FDNET: FOCAL DECOMPOSED NETWORK FOR EFFICIENT, ROBUST AND PRACTICAL TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents FDNet: a ***F***ocal ***D***ecomposed ***Net***work for efficient, robust and practical time series forecasting. We break away from conventional deep time series forecasting formulas which obtain prediction results from universal feature maps of input sequences. In contrary, FDNet neglects universal correlations of input elements and only extracts fine-grained local features from input sequence. We show that: (1) Deep time series forecasting with only fine-grained local feature maps of input sequence is feasible and competitive upon theoretical basis. (2) By abandoning global coarse-grained feature maps, FDNet overcomes distribution shift problem caused by changing local dynamics of time series which is common in real-world applications. (3) FDNet is not dependent on any assumption or priori knowledge of time series except basic auto-regression, which makes it general and practical. Moreover, we propose focal input sequence decomposition method which decomposes input sequence in a focal manner for efficient and robust forecasting when facing LSTI problem. FDNet achieves promising forecasting performances on five benchmark datasets and reduces prediction MSE by 38.4% on average compared with other seven SOTA forecasting baselines.

1 INTRODUCTION

Deep time series forecasting develops rapidly in recent years owing to more pressing demands Qu et al. (2019); Alassafi et al. (2022); Kumar & Susan (2020); Shuvo et al. (2021) of handling complicated non-stationary time series Kim et al. (2022); Woo et al. (2022b). At present, there exist deep time series forecasting networks in diverse formulas, including networks based on RNN Lai et al. (2018); Salinas et al. (2020)/CNN Wang et al. (2021); Liu et al. (2021)/Transformer Li et al. (2019); Liu et al. (2022), networks based on end-to-end Chen et al. (2021); Madhusudhanan et al. (2021); Cirstea et al. (2022a)/self-supervised Yue et al. (2022); Woo et al. (2022a) forecasting format, etc. However, they all obey similar forecasting procedures which can be simply divided into 3 steps as shown in Figure 1: (a) Embedding input sequence into latent space; (b) Feature extraction of input sequence; (c) Project input sequence latent representation into prediction sequence. Within the second step, nearly all methods have mechanisms to extract universal correlation information of different input elements to seek universal/global features of input sequences such as attention mechanism Zhou et al. (2021); Wu et al. (2021); Zhou et al. (2022), dilated convolution Wang et al. (2021); Liu et al. (2021); Yue et al. (2022), etc. We call them ***I***nput ***C***orrelation ***O***riented ***M***echanism(s) and ICOM for short. However, time series forecasting task is intended to pursue ***connections of previous and future sequence*** instead of only concerning ***the correlation information***

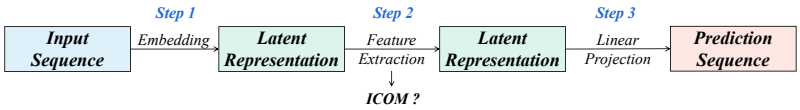

Figure 1: An overview of the similar forecasting procedure of all deep time series forecasting models. It contains three steps and ICOM is employed in the second step if needed.

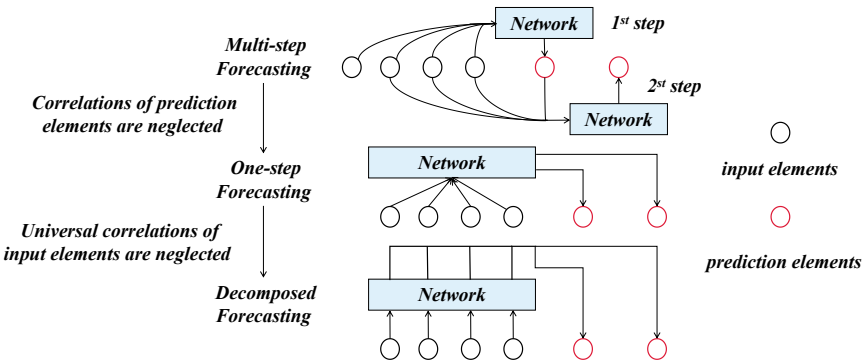

Figure 2: Connections and differences among three forecasting formulas. Correlations of prediction elements are neglected by one-step forecasting formula while universal correlations of input elements are further neglected by our proposed decomposed forecasting formula.

or universal features of previous sequences. So here comes the question: *Does ICOM necessary for time series forecasting?* We analyze this question from three perspectives in the following Section and show that network without ICOM is still capable of doing time series forecasting and even can do better. Therefore, we propose FDNet, a ***F***ocal ***D***ecomposed time series forecasting network. FDNet uses *decomposed forecasting formula* and its differences with existing multi-step Liu et al. (2021); Zhou et al. (2021); Woo et al. (2022b) and one-step Lai et al. (2018); Salinas et al. (2020); Wang et al. (2021) forecasting formulas are illustrated in Figure 2. Built upon one-step forecasting formula where forecasting processes of prediction elements are decomposed, decomposed forecasting formula further decomposes feature extraction processes of input elements. Hence, FDNet is composed of basic MLPs to extract local fine-grained feature maps of input sequence and canonical convolutions to stabilize feature extraction processes when handling outliers of input sequences.

Apart from the necessity of ICOM, currently there also exists another problem which is often ignored, i.e., the Long Sequence Time series Input (LSTI) problem Stoller et al. (2020); Aicher et al. (2019); Cao & Xu (2020). Though it is believed that networks which are able to extract long-term dependencies, e.g., Time Series Forecasting Transformers (TSFTs) Zhou et al. (2022); Woo et al. (2022b); Cirstea et al. (2022a), have already gotten rid of LSTI problem, Shen et al. (2022) points out that even TSFTs will suffer performance drop if excessively prolonging input sequences over a certain borderline as the problem of overfitting will overwhelm benefits of obtaining long-term dependency. Obviously, a qualified forecasting network which can capture potential long-term dependency shall at least not suffer performance drop if prolonging input sequence. Moreover, it is unacceptably time-consuming and expensive if input sequences are too long for most of forecasting networks, esp. TSFTs Li et al. (2019); Zhou et al. (2021); Madhusudhanan et al. (2021). Therefore, a novel input sequence decomposition strategy which not only can deal with LSTI problem but also can limit parameter explosion with the prolonging of input sequence is needed. Motivated from the discovery of Shen et al. (2022) that later input sequence elements are more related to prediction sequences and Focal Transformer Yang et al. (2021), we propose *focal input sequence decomposition method* to help networks deal with LSTI problem. Focal input sequence decomposition divides input sequence into several consecutive sub-sequences in a focal manner according to their temporal distances with prediction elements. Closer a sub-sequence is to prediction elements, shorter it is and more feature extraction layers it has. As a result, connections of input and prediction elements will become weaker and shallower as their temporal distances get farther. Moreover, with the prolonging of input sequence, extra parameters of extra input sequence will also not suffer parameter explosion in that they own fewer feature extraction layers.

Our main contributions are summarized as below:

1. We propose a novel decomposed forecasting formula. Built upon one-forward forecasting formula, it decomposes both forecasting processes of prediction elements and feature extraction processes of input elements.

2. We propose FDNet which uses decomposed forecasting formula. It is only composed of basic MLP and CNN, thus its architecture is very simple. However, FDNet is accurate, efficient and robust in time series forecasting.
3. We propose focal input sequence decomposition method to deal with long-standing LSTI problem. It gives forecasting networks capability of handling extremely long input sequences without suffering overfitting problem, performance drop and parameter explosion.
4. Extensive experiments over 5 benchmark datasets show that FDNet outperforms SOTA forecasting methods by 37.1%/39.6% for multivariate/univariate forecasting on average.
5. Ablation study of focal input sequence decomposition method demonstrates that it is competitive in dealing with LSTI problems. Moreover, it is general enough to couple with not only decomposed forecasting formula but also other forecasting formulas owning ICOM.

2 NECESSITY ANALYSIS OF ICOM

Above all, we discuss whether a time series forecasting model is still established if removing its ICOM from three perspectives to demonstrate the rationality of decomposed forecasting formula.

Time Series Forecasting Definition Time series forecasting is defined as the task of predicting future time series through current and previous time series. Given input sequence $\{z_{i,1:t_0}\}_{i=1}^{N}$, the task is to obtain the prediction sequence $\{z_{i,t_0+1:T}\}_{i=1}^{N}$. N is the number of variates; t_0 denotes the length of input sequence; $(T - t_0)$ refers to the length of prediction sequence. It can be observed that a time series forecasting model without ICOM is still able to do the forecasting task in that only the projection of input sequence to prediction sequence, i.e., step 3 in Figure 1, is prerequisite. In other words, ICOM is only a feature extraction technique of input sequence instead of a necessary component of a forecasting network though it is widely used.

