

DRFormer: Multi-Scale Transformer Utilizing Diverse Receptive Fields for Long Time-Series Forecasting

Ruixin Ding*

East China Normal University
Shanghai, China
51265901027@stu.ecnu.edu.cn

Yuqi Chen*

Fudan University
Shanghai, China
chenyuqi21@m.fudan.edu.cn

Yu-Ting Lan

Shanghai Jiao Tong University
Shanghai, China
lanyuting8806@sjtu.edu.cn

Wei Zhang†

East China Normal University
Shanghai, China
zhangwei.thu2011@gmail.com

Abstract

Long-term time series forecasting (LTSF) has been widely applied in finance, traffic prediction, and other domains. Recently, patch-based transformers have emerged as a promising approach, segmenting data into sub-level patches that serve as input tokens. However, existing methods mostly rely on predetermined patch lengths, necessitating expert knowledge and posing challenges in capturing diverse characteristics across various scales. Moreover, time series data exhibit diverse variations and fluctuations across different temporal scales, which traditional approaches struggle to model effectively. In this paper, we propose a dynamic tokenizer with a dynamic sparse learning algorithm to capture diverse receptive fields and sparse patterns of time series data. In order to build hierarchical receptive fields, we develop a multi-scale Transformer model, coupled with multi-scale sequence extraction, capable of capturing multi-resolution features. Additionally, we introduce a group-aware rotary position encoding technique to enhance intra- and inter-group position awareness among representations across different temporal scales. Our proposed model, named *DRFormer*, is evaluated on various real-world datasets, and experimental results demonstrate its superiority compared to existing methods. Our code is available at: <https://github.com/ruixindingECNU/DRFormer>.

CCS Concepts

• Applied computing → Forecasting; • Computing methodologies → Neural networks.

Keywords

time series forecasting, multi-scale transformer, dynamic sparse network, position information encoding

*Both authors contributed equally to this research.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).
CIKM '24, October 21–25, 2024, Boise, ID, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0436-9/24/10
<https://doi.org/10.1145/3627673.3679724>

ACM Reference Format:

Ruixin Ding, Yuqi Chen, Yu-Ting Lan, and Wei Zhang. 2024. DRFormer: Multi-Scale Transformer Utilizing Diverse Receptive Fields for Long Time-Series Forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*, October 21–25, 2024, Boise, ID, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3627673.3679724>

1 Introduction

Time series forecasting is crucial in various domains such as finance [10, 13], traffic prediction [5, 16, 44, 47], etc. The ability to accurately predict future values in time series data has significant implications for decision-making and planning [4, 40, 49]. The rapid advancement of deep learning has fueled remarkable progress in time series forecasting [1, 12, 17]. Among various deep learning approaches, Transformer [25, 40, 46, 50] and MLP-based [43, 48] models have demonstrated superior performance due to their ability to capture long-term dependency. Furthermore, recent works have witnessed a significant breakthrough in patch-based transformers [6, 25, 29] for the long-term time series forecasting (LTSF) task. These approaches divide time-series data into sub-level patches and utilize Transformer models [37] to generate meaningful input features. However, existing methods are mostly designed to break the time series into patches of a fixed length [6, 25] or with a set of predetermined patch lengths [29]. This static patching with fixed patch length requires expert knowledge and poses challenges for extracting temporal features and dependencies from various scales of temporal intervals.

To illustrate these challenges more comprehensively, it is essential to consider the following aspects: (i) The optimal sizes for patch division are influenced by the complex inherent characteristics and dynamic patterns of time series data, such as periodicity and trends. These intricate temporal patterns involve diverse variations and fluctuations across different temporal scales [2]. Currently, no established rules exist that can be validated either experimentally or theoretically to determine the optimal patch length. (ii) Real-world time series usually present multi-periodicity, such as daily, weekly, and monthly variations for traffic conditions [15, 21], or weekly and quarterly variations for electricity consumption [39]. These short-term and long-term recurring patterns contribute to the complexity of the forecasting task. (iii) The overall trend across the entire period and the specific time points of the learned sparse

patterns are significant for the LTSF task. The morning and evening peaks typically offer crucial information for traffic prediction. These characteristics require careful model design to introduce proper inductive bias.

To address these challenges, we propose a novel dynamic patching strategy coupled with a group-aware Roformer [33] network for LTSF. The proposed dynamic patching approach incorporates a dynamic sparse learning algorithm [41], which overcomes the need for expert knowledge by learning diverse receptive fields and extracts sparse patterns to identify critical points, thereby making it more applicable to real-world scenarios. To capture the inherent multi-resolution features, we introduce a Transformer model that enables multiple scales of temporal modeling. Additionally, we present a novel group-aware RoPE [33] method, named gRoPE, to enhance intra- and inter-group position awareness among representations with different temporal scales. By incorporating group awareness, *DRFormer* can effectively capture complex dependencies and interactions among different groups of representations, leading to improved forecasting performance. The contributions of the paper are as below:

- We propose a multi-scale Transformer model, named *DRFormer*, which employs a dynamic tokenizer to learn diverse receptive fields and utilizes multi-scale sequence extraction to capture inherent multi-resolution features.
- We introduce a group-aware rotary position encoding technique for learning intra- and inter-group relative position embedding. With such a design, *DRFormer* excels at capturing intricate dependencies among representations with distinct temporal scales.
- We conduct extensive experiments to demonstrate the superiority of *DRFormer* over various baseline models in diverse real-world scenarios.

2 Related Work

In this section, we discuss the related studies from the following aspects: transformer for long-term time series forecasting, CNNs for time-series forecasting, and relative position embedding.

2.1 Transformer for Long-term Time Series Forecasting

The adoption of Transformer-based models has emerged as a promising approach for long-term time series forecasting [20, 22, 24, 40, 45, 49, 50]. Among these models, Reformer [20] proposes locality-sensitive hashing attention for efficient and scalable sequence modeling. Informer [49] employs ProbSparse self-attention to extract important keys efficiently. Autoformer [40] introduces a novel decomposition framework, along with an auto-correlation attention mechanism. FEDformer [50] utilizes Fourier transformation to model temporal characteristics and dynamics. Patch-based transformers [6, 25, 29], dividing time-series data into sub-level patches, have yielded significant enhancements in forecasting accuracy and complexity reduction. However, existing methods mainly model time series within limited or fixed patch lengths, which necessitate expert knowledge to select the optimal patch lengths and pose challenges in capturing diverse characteristics across varying scales. Very recently, [2] developed a multi-scale Transformer that divides the time series into different temporal resolutions. However, it

fails to learn a wide range of receptive fields given the selection of multiple patch lengths and static patching. In contrast, our proposed *DRFormer* can adeptly learn from a wide range of receptive fields, capture both overarching trends and nuanced variations, and extract the inherent multi-resolution properties of the data.

2.2 CNNs for Time-Series Forecasting

In addition to Transformers, convolutional neural networks (CNNs) are highly regarded by researchers in the time-series community [8, 11, 14, 32]. To enhance the generalization capabilities of time-series tasks and encompass diverse receptive fields, [26] proposed the use of dilated convolution kernels as a structure-based low bandpass filter. Moreover, OS-CNN [34] introduced the Omni-Scale block (OS-block) for 1D-CNNs, enabling the model to learn a range of diverse receptive fields. Additionally, DSN [41] presented a dynamic sparse network that can adaptively cover various receptive fields without the need for extensive hyperparameter tuning. Drawing inspiration from CNNs, we integrate dynamic sparse networks and multi-scale modeling into the Transformer structure, enabling the model to leverage the advantages of CNNs.

2.3 Relative Position Embedding

In the realm of natural language processing, several approaches for relative position embedding (RPE) have been proposed [7, 18, 28, 30, 33]. Among these, RoPE [33] is a representative approach that encodes relative position by multiplying the context representations with a rotation matrix. Additionally, the adoption of RoPE has been widespread among large language models as a means to extend the context windows [3, 35]. In this paper, we apply RoPE to enhance position awareness in the LTSF task. Besides, we propose group-aware rotary position embedding to encode intra- and inter-group relative position information into the attention mechanism, which is better suited for extracting multi-scale characteristics in time series data.

3 Methodology

In this section, we detailedly describe our method, *DRFormer*, which captures multi-scale characteristics with diverse receptive fields and multi-resolution representations for LTSF, as shown in Figure 1.

