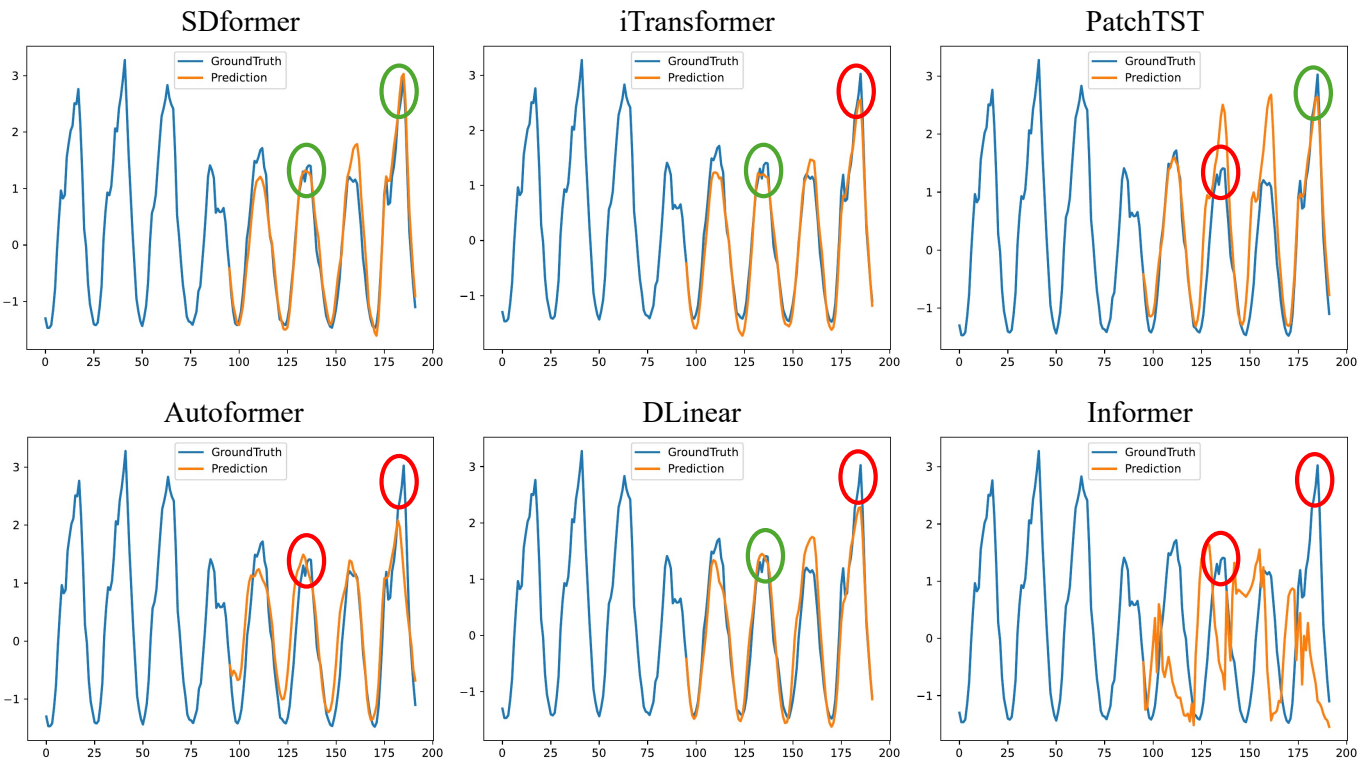

Fig. 9. Visualization of the results of setting input-96-predict-96 on the traffic dataset.

