Towards Foundational Models for Times Series

张 绍 群
Email: zhangsq@lamda.nju.edu.cn
Nanjing University

Content

I. Why do we need foundational models for time series?

a) What is a time series?

b) What is a foundational model?

c) Can we transform a pre-trained LLM for handling time series?

d) Why do we need to train foundational models for time series from scratch?

II. Our Planning

a) Open source of time series

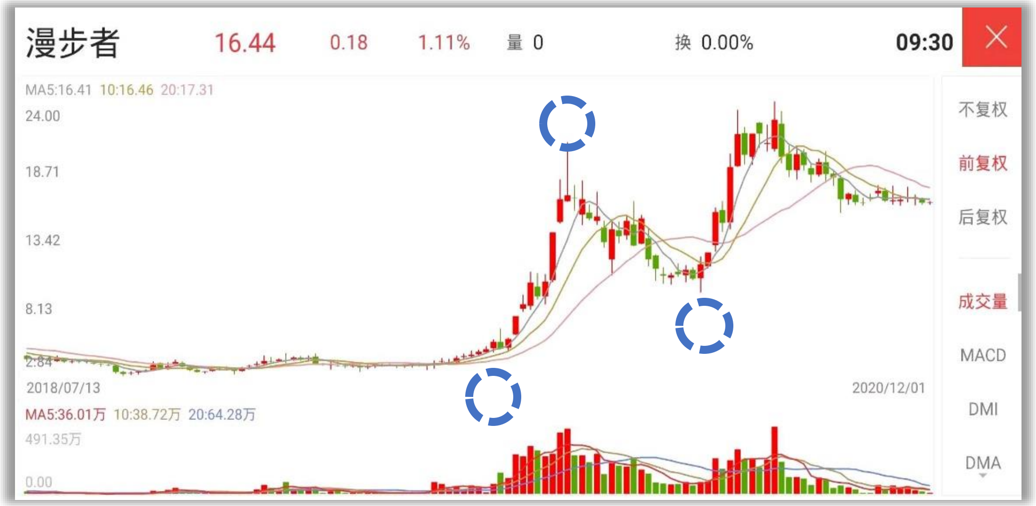
b) Time-Series VS Sequential models

Content

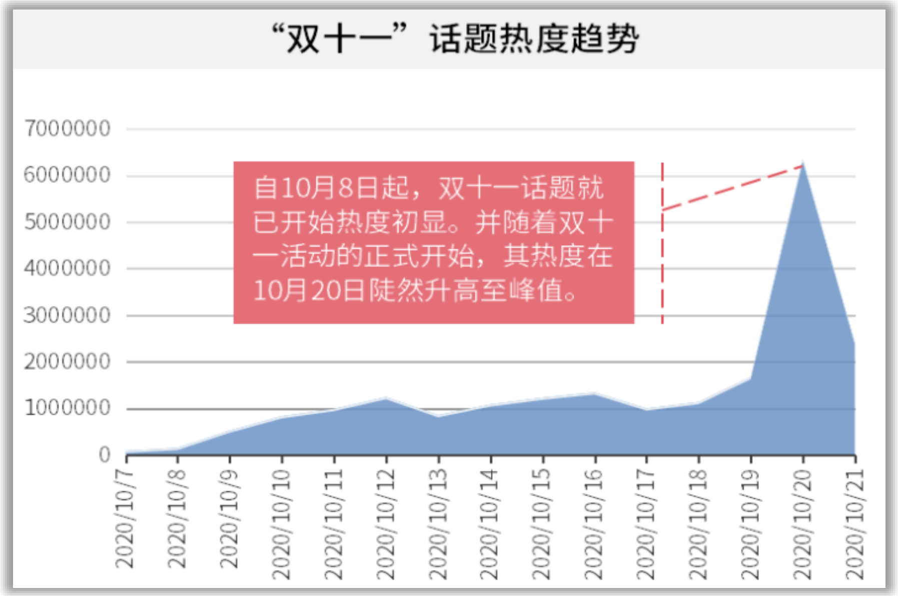
a) What is a time series?

In mathematics, a time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. [WIKI]

Stock



Topic popularity trend



a.1) Time series vs static data

Formulate a sample of time series **X** in math or computing :

Timestamp	t_1	t_2	t_3	…
Feature 1	X_{11}	X_{12}	X_{13}	
Feature 2	X_{21}	X_{22}	X_{23}	
…				
Feature n	X_{n1}	X_{n2}	X_{33}	

Spatial → Static

position, shape, size, etc.