We briefly introduce the intuition of our *DRFormer*. As aforementioned, time series data is characterized by multi-scale properties and identifying critical timestamps provides crucial insights for prediction. To model such inductive bias, we propose a novel dynamic patching strategy coupled with a multi-scale Transformer to inject such characteristics priors into the forecasting pipelines. As illustrated in Figure 2, *DRFormer* first incorporates a dynamic sparse network within the tokenizer, which simultaneously learns adaptive receptive fields and sparse patterns of the data. Next, we propose to transform time-series sequences into multi-scale sequences, allowing each token to represent features at multiple granularities. Finally, to capture cross-group interactions, we introduce a group-aware rotary position encoding technique for learning intra- and inter-group relative position embeddings.

In the following, we first formulate the problem and give an overview of our method. Then, we delve into the details of the

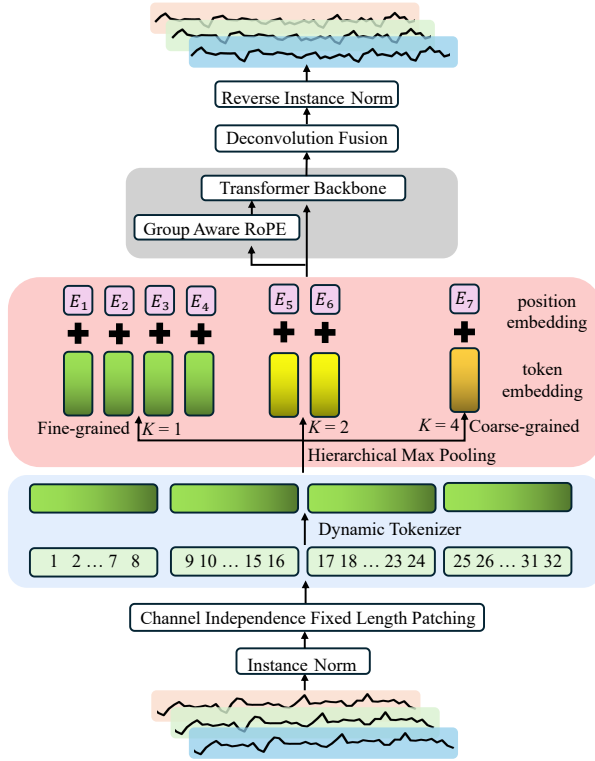


Figure 1: Overview of our DRFormer. DRFormer first utilizes a dynamic tokenizer to capture diverse receptive fields of each tokenizer. A hierarchical max pooling operation is then applied to leverage the multi-resolution property inherent in time series data. The multi-resolution time series data is then encoded by a group-aware Transformer model and finally processed by a deconvolution operation.

dynamic tokenizer, multi-scale sequence extraction and the multi-scale Transformer with group-aware rotary embedding.

3.1 Problem Formulation

The task of time series forecasting involves predicting a future series of length- O with the highest probability, based on a given past series of length- I , denoted as *input- I -predict- O* . In the context of long-term forecasting, the objective is to predict the future over a longer time horizon, specifically with a larger value of O . Given a multivariate time series, i.e., $X^{1:I} \in \mathbb{R}^{I \times C}$, where I denotes the length of the time series, and C denotes the number of variates. The general objective of this research is to predict $X^{I+1:I+O} \in \mathbb{R}^{O \times C}$ with $X^{1:I}$ as input. Note that our model is conducted on each variate of time series, i.e., channel independence. Thus, we denote x_i as the time series for the i -th variate and omit the variate of time series for simplicity.

3.2 Dynamic Tokenizer

In this section, we describe how we discover and exploit various receptive fields adaptively with the dynamic sparse network. As aforementioned, static patching requires expertise to determine the

length of temporal patch, in which complex inherent characteristics and dynamic patterns of time series data should be considered. Moreover, integrating fine-grained and coarse-grained features is crucial to model diverse variations and fluctuations, which is challenging for the pre-defined static model. To address this, we involve a novel dynamic tokenizer to dynamically capture the optimal scale features through a sparse learning strategy.

3.2.1 Data Normalization. To migrate the distribution shift between the training and testing data [19, 25, 36], we employ instance normalization on the input data. Specifically, we normalize each variable x_i by subtracting its mean and dividing by its standard deviation before applying patching. After the output prediction, we add back the normalized values to restore the original distribution with the mean and standard deviation.

3.2.2 Static Patching. The proposed dynamic tokenizer, as depicted in Figure 2, first adopts a static patching operation to transform the input time series into sub-level patches [25]. Next, a dynamic linear transformation is applied to obtain dynamic representations, which serve as the input token embeddings. Specifically, we denote the patch length as P and the stride as S . Each input univariate time series x_i is first divided into patches $p_i \in \mathbb{R}^{P \times N}$ where N is the number of patches, $N = \lfloor \frac{(I-P)}{S} \rfloor + 2$.

3.2.3 Dynamic Linear Transformation. Previous works adopt linear transformation to obtain input tokens [25]. Assume that $w^E \in \mathbb{R}^{P \times D}$ and $b^E \in \mathbb{R}^D$, where D is the number of hidden dimensions of the model. The embeddings are obtained by $e_i = p_i^\top w^E + b^E$. However, these embeddings are limited by a fixed receptive field, i.e., all dimensions of each token have the same receptive field size of P . To address the limitation, we introduce a learnable sparse mask, i.e.,

$$f_i = p_i^\top (w^E \odot I(w^E)) + b^E, \quad (1)$$

where $I(\cdot) : \mathbb{R}^{P \times D} \rightarrow \{0, 1\}^{P \times D}$ denotes an indicator function [41], \odot denotes the element-wise product. A dynamic linear layer with sparse ratio SR satisfies that $\|I(w^E)\|_0 \leq (1 - SR) \times P \times D$. In Figure 2, a dynamic linear layer is depicted, showcasing the first, second, and last groups. The first dimension of each group learns receptive fields of sizes 3, 7, and 14, respectively. By definition of the token receptive field (tRF, as defined below), a dynamic linear layer is inherently designed to capture a comprehensive set of receptive fields, denoted as $RF = \{0, 1, \dots, P\}$.

Remark: Token Receptive Field (tRF) for dynamic linear layer.

The receptive field (RF) in CNN layers is defined as the region in the input that the feature is looking at. In the context of the dynamic linear layer, tRF is defined as the region in the input that a token is looking at. Mathematically, assume that the indicator function of the weight vector w_i^E is defined as $Ind = I(w_i^E) \in \{0, 1\}^P$. Let \mathcal{S} be the set of indices where $Ind_j = 1$, i.e., $\mathcal{S} = \{Ind_j = 1 | 1 \leq j \leq P\}$, tRF is calculated as

$$tRF = \begin{cases} \max(\mathcal{S}) - \min(\mathcal{S}) + 1, & \text{if } Ind \neq \mathbf{0} \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

3.2.4 Group Partition. By design, during the training phase, the total number of activated weights must not exceed $(1 - SR) \times P \times D$. However, a larger token receptive field occupies the majority of the

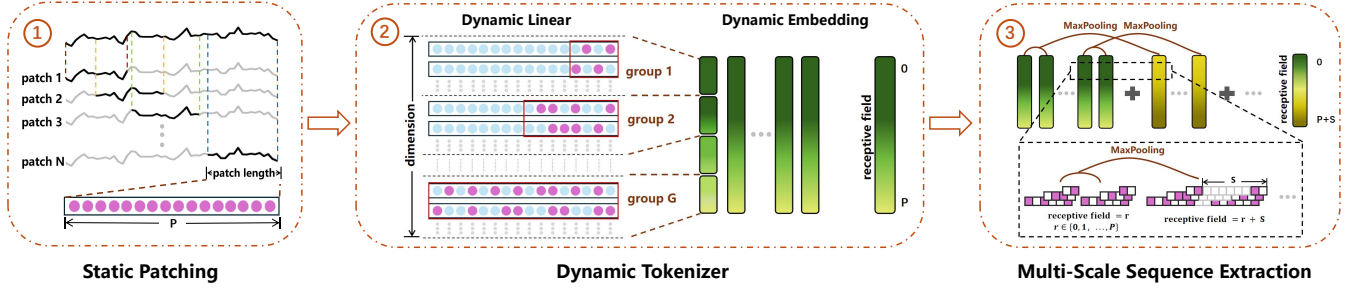


Figure 2: Illustration of static patching, dynamic tokenizer and multi-scale sequence extraction. ① Taking $P = 16$ as an example, the input sequence is transformed into N patches. ② The dynamic linear layer is divided into G groups and the corresponding exploration regions for each group are shown in the red boxes. The number of group G is set to 4 and the sparse ratio SR is set to 0.5. Purple (blue) circles indicate activated (inactivated) weights. ③ Hierarchical max-pooling on patched tokens yields multi-group representations with a more comprehensive set of receptive fields as shown in Equation 6.

tokens, especially as the sparsity ratio SR decreases [41], leading to a leak of local patterns being captured. To address this problem, we utilize a group partition strategy. In this approach, the dynamic linear layer is divided into several groups, whose corresponding exploration regions are of different sizes. Specifically, the weights $w^E \in \mathbb{R}^{P \times D}$ are split into G groups along the output channel, that is, $w_1^E, \dots, w_G^E \in \mathbb{R}^{P \times \frac{D}{G}}$. For the i -th group, the exploration region comprises the last $\lceil \frac{P}{G} \rceil$ positions, thereby ensuring that activated weights only appear within these positions and the number of activated weights must not exceed $(1 - SR) \times \lceil \frac{P}{G} \rceil \times \frac{D}{G}$. Additionally, we define the candidate set C as the set of weights that can be activated. Figure 2 illustrates a dynamic tokenizer with four groups, where the first group allows activation only for the last four positions.

