Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1], Muxi Chen[1], Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)

Joanne Wardell Mar 1 2024

The Discussion Around Transformers for TSF

Do We Really Need Deep Learning Models for Time Series Forecasting?

Shereen Elsayed*, Daniela Thyssens*, Ahmed Rashed, Hadi Samer Jomaa, and Lars Schmidt-Thieme

Department of Computer Science
University of Hildesheim
31141 Hildesheim, Germany

TSMixer: An All-MLP Architecture for Time Series Forecasting

Large Language Models Are Zero-Shot Time Series Forecasters

Nate Gruver* NYU · Marc Finzi* CMU · Shikai Qiu* NYU · Andrew Gordon Wilson NYU

Valeriy M., PhD, MBA, CQF @predict_addict · Jan 31 ·

🎉 Exciting News! 🎉

My recent tweet on "**Are Transformers Effective for Time Series Forecasting?**" has taken off, reaching over 130K views in a very short time! The response has been nothing short of spectacular.

Inspired by this, I've updated my repo...
Show more

Tranformers_Are_What_You_Dont_Need

The best repository showing why transformers don't work in time series forecasting

1. Are Transformers Effective for Time Series Forecasting? by Ailing Zeng, Muxi Chen, Lei Zhang, Qiang Xu (The Chinese University of Hong Kong, International Digital Economy Academy (IDEA), 2022) code 🔥🔥🔥🔥🔥
2. LLMs and foundational models for time series forecasting: They are not (yet) as good as you may hope by Christoph Bergmeir (2023) 🔥🔥🔥🔥🔥
3. Transformers Are What You Do Not Need by Valeriy Manokhin (2023) 🔥🔥🔥🔥🔥
4. Deep Learning is What You Do Not Need by Valeriy Manokhin (2022) 🔥🔥🔥🔥🔥
5. Why do Transformers suck at Time Series Forecasting by Devansh (2023)
6. Frequency-domain MLPs are More Effective Learners in Time Series Forecasting by Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Defu Lian, Ning An, Longbing Cao, Zhendong Niu (Bejing Institute of Technology, Tongji University, University of Oxford, Universuty of Technology Sydney, University of Macau, HeFei University of Technology, Macquarie University) (2023) 🔥🔥🔥🔥🔥
7. Forecasting CPI inflation under economic policy and geo-political uncertainties by Shovon Sengupta, Tanujit Chakraborty, Sunny Kumar Singh (Fidelity Investments, Sorbonne University, BITS Pilani Hyderabad). (2024) 🔥🔥🔥🔥🔥
8. Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping by Zhe Li, Shiyi Qi, Yiduo Li, Zenglin Xu (Harbin Institute of Technology, Shenzhen, 2023) code
9. SCINet: Time Series Modeling and Forecasting with Sample Convolution and Interaction by Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, Qiang Xu (The Chinese University of Hong

5 · 59 · 323 · 35K

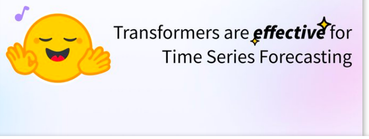
Yes, Transformers are Effective for Time Series Forecasting (+ Autoformer)

Published June 16, 2023

Update on GitHub

elisim Eli Simhayev guest · kashif Kashif Rasul · nielsr Niels Rogge

Open in Colab



Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1]*, Muxi Chen[1]*, Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)
{alzeng, mxchen21, qxu}@cse.cuhk.edu.hk
{leizhang}@idea.edu.cn

Abstract

Recently, there has been a surge of Transformer-based solutions for the long-term time series forecasting (LTSF) task. Despite the growing performance over the past few years, we question the validity of this line of research in this work. *Specifically, Transformers is arguably the most successful solution to extract the semantic correlations among the elements in a long sequence. However, in time series modeling, we are to extract the temporal relations in* an ordered set of continuous points. *While employing positional encoding and using tokens to embed sub-series in Transformers facilitate preserving some ordering information, the nature of the* permutation-invariant *self-attention mechanism inevitably results in temporal information loss.*

ergy management, and financial investment. Over the past several decades, TSF solutions have undergone a progression from traditional statistical methods (e.g., ARIMA [1]) and machine learning techniques (e.g., GBRT [11]) to deep learning-based solutions, e.g., Recurrent Neural Networks [15] and Temporal Convolutional Networks [3, 17].

Transformer [26] is arguably the most successful sequence modeling architecture, demonstrating unparalleled performances in various applications, such as natural language processing (NLP) [7], speech recognition [8], and computer vision [19, 29]. Recently, there has also been a surge of Transformer-based solutions for time series analysis, as surveyed in [27]. Most notable models, which focus on the less explored and challenging long-term time series forecasting (LTSF) problem, include LogTrans [16]

Simon Judge @mrprediction3 · Jun 23, 2023

The paper '**Are Transformers Effective for Time Series Forecasting?**' asked whether **Transformers** really **are** required when much simpler models **are** better. There's now a rebuttal showing that **Transformers are** better than DLinear.
mrprediction.com/transformers-f...
#machinelearning

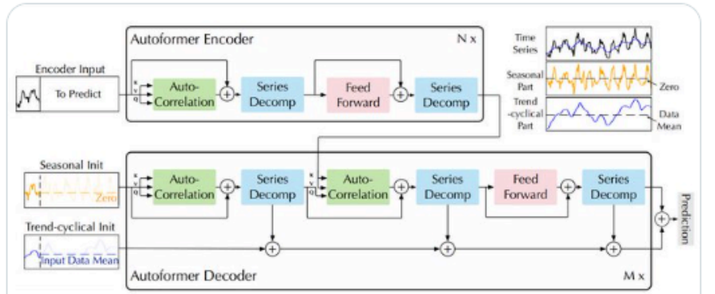


Figure 1: Autoformer architecture. The encoder eliminates the long-term trend-cyclical part by series decomposition blocks (blue blocks) and focuses on seasonal patterns modeling. The decoder accumulates the trend part extracted from hidden variables progressively. The past seasonal information from encoder is utilized by the encoder-decoder Auto-Correlation (center green block in decoder).

Paper Preview

Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1]*, Muxi Chen[1]*, Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy (IDEA)
{alzeng, mxchen21, qxu}@cse.cuhk.edu.hk
{leizhang}@idea.edu.cn

Abstract

Recently, there has been a surge of Transformer-based solutions for the long-term time series forecasting (LTSF) task. Despite the growing performance over the past few years, we question the validity of this line of research in this work. *Specifically, Transformers is arguably the most successful solution to extract the semantic correlations among the elements in a long sequence. However, in time series modeling, we are to extract the temporal relations in* an ordered set of continuous points. *While employing positional encoding and using tokens to embed sub-series in Transformers facilitate preserving some ordering information, the nature of the* permutation-invariant *self-attention mechanism inevitably results in temporal information loss.*

ergy management, and financial investment. Over the past several decades, TSF solutions have undergone a progression from traditional statistical methods (e.g., ARIMA [1]) and machine learning techniques (e.g., GBRT [11]) to deep learning-based solutions, e.g., Recurrent Neural Networks [15] and Temporal Convolutional Networks [3, 17].

Transformer [26] is arguably the most successful sequence modeling architecture, demonstrating unparalleled performances in various applications, such as natural language processing (NLP) [7], speech recognition [8], and computer vision [19, 29]. Recently, there has also been a surge of Transformer-based solutions for time series analysis, as surveyed in [27]. Most notable models, which focus on the less explored and challenging long-term time series forecasting (LTSF) problem, include LogTrans [16]

With the above, we conclude that *the temporal modeling capabilities of Transformers for time series are exaggerated, at least for the existing LTSF benchmarks.* At the

Claim: Transformers' MHA results in temporal information loss, impacting LTSF performance.