Temporal → Dynamic

frequency, periodicity, stability, trend, etc.

a.1) Time series vs static data

Characteristic	Spatial Data	Spatiotemporal Data
Temporal Dimension	None	Present
Dynamic Nature	Static	Dynamic
Data Structure	Simple (points, lines, polygons)	Complex (space + time)
Analysis Methods	Static spatial analysis	Dynamic spatiotemporal analysis
Application Scenarios	Static mapping, spatial feature studies	Dynamic prediction, trajectory analysis, spatiotemporal change monitoring

b) What is a foundational model?

Introduce foundational model

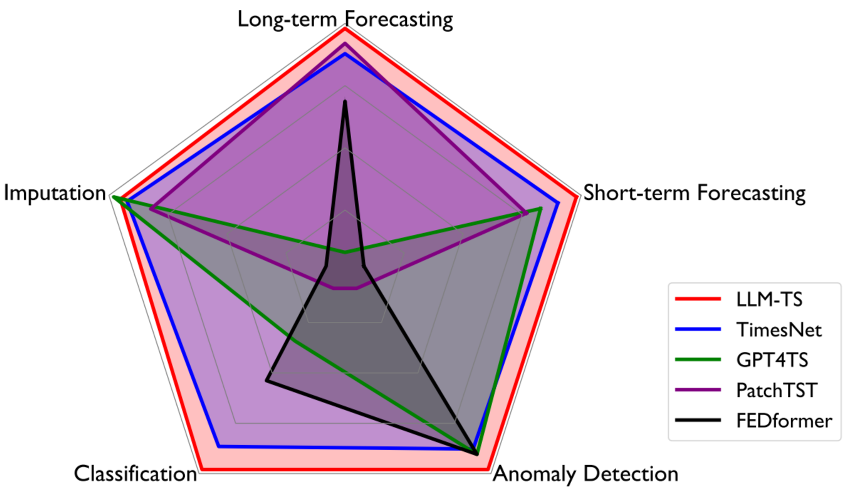
Scaling law: Exploring the ceiling of data-driven learning

How to improve the performance of data-driven learning? Data and Model Expressivity

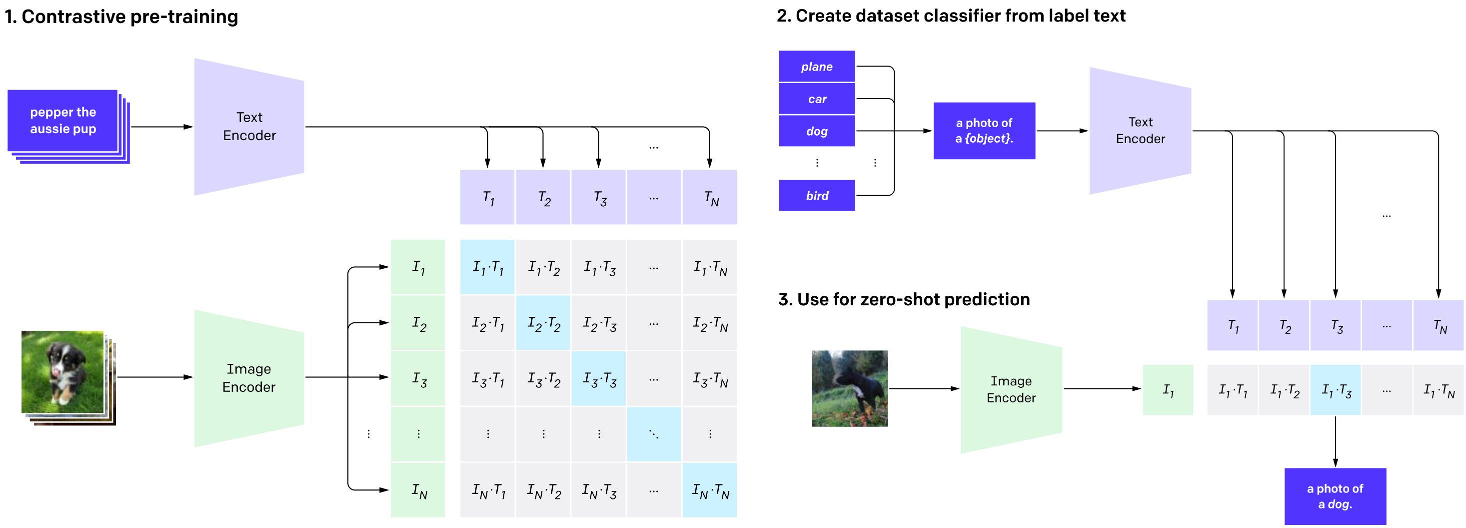
c) Can we transform a pre-trained LLM for handling time series?