Algorithm 1: Training algorithm for the indicator

Input: Dataset \mathcal{D} , learning rate α , initial weight w^E , candidate region C .

```

1 for  $t \leftarrow 1$  to  $T$  do
2   Sample a Batch  $B_t \sim \mathcal{D}$ ;
3    $L_t = \sum_{i \in B_t} L(f_\theta(x_i), y_i)$ ;
4   Update  $w^E$  and the network using gradient descent;
5   if  $t \bmod \Delta t = 0$  then
6     Calculate  $n$  using Eq. (3) with  $t, T, \alpha$  as inputs;
7      $\mathbb{I}_{\text{prune}} = \text{ArgTopK}(-|w^E|, n)$ ;
8      $\mathbb{I}_{\text{grow}} = \text{RandomK}(C \cap [I(w^E) = 0], n)$ ;
9      $\mathbf{I}^l(\cdot) \leftarrow \text{Update } \mathbf{I}^l(\cdot) \text{ using } \mathbb{I}_{\text{prune}} \text{ and } \mathbb{I}_{\text{grow}}$ ;
```

3.2.5 Training the Indicator. Updating the indicator directly through backpropagation is a non-differentiable operation. We adopt a heuristic algorithm to explore and update the weights [41]. In the selection of the masking strategy, various possibilities were explored. We ultimately determine masking out weights with small magnitudes as the masking strategy since it is intuitive and has been experimentally proven to be the most effective (more details in Section 4.4.3). The whole algorithm is listed in Algorithm 1.

Specifically, assume we have a total of T training iterations. For every Δt iteration, we perform one step of update. At iteration t , since weights with smaller magnitudes contribute insignificantly or negligibly to the overall computation, we select n weights with the smallest absolute values from the candidate set and set these weights to 0, effectively deactivating them. To ensure recoverability from pruning, we randomly reintroduce n weights, matching the number of pruned weights, to facilitate better exploration of activated weights. This dynamic and plastic weight exploration approach allows for adaptive exploration during the training process. The value of n is controlled by the annealing function, which adjusts the pruning rate over time:

$$n = \frac{\alpha}{2} \left(1 + \cos\left(\frac{t\pi}{T}\right) \right) \times \|I(w^E)\|_0, \quad (3)$$

where α is a hyper-parameter to control the learning rate.

3.3 Multi-Scale Sequence Extraction

Time series data is characterized by both fine-grained local details and coarse-grained global composition, and capturing both aspects is crucial for comprehensive modeling. To address this, we propose a multi-scale approach that utilizes multi-group representations through hierarchical max-pooling on patched tokens. Specifically, we denote $f_i \in \mathbb{R}^{D \times N}$ as a latent representation of patching of the dynamic tokenizer. The hierarchical max-pooling strategy involves the application of max-pooling from fine-grained to coarse-grained with non-overlapping windows of diverse on consecutive patches to generate multi-resolution representations as follows:

$$\mathcal{F}_i^0 = \{f_i^{K_1}, f_i^{K_2}, \dots, f_i^{K_k}\} \quad (4)$$

where k denotes the number of multi-scale sequences and $f_i^{K_1}$ is the original sequence from the dynamic tokenizer. Here, $f_i^{K_j} \in \mathbb{R}^{D \times \lceil \frac{N}{K_j} \rceil}$, $j \in \{1, 2, \dots, k\}$ denotes the representation after max pooling operation, i.e.,

$$f_{i,p}^{K_j} = \text{MaxPooling}(f_{i,p}, f_{i,p+1}, \dots, f_{i,p+K_j-1}), \quad (5)$$

where $p + K_j - 1 \leq N$ and $f_{i,p}$ denotes the p -th token in f_i . Besides, we denote $S_k = \{K_1, \dots, K_k\}$ as the set of different kernels. As is shown in Figure 1, we design the S_k as a set of power two, i.e., $S_k =$

$\{1, 2, \dots, 2^{k-1}\}$, to empower the model with multi-scale ability and obtain a comprehensive representation from fine-grained to coarse-grained temporal information. With such a design, the tokens can capture a more comprehensive set of receptive fields of:

$$\widehat{RF} = \{0, 1, \dots, P + (2^{k-1} - 1) \cdot S\}. \quad (6)$$

3.4 Multi-Scale Transformers

In this section, we formulate our multi-scale transformer. To overcome the limitations of position awareness of multi-scale representations for the transformer model, we propose a group-aware relative position encoding technique, which empowers our model to effectively capture intricate dependencies and interactions among different groups of representations, resulting in enhanced forecasting performance.

3.4.1 Group-Aware Rotary Position Encoding. Instead of using traditional absolute or relative position encoding, which ignores the inductive bias of intra and inter-group relations and treats different group embedding equally, we propose a novel group-aware rotary position encoding technique to capture intricate dependencies and interactions among different representation groups. We follow Roformer [33] and formulate the position encoding as the rotary matrix with pre-defined angle parameters. We derive the group-aware rotary position encoding for $f_i^{K_j} \in \mathbb{R}^{D \times \lceil \frac{N}{K_j} \rceil}$. Let $f_{i,m}^{K_j}$ be the m -th embedding for i -th group and the intra-group rotary position encoding for $f_{i,m}^{K_j}$ can be formulated as:

$$R_{\Theta,i,m}^{d,intra} = \begin{pmatrix} \cos \hat{m}\theta_1 & -\sin \hat{m}\theta_1 & 0 & 0 & \dots & 0 & 0 \\ \sin \hat{m}\theta_1 & \cos \hat{m}\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos \hat{m}\theta_2 & -\sin \hat{m}\theta_2 & \dots & 0 & 0 \\ 0 & 0 & \sin \hat{m}\theta_2 & \cos \hat{m}\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos \hat{m}\theta_{d/2} & -\sin \hat{m}\theta_{d/2} \\ 0 & 0 & 0 & 0 & \dots & \sin \hat{m}\theta_{d/2} & \cos \hat{m}\theta_{d/2} \end{pmatrix}, \quad (7)$$

where $\hat{m} = m / \lceil \frac{N}{K_i} \rceil$, indicates the relative position of m within the sequence, and pre-defined parameters:

$$\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \dots, d/2]\}. \quad (8)$$

Since $R_{\Theta,i,m}^{d,intra}$ ignores the group information, we define another inter-group rotary position encoding as:

$$R_{\Theta,i,m}^{d,inter} = \begin{pmatrix} \cos i\theta_1 & -\sin i\theta_1 & 0 & 0 & \dots & 0 & 0 \\ \sin i\theta_1 & \cos i\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos i\theta_2 & -\sin i\theta_2 & \dots & 0 & 0 \\ 0 & 0 & \sin i\theta_2 & \cos i\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos i\theta_{d/2} & -\sin i\theta_{d/2} \\ 0 & 0 & 0 & 0 & \dots & \sin i\theta_{d/2} & \cos i\theta_{d/2} \end{pmatrix}. \quad (9)$$

Here, intra- and inter-group rotary position encoding share the same parameters.