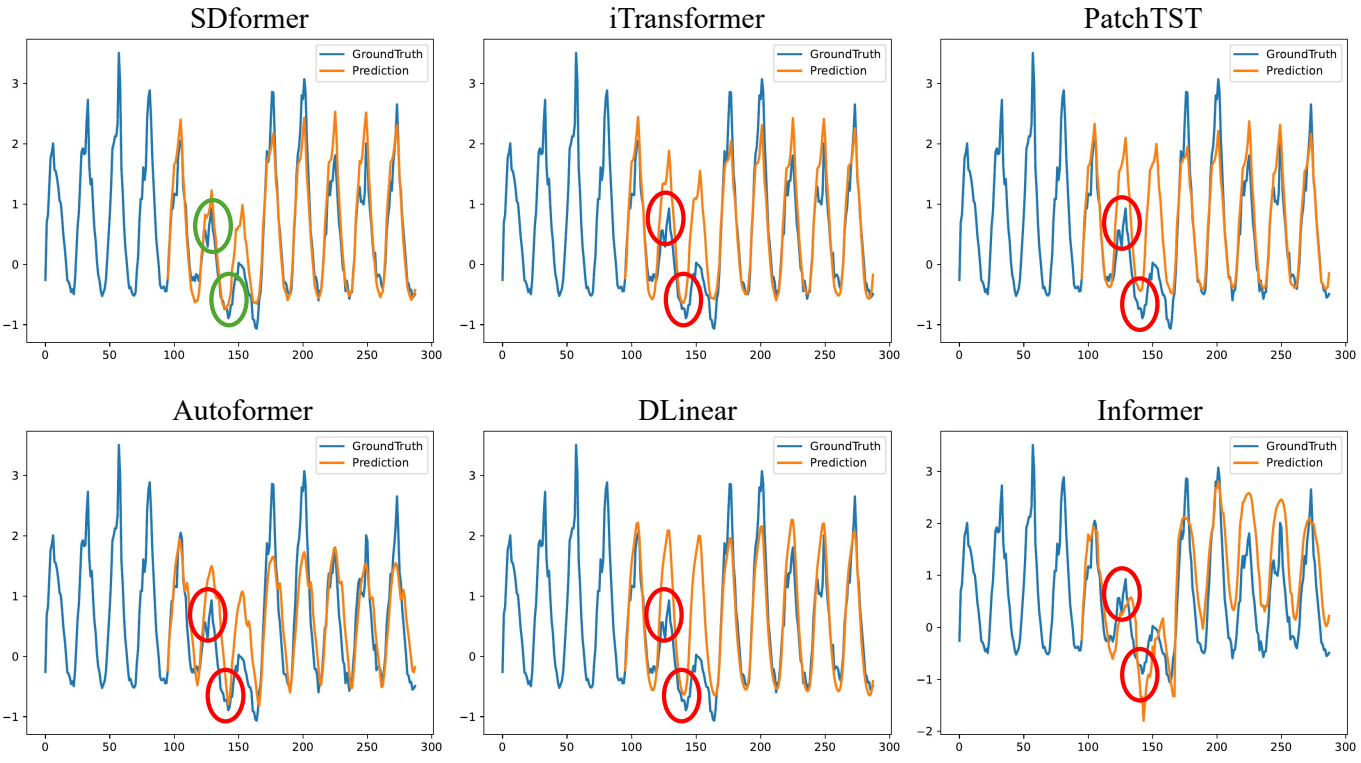

Fig. 10. Visualization of the results of setting input-96-predict-192 on the ECL dataset.

significantly better than PatchTST. Additionally, SDformer without the Transformer block also showed robust performance, with a training speed notably faster than the other models, similar to DLinear. As shown in Fig. 7 (right), in the low-dimensional ILI dataset, the training speed and memory consumption of SDformer were comparable to those of iTransformer and PatchTST, while its predictive performance was significantly better than that of the other models. Furthermore,