Their Model(s): LTSF-Linear

Experiments: 9X datasets

Additional Study: TSF Transformers design elements

Time Series Forecasting (TSF)

- Time series - ubiquitous in today's world

- TSF Solutions:
  - Statistical Methods -
    - ARIMA (1960s)
  - Machine Learning Techniques -
    - GBRT (2000s)
  - Deep Learning Techniques -
    - RNNs (1980s), Temporal CNNs (2010s)

- Transformers
  - Sequence modeling
    - NLP, speech recognition, computer vision

  - Time series analysis
    - LogTrans, Informer, Pyraformer, Triformer, FEDformer

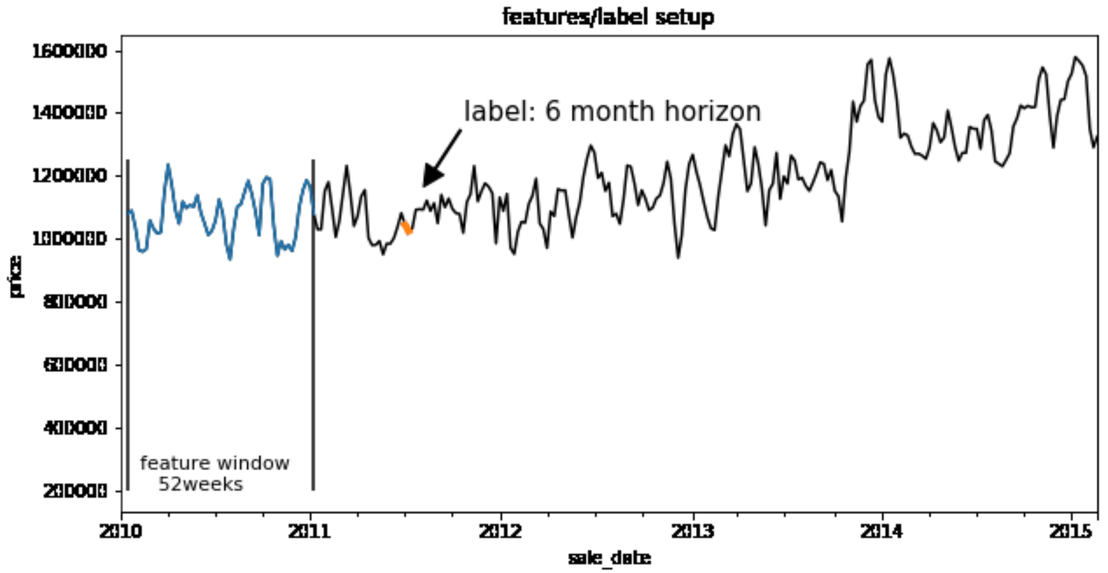
Model Type	Short Description	Date	Reference
BaselineHA	Historical Average Baseline	.	.
BaselineRW	Random Walk Baseline - guess the previous value	.	.
Regression4TS	Time-Series Regression with Lagged Variables	1949	[19]
ARIMA	AutoRegressive, Integrated, Moving-Average	1951	[9, 125]
SES	Simple Exponential Smoothing	1957	[36]
VAR	Vector AutoRegressive	1957	[96]
VARMA	Vector AutoRegressive, Moving-Average	1957	[96]
VECM	Vector Error Correction Model	1957	[96]
ES	Exponential Smoothing (Holt-Winters)	1960	[127]
SARIMA	Seasonal ARIMA	1967	[11]
SARIMAX	SARIMA, with eXogenous variables	1970	[10]
NAR/NARX	Nonlinear AutoRegressive, Exogenous	1978	[46]
Neural Network	Neural Network	1988	[108]
RNN	Recurrent Neural Network	1989	[126]
FDA/FTSA	Functional Data/Time-Series Analysis	1991	[98]
CNN	Convolutional Neural Network	1995	[60]
SVR	Support Vector Regression	1997	[85]
LSTM	Long, Short-Term Memory	1997	[35]
ELM	Extreme Learning Machine	2007	[107]
GRU	Gated Recurrent Unit	2014	[16]
Encoder-Decoder	Encoder-Decoder with Attention	2014	[17]
MGU	Minimal Gated Unit	2016	[149]
TCN	Temporal Convolutional Network	2016	[59]
GNN/GCN	Graph Neural/Convolutional Network	2016	[56]
vTRF	(Vanilla) Transformer	2017	[118]

Miller et. al. "A Survey of Deep Learning and Foundation Models for Time Series Forecasting", 2024

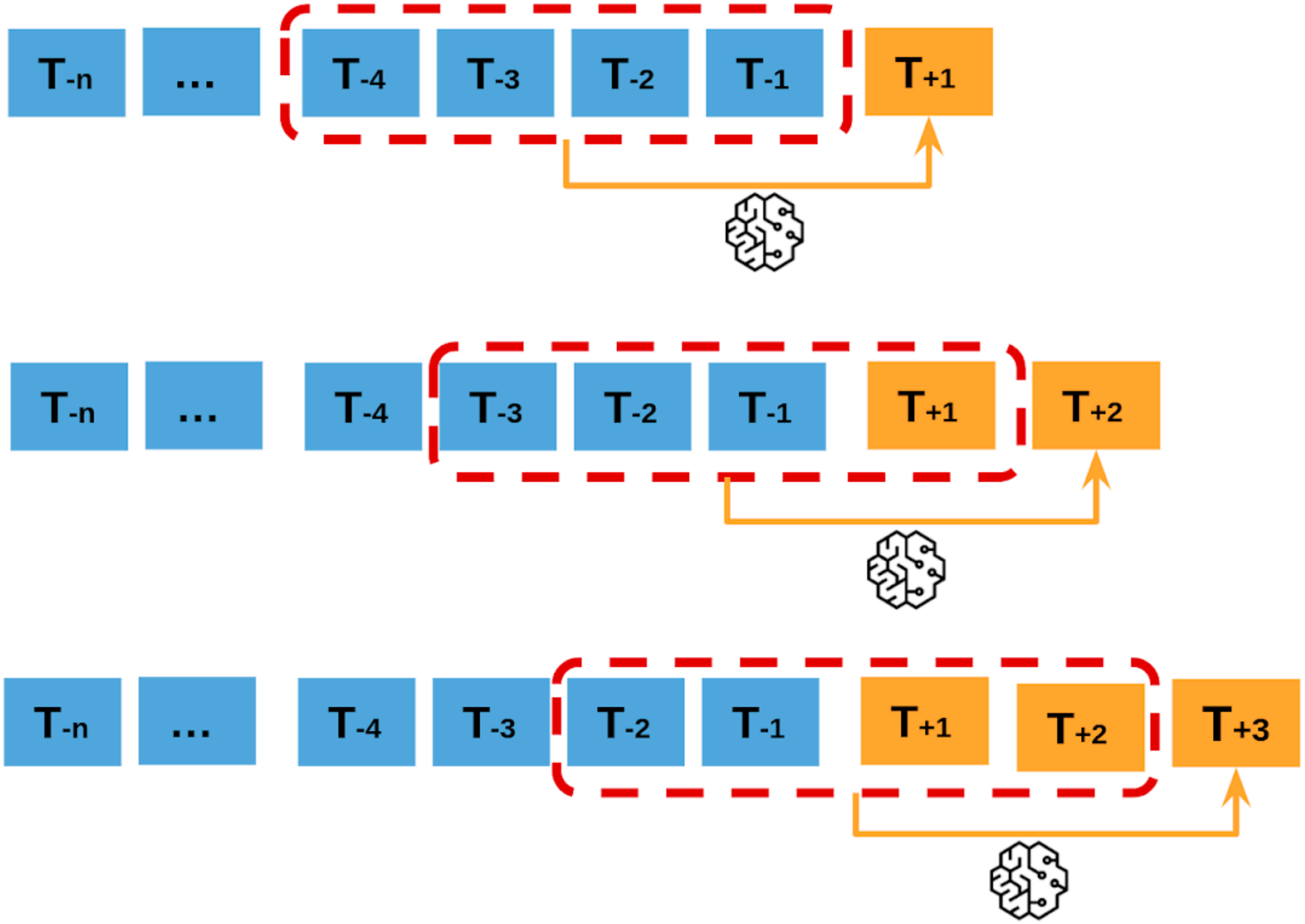
TSF Problem Formulation

- Given:
  - Time series containing C variates, denoted as $X = \{X_1^t, \ldots, X_C^t\}_{t=1}^{L}$
  - L is the look-back window size
  - X_i^t is the value of the i_{th} variate at the t_{th} time step.
- Find:
  - Predicted values $\hat{X} = \{\hat{X}_1^t, \ldots, \hat{X}_C^t\}_{t=L+1}^{L+T}$ at the T future time steps.