3.4.2 Transformer Backbone. The Transformer model is widely recognized for its effectiveness in sequence modeling tasks. However, to further improve its capacity to capture both inter-group and intra-group correlations, we introduce a novel group-aware rotary position encoding technique. Specifically, given the multi-scale inputs \mathcal{F}_i^l and corresponding inter- and intra-group rotary

position encoding, i.e., $R_{\Theta}^{d,inter}, R_{\Theta}^{d,intra}$. We first calculate keys and queries, i.e.,

$$\{Q, K\}^{\{inter,intra\}} = \mathcal{F}_i^l W_{\{Q,K\}} R_{\Theta}^{d,\{inter,intra\}}, \quad (10)$$

where $W_{\{Q,K\}}$ represents the transformation matrices for queries and keys, respectively. Next, we define the group-aware attention:

$$\text{Attn}(\mathcal{F}_i^l) = \text{softmax} \left(\frac{\left((Q^{inter} K^{inter})^T + (Q^{intra} K^{intra})^T \right)}{\sqrt{d_k}} \right) \mathcal{F}_i^l W_V. \quad (11)$$

The multi-scale Transformer is a highly efficient model that effectively extracts multi-scale information from time series data while also capturing group awareness. Additionally, each Transformer layer incorporates a feed-forward network and layer normalization [37]. The mathematical formulations are as follows:

$$\begin{aligned} \mathcal{F}^{l,1} &= \mathcal{F}^{l-1} + \text{LN} \left(\text{Attn}(\mathcal{F}_i^{l-1}) \right), \\ \mathcal{F}^l &= \mathcal{F}^{l,1} + \text{LN} \left(\text{FFN}(\mathcal{F}_i^{l,1}) \right), \end{aligned} \quad (12)$$

where \mathcal{F}^l indicates the output for the l -th Transformer layer with the input \mathcal{F}^0 defined in Eq. (4), LN and FFN represent the layer normalization operation and the feed-forward network, respectively [37].

3.4.3 Representation Fusion with Deconvolution. One possible approach is to use these embeddings for prediction directly. However, to achieve predictions that incorporate both fine-grained local details and coarse-grained global composition, we propose a fusion technique that combines these representations using deconvolution operations. Specifically, the output from the Transformer backbone is first split into multi-scale sequences:

$$o_i^m = \{o_i^{K_1}, o_i^{K_2}, \dots, o_i^{K_k}\}. \quad (13)$$

We then perform a deconvolution operation [42], which is a technique that upsamples features:

$$de_i^{K_j} = \text{Deconv} \left(o_i^{K_j}, K_j \right), \quad (14)$$

where $de_i^{K_j} \in \mathbb{R}^{N \times d}$. Finally, the output is obtained by:

$$O_i = \sum_{j=1}^k de_i^{K_j}. \quad (15)$$

3.5 Loss Function

We adopt the Mean Squared Error (MSE) loss to measure the discrepancy between the forecasting results and the ground truth observations. Let $\hat{X}^{I+1:I+O}$ and $X^{I+1:I+O}$ be the predictions and real observations from time $I+1$ to $I+O$. We denote $\hat{x}_i^{I+1:I+O}$ and $x_i^{I+1:I+O}$ be the predictions and real observations from the i -th variate. The training loss is defined as:

$$\mathcal{L} = \mathbb{E}_{X \sim \mathcal{D}} \left[\frac{1}{C} \sum_{i=1}^C \left\| \hat{x}_i^{I+1:I+O} - x_i^{I+1:I+O} \right\|_2^2 \right]. \quad (16)$$

Table 1: Experimental results for multivariate time series forecasting. Bold (Underlined) values indicate the best (second-best) performance. The input length is 96 for each dataset, and the prediction lengths for the ECL, Traffic, and ETT datasets are {96, 192, 336, 720}, while {24, 36, 48, 60} for the ILI dataset. (avg for the averaged results on the four different prediction lengths)