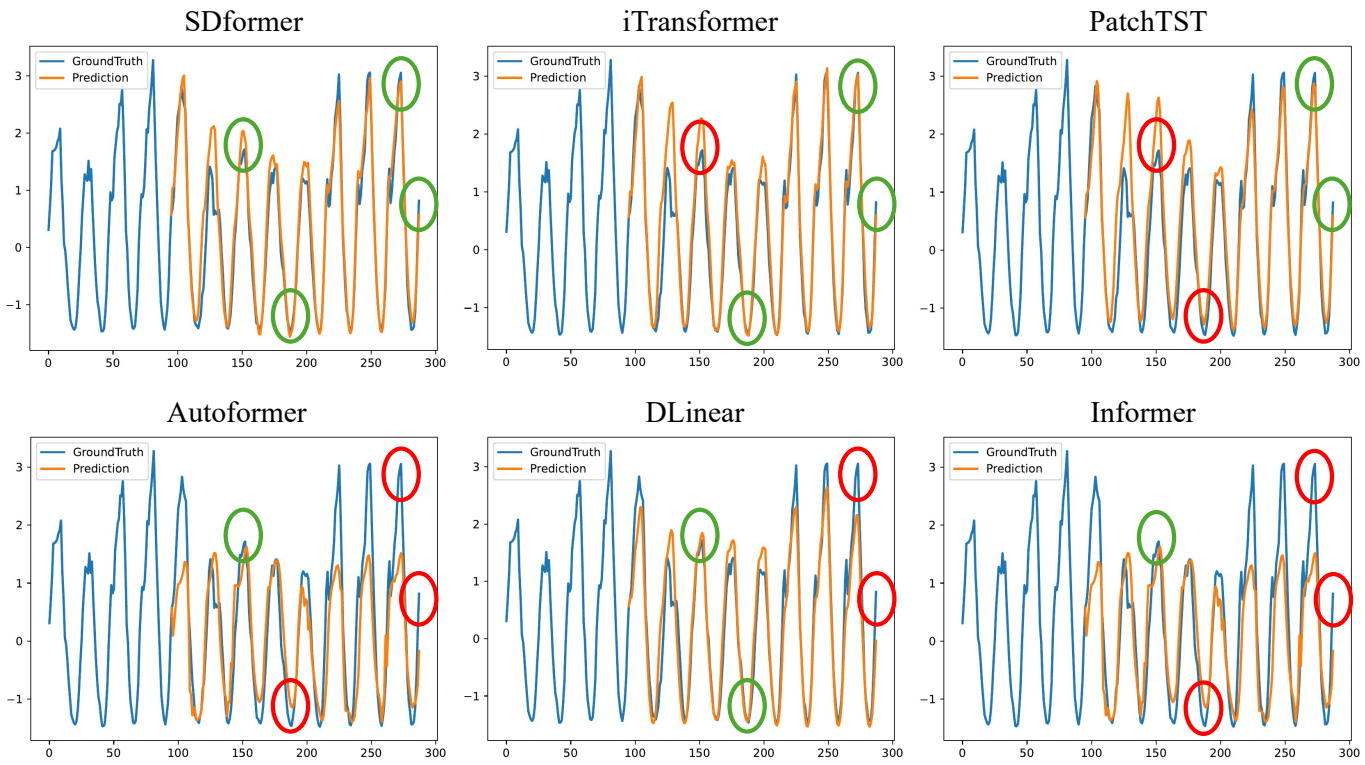

Fig. 11. Visualization of the results of setting input-96-predict-192 on the traffic dataset.

the SDformer variant without the Transformer block achieves performance on par with iTransformer, while attaining faster training speed than all Transformer-based baselines, second only to DLinear.

I. Visualization of prediction results

To visually demonstrate the predictive performance of SDformer in comparison to other models, we conducted tests on several sequences from the ECL and Traffic datasets and presented the prediction results. The models compared include iTransformer, PatchTST, Autoformer, DLinear, and Informer. Fig. 8 and Fig. 9 depict the prediction outcomes on the ECL and Traffic datasets, respectively, with a lookback length of 96 and a prediction length of 96. It is evident that SDformer achieves superior predictive accuracy at this prediction length. For instance, in Fig. 8, around the x-coordinates 125 and 150, SDformer's predictions are closer to the actual values. This improvement is likely due to the integration of time series decomposition information in the Encoder structure, which enhances SDformer's sensitivity to trends in the time series. Fig. 10 and Fig. 11 illustrate the prediction results on the ECL and Traffic datasets, respectively, with a lookback length of 96 and a prediction length of 192. The results indicate that even with an extended prediction length, SDformer consistently outperforms other models, aligning more closely with the actual value fluctuations.

V. Conclusion

This paper presents a comprehensive review of existing Transformer-based time series forecasting models and their utilization of time series decomposition techniques. In response, we propose a novel approach to leverage time series decomposition technology and introduce the SDE block. Experimental results demonstrate that the SDE block can effectively capture inter-variable correlations and extract more prominent temporal features. Moreover, integrating the SDE block into many time series forecasting models significantly enhances their predictive performance.