Network Expression Skills We compare expression skills of forecasting networks with/without ICOM to check effects of ICOM. As a downstream task of time series analysis, time series forecasting focuses on finding the correlation of input and prediction sequences, as mentioned in the former perspective. As a result, expression skill here denotes the expression skill of prediction sequences by input sequences. Expression skills of networks with/without ICOM have different dominant domains in time series forecasting which are determined by whether universal feature map of input sequence exists or is beneficial for forecasting. However, the existence of universal feature map relies on the inductive bias or manual assumption of time series properties, esp. those networks assume the season-trend decomposition of time series Wu et al. (2021); Zhou et al. (2022); Woo et al. (2022b). Networks with ICOM will leverage from these manual assumptions if time series dealt with really have supposed properties, otherwise they will suffer from severe forecasting performance turbulence and over-fitting problem. On contrary, networks without ICOM do not leverage from any assumptions of time series properties except basic auto-regression, making it less specific but more general and robust, esp. when dealing with real-world time series shown as below.

Real-world Time Series Finally, we show that networks without ICOM are more practical in dealing with real-world time series. Recent researches Kim et al. (2022); Woo et al. (2022b) have discovered that non-stationary time series, which most of real-world time series are, have different statistical properties or dynamics for local windows spanning different time stamps. We also verify this statement here by Kolmogorov-Smirnov Test Kolmogorov-Smirnov et al. (1933); Smirnov (1939) on target variates of five real-world benchmark datasets. More details of this experiment are shown in Appendix C. We randomly select 1000 sub-sequences of length 96 for each dataset and separately calculate Kolmogorov-Smirnov statistics, i.e., P-values, of the first selected sub-sequence and the rest. Results are shown in Table 1. Using a 0.05 P-value as margin statistics, it could be observed that all five datasets have extremely different local dynamics as reject rates are all very high and standard deviations of P-values are also very big compared with the margin value 0.05. This means that universal representations for all local windows are formidable or even impossible to extract for real-world time series. Therefore, networks with ICOM are easier to get stuck at local optimum, thus their prediction performances are easily affected by random weight initialization. When statistical properties or dynamics they extract are suitable for most of local windows, their

Table 1: Results of KS test

Dataset	$ETTh_1$			$ETTm_2$			ECL			Traffic			weather		
Metrics	Reject rate	Mean	Std	Reject rate	Mean	Std	Reject rate	Mean	Std	Reject rate	Mean	Std	Reject rate	Mean	Std
Values	98.2%	0.012	0.103	98.4%	0.009	0.089	66.4%	0.108	0.221	92.2%	0.031	0.118	86.0%	0.045	0.164

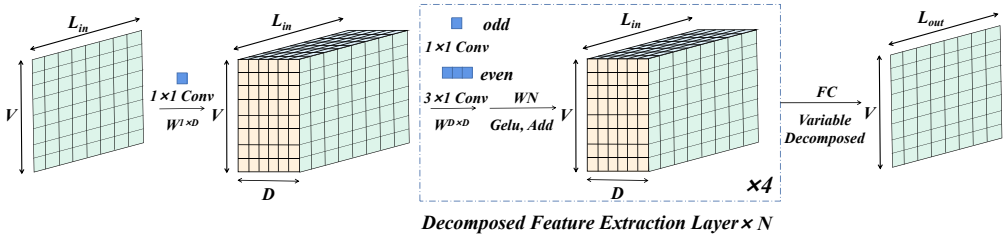

Figure 3: An overview of the architecture of FDNet. It decomposes feature extraction processes of different input elements and different variates. Its main components are N decomposed feature extractor layers (blue trapezoid), each containing four 2D convolutional layers. Weight Normalization Salimans & Kingma (2016), Gelu activation Hendrycks & Gimpel (2016) and res-connection He et al. (2016) are combined with each convolutional layer. L_{in}: the length of input sequence; L_{out}: the length of prediction sequence; V: the number of variables; D: the dimension of embedding.

forecasting performances will be better in general, otherwise their forecasting performances will be worse and unstable. Though some advanced methods have been proposed to alleviate this problem Kim et al. (2022); Shen et al. (2022); Cirstea et al. (2022a), they cannot completely solve this problem in that they do not change the forecasting formula essentially. However, networks without ICOM do not have such problems as they abandon the process of extracting global representations of input sequence, which solves this problem from the source.

From above three perspectives of analysis, it can be inferred that networks without ICOM are feasible, general, robust and practical so that decomposed forecasting formula is rational.

3 Model Architecture

FDNet without ICOM The architecture of FDNet with decomposed forecasting formula is mainly composed of MLPs. An overview of it is shown in Figure 3. The whole network has $(N+2)$ layers where the first layer is the embedding layer, the last layer is the projection layer and the rest are N decomposed feature extraction layers. Each feature extraction layer contains four convolutional layers. Detailed components of decomposed feature extraction layer are shown in Figure 4. Odd layers are 1×1 convolutional layers which has the same function with Perceptron. We use 2D convolution, where two dimensions respectively correspond to temporal dimension and variate dimension, instead of commonly used 1D convolution. The kernel size of the variate dimension will always be one to make sure that element values of different variates will not influence each other. This replacement is motivated by variable-specific forecasting methods Cirstea et al. (2022a; 2021; 2022c); Bai et al. (2020) which treat sequences from different variables as different instances. They point out that sequences of different variates will have different properties with each other in reality. However, employing different projection matrices for different variables is very expensive for forecasting conditions with hundreds of variates. FDNet does not mix values of different variates but gives them the same weight matrices through specific 2D convolution. This is a balance of variable-specific methods which completely splits different vari-

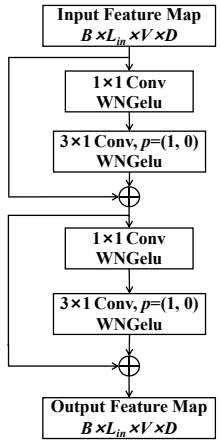

Figure 4: The architecture of decomposed feature extraction layer. B: batch size; p: padding.

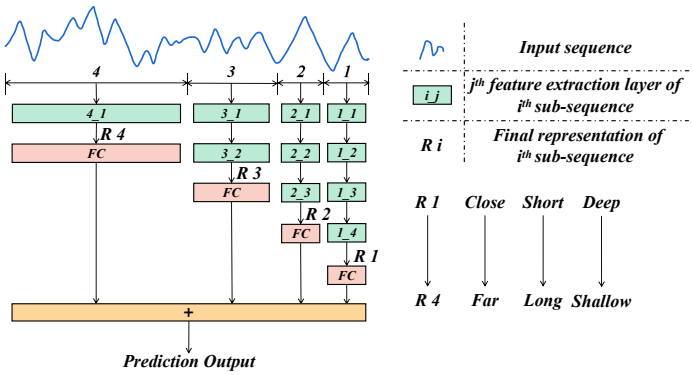

Figure 5: The architecture of focal input sequence decomposition. Final representations of different sub-sequences are from temporally close to far; short to long; deep to shallow.

ates and the opposite variable-agnostic methods Wu et al. (2021); Zhou et al. (2021); Kitaev et al. (2020) which completely mixes different variates during forecasting. Even layers are 3×1 convolutional layers which is a little contradictory to the concept of decomposed forecasting formula. However, pure element-wise feature extraction of input elements will make model susceptible to outliers Ziegler et al. (2019); Wibawa et al. (2022). Convolutional layers used here only have stride of 1 to enhance the locality and smooth anomalies. It is a tradeoff design between decomposed forecasting formula and realistic forecasting situations with considerable outliers. Besides, several convolutions will only make receptive fields of input elements contain few adjacent elements, which still ensures the local fine-grained feature extraction of input sequences. Consequently, feature extractions of elements from different variables and time stamps are all decomposed in FDNet. Finally, a FC layer is used to obtain separate prediction results for separate variates.

Focal Input Sequence Decomposition How focal input sequence decomposition works with forecasting networks is depicted in Figure 5. The latest sub-sequence of input sequence has the shortest length but has the most feature extraction layers. When it goes to farther regions, decomposed subsequence gets longer and feature map extracted from it gets shallower. Proportions of input sequence comprised by different sub-sequences approximately form a geometric series with common ratio of 0.5. For instance, if input sequence is consecutively splitted into 4 parts by focal decompostion method like Figure 5, then proportions will be $\{1/2, 1/4, 1/8, 1/8\}$. The latest sub-sequence takes the proportion of 1/8 instead of 1/16 in order to make the sum of proportions be 1. Furthermore, feature extractions of different sub-sequences and projections of them to output prediction sequence are all mutually independent. As a result, focal input sequence decomposition method effectively allocates complexity levels to different input sub-sequence independently according to their temporal distances with prediction sequence. Networks with focal input sequence decomposition method is now able to deal with LSTI problem without gaining considerable parameters and suffering performance drop with prolonging the input sequence length. When combining FDNet with focal input sequence decomposition, decomposed feature extraction layers in Figure 3 will take formats in Figure 5.