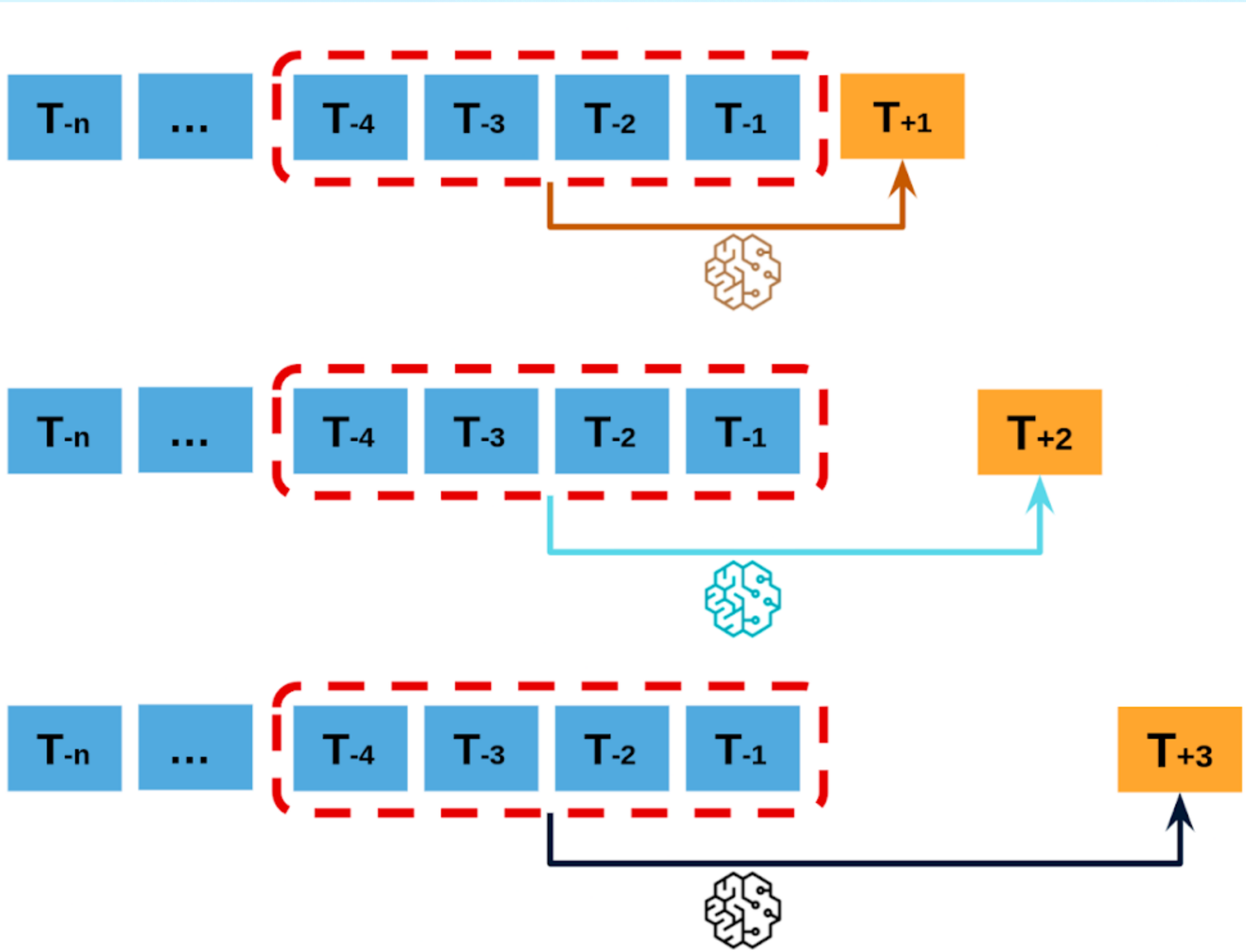
- When T > 1, can use iterated multi-step (IMS) forecasting.
- Alternatively, direct multi-step (DMS) is used.

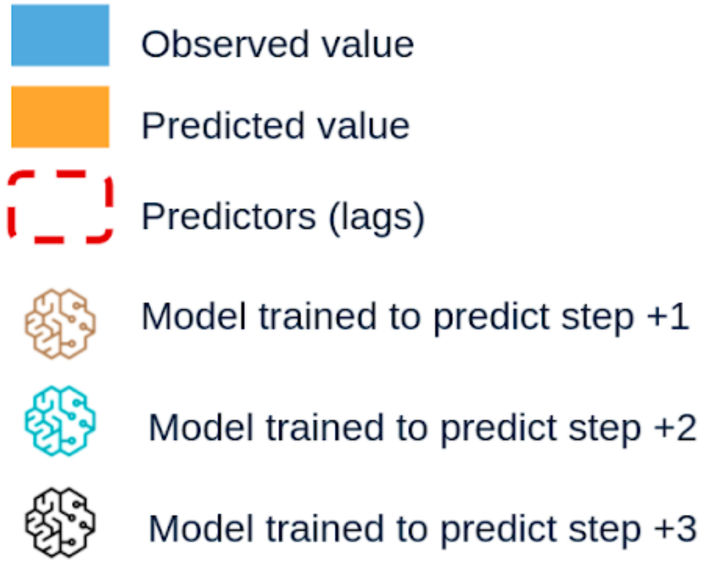


IMS Forecasting



DMS Forecasting





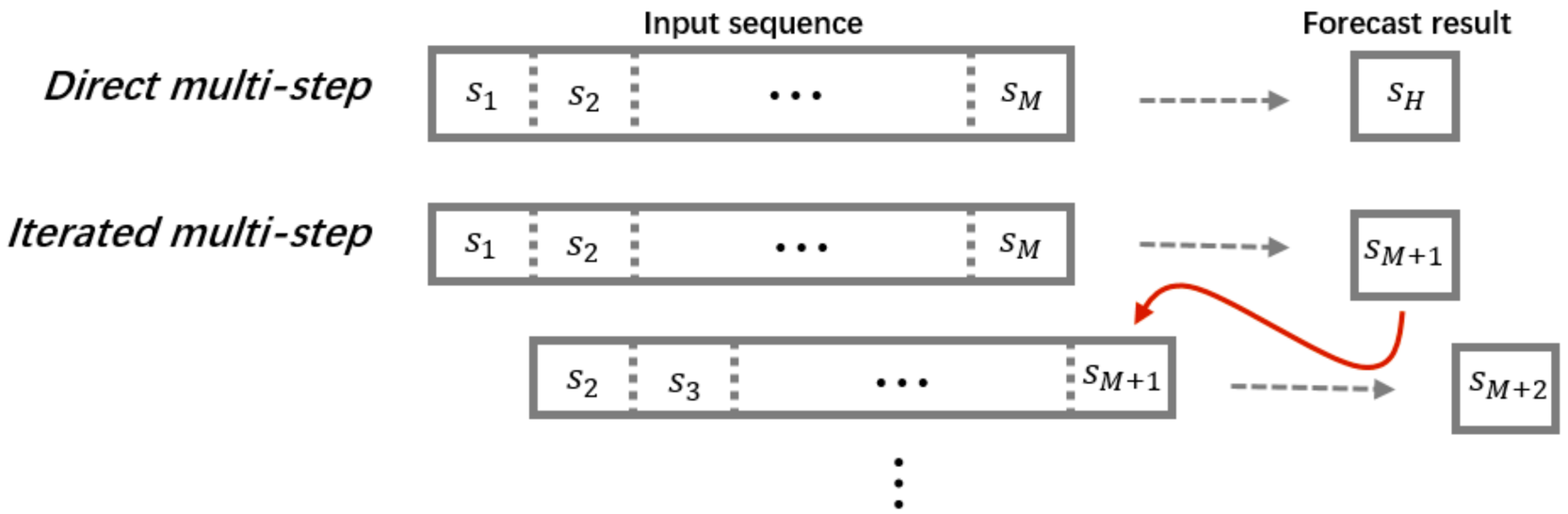
IMS vs DMS Forecasting

IMS

- Autoregressive estimation procedure
- Smaller variance
- Error accumulation effects
- Preferred when T is relatively small
- Needs highly accurate single-step forecaster

DMS

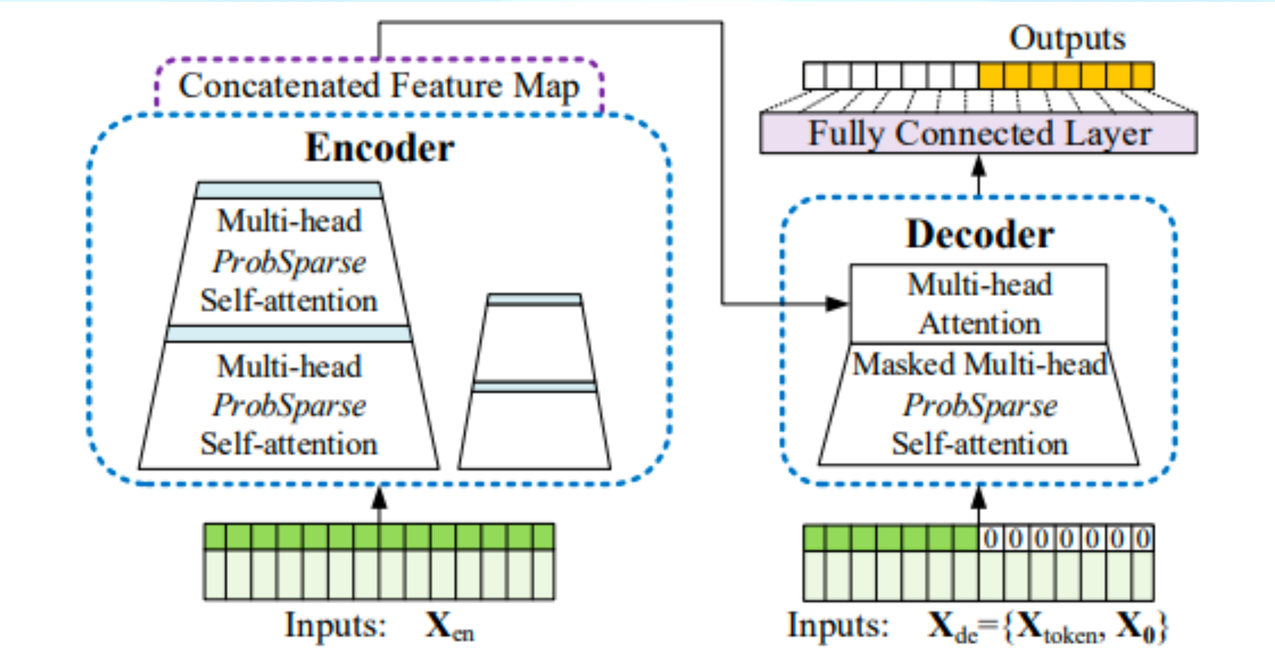
- More accurate predictions
- Preferred when T is small
- Doesn't need unbiased single-step forecaster