In summary, this paper introduces and validates the effectiveness of the SDE block in addressing time series forecasting problems, offering a fresh perspective for future research endeavors. However, it is worth noting that the applicability of SDE is currently limited to long-term time series forecasting tasks, and its performance has not been extensively validated on other types of datasets such as classification, anomaly detection, and imputation. Future work may extend the use of SDE block to broader contexts, particularly in scenarios involving missing data, where decomposition-based priors could prove beneficial. These directions could further expand the utility and generalizability of the SDformer in more complex and practical time series applications.

References

[1] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.

[2] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4356–4364.

[3] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 6913–6925, 2023.

[4] A. Nandi, A. De, A. Mallick, A. I. Middya, and S. Roy, "Attention based long-term air temperature forecasting network: ALTF Net," *Knowledge-Based Systems*, vol. 252, p. 109442, 2022.

[5] C.-X. Zhang, J. Li, X.-F. Huang, J.-S. Zhang, and H.-C. Huang, "Forecasting stock volatility and value-at-risk based on temporal convolutional networks," *Expert Systems with Applications*, vol. 207, p. 117951, 2022.

[6] X. Chen, Z. Su, L. Jin, and S. Li, "A correntropy-based echo state network with application to time series prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 2, pp. 425–435, 2025.

[7] 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.

[8] D. Luo and X. Wang, "ModernTCN: A modern pure convolution structure for general time series analysis," in *The twelfth international conference on learning representations*, 2024.

[9] 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.

[10] A. Das, W. Kong, A. Leach, S. K. Mathur, R. Sen, and R. Yu, "Long-term forecasting with TiDE: Time-series dense encoder," *Transactions on Machine Learning Research*, 2023.

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

[12] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-Variation modeling for general time series analysis," in *International Conference on Learning Representations*, 2023.

[13] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[14] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

[15] X. Cai and D. Li, "M-EDEM: A MNN-based empirical decomposition ensemble method for improved time series forecasting," *Knowledge-Based Systems*, vol. 283, p. 111157, 2024.

[16] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning *et al.*, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[17] H. Abbasimehr, A. Behboodi, and A. Bahrini, "A novel hybrid model to forecast seasonal and chaotic time series," *Expert Systems with Applications*, vol. 239, p. 122461, 2024.

[18] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Y. Zhang, Y. Liang, G. Pang, D. Song, and S. Pan, "Self-Supervised learning for time series analysis: Taxonomy, progress, and prospects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.

[19] 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 (ICML)*. PMLR, 2022, pp. 27 268–27 286.

[20] Z. Wang, S. Ruan, T. Huang, H. Zhou, S. Zhang, Y. Wang, L. Wang, Z. Huang, and Y. Liu, "A lightweight multi-layer perceptron for efficient multivariate time series forecasting," *Knowledge-Based Systems*, vol. 288, p. 111463, 2024.

[21] R. Wang, X. Pei, J. Zhu, Z. Zhang, X. Huang, J. Zhai, and F. Zhang, "Multivariable time series forecasting using model fusion," *Information Sciences*, vol. 585, pp. 262–274, 2022.

[22] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[23] X. Wang, Y. Kang, R. J. Hyndman, and F. Li, "Distributed arima models for ultra-long time series," *International Journal of Forecasting*, vol. 39, no. 3, pp. 1163–1184, 2023.

[24] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[25] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[26] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowledge-Based Systems*, vol. 181, p. 104785, 2019.

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

[28] Z. Zhang, Z. Lei, M. Zhou, H. Hasegawa, and S. Gao, "Complex-valued convolutional gated recurrent neural network for ultrasound beamforming," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2024.

[29] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based models for speech recognition," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[30] X. Li, Y. Liu, K. Wang, and F.-Y. Wang, "A recurrent attention and interaction model for pedestrian trajectory prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1361–1370, 2020.

[31] C. Luo, X. Zhao, Y. Sun, X. Li, and Y. Ye, "PredRANN: The spatiotemporal attention convolution recurrent neural network for precipitation nowcasting," *Knowledge-Based Systems*, vol. 239, p. 107900, 2022.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[34] Z. Zhang, Z. Lei, M. Omura, H. Hasegawa, and S. Gao, "Dendritic learning-incorporated vision transformer for image recognition," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 539–541, 2024.