FDNet vs other TCNs As components of FDNet contain convolutional layers, it is necessary to present differences of FDNets with other Temporal Convolution Networks (TCNs) Wang et al. (2021); Liu et al. (2021); Yue et al. (2022). The biggest one is that their usages of convolutions are different. Previous works focus on modifying convolutional layers or combining them with different other techniques such as representation learning Yue et al. (2022); Woo et al. (2022a), binary tree Liu et al. (2021), etc, to obtain universal/global feature maps of input sequence. The most well-known and widely used modification is causal dilated convolution Wang et al. (2021); Yue et al. (2022); Woo et al. (2022a). In contrary, FDNet only uses the basic function of convolution, i.e., enhancing locality of neural networks. Determined by its decomposed forecasting formula, FDNet does not extract global/universal feature maps which is another big difference of it with other TCNs.

Focal with ICOMs Though focal input sequence decomposition method absolutely tallies with decomposed forecasting formula, it does not mean that it cannot be applied to other forecasting

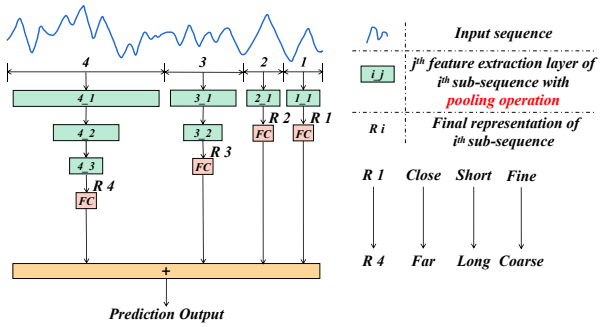

Figure 6: An application of focal input sequence decomposition method combined with ICOMs. Its feature extraction layers additionally contain pooling operations. Final representations of different sub-sequences are from temporally close to far; short to long; fine to coarse.

formulas owning ICOM. When combining forecasting formulas with ICOMs and focal input sequence decomposition, ICOMs can be applied within each sub-sequence to extract universal features of sub-sequences. We provide a possible application which is shown in Figure 6 and call this ***F**ocal **U**niversal **Net**work* (FUNet). Slightly different from the application of focal input sequence decomposition in decomposed forecasting formula, farther sub-sequences have more feature extraction layers. However, the core idea is invariant in that feature extraction layers here have additional pooling operations[1] so that final representations of farther sub-sequences are more global and coarse-grained. Pooling operations also reduce the dimension of sequence length, which makes parameter increasing rate controllable with the prolonging of input sequence. As a result, focal input sequence decomposition can also efficiently extract hierarchical features maps of input sequence when combined with other forecasting formulas with ICOM. Concrete architecture of feature extraction layer in Figure 6 is shown in Figure 7. We only use the canonical attention mechanism and maxpooling operation to perform experiments in later sections in order to emphasize the strength of focal input sequence decomposition.

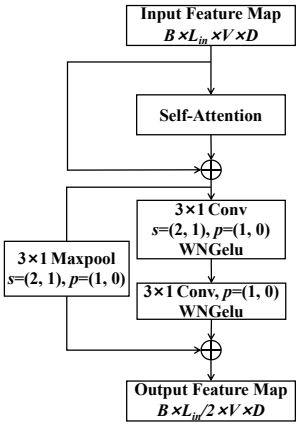

Figure 7: Architecture of feature extraction layer in Figure 6. Self-Attention is chosen as ICOM in this application and s refers to stride.

4 EXPERIMENT

Datasets and Baselines Extensive experiments are performed under five real-world datasets $\{\text{ETTh}_1, \text{ETTm}_2, \text{ECL}, \text{Traffic}, \text{weather}\}$ [2]. More details of them are shown in Appendix D.1.

Seven state-of-the-art time series forecasting models {FEDformer Zhou et al. (2022), Pyraformer Liu et al. (2022), ETSformer Woo et al. (2022b), Triformer Cirstea et al. (2022a), SCINet Liu et al. (2021), TS2Vec Yue et al. (2022), CoST Woo et al. (2022a)} are chosen as baselines.

Main Results We perform multivariate/univariate forecasting experiments to compare the forecasting capability of FDNet with those of mentioned baselines under five datasets. The prediction length group is {96, 192, 336, 720}, which follows Zhou et al. (2022); Woo et al. (2022b). Results of seven baselines are borrowed from their papers if exist, other experiments are performed following their default settings. Results of FEDformer are average results of its two versions {FEDformer-f, FEDformer-w}. Input lengths of FDNet are all set to 672 and input sequences are all divided into 5 parts by focal decomposition method. All experiments are repeated 10 times and means of metrics are used. Other settings are introduced in Appendix D.2. Multivariate/Univariate forecasting results

[1] Additional pooling operations contain convolutional layers with stride=2 and maxpooling.

[2] These five datasets were acquired at: `https://drive.google.com/drive/folders/1ZOYpTUa82_jCcxIdTmyr0LXQfvaM9vIy?usp=sharing`

Table 2: Results of multivariate forecasting

Methods		FDNet		FEDformer		Pyraformer		ETSformer		Triformer		SCINet		TS2Vec		CoST	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh_1$	96	***0.365***	***0.397***	0.419	0.459	0.662	0.611	0.511	0.487	0.419	0.446	0.531	0.503	0.670	0.588	0.499	0.498
	192	***0.400***	***0.419***	0.461	0.483	0.791	0.683	0.561	0.513	0.484	0.486	0.535	0.513	0.781	0.651	0.652	0.583
	336	***0.427***	***0.438***	0.530	0.523	0.902	0.734	0.599	0.529	0.513	0.489	0.584	0.560	0.911	0.718	0.804	0.672
	720	***0.457***	***0.482***	0.686	0.606	0.974	0.780	0.588	0.541	0.711	0.638	0.685	0.634	1.059	0.794	0.973	0.772
$ETTm_2$	96	***0.168***	***0.260***	0.204	0.288	0.378	0.456	0.189	0.280	0.240	0.326	0.312	0.415	0.360	0.426	0.289	0.399
	192	***0.237***	***0.316***	0.293	0.346	1.192	0.870	0.253	0.319	0.387	0.449	0.573	0.591	0.534	0.537	0.509	0.536
	336	***0.310***	0.369	0.342	0.377	1.176	1.033	0.314	***0.357***	0.545	0.532	1.870	1.078	0.833	0.694	0.800	0.686
	720	0.417	0.437	0.427	0.424	6.720	2.077	***0.414***	***0.413***	1.928	0.924	3.462	1.753	1.906	1.054	1.657	1.000
ECL	96	***0.142***	***0.242***	0.188	0.303	0.418	0.460	0.187	0.304	—	—	0.210	0.333	0.336	0.412	0.163	0.267
	192	***0.155***	***0.254***	0.198	0.312	0.408	0.454	0.199	0.315	—	—	0.234	0.345	0.337	0.415	0.172	0.275
	336	***0.170***	***0.271***	0.213	0.321	0.410	0.457	0.212	0.329	—	—	0.227	0.340	0.350	0.426	0.196	0.296
	720	***0.204***	***0.301***	0.239	0.349	0.407	0.456	0.233	0.345	—	—	0.269	0.373	0.375	0.438	0.232	0.327
Traffic	96	***0.402***	***0.276***	0.575	0.358	0.938	0.490	0.607	0.392	—	—	0.581	0.423	0.941	0.550	0.453	0.330
	192	***0.412***	***0.280***	0.583	0.360	0.939	0.488	0.621	0.399	—	—	0.595	0.429	—	—	0.459	0.327
	336	***0.424***	***0.286***	0.596	0.353	0.948	0.488	0.622	0.396	—	—	—	—	—	—	—	—
	720	***0.466***	***0.306***	0.611	0.375	—	—	0.632	0.396	—	—	—	—	—	—	—	—
weather	96	***0.159***	***0.211***	0.222	0.300	0.212	0.296	0.197	0.281	0.174	0.242	0.179	0.255	0.906	0.627	0.355	0.410
	192	***0.200***	***0.248***	0.286	0.350	0.246	0.321	0.237	0.312	0.219	0.290	0.230	0.299	0.980	0.678	0.501	0.507
	336	***0.247***	***0.286***	0.360	0.398	0.287	0.349	0.298	0.353	0.272	0.323	0.280	0.331	1.252	0.794	0.654	0.598
	720	***0.309***	***0.333***	0.414	0.431	0.358	0.400	0.352	0.388	0.357	0.378	0.358	0.392	1.704	0.969	0.884	0.717