Transformers as LTSF Solutions

- Unparalleled performance in AI tasks
- Triggered research into Transformers
- Lots of research into Transformers for LTSF tasks

- Vanilla Transformer
  - time/memory complexity
    - Autoregressive decoder design
  - Informer - novel architecture
    - Reduced complexity
    - DMS forecasting strategy



Informer Model Overview

Existing Transformer-based TSF Pipelines

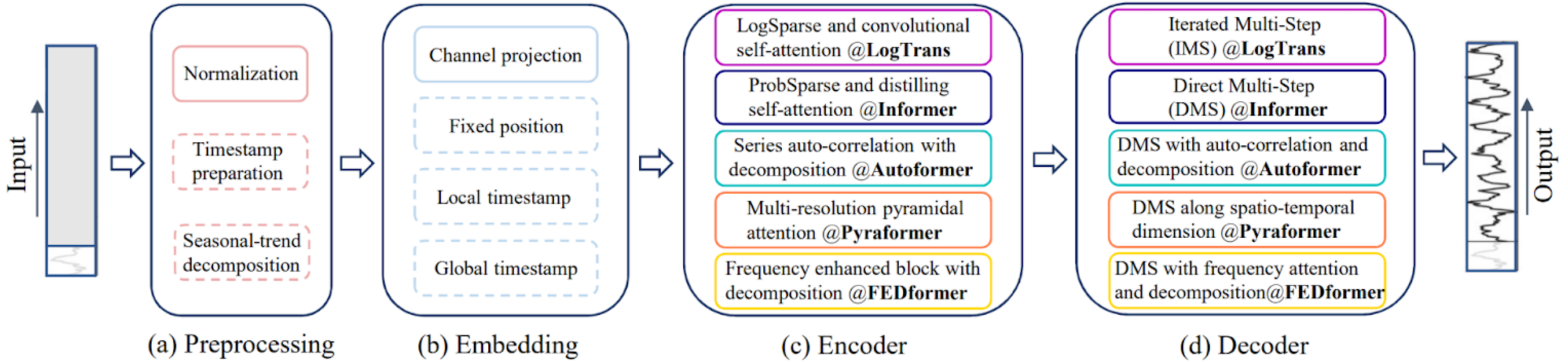


Figure 1. The pipeline of existing Transformer-based TSF solutions. In (a) and (b), the solid boxes are essential operations, and the dotted boxes are applied optionally. (c) and (d) are distinct for different methods [16, 18, 28, 30, 31].

An Embarrassingly Simple Baseline

- LTSF Transformers compared to IMS forecasting
  - IMS - known for error accumulation effects
- Hypothesis:
  - Performance improvement mainly due to DMS forecasting in LTSF Trans.

- Baseline: Simplest DMS model
  - Temporal linear layer
  - Directly regresses historical TS for future prediction via weighted sum
  - $\hat{X}_i = W X_i,\ W \in \mathbb{R}^{T \times L}$

W - linear layer along temporal axis

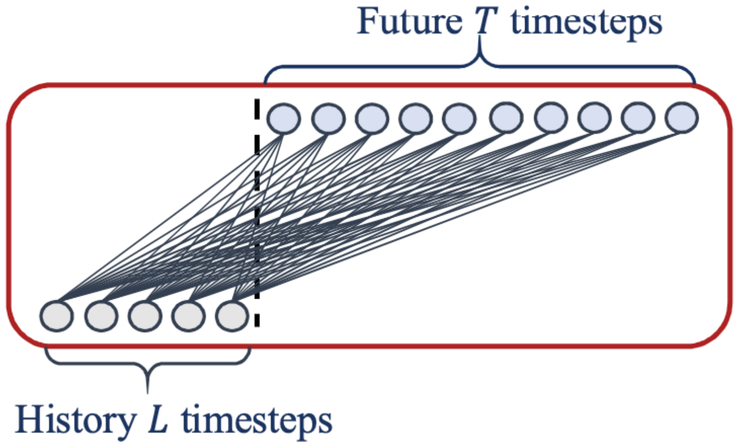


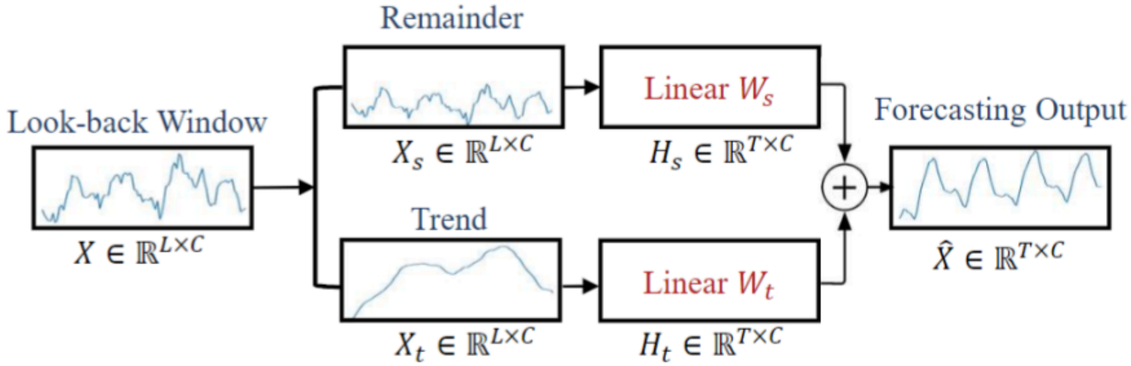
Figure 2. Illustration of the basic linear model.

LTSF Linear (Set of Linear Models)

- Vanilla Linear
  - One layer linear model

- DLinear (Handles Trend)
  - Based on Autoformer & FEDformer decomposition scheme
  - Decomp. input data to a trend component (Mv. Avg. Kernel)
  - Two 1-layer linear layers applied to each comp.
  - Sum features to get final result

- NLinear (Handles Shift)
  - Subtracts input by the last value of the sequence
  - Puts through Linear Layer
  - Adds subtracted part back before final prediction



(a) The whole structure of *DLinear*

Experimental Settings

- Datasets
  - Electricity Transformer Temperature (ETT)
    - ETTh1, ETTh2, ETTm1, ETTm2
  - Traffic, Electricity, Weather, ILI, Exchange-Rate

- Evaluation Metric
  - Mean Squared Error (MSE)
  - Mean Absolute Error (MAE)

Compared Methods

- Transformers
  - FEDformer-f
    - Fourier TX variant
  - Autoformer
  - Informer
  - Pyraformer
  - LogTrans

- DMS Methods
  - Closest Repeat (Repeat)

Datasets	ETTh1&ETTh2	ETTm1 &ETTm2	Traffic	Electricity	Exchange-Rate	Weather	ILI
Variates	7	7	862	321	8	21	7
Timesteps	17,420	69,680	17,544	26,304	7,588	52,696	966
Granularity	1hour	5min	1hour	1hour	1day	10min	1week

Table 1. The statistics of the nine popular datasets for the LTSF problem.