| Models | | <i>DRFormer</i> | | Koopa | | PatchTST | | TimesNet | | Dlinear | | ETSformer | | Autoformer | | Informer | | Reformer | |
|---------|-----|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|-------|------------|-------|----------|-------|----------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ECL | 96 | <u>0.163</u> | <u>0.254</u> | 0.147 | 0.247 | 0.178 | 0.264 | 0.168 | 0.272 | 0.197 | 0.282 | 0.187 | 0.304 | 0.201 | 0.317 | 0.274 | 0.368 | 0.312 | 0.402 |
| | 192 | 0.174 | 0.264 | <u>0.181</u> | 0.276 | 0.184 | <u>0.270</u> | 0.184 | 0.289 | 0.196 | 0.285 | 0.199 | 0.315 | 0.222 | 0.334 | 0.296 | 0.386 | 0.348 | 0.433 |
| | 336 | 0.193 | 0.282 | <u>0.195</u> | 0.290 | 0.201 | <u>0.286</u> | 0.198 | 0.300 | 0.209 | 0.301 | 0.212 | 0.329 | 0.231 | 0.338 | 0.300 | 0.394 | 0.350 | 0.433 |
| | 720 | 0.232 | <u>0.317</u> | <u>0.229</u> | 0.316 | 0.241 | 0.319 | 0.220 | 0.320 | 0.245 | 0.333 | 0.233 | 0.345 | 0.254 | 0.361 | 0.373 | 0.439 | 0.340 | 0.420 |
| | avg | <u>0.191</u> | 0.279 | 0.188 | <u>0.282</u> | 0.201 | 0.285 | 0.193 | 0.295 | 0.212 | 0.300 | 0.208 | 0.323 | 0.227 | 0.338 | 0.311 | 0.397 | 0.338 | 0.422 |
| Traffic | 96 | 0.414 | 0.267 | 0.477 | 0.317 | <u>0.454</u> | <u>0.290</u> | 0.593 | 0.321 | 0.650 | 0.396 | 0.607 | 0.392 | 0.613 | 0.388 | 0.719 | 0.391 | 0.732 | 0.423 |
| | 192 | 0.427 | 0.271 | 0.500 | 0.339 | <u>0.461</u> | <u>0.291</u> | 0.617 | 0.336 | 0.598 | 0.370 | 0.621 | 0.399 | 0.616 | 0.382 | 0.696 | 0.379 | 0.733 | 0.420 |
| | 336 | 0.440 | 0.278 | 0.531 | 0.349 | <u>0.477</u> | <u>0.299</u> | 0.629 | 0.336 | 0.605 | 0.373 | 0.622 | 0.396 | 0.622 | 0.337 | 0.777 | 0.420 | 0.742 | 0.420 |
| | 720 | 0.474 | 0.296 | 0.566 | 0.366 | <u>0.510</u> | <u>0.316</u> | 0.640 | 0.350 | 0.645 | 0.394 | 0.632 | 0.396 | 0.660 | 0.408 | 0.864 | 0.472 | 0.755 | 0.423 |
| | avg | 0.439 | 0.278 | 0.519 | 0.343 | <u>0.476</u> | <u>0.299</u> | 0.620 | 0.336 | 0.625 | 0.383 | 0.621 | 0.396 | 0.628 | 0.379 | 0.764 | 0.416 | 0.741 | 0.422 |
| ETTh1 | 96 | 0.378 | 0.398 | 0.384 | 0.407 | 0.396 | 0.408 | <u>0.384</u> | 0.402 | 0.386 | <u>0.400</u> | 0.494 | 0.479 | 0.449 | 0.459 | 0.865 | 0.713 | 0.837 | 0.728 |
| | 192 | 0.425 | 0.429 | 0.447 | 0.435 | 0.445 | 0.440 | <u>0.436</u> | 0.429 | 0.437 | <u>0.432</u> | 0.538 | 0.504 | 0.500 | 0.482 | 1.008 | 0.792 | 0.923 | 0.766 |
| | 336 | 0.467 | 0.453 | 0.493 | 0.461 | 0.486 | 0.464 | 0.491 | 0.469 | <u>0.481</u> | <u>0.459</u> | 0.574 | 0.521 | 0.521 | 0.496 | 1.107 | 0.809 | 1.097 | 0.835 |
| | 720 | 0.491 | 0.478 | <u>0.512</u> | <u>0.487</u> | 0.491 | 0.490 | 0.521 | 0.500 | 0.519 | <u>0.516</u> | 0.562 | 0.535 | 0.514 | 0.512 | 1.181 | 0.865 | 1.257 | 0.889 |
| | avg | 0.440 | 0.440 | 0.459 | <u>0.448</u> | <u>0.455</u> | 0.451 | 0.458 | 0.450 | 0.456 | 0.452 | 0.542 | 0.510 | 0.496 | 0.487 | 1.040 | 0.795 | 1.029 | 0.805 |
| ETTh2 | 96 | 0.290 | 0.345 | 0.314 | 0.357 | <u>0.298</u> | <u>0.347</u> | 0.340 | 0.374 | 0.333 | 0.387 | 0.340 | 0.391 | 0.346 | 0.388 | 3.755 | 1.525 | 2.626 | 1.317 |
| | 192 | 0.367 | 0.393 | 0.378 | 0.398 | 0.382 | 0.396 | 0.402 | 0.414 | 0.477 | 0.476 | 0.430 | 0.439 | 0.456 | 0.452 | 5.602 | 1.931 | 11.120 | 2.979 |
| | 336 | 0.414 | 0.427 | <u>0.419</u> | 0.491 | 0.420 | <u>0.431</u> | 0.452 | 0.452 | 0.594 | 0.541 | 0.485 | 0.497 | 0.482 | 0.486 | 2.723 | 1.340 | 4.028 | 1.688 |
| | 720 | 0.426 | 0.446 | 0.445 | 0.456 | <u>0.433</u> | <u>0.449</u> | 0.462 | 0.468 | 0.831 | 0.657 | 0.500 | 0.497 | 0.515 | 0.511 | 3.467 | 1.473 | 5.381 | 2.015 |
| | avg | 0.374 | 0.403 | 0.389 | 0.426 | <u>0.383</u> | <u>0.406</u> | 0.414 | 0.427 | 0.559 | 0.515 | 0.439 | 0.456 | 0.450 | 0.459 | 3.887 | 1.567 | 5.789 | 2.000 |
| ETTm1 | 96 | 0.328 | 0.368 | 0.330 | 0.363 | 0.355 | 0.383 | 0.338 | 0.375 | 0.345 | 0.372 | 0.375 | 0.398 | 0.505 | 0.475 | 0.672 | 0.571 | 0.538 | 0.528 |
| | 192 | 0.364 | 0.387 | 0.379 | 0.393 | 0.393 | 0.400 | <u>0.374</u> | 0.387 | 0.380 | 0.389 | 0.408 | 0.410 | 0.553 | 0.496 | 0.795 | 0.669 | 0.658 | 0.592 |
| | 336 | 0.390 | 0.405 | <u>0.402</u> | 0.412 | 0.424 | 0.417 | 0.410 | <u>0.411</u> | 0.413 | 0.413 | 0.435 | 0.428 | 0.621 | 0.537 | 1.212 | 0.871 | 0.898 | 0.721 |
| | 720 | 0.449 | 0.439 | 0.475 | 0.448 | 0.477 | <u>0.446</u> | 0.478 | 0.450 | <u>0.474</u> | 0.453 | 0.499 | 0.462 | 0.670 | 0.561 | 1.166 | 0.823 | 1.102 | 0.841 |
| | avg | 0.383 | 0.400 | 0.397 | <u>0.404</u> | 0.412 | 0.412 | 0.400 | 0.406 | 0.403 | 0.407 | 0.429 | 0.425 | 0.587 | 0.517 | 0.961 | 0.734 | 0.799 | 0.671 |
| ETTm2 | 96 | 0.175 | 0.259 | <u>0.179</u> | <u>0.261</u> | <u>0.179</u> | 0.263 | 0.187 | 0.267 | 0.193 | 0.292 | 0.189 | 0.280 | 0.255 | 0.339 | 0.365 | 0.453 | 0.658 | 0.619 |
| | 192 | 0.241 | <u>0.303</u> | 0.245 | 0.306 | <u>0.244</u> | 0.302 | 0.249 | 0.309 | 0.284 | 0.362 | 0.253 | 0.319 | 0.281 | 0.340 | 0.533 | 0.563 | 1.078 | 0.827 |
| | 336 | 0.305 | 0.345 | 0.304 | 0.345 | 0.304 | 0.345 | 0.321 | 0.351 | 0.369 | 0.427 | 0.314 | 0.357 | 0.339 | 0.372 | 1.363 | 0.887 | 1.549 | 0.972 |
| | 720 | <u>0.408</u> | 0.400 | 0.406 | <u>0.402</u> | <u>0.408</u> | 0.405 | <u>0.408</u> | 0.403 | 0.554 | 0.522 | 0.414 | 0.413 | 0.433 | 0.432 | 3.379 | 1.388 | 2.631 | 1.242 |
| | avg | 0.282 | 0.327 | <u>0.284</u> | <u>0.329</u> | <u>0.284</u> | <u>0.329</u> | 0.291 | 0.333 | 0.350 | 0.401 | 0.293 | 0.342 | 0.327 | 0.371 | 1.410 | 0.823 | 1.479 | 0.915 |
| ILI | 24 | 1.887 | 0.863 | <u>1.974</u> | 0.900 | 2.101 | <u>0.866</u> | 2.317 | 0.934 | 2.398 | 1.040 | 2.527 | 1.020 | 3.483 | 1.287 | 5.764 | 1.677 | 4.400 | 1.382 |
| | 36 | <u>2.117</u> | 0.892 | 2.278 | 0.928 | 2.647 | 0.978 | 1.972 | <u>0.920</u> | 2.646 | 1.088 | 2.615 | 1.007 | 3.103 | 1.148 | 4.755 | 1.467 | 4.783 | 1.448 |
| | 48 | 1.929 | 0.827 | 2.156 | 0.910 | <u>2.056</u> | <u>0.882</u> | 2.238 | 0.940 | 2.614 | 1.086 | 2.359 | 0.972 | 2.669 | 1.085 | 4.763 | 1.469 | 4.832 | 1.465 |
| | 60 | 1.903 | 0.906 | 2.267 | 0.956 | 2.329 | 1.020 | <u>2.027</u> | <u>0.928</u> | 2.804 | 1.146 | 2.487 | 1.016 | 2.770 | 1.125 | 5.264 | 1.564 | 4.882 | 1.483 |
| | avg | 1.959 | 0.872 | 2.169 | <u>0.924</u> | 2.283 | 0.937 | <u>2.139</u> | 0.931 | 2.616 | 1.090 | 2.497 | 1.004 | 3.006 | 1.161 | 5.137 | 1.544 | 4.724 | 1.445 |

4 Experiments

4.1 Experimental Setting

4.1.1 Dataset Description. We conducted extensive experiments on time-series benchmark datasets. These datasets cover a variety of applications, including ETT and Electricity for electricity prediction, Exchange for financial applications, ILI for disease prediction, and Traffic for traffic prediction.

4.1.2 Baselines. We compare *DRFormer* with several Transformer-based models, including

- **Reformer** [20], which proposes a locality-sensitive hashing mechanism to reduce the time cost of self-attention calculation.
- **Informer** [49], which proposes a ProbSparse self-attention with distilling techniques to extract the most important keys.

- **Autoformer** [40], which proposes an auto-correlation attention mechanism and a novel decomposition architecture.
- **FEDformer** [50], which proposes to combine Fourier analysis with the Transformer-based method.
- **ETSformer** [38], which exploits the principle of exponential smoothing and performs a layer-wise level, growth, and seasonal decomposition.
- **PatchTST** [25], which divides the time-series data into sub-level patches to generate meaningful input features inspired by a patch-based Transformer on images in [9].

We also consider several non-Transformer models, including

- **DLinear** [43], a simple linear model that only adopts a one-layer MLP model on the temporal dimension.

Table 2: Experimental results for univariate time series forecasting. Bold (Underlined) values indicate the best (second-best) performance. The input length is 96 for each dataset, and the prediction lengths for ETTh1, ETTh2, ETTm1, and ETTm2 datasets are {96, 192, 336, 720}, while {24, 36, 48, 60} for the ILI dataset. (avg for the averaged results on four different prediction lengths)