Table 3: Results of univariate forecasting

Methods		FDNet		FEDformer		Pyraformer		ETSformer		Triformer		SCINet		TS2Vec		CoST	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh_1$	96	0.067	0.200	0.115	0.266	0.143	0.309	***0.060***	***0.190***	0.153	0.324	0.119	0.269	0.098	0.241	0.080	0.214
	192	0.084	0.226	0.137	0.292	0.159	0.322	***0.081***	***0.221***	0.177	0.347	0.129	0.280	0.153	0.302	0.104	0.247
	336	0.099	0.248	0.142	0.295	0.196	0.372	***0.098***	***0.248***	0.169	0.336	0.160	0.322	0.169	0.326	0.121	0.268
	720	0.167	0.331	0.144	0.302	0.230	0.410	***0.119***	***0.282***	0.271	0.453	0.243	0.414	0.164	0.327	0.302	0.485
$ETTm_2$	96	0.069	***0.196***	***0.068***	0.198	0.461	0.527	0.080	0.213	0.083	0.221	0.076	0.210	0.088	0.224	0.076	0.203
	192	***0.099***	***0.241***	0.106	0.249	0.781	0.683	0.110	0.252	0.124	0.271	0.102	0.248	0.122	0.271	0.112	0.254
	336	***0.129***	***0.277***	0.139	0.290	1.372	0.913	0.136	0.283	0.157	0.310	0.129	0.280	0.158	0.314	0.145	0.295
	720	***0.173***	***0.325***	0.199	0.347	5.780	1.878	0.185	0.333	0.269	0.408	0.176	0.328	0.200	0.357	0.216	0.348
ECL	96	***0.203***	***0.313***	0.258	0.374	0.347	0.432	0.726	0.656	0.358	0.424	0.312	0.411	0.315	0.419	0.208	0.329
	192	0.236	***0.337***	0.299	0.398	0.436	0.493	0.667	0.625	0.360	0.433	0.314	0.416	0.333	0.430	***0.233***	0.348
	336	***0.267***	***0.362***	0.354	0.438	0.493	0.526	0.770	0.677	0.399	0.456	0.332	0.427	0.347	0.440	0.289	0.375
	720	***0.305***	***0.409***	0.435	0.493	0.614	0.605	0.766	0.674	0.446	0.499	0.364	0.451	0.350	0.447	0.327	0.419
Traffic	96	***0.134***	***0.222***	0.189	0.288	0.501	0.512	0.243	0.355	0.285	0.367	0.217	0.330	0.357	0.431	0.156	0.243
	192	***0.135***	***0.224***	0.189	0.289	0.541	0.532	0.241	0.352	0.284	0.363	0.299	0.397	0.359	0.433	0.158	0.245
	336	***0.135***	***0.227***	0.199	0.295	0.557	0.541	0.240	0.353	0.308	0.383	0.259	0.365	0.368	0.440	0.163	0.252
	720	***0.162***	***0.259***	0.216	0.315	0.596	0.561	0.252	0.362	0.405	0.457	0.278	0.379	0.380	0.447	0.182	0.268
weather	96	***0.002***	***0.030***	0.005	0.054	0.006	0.061	0.008	0.078	0.004	0.050	0.008	0.068	0.015	0.089	0.010	0.077
	192	***0.002***	***0.034***	0.006	0.061	0.007	0.068	0.006	0.065	0.006	0.062	0.008	0.069	0.010	0.074	0.009	0.073
	336	***0.002***	***0.033***	0.006	0.061	0.007	0.070	0.006	0.065	0.006	0.063	0.008	0.070	0.009	0.070	0.009	0.073
	720	***0.003***	***0.037***	0.010	0.075	0.007	0.072	0.005	0.062	0.007	0.070	0.008	0.071	0.011	0.078	0.009	0.075

are shown in Table 2/3. The lowest MSE/MAE are highlighted in bold and italic. '—' denotes that models fail for out-of-memory (24GB) even when batch size = 1.

It could be observed from Table 2/3 that FDNet surpasses other baselines in most of situations. When compared with FEDformer/Pyarformer/ETSformer/Triformer/SCINet/TS2Vec/CoST, FDNet yields 21.7%/51.6%/19.7%/26.1%/34.4%/61.4%/39.4% relative MSE reduction during multivariate forecasting and yields 29.7%/64.5%/33.3%/44.5%/36.3%/42.7%/26.2% relative MSE reduction during univariate forecasting in general, which shows the superior forecasting capability of FDNet.

Universal/Local Feature Extraction Methods Other seven baselines all own ICOMs and intend to extract universal feature maps of input sequence, however their general performances are worse than that of FDNet extracting only local features in Table 2/3. It empirically demonstrates that

Table 4: Results of CoST with different kernel sizes during weather univariate forecasting

Pred	96			192			336			720		
Kernel size	MSE	MAE	Rank	MSE	MAE	Rank	MSE	MAE	Rank	MSE	MAE	Rank
{1, 2, 4, 8, 16, 32, 64, 128}	0.010	0.077	7	0.009	0.073	7	0.009	0.073	7	0.009	0.075	6
{1, 2, 4}	***0.006***	***0.061***	***5***	***0.006***	***0.061***	***3***	***0.006***	***0.062***	***5***	***0.007***	***0.065***	***3***

Table 5: MSE of ablation study on focal decomposition method and usage of convolutions

Focal	Traffic (Univariate)				Traffic (Multivariate)			
Methods	96	192	336	720	96	192	336	720
Initial	0.139	0.138	0.149	0.173	0.418	0.430	0.442	0.473
Focal	***0.133***	***0.135***	***0.135***	***0.162***	***0.402***	***0.412***	***0.424***	***0.466***
Pyramid	0.134	0.135	0.136	0.162	0.418	0.430	0.441	0.476
Patch	0.138	0.138	0.142	0.165	0.406	0.415	0.428	0.470
Conv	weather (Univariate)				weather (Multivariate)			
Methods	96	192	336	720	96	192	336	720
Conv	***1.8e-3***	***2.2e-3***	***2.1e-3***	***2.6e-3***	***0.159***	***0.200***	***0.247***	***0.309***
MLP	2.4e-3	2.5e-3	2.7e-3	3.0e-3	0.160	0.201	0.248	0.310

decomposed forecasting formula is more suitable for real-world forecasting conditions. In other words, local feature extraction method seems more practical than the universal one in time series forecasting networks. Specially, CoST Woo et al. (2022a) extracts global-local feature maps of input sequence through a mixture of experts owning different convolution kernel sizes within $\{1, 2, 4, 8, 16, 32, 64, 128\}$. However, its performance is still worse than that of FDNet, indicating that universal feature maps might bring pernicious effects. To validate this statement, we use a smaller convolution group within $\{1, 2, 4\}$ and remove its dilated convolution architecture to redo univariate forecasting experiment under weather dataset. Results are shown in Table 4. It could be observed that the forecasting accuracy of CoST rises a lot and its forecasting rank rises among all eight methods after our transformation. This result once more demonstrates that extracting local fine-grained features is more practical and useful for time series forecasting task.

Ablation Study We perform ablation study on focal decomposition method and 3×1 convolutional layer used in decomposed feature extraction layer to verify their corresponding functions.

As for focal decomposition method, four ablation variants are tested under Traffic dataset: (1) Initial: FDNet without any decomposition method; (2) Focal: FDNet with focal decomposition method; (3) Pyramid: FDNet with pyramid decomposition method Shen et al. (2022); (4) Patch: FDNet with patch decomposition method Cirstea et al. (2022a). Number of pyramids/patches are set as 3/4 following their default settings. Experiment results are shown in the upper part of Table 5. It could be observed that all three decomposition methods improve forecasting performances of FDNet. However, focal decomposition method behaves better in every single experiment compared with other two decomposition methods, showing that it is more competitive in forecasting tasks.

As for the usage of convolutions, two variants are tested under weather dataset: (1) Conv: Initial FDNet in Figure 3; (2) MLP: FDNet with decomposed feature extractor layers containing only 1×1 convolutional layers. Experiment results are shown in the bottom part of Table 5. It could be observed that forecasting performances of FDNet drop obviously without the usage of convolutions, demonstrating the function of convolutions used in decomposed feature extraction layers.

Focal with ICOM We perform experiments on FUNet shown in Figure 6 to validate that focal input sequence decomposition method with ICOM can compensate the shortcoming of FDNet when global feature maps exist and are beneficial. Experiments are conducted under univariate forecasting of ETTh_1 where FDNet is not so competitive with some selected baselines, e.g., ETSformer and FEDformer. Results are shown in Table 6. Bold and italic MSEs mean that they are lower than those of all other baselines in Table 2. It shows that under those rare occasions where universal

feature maps are needed, focal input sequence decomposed method could transform FDNet to FUNet to outperform those SOTA forecasting methods with elaborately designed ICOMs even only using the most basic ICOMs including canonical attention and convolution. We additionally supply multivariate forecasting results of FUNet in Table 6. MSEs with underline mean that they are higher than MSEs of FDNet but lower than MSEs of other baselines in Table 3. It could be inferred that though FUNet performs better than other baselines, it fails to challenge FDNet in general. This once more illustrates that local feature extraction methods are more practical and general for time series forecasting and the outstanding forecasting capability of FDNet architecture regardless of armed with ICOM or not.