<u>Key Idea</u>: Training a time-series small model by exploiting pre-trained LLM

☐ Observation 1

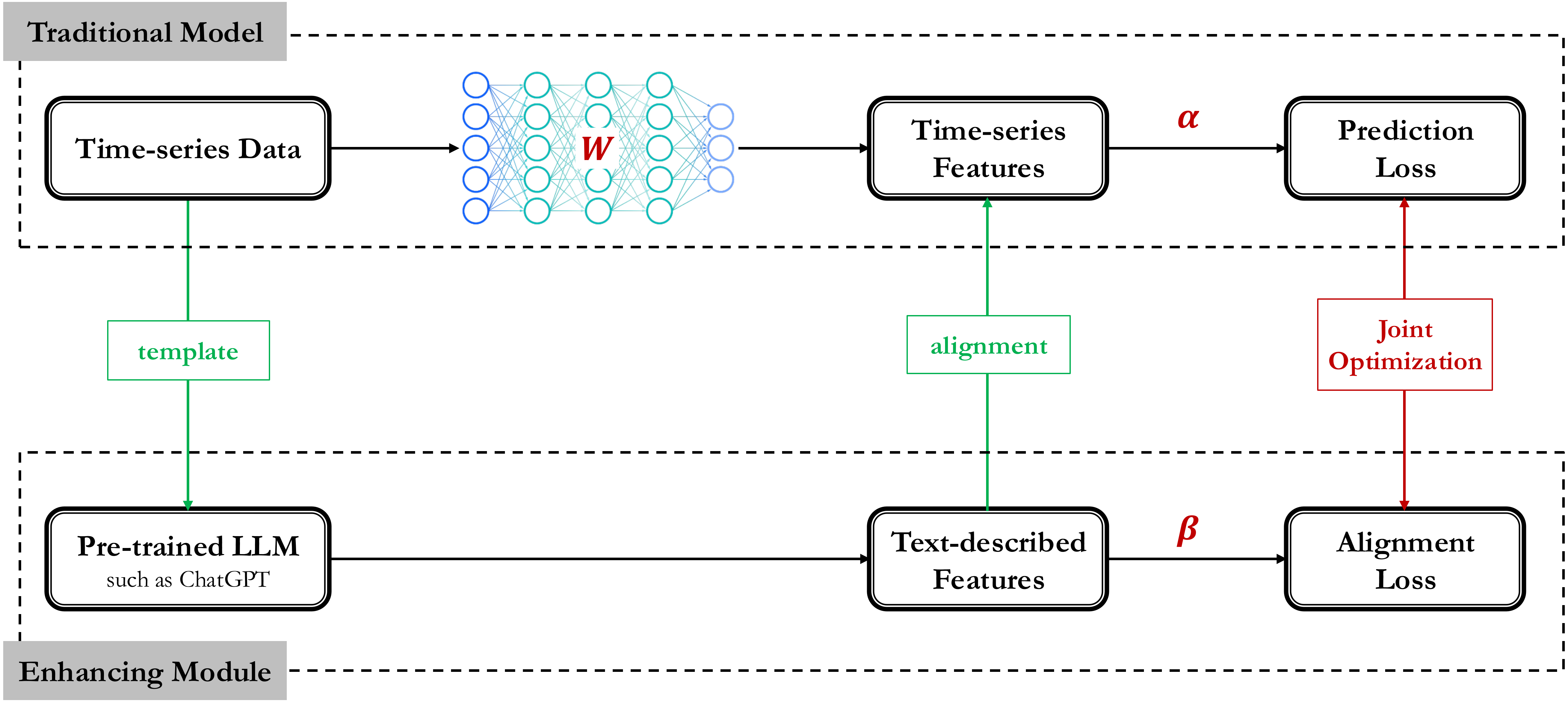


☐ Observation 2: CLIP



[Draft 2024]

c.1) Our method



[Draft 2024]

c.1) Our method

Implementations

- modify a github template as Template
- employ K-complexity or entropy as Alignment
- α and β are two re-weighted vectors, here we implement they using a MLP network
- iteratively optimize the parameters of $\mathbf{W}, \alpha, \beta$

Table 1: Short-term M4 forecasting. The prediction lengths are in [6, 48] and results are obtained by weighting averages across multiple datasets with varying sampling intervals. Full results are in Appendix A.6.