Comparison with Transformers - Quantitative results

Methods		IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer		Pyraformer*		LogTrans		*Repeat**	
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	27.40%	**0.140**	**0.237**	0.141	**0.237**	**0.140**	**0.237**	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	0.258	0.357	1.588	0.946
	192	23.88%	**0.153**	0.250	0.154	**0.248**	**0.153**	0.249	0.201	0.315	0.222	0.334	0.296	0.386	0.386	0.443	0.266	0.368	1.595	0.950
	336	21.02%	**0.169**	0.268	0.171	**0.265**	**0.169**	0.267	0.214	0.329	0.231	0.338	0.300	0.394	0.378	0.443	0.280	0.380	1.617	0.961
	720	17.47%	**0.203**	0.301	0.210	**0.297**	**0.203**	0.301	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	0.283	0.376	1.647	0.975
Exchange	96	45.27%	0.082	0.207	0.089	0.208	**0.081**	0.203	0.148	0.278	0.197	0.323	0.847	0.752	0.376	1.105	0.968	0.812	**0.081**	**0.196**
	192	42.06%	0.167	0.304	0.180	0.300	**0.157**	0.293	0.271	0.380	0.300	0.369	1.204	0.895	1.748	1.151	1.040	0.851	0.167	**0.289**
	336	33.69%	0.328	0.432	0.331	0.415	**0.305**	0.414	0.460	0.500	0.509	0.524	1.672	1.036	1.874	1.172	1.659	1.081	**0.305**	**0.396**
	720	46.19%	0.964	0.750	1.033	0.780	**0.643**	**0.601**	1.195	0.841	1.447	0.941	2.478	1.310	1.943	1.206	1.941	1.127	0.823	0.681
Traffic	96	30.15%	**0.410**	0.282	**0.410**	**0.279**	**0.410**	0.282	0.587	0.366	0.613	0.388	0.719	0.391	2.085	0.468	0.684	0.384	2.723	1.079
	192	29.96%	**0.423**	0.287	**0.423**	**0.284**	**0.423**	0.287	0.604	0.373	0.616	0.382	0.696	0.379	0.867	0.467	0.685	0.390	2.756	1.087
	336	29.95%	0.436	0.295	**0.435**	**0.290**	0.436	0.296	0.621	0.383	0.622	0.337	0.777	0.420	0.869	0.469	0.734	0.408	2.791	1.095
	720	25.87%	0.466	0.315	**0.464**	**0.307**	0.466	0.315	0.626	0.382	0.660	0.408	0.864	0.472	0.881	0.473	0.717	0.396	2.811	1.097
Weather	96	18.89%	**0.176**	0.236	0.182	**0.232**	**0.176**	0.237	0.217	0.296	0.266	0.336	0.300	0.384	0.896	0.556	0.458	0.490	0.259	0.254
	192	21.01%	**0.218**	0.276	0.225	**0.269**	0.220	0.282	0.276	0.336	0.307	0.367	0.598	0.544	0.622	0.624	0.658	0.589	0.309	0.292
	336	22.71%	**0.262**	0.312	0.271	**0.301**	0.265	0.319	0.339	0.380	0.359	0.395	0.578	0.523	0.739	0.753	0.797	0.652	0.377	0.338
	720	19.85%	0.326	0.365	0.338	**0.348**	**0.323**	0.362	0.403	0.428	0.419	0.428	1.059	0.741	1.004	0.934	0.869	0.675	0.465	0.394
ILI	24	47.86%	1.947	0.985	**1.683**	**0.858**	2.215	1.081	3.228	1.260	3.483	1.287	5.764	1.677	1.420	2.012	4.480	1.444	6.587	1.701
	36	36.43%	2.182	1.036	**1.703**	**0.859**	1.963	0.963	2.679	1.080	3.103	1.148	4.755	1.467	7.394	2.031	4.799	1.467	7.130	1.884
	48	34.43%	2.256	1.060	**1.719**	**0.884**	2.130	1.024	2.622	1.078	2.669	1.085	4.763	1.469	7.551	2.057	4.800	1.468	6.575	1.798
	60	34.33%	2.390	1.104	**1.819**	**0.917**	2.368	1.096	2.857	1.157	2.770	1.125	5.264	1.564	7.662	2.100	5.278	1.560	5.893	1.677
ETTh1	96	0.80%	0.375	0.397	**0.374**	**0.394**	0.375	0.399	0.376	0.419	0.449	0.459	0.865	0.713	0.664	0.612	0.878	0.740	1.295	0.713
	192	3.57%	0.418	0.429	0.408	**0.415**	**0.405**	0.416	0.420	0.448	0.500	0.482	1.008	0.792	0.790	0.681	1.037	0.824	1.325	0.733
	336	6.54%	0.479	0.476	**0.429**	**0.427**	0.439	0.443	0.459	0.465	0.521	0.496	1.107	0.809	0.891	0.738	1.238	0.932	1.323	0.744
	720	13.04%	0.624	0.592	**0.440**	**0.453**	0.472	0.490	0.506	0.507	0.514	0.512	1.181	0.865	0.963	0.782	1.135	0.852	1.339	0.756
ETTh2	96	19.94%	0.288	0.352	**0.277**	**0.338**	0.289	0.353	0.346	0.388	0.358	0.397	3.755	1.525	0.645	0.597	2.116	1.197	0.432	0.422
	192	19.81%	0.377	0.413	**0.344**	**0.381**	0.383	0.418	0.429	0.439	0.456	0.452	5.602	1.931	0.788	0.683	4.315	1.635	0.534	0.473
	336	25.93%	0.452	0.461	**0.357**	**0.400**	0.448	0.465	0.496	0.487	0.482	0.486	4.721	1.835	0.907	0.747	1.124	1.604	0.591	0.508
	720	14.25%	0.698	0.595	**0.394**	**0.436**	0.605	0.551	0.463	0.474	0.515	0.511	3.647	1.625	0.963	0.783	3.188	1.540	0.588	0.517
ETTm1	96	21.10%	0.308	0.352	0.306	0.348	**0.299**	**0.343**	0.379	0.419	0.505	0.475	0.672	0.571	0.543	0.510	0.600	0.546	1.214	0.665
	192	21.36%	0.340	0.369	0.349	0.375	**0.335**	**0.365**	0.426	0.441	0.553	0.496	0.795	0.669	0.557	0.537	0.837	0.700	1.261	0.690
	336	17.07%	0.376	0.393	0.375	0.388	**0.369**	**0.386**	0.445	0.459	0.621	0.537	1.212	0.871	0.754	0.655	1.124	0.832	1.283	0.707
	720	21.73%	0.440	0.435	0.433	0.422	**0.425**	**0.421**	0.543	0.490	0.671	0.561	1.166	0.823	0.908	0.724	1.153	0.820	1.319	0.729
ETTm2	96	17.73%	0.168	0.262	**0.167**	**0.255**	**0.167**	0.260	0.203	0.287	0.255	0.339	0.365	0.453	0.435	0.507	0.768	0.642	0.266	0.328
	192	17.84%	0.232	0.308	**0.221**	**0.293**	0.224	0.303	0.269	0.328	0.281	0.340	0.533	0.563	0.730	0.673	0.989	0.757	0.340	0.371
	336	15.69%	0.320	0.373	**0.274**	**0.327**	0.281	0.342	0.325	0.366	0.339	0.372	1.363	0.887	1.201	0.845	1.334	0.872	0.412	0.410
	720	12.58%	0.413	0.435	**0.368**	**0.384**	0.397	0.421	0.421	0.415	0.433	0.432	3.379	1.338	3.625	1.451	3.048	1.328	0.521	0.465

- Methods* are implemented by us; Other results are from FEDformer [31].

Table 2. Multivariate long-term forecasting errors in terms of MSE and MAE, the lower the better. Among them, ILI dataset is with forecasting horizon $T \in \{24, 36, 48, 60\}$. For the others, $T \in \{96, 192, 336, 720\}$. *Repeat* repeats the last value in the look-back window. The **best results** are highlighted in **bold** and the best results of Transformers are highlighted with a underline. Accordingly, IMP. is the best result of linear models compared to the results of Transformer-based solutions.