| Methods | | DRFormer | | Koopaa | | PatchTST | | TimesNet | | FEDformer | | ETSformer | | Autoformer | | Informer | | Reformer | |
|---------|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|--------------|-------|----------|-------|--------------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | 0.056 | <u>0.183</u> | 0.058 | 0.184 | <u>0.057</u> | 0.179 | 0.058 | 0.185 | 0.079 | 0.215 | 0.063 | 0.194 | 0.071 | 0.206 | 0.193 | 0.377 | 0.532 | 0.569 |
| | 192 | 0.072 | 0.206 | <u>0.073</u> | <u>0.207</u> | 0.075 | 0.209 | 0.077 | 0.213 | 0.104 | 0.245 | 0.085 | 0.227 | 0.114 | 0.262 | 0.217 | 0.395 | 0.568 | 0.575 |
| | 336 | 0.084 | 0.230 | <u>0.089</u> | <u>0.232</u> | 0.089 | 0.233 | <u>0.088</u> | <u>0.232</u> | 0.119 | 0.270 | 0.100 | 0.251 | 0.107 | 0.258 | 0.202 | 0.381 | 0.635 | 0.589 |
| | 720 | 0.088 | 0.233 | <u>0.094</u> | <u>0.241</u> | 0.097 | 0.245 | <u>0.095</u> | 0.242 | 0.142 | 0.299 | 0.100 | 0.250 | 0.126 | 0.283 | 0.183 | 0.355 | 0.762 | 0.666 |
| | avg | 0.075 | 0.213 | <u>0.079</u> | <u>0.216</u> | 0.080 | 0.217 | <u>0.080</u> | 0.218 | 0.111 | 0.257 | 0.087 | 0.231 | 0.105 | 0.252 | 0.199 | 0.377 | 0.624 | 0.600 |
| ETTh2 | 96 | <u>0.133</u> | <u>0.279</u> | 0.139 | 0.290 | 0.137 | 0.285 | 0.141 | 0.293 | 0.128 | 0.271 | 0.157 | 0.310 | 0.153 | 0.306 | 0.213 | 0.373 | 1.411 | 0.838 |
| | 192 | 0.183 | <u>0.336</u> | 0.183 | <u>0.336</u> | 0.187 | 0.340 | 0.193 | 0.347 | 0.185 | 0.330 | 0.211 | 0.364 | 0.204 | 0.351 | 0.227 | 0.387 | 5.658 | 1.671 |
| | 336 | 0.218 | 0.373 | <u>0.219</u> | <u>0.385</u> | 0.223 | <u>0.378</u> | 0.233 | 0.386 | 0.231 | <u>0.378</u> | 0.258 | 0.409 | 0.246 | 0.389 | 0.242 | 0.401 | 4.777 | 1.582 |
| | 720 | 0.227 | 0.383 | <u>0.238</u> | <u>0.393</u> | 0.244 | 0.397 | 0.255 | 0.405 | 0.278 | 0.420 | 0.289 | 0.434 | 0.268 | 0.409 | 0.291 | 0.439 | 2.042 | 1.039 |
| | avg | 0.190 | 0.343 | <u>0.195</u> | <u>0.351</u> | 0.198 | 0.350 | 0.206 | 0.358 | 0.206 | <u>0.350</u> | 0.229 | 0.379 | 0.218 | 0.364 | 0.243 | 0.400 | 3.472 | 1.283 |
| ETTm1 | 96 | 0.028 | 0.125 | <u>0.030</u> | <u>0.130</u> | <u>0.031</u> | <u>0.127</u> | <u>0.030</u> | 0.129 | 0.033 | 0.140 | 0.032 | 0.135 | 0.056 | 0.183 | 0.109 | 0.277 | 0.296 | 0.355 |
| | 192 | 0.043 | 0.159 | 0.045 | <u>0.161</u> | <u>0.044</u> | 0.165 | 0.047 | 0.163 | 0.058 | 0.186 | 0.046 | 0.167 | 0.081 | 0.216 | 0.151 | 0.310 | 0.429 | 0.474 |
| | 336 | 0.057 | 0.184 | <u>0.060</u> | <u>0.188</u> | <u>0.060</u> | 0.187 | 0.063 | 0.193 | 0.084 | 0.231 | <u>0.060</u> | 0.188 | 0.076 | 0.218 | 0.427 | 0.591 | 0.585 | 0.583 |
| | 720 | 0.081 | <u>0.219</u> | 0.081 | 0.218 | <u>0.083</u> | 0.221 | 0.085 | 0.226 | 0.102 | 0.250 | <u>0.087</u> | 0.226 | 0.110 | 0.267 | 0.438 | 0.586 | 0.782 | 0.730 |
| | avg | 0.052 | 0.172 | <u>0.054</u> | <u>0.174</u> | 0.055 | 0.175 | 0.056 | 0.178 | 0.069 | 0.202 | 0.056 | 0.179 | 0.081 | 0.221 | 0.281 | 0.441 | 0.523 | 0.536 |
| ETTh2 | 96 | 0.064 | 0.182 | 0.067 | <u>0.186</u> | 0.073 | 0.200 | 0.075 | 0.202 | 0.072 | 0.206 | 0.080 | 0.212 | <u>0.065</u> | 0.189 | 0.088 | 0.225 | 0.076 | 0.214 |
| | 192 | 0.099 | 0.233 | <u>0.101</u> | <u>0.238</u> | 0.105 | 0.243 | 0.109 | 0.250 | 0.102 | 0.245 | 0.150 | 0.302 | 0.118 | 0.256 | 0.132 | 0.283 | 0.132 | 0.290 |
| | 336 | 0.129 | 0.273 | <u>0.134</u> | <u>0.279</u> | 0.136 | 0.281 | 0.142 | 0.290 | <u>0.130</u> | <u>0.279</u> | 0.175 | 0.334 | 0.154 | 0.305 | 0.180 | 0.336 | 0.160 | 0.312 |
| | 720 | 0.180 | <u>0.328</u> | 0.182 | <u>0.332</u> | 0.185 | 0.334 | 0.190 | 0.341 | <u>0.178</u> | 0.325 | 0.224 | 0.379 | 0.182 | 0.335 | 0.300 | 0.435 | 0.168 | 0.335 |
| | avg | 0.118 | 0.254 | 0.121 | <u>0.259</u> | 0.125 | 0.265 | 0.129 | 0.271 | <u>0.121</u> | 0.264 | 0.157 | 0.307 | 0.130 | 0.271 | 0.175 | 0.320 | 0.134 | 0.288 |
| ILI | 24 | <u>0.809</u> | <u>0.657</u> | 0.918 | 0.667 | 0.810 | 0.674 | 0.828 | 0.662 | 0.708 | 0.627 | 1.161 | 0.748 | 0.948 | 0.732 | 5.282 | 2.050 | 3.838 | 1.720 |
| | 36 | <u>0.786</u> | <u>0.684</u> | 0.938 | 0.713 | 0.705 | <u>0.645</u> | 0.820 | 0.698 | 0.584 | 0.617 | 0.759 | 0.688 | <u>0.634</u> | 0.650 | 4.554 | 1.916 | 2.934 | 1.520 |
| | 48 | 0.715 | 0.679 | 0.786 | 0.705 | 0.764 | 0.726 | 0.730 | <u>0.686</u> | <u>0.717</u> | 0.697 | 1.017 | 0.839 | 0.791 | 0.752 | 4.273 | 1.846 | 3.755 | 1.749 |
| | 60 | 0.714 | 0.691 | <u>0.735</u> | <u>0.703</u> | 0.786 | 0.744 | 0.759 | <u>0.713</u> | 0.855 | 0.774 | 1.022 | 0.823 | 0.874 | 0.797 | 5.214 | 2.057 | 4.162 | 1.847 |
| | avg | <u>0.756</u> | 0.678 | 0.844 | 0.697 | 0.766 | 0.697 | 0.784 | 0.690 | 0.716 | <u>0.679</u> | 0.990 | 0.775 | 0.812 | 0.733 | 4.831 | 1.967 | 3.672 | 1.709 |

Table 3: Ablation study of DRFormer on the Traffic dataset. MS stands for multi-scale feature and DT for dynamic tokenizer.

| Model | Transformer | | | | Transformer+RoPE | | | | Transformer+MS+RoPE | | | | Transformer+MS+gRoPE | | | |
|---------|-------------|-------|-------|-------|------------------|-------|-------|-------|---------------------|-------|-------|--------------|----------------------|-------|--------------|--------------|
| Dynamic | w/o DT | | w/ DT | | w/o DT | | w/ DT | | w/o DT | | w/ DT | | w/o DT | | w/ DT | |
| Metric | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| 96 | 0.426 | 0.270 | 0.424 | 0.271 | 0.420 | 0.270 | 0.415 | 0.270 | 0.421 | 0.270 | 0.416 | 0.266 | 0.418 | 0.268 | 0.414 | 0.267 |
| 192 | 0.439 | 0.276 | 0.432 | 0.272 | 0.432 | 0.274 | 0.431 | 0.274 | 0.432 | 0.274 | 0.432 | 0.272 | 0.429 | 0.272 | 0.427 | 0.271 |
| 336 | 0.452 | 0.283 | 0.447 | 0.280 | 0.445 | 0.280 | 0.444 | 0.278 | 0.448 | 0.281 | 0.442 | 0.277 | 0.445 | 0.281 | 0.440 | 0.278 |
| 720 | 0.487 | 0.301 | 0.483 | 0.297 | 0.481 | 0.300 | 0.478 | 0.298 | 0.484 | 0.301 | 0.477 | 0.297 | 0.478 | 0.298 | 0.474 | 0.296 |
| avg | 0.451 | 0.283 | 0.447 | 0.280 | 0.445 | 0.281 | 0.442 | 0.280 | 0.446 | 0.282 | 0.442 | 0.278 | 0.443 | 0.280 | 0.439 | 0.278 |

- **TimesNet** [39], which transforms the 1D time series into 2D space and extract the complex temporal variations from transformed 2D tensors by a parameter-efficient inception block.
- **Koopaa** [23], which tackles non-stationary time series with modern Koopman theory that fundamentally considers the underlying time-variant dynamics.