Table 6: MSE of FUNet under ETTh_1

Methods	ETTh_1 (Univariate)				ETTh_1 (Multivariate)			
	96	192	336	720	96	192	336	720
FUNet	***0.059***	***0.071***	***0.088***	***0.112***	0.402	0.458	0.501	0.582
FDNet	0.067	0.084	0.099	0.167	***0.365***	***0.400***	***0.427***	***0.457***

LSTI Problem Handling Capability To verify that FDNet is more accurate, efficient and robust in handling LSTI problems. We conduct experiments under univariate forecasting of ECL. FEDformer-w and ETSformer are chosen as baselines due to their generally outstanding performances in Table 2/3. Input sequence lengths are chosen within $\{96, 672, 1344\}$ and prediction sequence is set to 96. Input sequences of FDNet are respectively divided into $\{4, 5, 6\}$ parts by focal decomposition method. Each sub-experiment is done for 20 times. Means and standard deviations (Stds) of all forecasting MSEs are shown in Table 7, together with their GPU memory occupations (GPU), average training time per epoch (ATPE) and total inference time (TIT). Best results are highlighted in bold and italic. It could be observed from Table 7 that FDNet performs worse than FEDformer-w when it comes to the shortest input sequence condition. However, it performs better than other two baselines in the rest of conditions proving that FDNet is better at handling LSTI problem and more accurate. Specially, forecasting MSEs of FDNet with input sequence lengths 672 or 1344 are only slightly different while MSE of FEDformer-w grows apparently when input sequence length changes from 672 to 1344. Moreover, Stds of FDNet are much lower than those of other two baselines in any forecasting condition, demonstrating the robustness of FDNet. What's more, except the shortest input sequence condition where ATPE of FDNet is bigger than that of ETSformer, all GPU/ATPE/TIT of FDNet is smaller than those of other baselines, illustrating better computation efficiency of FDNet. In conclusion, FDNet is a more accurate, efficient and robust method in dealing with LSTI problems even compared with those state-of-the-art.

Table 7: Results on LSTI problem handling capability under ETTh_1

Input length	Methods	Mean	Std	GPU/MB	ATPE/s	TIT/s
96	FDNet	0.391	***1.0e-3***	***1516***	35.217	***1.986***
	ETSformer	0.726	1.7e-3	2252	***26.317***	3.381
	FEDformer-w	***0.268***	1.8e-3	6445	444.031	19.483
672	FDNet	***0.204***	***1.3e-3***	***1535***	***65.262***	***4.991***
	ETSformer	0.891	2.0e-3	3506	68.338	5.736
	FEDformer-w	0.294	1.9e-3	7885	583.146	25.572
1344	FDNet	***0.209***	***1.5e-3***	***1551***	***84.495***	***6.579***
	ETSformer	0.893	3.9e-3	4954	92.695	9.006
	FEDformer-w	0.323	11e-3	10059	697.448	30.584

5 CONCLUSION

In this paper, we propose FDNet whose core ideas contain decomposed forecasting formula and focal input sequence decomposition method. Built upon decomposed forecasting formula, FDNet is designed to only extract local fine-grained feature maps of input sequence, which is proved to be effective and feasible both theoretically and empirically. In addition, focal input sequence decomposition method solves long-standing LSTI problem by consecutively splitting and processing input sequence in a focal manner. Extensive experiments demonstrate that FDNet is simple but accurate, efficient, robust and practical for real-world time series forecasting.

REFERENCES

Waqar Ahmad, Bibi Misbah Kazmi, and Hazrat Ali. Human activity recognition using multi-head cnn followed by lstm. In *2019 15th international conference on emerging technologies (ICET)*, pp. 1–6. IEEE, 2019.

Christopher Aicher, Nicholas J. Foti, and Emily B. Fox. Adaptively truncating backpropagation through time to control gradient bias. *ArXiv*, abs/1905.07473, 2019.

Madini O. Alassafi, Mutasem Jarrah, and Reem Alotaibi. Time series predicting of covid-19 based on deep learning. *Neurocomputing*, 468:335–344, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.10.035. URL https://www.sciencedirect.com/science/article/pii/S0925231221015150.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

George EP Box and Gwilym M Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. *ArXiv*, abs/2103.07719, 2020.

Yanshuai Cao and Peng Xu. Better long-range dependency by bootstrapping a mutual information regularizer. In *AISTATS*, 2020.

Weidong Chen, Xiaofeng Xing, Xiangmin Xu, Jichen Yang, and Jianxin Pang. Key-sparse transformer with cascaded cross-attention block for multimodal speech emotion recognition. *arXiv preprint arXiv:2106.11532*, 2021.

Razvan-Gabriel Cirstea, Tung Kieu, Chenjuan Guo, Bin Yang, and Sinno Jialin Pan. Enhancenet: Plugin neural networks for enhancing correlated time series forecasting. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1739–1750, 2021. doi: 10.1109/ICDE51399.2021.00153.

Razvan-Gabriel Cirstea, Chenjuan Guo, B. Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting-full version. In *IJCAI*, 2022a.

Razvan-Gabriel Cirstea, B. Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting-full version. *ArXiv*, abs/2203.15737, 2022b.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2900–2913, 2022c. doi: 10.1109/ICDE53745.2022.00262.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. Forecasting with exponential smoothing: The state space approach. 2008.

Mahnoor Khan, Nadeem Javaid, Muhammad Nabeel Iqbal, Muhammad Bilal, Syed Farhan Ali Zaidi, and Rashid Ali Raza. Load prediction based on multivariate time series forecasting for energy consumption and behavioral analytics. In Leonard Barolli, Nadeem Javaid, Makoto Ikeda, and Makoto Takizawa (eds.), *Complex, Intelligent, and Software Intensive Systems*, pp. 305–316, Cham, 2019. Springer International Publishing. ISBN 978-3-319-93659-8.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

AN Kolmogorov-Smirnov, Andrej Nikolajevi_vc Kolmogorov, and M Kolmogorov. Sulla determinazione empírica di uma legge di distribuzione. 1933.

Naresh Kumar and Seba Susan. Covid-19 pandemic prediction using time series forecasting models. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, 2020. doi: 10.1109/ICCCNT49239.2020.9225319.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32:5243–5253, 2019.

Minhao Liu, Ailing Zeng, Zhijian Xu, Qiuxia Lai, and Qiang Xu. Time series is a special sequence: Forecasting with sample convolution and interaction. *arXiv preprint arXiv:2106.09305*, 2021.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0EXmFzUn5I.

Kiran Madhusudhanan, Johannes Burchert, Nghia Duong-Trung, Stefan Born, and Lars Schmidt-Thieme. Yformer: U-net inspired transformer architecture for far horizon time series forecasting. *arXiv preprint arXiv:2110.08255*, 2021.

Licheng Qu, Wei Li, Wenjing Li, Dongfang Ma, and Yinhai Wang. Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with Applications*, 121:304–312, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.12.031. URL https://www.sciencedirect.com/science/article/pii/S0957417418308017.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *CoRR*, abs/1602.07868, 2016. URL http://arxiv.org/abs/1602.07868.

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

Li Shen, Yuning Wei, and Yangzhu Wang. Respecting time series properties makes deep time series forecasting perfect. *ArXiv*, abs/2207.10941, 2022.

Mohammmad Asifur Rahman Shuvo, Muhtadi Zubair, Afsara Tahsin Purnota, Sarowar Hossain, and Muhammad Iqbal Hossain. Traffic forecasting using time-series analysis. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pp. 269–274, 2021. doi: 10.1109/ICICT50816.2021.9358682.

Nikolai V Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2):3–14, 1939.

Daniel Stoller, Mi Tian, Sebastian Ewert, and Simon Dixon. Seq-u-net: A one-dimensional causal u-net for efficient sequence modelling. In *IJCAI*, 2020.

Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. Tcn: Table convolutional network for web table interpretation. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 4020–4032, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450090. URL https://doi.org/10.1145/3442381.3450090.

Aji Prasetya Wibawa, Agung Bella Putra Utama, Hakkun Elmunsyah, Utomo Pujianto, Felix Andika Dwiyanto, and Leonel Hernandez. Time-series analysis with smoothed convolutional neural network. *Journal of Big Data*, 9(1):44, Apr 2022.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=PilZY3omXV2.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *ArXiv*, abs/2202.01381, 2022b.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *ArXiv*, abs/2107.00641, 2021.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yu Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *AAAI*, 2022.

Yun Zhao, Yuqing Wang, Junfeng Liu, Haotian Xia, Zhenni Xu, Qinghang Hong, Zhiyang Zhou, and Linda Petzold. Empirical quantitative analysis of covid-19 forecasting models. *arXiv preprint arXiv:2110.00174*, 2021.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

Thomas Ziegler, Manuel Fritsche, Lorenz Kuhn, and Konstantin Donhauser. Efficient smoothing of dilated convolutions for image segmentation. *CoRR*, abs/1903.07992, 2019. URL http://arxiv.org/abs/1903.07992.