Comparison with Transformers - Qualitative results

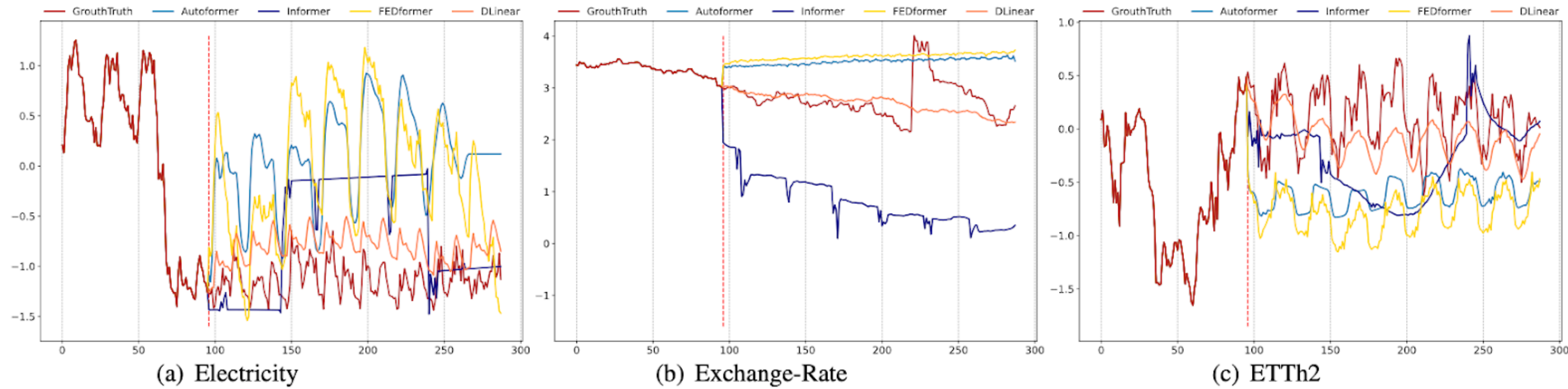
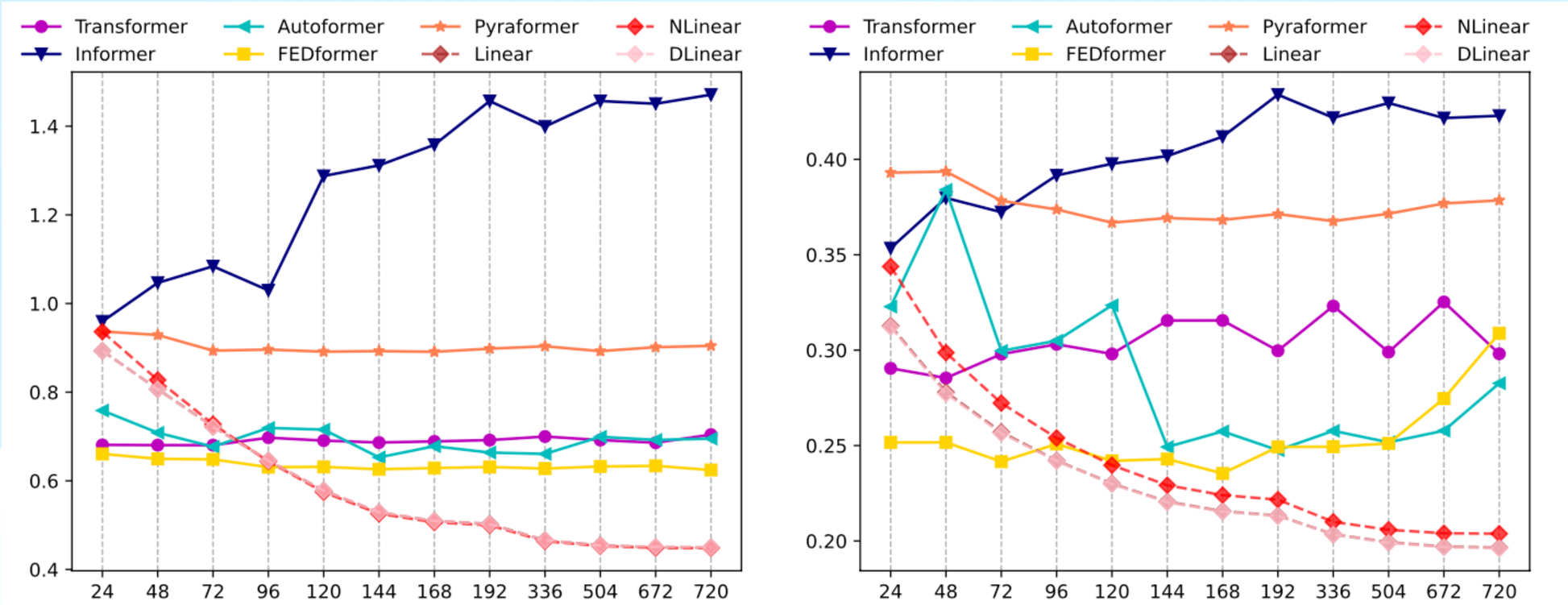


Figure 3. Illustration of the long-term forecasting output (Y-axis) of five models with an input length L=96 and output length T=192 (X-axis) on Electricity, Exchange-Rate, and ETTh2, respectively.

- Fail to capture scale & bias of future data (Electricity, ETTh2)
- Fail to predict trends on aperiodic data (Exchange-Rate)

Can LTSF-Transformers extract temporal relations well from longer input sequences?



(a) **720** steps-**Traffic** (b) **720** steps-**Electricity**

Figure 4. The MSE results (Y-axis) of models with different look-back window sizes (X-axis) of long-term forecasting (T=**720**) on the Traffic and Electricity datasets.

What can be learned for long-term forecasting?

- Short term forecasting
  - Temporal dynamics in look back window

- Long term forecasting
  - Hypothesis: trend & periodicity
  - Farther forecasting horizon, less impact look back window has

Methods	FEDformer		Autoformer	
Input	*Close*	*Far*	*Close*	*Far*
Electricity	0.251	0.265	0.255	0.287
Traffic	0.631	0.645	0.677	0.675

Table 3. Comparison of different input sequences under the MSE metric to explore what LTSF-Transformers depend on. If the input is *Close*, we use the $96_{th}, ..., 191_{th}$ time steps as the input sequence. If the input is *Far*, we use the $0_{th}, ..., 95_{th}$ time steps. Both of them forecast the $192_{th}, ..., (192 + 720)_{th}$ time steps.

Are the self-attention scheme effective for LTSF?

- Gradually transformed Informer to Linear
  - Replace self-attention layers w/ linear layer (Att.-Linear)
  - Discard other auxiliary designs (e.g. FFN)
    - Embedding and linear layers remaining (Embed + Linear)
  - Simplify model to one linear layer

- Informer Performance Grows w/ Simplification

Methods		Informer	*Att.-Linear*	*Embed + Linear*	Linear
Exchange	96	0.847	1.003	0.173	0.084
	192	1.204	0.979	0.443	0.155
	336	1.672	1.498	1.288	0.301
	720	2.478	2.102	2.026	0.763
ETTh1	96	0.865	0.613	0.454	0.400
	192	1.008	0.759	0.686	0.438
	336	1.107	0.921	0.821	0.479
	720	1.181	0.902	1.051	0.515

Table 4. The MSE comparisons of gradually transforming Informer to a Linear from the left to right columns. *Att.-Linear* is a structure that replaces each attention layer with a linear layer. *Embed + Linear* is to drop other designs and only keeps embedding layers and a linear layer. The look-back window size is 96.

Can existing LTSF-Transformers preserve temporal order well?

Methods		Linear			FEDformer			Autoformer			Informer		
Predict Length		*Ori.*	*Shuf.*	*Half-Ex.*	*Ori.*	*Shuf.*	*Half-Ex.*	*Ori.*	*Shuf.*	*Half-Ex.*	*Ori.*	*Shuf.*	*Half-Ex.*
Exchange	96	0.080	0.133	0.169	0.161	0.160	0.162	0.152	0.158	0.160	0.952	1.004	0.959
	192	0.162	0.208	0.243	0.274	0.275	0.275	0.278	0.271	0.277	1.012	1.023	1.014
	336	0.286	0.320	0.345	0.439	0.439	0.439	0.435	0.430	0.435	1.177	1.181	1.177
	720	0.806	0.819	0.836	1.122	1.122	1.122	1.113	1.113	1.113	1.198	1.210	1.196
	Average Drop	N/A	27.26%	46.81%	N/A	-0.09%	0.20%	N/A	0.09%	1.12%	N/A	-0.12%	-0.18%
ETTh1	96	0.395	0.824	0.431	0.376	0.753	0.405	0.455	0.838	0.458	0.974	0.971	0.971
	192	0.447	0.824	0.471	0.419	0.730	0.436	0.486	0.774	0.491	1.233	1.232	1.231
	336	0.490	0.825	0.505	0.447	0.736	0.453	0.496	0.752	0.497	1.693	1.693	1.691
	720	0.520	0.846	0.528	0.468	0.720	0.470	0.525	0.696	0.524	2.720	2.716	2.715
	Average Drop	N/A	81.06%	4.78%	N/A	73.28%	3.44%	N/A	56.91%	0.46%	N/A	1.98%	0.18%

Table 5. The MSE comparisons of models when shuffling the raw input sequence. *Shuf.* randomly shuffles the input sequence. *Half-EX.* randomly exchanges the first half of the input sequences with the second half. Average Drop is the average performance drop under all forecasting lengths after shuffling. All results are the average test MSE of five runs.

- Transformer performance not fluctuating after shuffling
- LTSF-Linear performance damaged significantly

How effective are different embedding strategies?