[35] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[36] 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*, 2022.

[37] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *International Conference on Learning Representations*, 2023.

[38] P. Zeng, G. Hu, X. Zhou, S. Li, P. Liu, and S. Liu, "Muformer: A long sequence time-series forecasting model based on modified multi-head attention," *Knowledge-Based Systems*, vol. 254, p. 109584, 2022.

[39] K. Fu, H. Li, and X. Shi, "CTF-former: A novel simplified multi-task learning strategy for simultaneous multivariate chaotic time series prediction," *Neural Networks*, vol. 174, p. 106234, 2024.

[40] Y. Wang, H. Wu, J. Dong, Y. Liu, Y. Qiu, H. Zhang, J. Wang, and M. Long, "Timexer: Empowering transformers for time series forecasting with exogenous variables," *Advances in Neural Information Processing Systems*, 2024.

[41] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, "TimeMixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations*, 2024.

[42] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *International Conference on Machine Learning (ICML)*, vol. 235. PMLR, 2024, pp. 10 041–10 071.

[43] J. Li, Z. Lei, Z. Zhang, H. Li, Y. Todo, and S. Gao, "Alternating excitation-inhibition dendritic computing for classification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 11, pp. 5431–5441, 2024.

[44] T. Liu, J. Li, Z. Zhang, H. Yu, and S. Gao, "FD-GRNet: A dendritic-driven gru framework for advanced stock market prediction," *IEEE Access*, vol. 13, pp. 28 265–28 279, 2025.

[45] C. Lv, Y. Wang, D. Han, X. Zheng, X. Huang, and D. Li, "Efficient and effective time-series forecasting with spiking neural networks," in *International Conference on Machine Learning (ICML)*, vol. 235. PMLR, 2024, pp. 33 624–33 637.

[46] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *International Conference on Learning Representations*, 2024.

[47] G. Dudek, "STD: A seasonal-trend-dispersion decomposition of time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10 339–10 350, 2023.

[48] K. Bandara, C. Bergmeir, and H. Hewamalage, "LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 4, pp. 1586–1599, 2020.
[49] C. Deng, Y. Huang, N. Hasan, and Y. Bao, "Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition," *Information Sciences*, vol. 607, pp. 297–321, 2022.
[50] H. He, S. Gao, T. Jin, S. Sato, and X. Zhang, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," *Applied Soft Computing*, vol. 108, p. 107488, 2021.
[51] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations*, 2020.

Jiayi Li received his M.E. degree from University of Toyama, Toyama, Japan in 2023, where he is currently pursuing his Ph.D. degree. His current research interests include machine learning, evolutionary computation, neural networks, and bioinformatics.

Zhang Zihang received the Ph.D. degree in science and engineering from the University of Toyama, Toyama, Japan, in 2025. He is currently an Assistant Professor with the Faculty of Engineering, University of Toyama, Toyama, Japan. His current research interests are neural network for real-world applications and computational intelligence.

Zhang Chao received his Ph.D. at Iwate University (Japan) in March 2017. He is now a specially appointed professor at the Faculty of Engineering, University of Toyama (Japan). His research interest is in computer vision, especially anomaly detection. He is a member of the IEEE, IEEJ, ITE, JSAI, and IEICE.

Jun Tang earned his M.E. degree from Nagoya University in 1988 and continued his doctoral studies there from 1988 to 1990. His research interests span signal processing, artificial intelligence (including neural networks and machine learning), and computer operating systems, among other areas. Currently, he holds the position of Chairman and Chief Architect at Wicresoft Co., Ltd., USA. Additionally, he served as a visiting professor at several universities, including Beijing University and Dalian University of Technology.

Shangce Gao received his Ph.D. degree in Innovative Life Science from University of Toyama, Toyama, Japan in 2011. He is currently a Professor with the Faculty of Engineering, University of Toyama, Japan. His current research interests include nature-inspired technologies, machine learning, and neural networks for real-world applications. He serves as an Associate Editor for many international journals such as IEEE Transactions on Neural Networks and Learning Systems, and IEEE/CAA Journal of Automatica Sinica.