A MEANINGS OF ABBREVIATIONS AND PHRASES

Meanings of mentioned abbreviations and phrases are shown in Table 8.

Table 8: Meanings of abbreviations and phrases

Abbr./Phrase	Meaning
LSTI	***L***ong ***S***equence ***T***ime series ***I***nput
ICOM	***I***nput ***C***orrelation ***O***riented ***M***echanism

B RELATED WORKS

Deep Time Series Forecasting Time series forecasting has wide application in various domains: e.g., electricity prediction Khan et al. (2019), traffic forecasting Shuvo et al. (2021); Cirstea et al. (2022b); Qu et al. (2019), sensor-based recognition Ahmad et al. (2019) and COVID-19 pandemic analysis Kumar & Susan (2020); Zhao et al. (2021). In order to handling long-term and complicated time series, machine learning techniques including neural networks are applied to time series forecasting gradually dominant the research direction. Time series forecasting networks are normally categorized by their network architectures (RNN Salinas et al. (2020); Lai et al. (2018), CNN Wang et al. (2021); Liu et al. (2021), Transformer Wu et al. (2021); Zhou et al. (2021); Woo et al. (2022b), GNN Cao et al. (2020); Wu et al. (2020), etc.) or learning algorithm (supervised end-to-end Liu et al. (2022); Zhou et al. (2022) or self-supervised representation learning Yue et al. (2022); Woo et al. (2022a)). However, we categorize them by three different criteria as shown below to discuss time series forecasting from other perspectives.

Multi-/One-step Forecasting Formula It is believed that time series forecasting starts from multi-step forecasting. Traditional models like ARIMA Box & Jenkins (1968); Box et al. (2015), State Space Estimation Durbin & Koopman (2012), ES Hyndman et al. (2008) are all multi-step forecasting models. Given input sequence $\{z_{i,1:t_0}\}_{i=1}^{N}$, the prediction sequence $\{z_{i,t_0+1:T}\}_{i=1}^{N}$ is obtained by rolling forecasting strategy which predicts the prediction sequence through $(T - t_0)$ steps. However, in virtue of the end-to-end property of neural network, deep time series forecasting network has another option, i.e., one-step forecasting formula Zhou et al. (2021); Liu et al. (2021); Wu et al. (2021). Given input sequence $\{z_{i,1:t_0}\}_{i=1}^{N}$, the prediction sequence $\{z_{i,t_0+1:T}\}_{i=1}^{N}$ is obtained by one-forward strategy which predicts the entire prediction sequence in one step. Comparing these two forecasting formulas, it can be deduced that one-step forecasting formula is more efficient in that it only needs one forward propagation process to obtain the entire prediction sequence. So one-step forecasting formula is also named one-forward forecasting formula. Moreover, inputs of one-step forecasting models are all known and definite while inputs of multi-step forecasting models are partially unknown and inferred from known inputs and models themselves. It is obvious that one-step forecasting models theoretically suffer slighter error accumulation so that one-step forecasting formula is more preferred and common in recent researches Lai et al. (2018); Zhou et al. (2022); Liu et al. (2021). Here we make specific explanations for one-step Time Series Forecasting Transformer (TSFT) Wu et al. (2021); Zhou et al. (2021); Li et al. (2019) as they own masked self-attention mechanisms which seems like inferring prediction elements from themselves and contradicts to our saying that one-step forecasting formula makes inference processes of different prediction elements decomposed and independent from each other. However, notice that inputs of their mask self-attention mechanisms are either zero-padded units Zhou et al. (2021); Li et al. (2019) or decomposed parts of input sequences Zhou et al. (2022); Woo et al. (2022b). Neither of these inputs can reflect essential relationships insider prediction sequences. Some state-of-the-art TSFTs Zhou et al. (2022); Woo et al. (2022b) also notice this phenomenon, thus they do not employ masked self-attention mechanisms within their networks but still achieve promising forecasting performances.

Variable-agnostic/-specific Forecasting Formula Multivariate forecasting starts to be practical after the involvement of machine learning. It has two multivariate oriented forecasting formulas categorized by Cirstea et al. (2022a), i.e., Variable-agnostic and Variable-specific forecasting formula.

Models with variable-agnostic forecasting formula Wu et al. (2021); Zhou et al. (2021); Kitaev et al. (2020) have the same projection matrices for all variables while models with variable-specific forecasting formula Cirstea et al. (2021; 2022c); Bai et al. (2020) have distinct (decomposed) projection matrices for them. In other words, variable-agnostic formula assumes that variables are closely related to each other and have the same statistical properties and variable-specific formula assumes the opposite. RTNet Shen et al. (2022) also points out that variable-agnostic formula only works if we have priori knowledge that variables are closely relevant throughout the whole time span otherwise this formula will make models suffer heavy over-fitting problem. Consequently, experiment results of variable-specific models Shen et al. (2022); Cirstea et al. (2022a) show that variable-specific formula behaves better in most of benchmark datasets/real-world applications.

Compound/Decomposed Forecasting Formula Time series forecasting networks can also be categorized by their input sequence feature extraction methods. Those methods which own ICOM belong to compound forecasting methods. Traditional forecasting methods Box & Jenkins (1968); Box et al. (2015); Durbin & Koopman (2012) and most of deep time series forecasting networks are compound forecasting methods. For instance, classical RNN forecasting networks including DeepAR Salinas et al. (2020), LSTNet Lai et al. (2018), CNN forecasting networks including TCN Wang et al. (2021), TS2Vec Yue et al. (2022) and TSFTs including Informer Zhou et al. (2021), LogTrans Li et al. (2019) are all compound forecasting methods. They may own hierarchical feature extraction mechanisms or pyramid networks which only extract feature maps from partial input sequence, i.e., ICOMs. However, they all have partial networks extracting universal feature maps from the whole input sequence. Recently, some researches point out that decomposing input sequence into season and trend Zhou et al. (2022) or even season, level and growth Woo et al. (2022b) gives networks more competitive forecasting performances. Obviously, these researches propose their methods based on inductive bias that time series can be decomposed like this, which means that their decompositions are beneficial under limited occasions. At least, non-stationary time series which are more common in real-world applications cannot easily be described like this so that these methods are not very practical. We have analyzed above statement in Section 2 of the main text. Moreover, their decompositions do not get rid of universal coarse-grained feature extraction. Though season and trend terms are decomposed from time series, they are still universal representations of specific properties of time series, i.e., these kinds of networks still own ICOM and belong to compound forecasting methods. As a result, a practical decomposition implementation form not limited in season-trend decompositions is needed, which is the motivation of this paper.

C SUPPLEMENTARY OF KS TEST

To examine whether extracting universal representations is possible for time series forecasting models under real-world time series forecasting tasks, we introduce Kolmogorov-Smirnov (KS) Test.

KS test is a nonparametric test to depict agreements between distributions of each two sequences. In essence, KS test describes the probability that they come from the same (but unknown) probability distribution. The Kolmogorov–Smirnov statistic measures a distance D between the empirical distribution function of them and the value of D is calculated as Equation 1.

$$D = \sup_x |F(\{x_i\}_{i=m}^{m+m_1}) - F(\{x_i\}_{i=n}^{n+n_1})| \tag{1}$$

Where $\{x_i\}_{i=m}^{m+m_1}$/$\{x_i\}_{i=n}^{n+n_1}$ refers to the sequence within timespan $[m, m+m_1]$/$[n, n+n_1]$, $F(\cdot)$ is the empirical distribution function and sup denotes the supremum function. For large samples, the null hypothesis is rejected at level α if the calculated value D satisfies Inequality 2 so that P-value is smaller than level α as Equation 3 shows. It means that if P-value is small, the null hypothesis is more likely to be rejected, i.e., these two distributions are more likely to be different.

$$D > \sqrt{-\frac{1}{2}\ln\frac{\alpha}{2}} \times \sqrt{\frac{m_1+n_1}{m_1 \cdot n_1}} \tag{2}$$

$$P - value = 2e^{-2D^2\frac{m_1 \cdot n_1}{m_1+n_1}} < \alpha \tag{3}$$

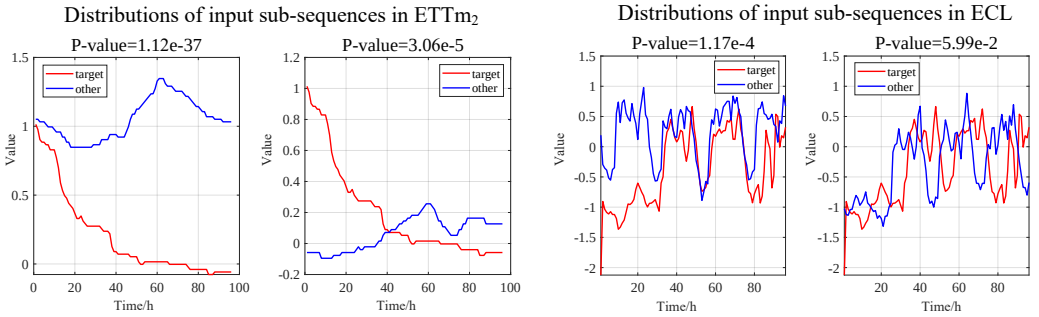

Figure 8: Distributions of two instances when experimenting with ETTm_2/ECL. Distributions of target and these two instances are quite different esp. in ETTm_2.