Methods	Embedding	Traffic 96	192	336	720
FEDformer	All	0.597	0.606	0.627	0.649
	wo/Pos.	**0.587**	**0.604**	**0.621**	**0.626**
	wo/Temp.	0.613	0.623	0.650	0.677
	wo/Pos.-Temp.	0.613	0.622	0.648	0.663
Autoformer	All	0.629	0.647	0.676	**0.638**
	wo/Pos.	**0.613**	**0.616**	**0.622**	0.660
	wo/Temp.	0.681	0.665	0.908	0.769
	wo/Pos.-Temp.	0.672	0.811	1.133	1.300
Informer	All	**0.719**	**0.696**	**0.777**	**0.864**
	wo/Pos.	1.035	1.186	1.307	1.472
	wo/Temp.	0.754	0.780	0.903	1.259
	wo/Pos.-Temp.	1.038	1.351	1.491	1.512

Table 6. The MSE comparisons of different embedding strategies on Transformer-based methods with look-back window size 96 and forecasting lengths $\{96, 192, 336, 720\}$.

Is training data size a limiting factor for LTSF- Transformers?

Unlike CV/NLP -
- TSF is performed on collected time series
- Difficult to scale up the training data size
- Size of the training data significant impact on performance

Experiments on Traffic
- Model trained on full dataset
  - 17,544*0.7 hours
  - Namd Ori.
- Model trained on shortened dataset
  - 8,760 hours, i.e., 1 year
  - Named Short

Result: Prediction errors w/ reduced training data
- Lower in most cases
- Why?
  - whole-year data more clear temporal features
    - Opp. to longer but incomplete data size

Methods	FEDformer		Autoformer	
Dataset	*Ori.*	*Short*	*Ori.*	*Short*
96	0.587	**0.568**	0.613	**0.594**
192	0.604	**0.584**	**0.616**	0.621
336	0.621	**0.601**	0.622	**0.621**
720	0.626	**0.608**	0.660	**0.650**

Table 7. The MSE comparison of two training data sizes

Is efficiency really a top-level priority?

Existing LTSF- Transformers claim

- Vanilla Trans. quad. complexity too high for LTSF
- Prove able to improve theoretical complexity
- Unclear if
  - Actual cost on devices improved
  - Memory issue unacceptable for today's GPU

Compared average practical efficiencies with 5 runs

- Most Transformer variants
  - incur similar or worse in practice
    - inference time
    - parameters
- These follow-ups
  - add more design elements
  - increase practical costs
- vanilla Trans. Memory cost
  - practically acceptable
  - even for output length L = 720,
    - weakens importance of developing a mem. efficient Transformer

Method	MACs	Parameter	Time	Memory
DLinear	**0.04G**	**139.7K**	**0.4ms**	**687MiB**
Transformer×	4.03G	13.61M	26.8ms	6091MiB
Informer	3.93G	14.39M	49.3ms	3869MiB
Autoformer	4.41G	14.91M	164.1ms	7607MiB
Pyraformer	0.80G	241.4M*	3.4ms	7017MiB
FEDformer	4.41G	20.68M	40.5ms	4143MiB

- × is modified into the same one-step decoder, which is implemented in the source code from Autoformer.
- ∗ 236.7M parameters of Pyraformer come from its linear decoder.

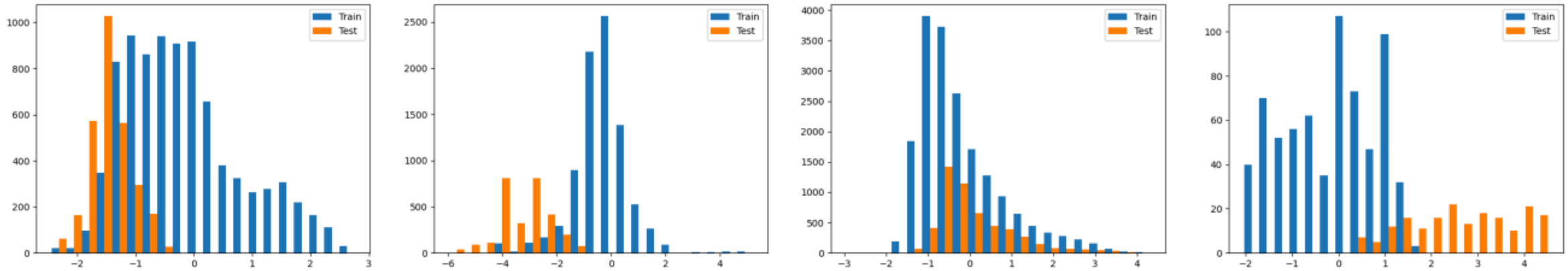
Table 8. Comparison of practical efficiency of LTSF-Transformers under L=96 and T=720 on the Electricity. MACs are the number of multiply-accumulate operations. We use Dlinear for comparison since it has the double cost in *LTSF-Linear*. The inference time averages 5 runs.

Comparison of Univariate Forecasting

Methods		Linear		NLinear		DLinear		FEDformer-f		FEDformer-w		Autoformer		Informer		LogTrans	
Metric		MSE	MAE	MSE	MSE	MAE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1$	96	0.189	0.359	**0.053**	**0.177**	0.056	0.180	0.079	0.215	0.080	0.214	0.071	0.206	0.193	0.377	0.283	0.468
	192	0.078	0.212	**0.069**	**0.204**	0.071	0.204	0.104	0.245	0.105	0.256	0.114	0.262	0.217	0.395	0.234	0.409
	336	0.091	0.237	**0.081**	**0.226**	0.098	0.244	0.119	0.270	0.120	0.269	0.107	0.258	0.202	0.381	0.386	0.546
	720	0.172	0.340	**0.080**	**0.226**	0.189	0.359	0.142	0.299	0.127	0.280	0.126	0.283	0.183	0.355	0.475	0.629
$ETTh2$	96	0.133	0.283	**0.129**	**0.278**	0.131	0.279	0.128	0.271	0.156	0.306	0.153	0.306	0.213	0.373	0.217	0.379
	192	0.176	0.330	**0.169**	**0.324**	0.176	0.329	0.185	0.330	0.238	0.380	0.204	0.351	0.227	0.387	0.281	0.429
	336	0.213	0.371	**0.194**	**0.355**	0.209	0.367	0.231	0.378	0.271	0.412	0.246	0.389	0.242	0.401	0.293	0.437
	720	0.292	0.440	**0.225**	**0.381**	0.276	0.426	0.278	0.420	0.288	0.438	0.268	0.409	0.291	0.439	0.218	0.387
$ETTm1$	96	0.028	0.125	**0.026**	**0.122**	0.028	0.123	0.033	0.140	0.036	0.149	0.056	0.183	0.109	0.277	0.049	0.171
	192	0.043	0.154	**0.039**	**0.149**	0.045	0.156	0.058	0.186	0.069	0.206	0.081	0.216	0.151	0.310	0.157	0.317
	336	0.059	0.180	**0.052**	**0.172**	0.061	0.182	0.084	0.231	0.071	0.209	0.076	0.218	0.427	0.591	0.289	0.459
	720	0.080	0.211	**0.073**	**0.207**	0.080	0.210	0.102	0.250	0.105	0.248	0.110	0.267	0.438	0.586	0.430	0.579
$ETTm2$	96	0.066	0.189	**0.063**	**0.182**	0.063	0.183	0.067	0.198	0.063	0.189	0.065	0.189	0.088	0.225	0.075	0.208
	192	0.094	0.230	**0.090**	**0.223**	0.092	0.227	0.102	0.245	0.110	0.252	0.118	0.256	0.132	0.283	0.129	0.275
	336	0.120	0.263	**0.117**	**0.259**	0.119	0.261	0.130	0.279	0.147	0.301	0.154	0.305	0.180	0.336	0.154	0.302
	720	0.175	0.320	**0.170**	**0.318**	0.175	0.320	0.178	0.325	0.219	0.368	0.182	0.335	0.300	0.435	0.160	0.321

Table 9. Univariate long sequence time-series forecasting results on ETT full benchmark. The **best results** are highlighted in **bold** and the best results of Transformers are highlighted with a underline.

Distribution of ETTh1, ETTh2, Electricity, and ILI dataset



(a) ETTh1 channel6 (b) ETTh2 channel3 (c) Electricity channel3 (d) ILI channel6

Figure 5. Distribution of ETTh1, ETTh2, Electricity, and ILI dataset. A clear distribution shift between training and testing data can be observed in ETTh1, ETTh2, and ILI.