4.1.3 Implementation details. All the experiments are implemented with PyTorch [27] and conducted on a single NVIDIA RTX 3090 GPU. The hidden dimension size D is set to 128 for ETT datasets while 512 for other datasets. We set the patch length P to 16 and the stride S to 4 for the ECL, Traffic, and ETT datasets while the patch length to 24 and the stride to 2 for ILI dataset. For dynamic linear, we set the number of groups G to 8 and the sparse ratio SR to 0.5.

Assuming we have i iterations for each epoch, we set the update frequency Δt to $\lfloor 30\% \times i \rfloor$. For the multi-scale transformer, we set k for the number of multi-view sequences to 3 and $S_K = \{1, 2, 4\}$.

4.2 Model Comparisons

4.2.1 Multivariate Forecasting Results. The multivariate forecasting results are shown in Table 1, indicating *DRFormer* achieves the state-of-the-art results on 6 datasets. Specifically, as compared to the best baselines, *DRFormer* reduces the MSE by **8.4%** ($0.476 \rightarrow 0.439$) on the Traffic dataset, and overall **2.7%** ($0.380 \rightarrow 0.370$) reduction on four subsets of the ETT dataset, and **9.2%** ($2.139 \rightarrow 1.959$) reduction on the ILI dataset. On average, *DRFormer* achieves a **6.20%** ($0.617 \rightarrow 0.581$) reduction on the MSE metric compared

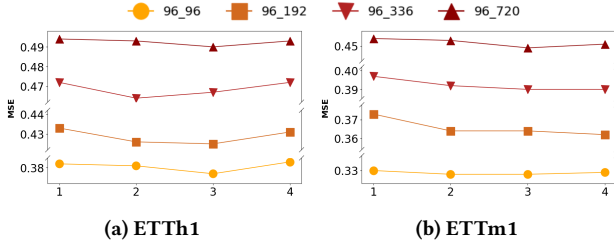


Figure 3: The performance of DRFormer on ETTh1 and ETTm1 across varying numbers of multi-scale sequences.

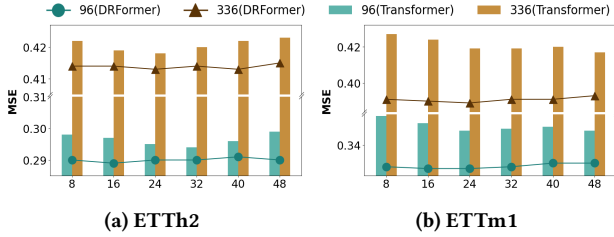


Figure 4: The performance of Transformer(w/o DT) and DRFormer on ETTh2 and ETTm1 datasets under different pre-determined patch lengths.

to the best baselines. Additionally, compared with the best non-Transformer method, i.e., Koopa [23], *DRFormer* improves **18.22%** on the Traffic dataset and an overall improvement of **8.26%** under the MSE metric. Compared with the best Transformer-based method, i.e., PatchTST [25], *DRFormer* improves **16.54%** on the ILI dataset and an overall improvement of **10.50%**.

4.2.2 Univariate Forecasting Results. We show the univariate forecasting results [40] in Table 2. As shown in the table, *DRFormer* achieves state-of-the-art results on four ETT datasets. Additionally, *DRFormer* achieves the second-best results for the MAE metric on the ILI dataset. Specifically, *DRFormer* achieves a **5.33%** reduction on the ETTh1 dataset under the MSE metric, a **3.85%** reduction on the ETTm1 dataset, a **2.63%** reduction on the ETTh2 dataset and a **2.54%** reduction on the ETTm2 dataset respectively.

4.3 Ablation Study

In this section, we delve into a comprehensive analysis of *DRFormer* to showcase the effectiveness of each component of the model.

4.3.1 The effectiveness of dynamic modeling. We employ a dynamic tokenizer technique to capture fine-grained features within the patch size, which brings about diverse receptive fields. We demonstrate the effectiveness of the dynamic tokenizer using different Transformer architectures. As shown in Table 3, the dynamic tokenizer can consistently decrease the prediction error, indicating the robustness of the dynamic tokenizer across various types of Transformer models.

4.3.2 The effectiveness of multi-scale Transformer. The consideration of multi-scale properties is a crucial aspect of time-series forecasting. To tackle this issue, we propose a hierarchical pooling strategy and a group-aware multi-scale Transformer model. By comparing the results of Transformer+RoPE and Transformer+MS+gRoPE, as listed in Table 3, we consider the design of multi-scale features effective when combined with group-aware RoPE. The comparison results between Transformer+RoPE and Transformer+MS+RoPE, with a few cases where performance decreases (MSE w/o DT), are reasonable. Transformer+MS+RoPE faces challenges in effectively aligning spatially close patches across varying scales and capturing intricate dependencies among different representation groups. In summary, the synergistic use of a multi-scale Transformer and gRoPE emerges as a requisite for optimal performance.

4.3.3 The effectiveness of relative position embedding. Incorporating relative position information in input sequences is crucial for Transformer-based models to overcome the weak sensitivity to the ordering of time series [43]. To tackle the issue, we first apply RoPE on the Transformer model. By comparing the results of the Transformer and Transformer+RoPE, as listed in Table 3, we can observe that RoPE improves the forecasting performance.

Furthermore, to overcome the limitations of position awareness of multi-scale representations for the transformer model, we propose a novel group-aware RoPE (gRoPE). By comparing the results of Transformer+MS+RoPE and Transformer+MS+gRoPE in Table 3, we can observe that multi-scale Transformer models with gRoPE perform better than those with RoPE.

4.4 Sensitivity Analysis

In this section, we study the sensitivity of *DRFormer* to its hyper-parameters and masking strategy.

4.4.1 The Influence of Multi-Scale Sequences. Using multi-scale sequences allows us to extract features at multiple scales by transforming original resolution time series into multi-scale representations. To examine the impact of parameter k on forecasting results, we varied k within the range $\{1, 2, 3, 4\}$ and evaluated the performance of *DRFormer* in terms of mean squared error (MSE) on the ETTh1 and ETTm1 datasets. The results, depicted in Figure 3, demonstrate relatively stable and consistent trends across four different prediction horizons $\{96, 192, 336, 720\}$. Notably, increasing the value of k leads to a significant reduction in MSE errors on both datasets, as long as k remains below 3. This improvement can be attributed to the incorporation of features at more diverse scales through an increased number of multi-scale sequences. However, it is important to note that the length of the resized sequence decreases rapidly with larger kernel sizes, which ultimately limits the potential for further enhancement in forecasting performance.

4.4.2 The Influence of Patch Length. To analyze the impact of patch length on ETTh2 and ETTm1 datasets, we select patch length from $\{8, 16, 24, 32, 40, 48\}$. Results from Figure 4 indicate that *DRFormer* exhibits significant insensitivity to changes in patch length compared to patch-based Transformers without dynamic tokenizer and multi-scale sequences. The accuracy of *DRFormer* on the test set remains consistently high across a wide range of patch-length

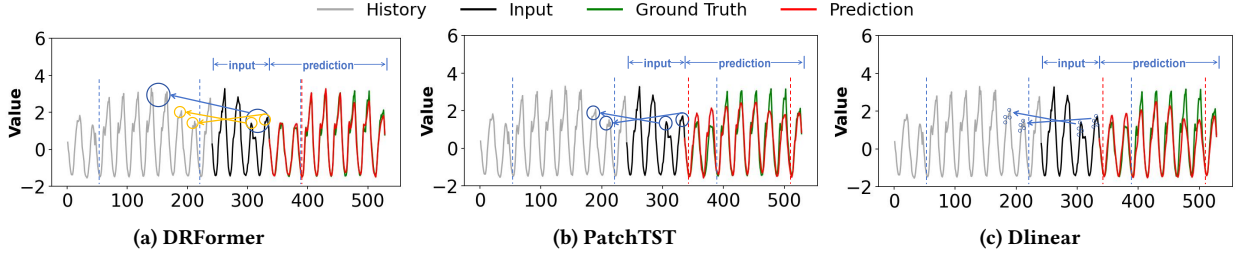


Figure 5: Visualization of forecasting results on the Traffic dataset with $I = 96$ and $O = 192$. The black (grey) lines stand for input sequences (sequences before input). The green (red) lines stand for the ground truth (prediction). The blue (red) dashed lines represent the periodicity of the ground truth (prediction). Different diameters of circles represent different receptive fields.

configurations, highlighting the advantage of capturing a set of receptive fields with a predetermined patch length.