Methods	LLM-TS	TimesNet	GPT4TS	TIME-LLM	TEST	PatchTST	N-HiTS	N-BEATS	FEDformer	Stationary	Autoformer
SMAPE	**11.819**	11.908	11.991	11.983	11.927	12.059	11.927	11.851	12.840	12.780	12.909
MASE	**1.588**	1.612	1.600	1.595	1.613	1.623	1.613	1.599	1.701	1.756	1.771
OWA	**0.851**	0.860	0.861	0.859	0.861	0.869	0.861	0.855	0.918	0.930	0.939

- Short-term forecasting
- Long-term forecasting
- Imputation
- Classification
- Anomaly detection

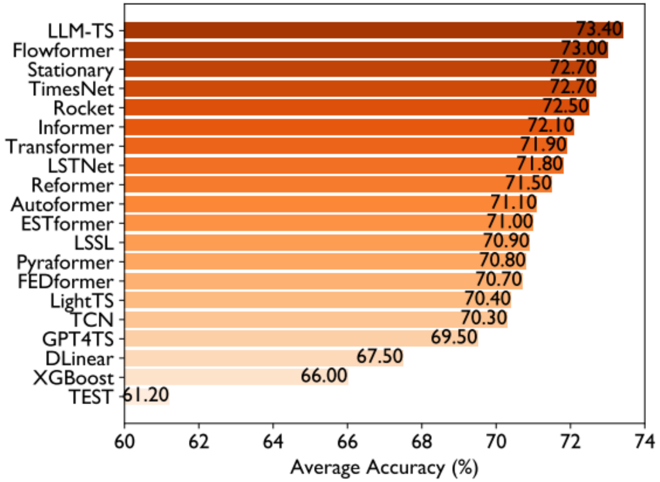


Table 2: Long-term forecasting: Averages over 4 prediction lengths: 24, 36, 48, 60 for ILI, and 96, 192, 336, 720 for others. Full results in Appendix A.7.

Methods	LLM-TS		TimesNet		TIME-LLM		DLinear		PatchTST		GPT4TS		FEDformer		TEST		Stationary		ETSformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MAE	MSE	MAE	MSE
Weather	**0.257**	**0.285**	0.265	0.290	0.279	0.296	0.265	0.317	0.265	**0.285**	0.275	0.292	0.309	0.360	0.291	0.315	0.288	0.314	0.271	0.334
ETTh1	0.454	**0.451**	0.470	0.462	0.474	0.459	0.456	0.452	0.516	0.484	0.473	**0.451**	**0.440**	0.460	**0.440**	0.460	0.570	0.537	0.542	0.510
ETTh2	0.396	0.413	0.413	0.426	0.398	0.415	0.559	0.515	0.391	**0.411**	**0.383**	0.410	0.437	0.449	0.414	0.432	0.526	0.516	0.439	0.452
ETTm1	**0.401**	0.409	0.414	0.418	0.437	0.421	0.403	0.407	0.406	0.407	0.408	**0.400**	0.448	0.452	0.402	0.411	0.481	0.456	0.429	0.425
ETTm2	0.295	**0.331**	0.294	**0.331**	0.298	0.342	0.350	0.401	**0.290**	0.334	**0.290**	0.335	0.305	0.349	0.323	0.359	0.306	0.347	0.293	0.342
ILI	**1.973**	**0.894**	2.266	0.974	2.726	1.098	2.616	1.090	2.184	0.906	5.117	1.650	2.847	1.144	3.324	1.232	2.077	0.914	2.497	1.004
ECL	0.194	0.299	0.198	0.298	0.229	0.315	0.212	0.300	0.216	0.318	0.206	**0.285**	0.214	0.327	0.237	0.324	**0.193**	0.296	0.208	0.323
Traffic	0.618	**0.333**	0.627	0.335	0.606	0.395	0.625	0.383	**0.529**	0.341	0.561	0.373	0.610	0.376	0.581	0.388	0.624	0.340	0.621	0.396
Average	**0.574**	**0.427**	0.618	0.442	0.681	0.468	0.686	0.483	0.600	0.436	0.964	0.525	0.701	0.489	0.756	0.491	0.633	0.465	0.662	0.473

Table 3: Imputation task: Randomly masked $\{12.5\%, 25\%, 37.5\%, 50\%\}$ of points in 96-length series, averaging results over 4 mask ratios. Full results are in Appendix A.8.