It can be deduced that if there exist universal properties of input sequences, statistical properties or dynamics of local input sequences will be similar to each other so that their P-values will be bigger compared with the margin P-value.

We perform KS test under five real-world time series benchmark datasets in Section 2 in the main text. We randomly select 1000 sub-sequences of length 96 for each dataset and separately calculate Kolmogorov-Smirnov statistics, i.e., P-values, of the first selected sub-sequence and the rest. Therefore, m_1, n_1 in Equation 1, 2 are all 96 and we take 0.05 as the margin P-value. Results in Table 1 of the main text has shown that universal representations for all local windows are formidable or even impossible to extract. To further examine this, we visualize distributions of two instances when experimenting with ETTm_2/ECL, which own highest/lowest reject rate, respectively as shown in Figure 8. Curves in Figure 8 clearly illustrate that different sub-sequences of both two datasets have different statistical properties or dynamics, which is identical to our aforementioned discovery that extracting universal representations is not necessary for time series forecasting models. Notice that even though the second instance of ECL owns P-value over the margin 0.05, distributions of these two curves have distinct differences.

D Supplementary Experiment

D.1 Brief Introductions of Datasets

ETT (Electricity Transformer Temperature) dataset, which consists of two versions subsets: 1-hour-level datasets $\{\text{ETTh}_1, \text{ETTh}_2\}$ and 15-min-level datasets $\{\text{ETTm}_1, \text{ETTm}_2\}$, is composed of 2-years data of two separated electric stations in China. 'OT' (oil temperature) is the target value. For averting unnecessary experiments, we perform experiments on ETTh_1, a 1-hour-level subset, and ETTm_2, one 15-min-level subset, among these four subsets. The train/val/test is 12/4/4 months.

ECL (Electricity Consuming Load) dataset contains the electricity consumption (Kwh) of 321 clients lasting for almost 2 years. It is converted into 2 years by informer Zhou et al. (2021). 'MT_321' is set as the target value following the settings of FEDformer Zhou et al. (2022), ETSformer Woo et al. (2022b), etc. The train/val/test is 70%/10%/20%.

Traffic dataset consists of road occupation rates in San Francisco Bay area freeways lasting for two years. It is collected hourly. The target is 'Sensor_861' and the train/val/test is 70%/10%/20%.

weather dataset is a 10-min-level dataset which describes 21 meteorological indicators in Germany during 2020. The target variate name is set to 'OT' following FEDformer,ETSformer Zhou et al. (2022); Woo et al. (2022b). The train/val/test is 70%/10%/20%.

Exchange [3] consists of daily exchange rates in eight countries from 1990 to 2016. The target is set to 'Singapore' and the train/val/test is 70%/10%/20%.

Numeral details of these datasets are shown in Table 9.

Table 9: Details of six datasets

Dataset	Size	Dimension	Frequency
$ETTh_1$	17420	7	1h
$ETTm_2$	69680	7	15min
ECL	26304	321	1h
Traffic	17544	862	1h
weather	52696	21	10min
Exchange	7588	8	1day

D.2 Experiment Details

Hyper-parameter/Setting details of FDNet are shown in Table 10. MSE ($\sum_{i=1}^{n}(x_i - \hat{x}_i)^2/n$) and MAE ($\sum_{i=1}^{n}|x_i - \hat{x}_i|/n$) are chosen as metrics. All experiments are repeated 10 times and means of metrics are used. Results of other baselines are directly borrowed from their papers if exist. We do their rest experiments according to their default settings.

Table 10: Details of hyper-parameters/settings

Hyper-parameters/Settings	Values/Mechanisms
Input length	672
The number of input sub-sequences divided by focal input sequence decomposition method	5
The number of decomposed feature extraction layers	5
Embedding dimension	8
The kernel size of Conv layers	$\{1 \times 1$ (odd), 3×1 (even)$\}$
Standardization	Z-score
Loss function of the second stage	MSE
Optimizer	Adam
Activation	Gelu
Dropout	0.1
Learning rate	1e-4
Learning rate decreasing rate	Half per epoch
Batch size	16
Random seed	4321 (if used)
Platform	Python 3.8.0 Pytorch 1.11.0
Device	A single NVIDIA GeForce RTX 3090 24GB GPU

D.3 Supplementary of Experiment Results

Main Results under Exchange We extra perform multivariate/univariate forecasting experiments under Exchange dataset for more comprehensive comparison and results are shown in Table 11/12.

[3] It was acquired at: https://drive.google.com/drive/folders/1ZOYpTUa82_jCcxIdTmyr0LXQfvaM9vIy?usp=sharing

The lowest MSE/MAE are highlighted in bold and italic. It could be observed that forecasting performances of ETSformer and CoST could challenge that of FDNet over half of situations under Exchange dataset. However, when compared with ETSformer/CoST, FDNet yields 16.4%/41.7% relative MSE reduction during univariate forecasting and yields 28.6%/22.1% relative MSE reduction during univariate forecasting in general under all six datasets. It illustrates that FDNet is more general when compared with ETSformer/CoST.

Table 11: Results of multivariate forecasting under Exchange

Methods	FDNet		FEDformer-f		FEDformer-w		Pyraformer		ETSformer		Triformer		SCINet		TS2Vec		CoST	
Pred	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.095	0.226	0.148	0.278	0.139	0.276	1.489	1.018	***0.085***	***0.204***	0.330	0.406	0.221	0.365	0.184	0.315	0.259	0.383
192	0.184	0.322	0.271	0.380	0.256	0.369	1.642	1.075	***0.182***	***0.303***	0.750	0.611	0.323	0.442	0.373	0.452	0.467	0.514
336	0.381	0.465	0.460	0.500	0.426	0.464	1.744	1.107	***0.348***	***0.428***	1.776	0.966	0.661	0.564	0.666	0.612	0.853	0.688
720	***0.806***	***0.679***	1.195	0.841	1.090	0.800	2.080	1.197	1.025	0.774	1.844	0.986	2.691	1.320	2.941	1.313	1.124	0.879

Table 12: Results of univariate forecasting under Exchange

Methods	FDNet		FEDformer-f		FEDformer-w		Pyraformer		ETSformer		Triformer		SCINet		TS2Vec		CoST	
Pred	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.109	0.264	0.154	0.304	0.131	0.284	0.627	0.639	***0.100***	***0.252***	0.393	0.505	0.209	0.366	0.184	0.315	0.107	0.263
192	***0.225***	0.388	0.286	0.420	0.277	0.420	1.010	0.820	0.226	***0.353***	1.255	0.927	0.347	0.475	0.373	0.452	0.225	0.381
336	0.439	0.525	0.511	0.555	0.426	0.511	1.227	0.915	0.434	***0.500***	2.025	1.194	0.575	0.604	0.666	0.612	***0.431***	0.512
720	***0.702***	***0.655***	1.301	0.879	1.162	0.832	1.742	1.134	0.990	0.821	2.074	1.105	1.378	0.939	2.941	1.313	0.778	0.682

Full Results of FEDformer Here we present full results of FEDformer in its two versions: {FEDformer-f using Fourier basis, FEDformer-w using Wavelet basis}. Their multivariate/univariate forecasting results are shown in 13/14. Results with underline mean that they are lower than corresponding results of FDNet. Obviously, only in one condition which is the univaritae forecasting under $ETTm_2$ with the prediction length of 96, FEDformer-w could outperform FDNet.