Visualizing Weights of LSTF-Linear on Different Datasets

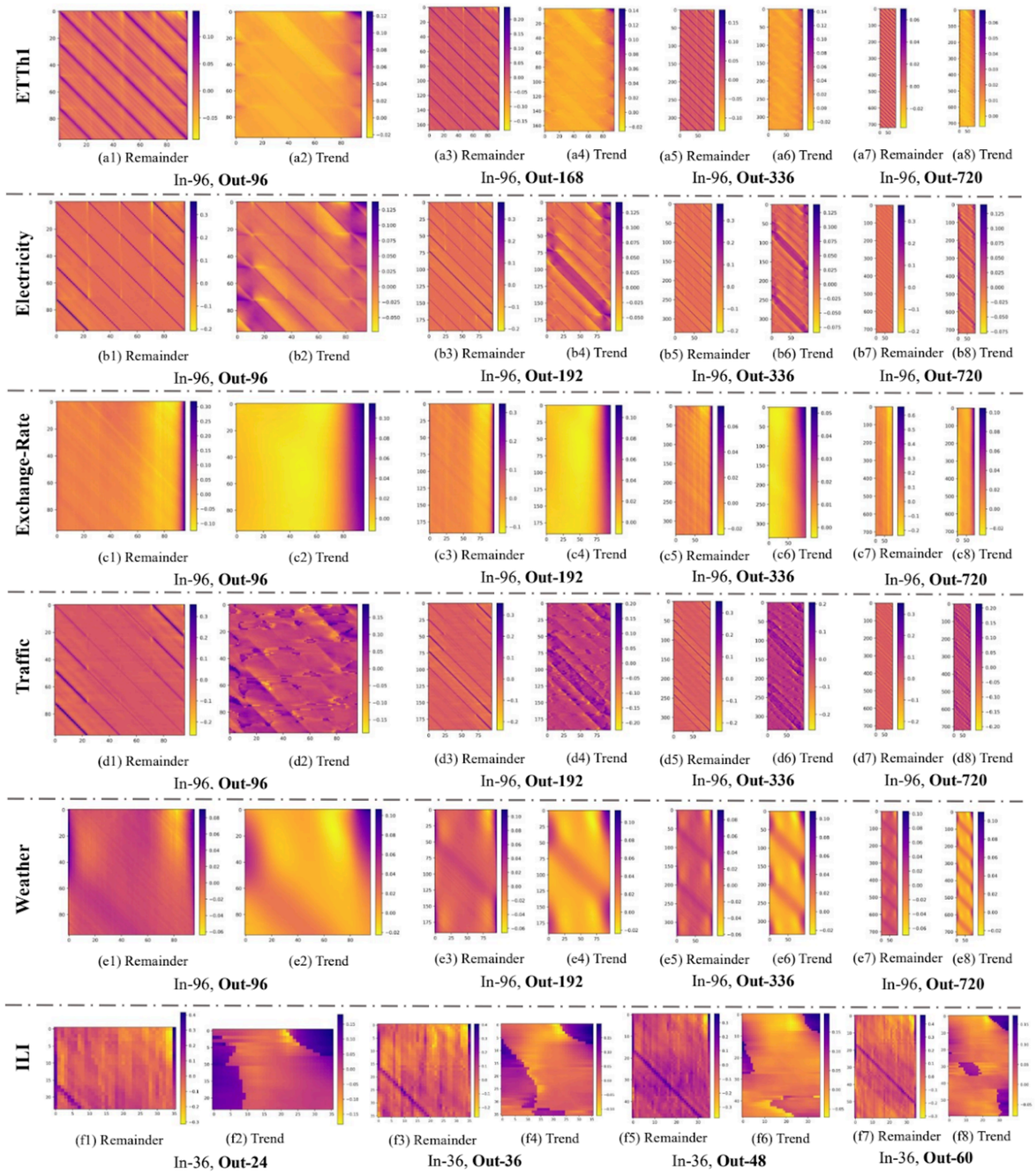
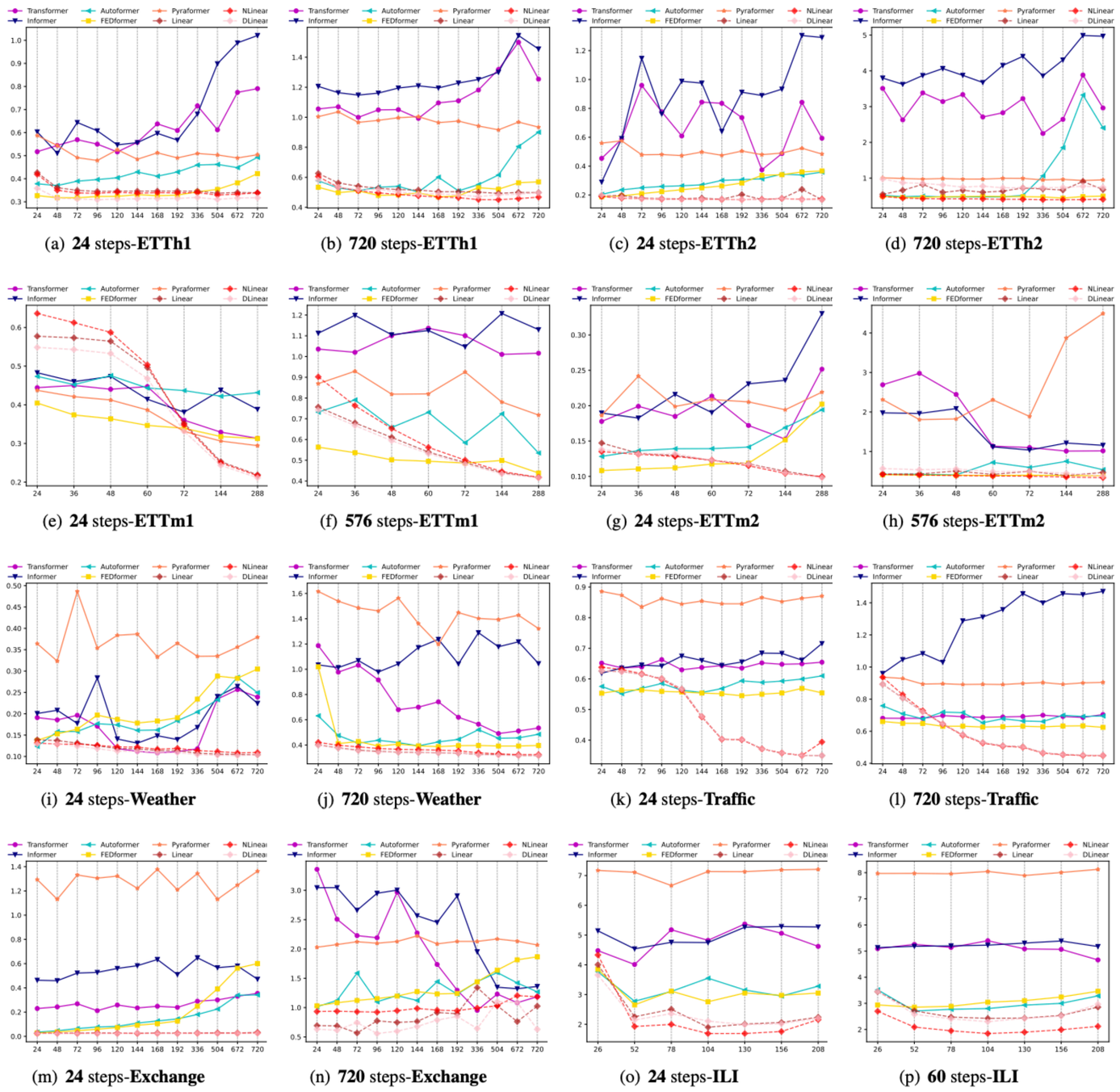


Figure 7. Visualization of the weights(T*L) of *LTSF-Linear* on several benchmarks. Models are trained with a look-back window L (X-axis) and different forecasting time steps T (Y-axis). We show weights in the remainder and trend layer.

The MSE results (Y-axis) of models with different look-back window sizes (X-axis) of the long-term forecasting



(a) **24** steps-**ETTh1** (b) **720** steps-**ETTh1** (c) **24** steps-**ETTh2** (d) **720** steps-**ETTh2**

(e) **24** steps-**ETTm1** (f) **576** steps-**ETTm1** (g) **24** steps-**ETTm2** (h) **576** steps-**ETTm2**

(i) **24** steps-**Weather** (j) **720** steps-**Weather** (k) **24** steps-**Traffic** (l) **720** steps-**Traffic**

(m) **24** steps-**Exchange** (n) **720** steps-**Exchange** (o) **24** steps-**ILI** (p) **60** steps-**ILI**

Figure 6. The MSE results (Y-axis) of models with different look-back window sizes (X-axis) of the long-term forecasting (e.g., **720**-time steps) and the short-term forecasting (e.g., **24** time steps) on different benchmarks.