4.4.3 The Influence of Masking Strategy. We explored various masking approaches, including masking out weights based on their magnitudes, both small and large, as well as masking weights according to the product of their magnitudes and gradients [31]. We ultimately chose masking out weights with small magnitudes as it is intuitive and has been experimentally proven to be the most effective as shown in Figure 6. It is widely recognized that the contribution of weights with smaller magnitudes is insignificant or even negligible.

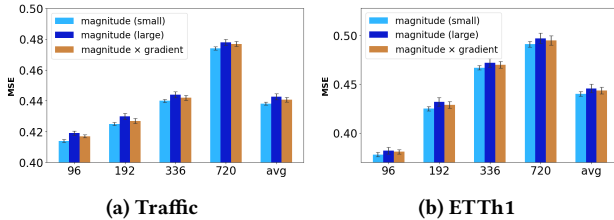


Figure 6: The performance of DRFormer on Traffic and ETTh1 datasets across distinct masking strategies.

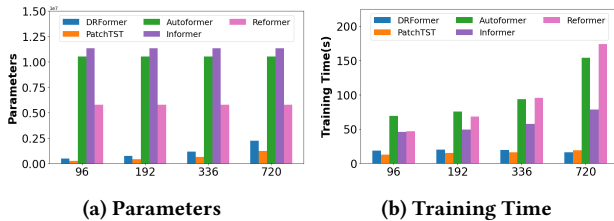


Figure 7: The comparison of parameters and training time between DRFormer and other transformer-based models.

4.5 Model Complexity Analysis

We conducted experiments to assess the complexity of DRFormer, focusing on two key metrics: parameters and training time. To ensure fairness, we maintained the same batch size for all models. As depicted in Figure 7, DRFormer demonstrates significant advantages in both metrics, trailing only PatchTST. This can be attributed to

the implementation of a multi-scale Transformer, which increases the total number of tokens by adding coarse-grained tokens via hierarchical max pooling. However, the additional resource requirements are deemed acceptable. In comparison to models Autoformer, Informer, and Reformer, DRFormer exhibits lower complexity.

4.6 Visualization

We select one test example from the Traffic dataset for case visualization. The ground truth and the predictions from DRFormer and other baselines, i.e., PatchTST, and DLinear, are shown in Figure 5, where DRFormer provides the best forecasting. Specifically, we observe that DRFormer, less affected by low amplitude at the end of the input sequence, relies on long-term trends to align accurately with corresponding segments in the historical sequence. Compared with PatchTST and DLinear, the diverse receptive fields in Figure 5a enable DRFormer to learn multi-scale temporal patterns, improving its ability to predict periodicity and long-term variation without sacrificing compromising the accuracy of details.

5 Conclusion

In this paper, we propose a multi-scale Transformer model coupled with a dynamic tokenizer, named DRFormer, for long-term time series forecasting. DRFormer is a patch-based Transformer with a dynamic tokenizer and multi-resolution representations. Additionally, we present a novel group-aware RoPE method, named gRoPE to enhance intra- and inter-group position awareness among representations with different temporal scales. Extensive experimental results on both multivariate and univariate time series forecasting demonstrate that DRFormer outperforms the previous state-of-the-art approaches. Dynamic tokenizer and multi-scale Transformer can be transferred easily to other patch-based models.

Limitations: DRFormer is designed under a channel-independent setting and it can be further explored to incorporate the correlation between different channels.

6 Acknowledgments

The authors would like to extend their heartfelt thanks to Zanwei Zhou and Jiajun Cui for their insightful feedback and valuable suggestions on paper writing. This work was supported in part by the National Key Research and Development Program of China (2021ZD0111000, 2019YFB2102600) and Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education.

References

- [1] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469* (2023).
- [2] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. 2024. Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *International Conference on Learning Representations*.
- [3] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595* (2023).
- [4] Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. 2023. ContiFormer: Continuous-Time Transformer for Irregular Time Series Modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- [5] Yuqi Chen, Hanyuan Zhang, Weiwei Sun, and Baihua Zheng. 2023. Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 829–842.
- [6] Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. 2022. Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1994–2001. <https://doi.org/10.24963/ijcai.2022/277> Main Track.
- [7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 2978–2988. <https://doi.org/10.18653/v1/P19-1285>
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv abs/2010.11929* (2020). <https://api.semanticscholar.org/CorpusID:225039882>
- [10] Dr M Durairaj and BH Krishna Mohan. 2022. A convolutional neural network based approach to financial time series prediction. *Neural Computing and Applications* 34, 16 (2022), 13319–13337.
- [11] Yuchen Fang, Kan Ren, Caihua Shan, Yifei Shen, You Li, Weinan Zhang, Yong Yu, and Dongsheng Li. 2023. Learning decomposed spatial relations for multivariate time-series modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7530–7538.
- [12] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820* (2023).
- [13] Qitong Guo, Shun Lei, Qing Ye, and Zhiyang Fang. 2021. MRC-LSTM: a hybrid approach of multi-scale residual CNN and LSTM to predict bitcoin price. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [15] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. 2020. HetETA: Heterogeneous information network embedding for estimating time of arrival. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2444–2454.
- [16] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4356–4364.
- [17] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).
- [18] Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking Positional Encoding in Language Pre-training. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=09-528y2Fgf>
- [19] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.
- [20] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).
- [21] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2017. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2017). <https://api.semanticscholar.org/CorpusID:4922476>
- [22] Yan Li, Xinjiang Lu, Haoyi Xiong, Jian Tang, Jiantao Su, Bo Jin, and Dejing Dou. 2023. Towards Long-Term Time-Series Forecasting: Feature, Pattern, and Distribution. *arXiv preprint arXiv:2301.02068* (2023).
- [23] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. 2023. Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors. *arXiv preprint arXiv:2305.18803* (2023).
- [24] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems* 35 (2022), 9881–9893.
- [25] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).
- [26] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
- [29] Mohammad Amin Shabani, Amir H. Abdi, Lili Meng, and Tristan Sylvain. 2023. Scaleformer: Iterative Multi-scale Refining Transformers for Time Series Forecasting. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=sCnllCtjoE>
- [30] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* (2018).
- [31] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*. <https://api.semanticscholar.org/CorpusID:3385018>
- [32] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [33] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yufeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.
- [34] Wensi Tang, Guodong Long, Lu Liu, Tianyi Zhou, Michael Blumenstein, and Jing Jiang. 2020. Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification. *arXiv preprint arXiv:2002.10061* (2020).
- [35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [36] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [38] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. 2022. ETSformer: Exponential Smoothing Transformers for Time-series Forecasting. *ArXiv abs/2202.01381* (2022). <https://api.semanticscholar.org/CorpusID:246485702>
- [39] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186* (2022).
- [40] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Neural Information Processing Systems*. <https://api.semanticscholar.org/CorpusID:235623791>
- [41] Qiao Xiao, Boqian Wu, Yu Zhang, Shiwei Liu, Mykola Pechenizkiy, Elena Mocanu, and Decebal Constantin Mocanu. 2022. Dynamic sparse network for time series classification: Learning what to “see”. *Advances in Neural Information Processing Systems* 35 (2022), 16849–16862.
- [42] Li Xu, Jimmy S Ren, Ce Liu, and Jiaya Jia. 2014. Deep convolutional neural network for image deconvolution. *Advances in neural information processing systems* 27 (2014).
- [43] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.
- [44] Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng. 2021. Traffic flow forecasting with spatial-temporal graph

- diffusion network. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 15008–15015.
- [45] Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. 2023. Multi-resolution Time-Series Transformer for Long-term Forecasting. *arXiv preprint arXiv:2311.04147* (2023).
 - [46] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.
 - [47] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1234–1241.
 - [48] Shuhan Zhong, Sizhe Song, Guanyao Li, Weipeng Zhuo, Yang Liu, and S-H Gary Chan. 2023. A Multi-Scale Decomposition MLP-Mixer for Time Series Analysis. *arXiv preprint arXiv:2310.11959* (2023).
 - [49] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.
 - [50] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*. PMLR, 27268–27286.