Table 13: Results of FEDformer during multivariate forecasting

Formats	Metrics	$ETTh_1$				$ETTm_2$				ECL				Traffic				Weather			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
Fourier	MSE	0.415	0.474	0.535	0.680	0.203	0.269	0.325	0.421	0.193	0.201	0.214	0.246	0.587	0.604	0.621	0.626	0.217	0.276	0.339	0.403
	MAE	0.453	0.493	0.524	0.593	0.287	0.328	0.366	0.415	0.308	0.315	0.329	0.355	0.366	0.373	0.383	0.382	0.296	0.336	0.380	0.428
Wavelet	MSE	0.423	0.448	0.525	0.691	0.204	0.316	0.359	0.433	0.183	0.195	0.212	0.231	0.562	0.562	0.570	0.596	0.227	0.295	0.381	0.424
	MAE	0.464	0.473	0.522	0.618	0.288	0.363	0.387	0.432	0.297	0.308	0.313	0.343	0.349	0.346	0.323	0.368	0.304	0.363	0.416	0.434

Table 14: Results of FEDformer during univariate forecasting

Formats	Metrics	$ETTh_1$				$ETTm_2$				ECL				Traffic				Weather			
		96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
Fourier	MSE	0.103	0.129	0.132	0.134	0.072	0.102	0.130	0.178	0.253	0.282	0.346	0.422	0.207	0.205	0.219	0.244	0.006	0.006	0.004	0.006
	MAE	0.252	0.285	0.291	0.293	0.206	0.245	0.279	0.325	0.370	0.386	0.431	0.484	0.312	0.312	0.323	0.344	0.062	0.062	0.050	0.059
Wavelet	MSE	0.126	0.144	0.151	0.154	0.063	0.110	0.147	0.219	0.262	0.316	0.361	0.448	0.170	0.173	0.178	0.187	0.004	0.005	0.008	0.015
	MAE	0.279	0.298	0.299	0.311	0.189	0.252	0.301	0.368	0.378	0.410	0.445	0.501	0.263	0.265	0.266	0.286	0.046	0.059	0.072	0.091

MAE of Ablation Study Additional MAEs of ablation study results are shown in Table 15. The lowest MAE is highlighted in bold and italic.

MAE of FUNet Additional MAEs of FUNet under $ETTh_1$ are shown in Table 16. Bold and italic MAEs mean that they are lower than those of all other selected baselines in Table 2/3. MAEs with underline in the right part mean that they are higher than corresponding MAEs of FDNet but lower than those of other selected baselines.

LSTI Problem Handling Capability Additional MAEs on LSTI problem handling capability under $ETTh_1$ is shown in Table 17.

It could be observed from table 15/16/17 that conclusions and discovery drawn in the main text will not change though considering both MSE and MAE.

Table 15: MAE of ablation study

Focal	Traffic (Univariate)				Traffic (Multivariate)			
Methods	96	192	336	720	96	192	336	720
Initial	0.229	0.228	0.241	0.260	0.288	0.293	0.300	0.317
Focal	0.222	0.224	***0.227***	0.259	***0.276***	***0.280***	***0.286***	***0.306***
Pyramid	***0.215***	***0.218***	0.225	***0.256***	0.279	0.284	0.288	0.306
Patch	0.230	0.228	0.235	0.261	0.279	0.282	0.288	0.308
Conv	weather (Univariate)				weather (Multivariate)			
Methods	96	192	336	720	96	192	336	720
Conv	***0.030***	***0.034***	***0.033***	***0.037***	***0.211***	***0.248***	***0.286***	***0.333***
MLP	0.036	0.037	0.039	0.040	0.213	0.248	0.287	0.334

Table 16: MAE of FUNet under $ETTh_1$

Methods	$ETTh_1$ (Univariate)				$ETTh_1$ (Multivariate)			
	96	192	336	720	96	192	336	720
FUNet	***0.190***	***0.202***	***0.230***	***0.263***	0.416	0.448	0.473	0.550
FDNet	0.200	0.226	0.248	0.331	***0.397***	***0.419***	***0.438***	***0.482***

Table 17: MAE on LSTI problem handling capability under $ETTh_1$

Methods	FDNet		ETSformer		FEDformer-w	
Input length	Mean	Std	Mean	Std	Mean	Std
96	***0.453***	***1.1e-3***	0.656	2.8e-3	0.377	1.3e-3
672	***0.314***	***1.2e-3***	0.765	3.0e-3	0.408	1.2e-3
1344	***0.324***	***1.0e-3***	0.770	3.9e-3	0.432	9.4e-3

D.4 Visualization of Final Representation by T-SNE

FDNet with focal input sequence decomposition method have two expectations for final representations of each input sequence. The first one aims to independently extract representations of different input sub-sequences. It means that final representations of different input sub-sequences shall be distant from each other. Moreover, when the sub-sequence is closer to prediction sequence, the number of feature extraction layers increases so that the number of convolutional layers increases.

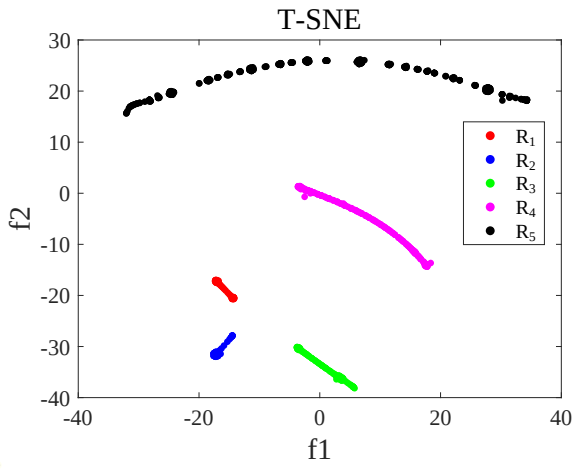

Figure 9: The visualization of final representations of elements in an input sequence by T-SNE under ETTm_2 univariate forecasting. The input sequence length is set to 672.

It means that if the sub-sequence is closer to prediction sequence, final representations of its input elements shall be relatively closed to each other.

To verify whether these are established in real-world applications, we visualize final representations of elements in an input sequence with length of 672 via T-SNE under ETTm_2 univariate forecasting as Figure 9 shows. This input sequence is randomly selected in the test dataset and it is divided into 5 sub-sequences by focal input sequence decomposition method. The prediction length is set to 96. Each data point in Figure 9 denotes the final representation of corresponding input element and its label R_i refers to the i-th input sub-sequence it belongs to. The number of label increases with temporal distances between the input sequence it represents and prediction elements getting farther.

It is obvious from Figure 9 that data points with different colors are distant from each other. It illustrates that feature maps of different sub-sequences are independent with each other, which fits the first expectation. Meanwhile, data points of certain sub-sequence become more sparse as the sub-sequence gets longer and farther from prediction elements, which fits the second expectation. Therefore, two expectations of final representations of FDNet with focal input sequence decomposition method are all achieved, proving the rationality of its design.

E AN OVERVIEW OF FDNET ARCHITECTURE IN DETAILS

An overview of FDNet/FUNet architecture in details is shown in Table 18/20. 'DFE-ICOM'/'DFE-initial' refers to decomposed feature extraction layer with/without ICOM and their detailed architectures are shown in Table 19/21 separately.

Table 18: An overview of FDNet architecture in details

Output sequence				
Add				
FC_5	FC_4	FC_3	FC_2	FC_1
DFE-initial×1	DFE-initial×2	DFE-initial×3	DFE-initial×4	DFE-initial×5
$Embedding_5$	$Embedding_4$	$Embedding_3$	$Embedding_2$	$Embedding_1$
$input_5$	$input_4$	$input_3$	$input_2$	$input_1$
1/2	1/4	1/8	1/16	1/16
Split				
Input sequence ($\{x_{i,1:t_0}\}_{i=1}^{N}$)				

Table 19: DFE-initial components in details

Input feature map
1×1 Conv
WN, Dropout ($p = 0.1$), Gelu
3×1 Conv padding=(1,0)
Add, WN, Dropout ($p = 0.1$), Gelu
1×1 Conv
WN, Dropout ($p = 0.1$), Gelu
3×1 Conv padding=(1,0)
Add, WN, Dropout ($p = 0.1$), Gelu
Output feature map

Table 20: An overview of FUNet architecture in details

Output sequence				
Add				
FC_5	FC_4	FC_3	FC_2	FC_1
DFE-ICOM×4	DFE-ICOM×3	DFE-ICOM×2	DFE-ICOM×1	DFE-ICOM×1
$Embedding_5$	$Embedding_4$	$Embedding_3$	$Embedding_2$	$Embedding_1$
$input_5$	$input_4$	$input_3$	$input_2$	$input_1$
1/2	1/4	1/8	1/16	1/16
Split				
Input sequence ($\{x_{i,1:t_0}\}_{i=1}^{N}$)				

Table 21: DFE-ICOM components in details

<table>
<tr><td colspan="2">Input feature map</td></tr>
<tr><td>Multi-head full attention ($d = 8, h = N$)</td><td rowspan="3"></td></tr>
<tr><td>1×1 Conv</td></tr>
<tr><td>Add, WN, Dropout ($p = 0.1$), Gelu</td></tr>
<tr><td>3×1 Conv stride=(2,1), padding=(1,0)</td><td rowspan="4">3×1 Maxpooling stride=(2,1), padding=(1,0)</td></tr>
<tr><td>WN, Dropout ($p = 0.1$), Gelu</td></tr>
<tr><td>3×1 Conv padding=(1,0)</td></tr>
<tr><td>WN, Dropout ($p = 0.1$), Gelu</td></tr>
<tr><td colspan="2">Add</td></tr>
<tr><td colspan="2">Output feature map</td></tr>
</table>