Methods	LLM-TS		TimesNet		GPT4TS		PatchTST		LightTS		DLinear		FEDformer		Stationary		Autoformer		Reformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.025**	**0.103**	0.028	0.109	0.028	0.108	0.047	0.140	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150	0.055	0.166
ETTm2	**0.021**	**0.087**	0.022	0.089	0.023	0.088	0.029	0.102	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105	0.157	0.280
ETTh1	0.087	0.198	0.090	0.199	**0.069**	**0.174**	0.115	0.224	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214	0.122	0.245
ETTh2	**0.050**	0.148	0.051	0.150	**0.050**	**0.144**	0.065	0.163	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156	0.234	0.352
ECL	0.094	0.211	0.095	0.212	0.091	0.207	**0.072**	**0.183**	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225	0.200	0.313
Weather	**0.030**	0.056	0.031	0.059	0.032	0.058	0.034	**0.055**	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057	0.038	0.087
Average	0.051	0.134	0.053	0.136	**0.049**	**0.130**	0.060	0.144	0.123	0.228	0.119	0.224	0.112	0.229	0.056	0.142	0.061	0.151	0.134	0.240

Table 4: Anomaly detection task. F1-score (as %) is calculated per dataset. $*$. in the Transformers represents the name of $*$former. Full results are in Appendix A.10.

Methods	LLM-TS	TimesNet	GPT4TS	PatchTS.	ETS.	FED.	LightTS	DLinear	Stationary	Auto.	Pyra.	Anomaly.**	In.	Re.	Trans.
SMD	84.69	84.57	84.32	84.62	83.13	85.08	82.53	77.10	84.72	85.11	83.04	85.49	81.65	75.32	79.56
MSL	81.11	80.34	81.73	78.70	85.03	78.57	78.95	84.88	77.50	79.05	84.86	83.31	84.06	84.40	78.68
SMAP	69.41	69.18	68.86	68.82	69.50	70.76	69.21	69.26	71.09	71.12	71.09	71.18	69.92	70.40	69.70
SWaT	93.23	93.12	92.59	85.72	84.91	93.19	93.33	87.52	79.88	92.74	91.78	83.10	81.43	82.80	80.37
PSM	97.43	97.27	97.34	96.08	91.76	97.23	97.15	93.55	97.29	93.29	82.08	79.40	77.10	73.61	76.07
Average	**85.17**	84.90	84.97	82.79	82.87	84.97	84.23	82.46	82.08	84.26	82.57	80.50	78.83	77.31	76.88

d) Why do we need to train foundational models for time series from scratch?

The limitations of pre-trained LLMs when handling time series:

智谱华章
Knowledge Atlas

The attention module can fit the cosine curves in the training phase, but **FAIL** to predict in the testing phase.

The combination of Thm 0.2 and Thm 0.3 shows that the task of deciding whether approximation of a given pattern is possible or not is **NP-hard** for a fixed $d > 1$.

On the Expressive Flexibility of Self-Attention Matrices

Valerii Likhosherstov[1]*, Krzysztof Choromanski[2]*, Adrian Weller[1,3]
[1]University of Cambridge
[2]Google Brain
[3]The Alan Turing Institute
vl304@cam.ac.uk

d.1) The expressivity of data-driven models

There is a theoretical paradigm of data-driven learning, that is,

the **PAC (Probably Approximately Correct)**,

presented by Leslie Valiant [1984, Turing Award 2021]

$$P(E(h) \leqslant \epsilon) \geqslant 1 - \delta$$

where h denotes a function expressed by a machine learning model, neural network, or foundational model, E is the error, and $\epsilon, \delta \in [0,1]$.

Only using data, the error of learning models always ALWAYS exist or CANNOT vanish.

d.1) The expressivity of data-driven models

$$P(E(h) \leqslant \epsilon) \geqslant 1 - \delta$$

In-depth, the error is caused by the gap between data and distribution, including

① the approximation gap between data and distribution, data noise, and **distribution changing**

② the expressive gap between ground-truth concept and the learning model

③ the error caused by optimization algorithms

Three examples with picture

Content

I. Why do we need foundational models for time series?

a) What is a time series?

b) What is a foundational model?

c) Can we transform a pre-trained LLM for handling time series?

d) Why do we need to train foundational models for time series from scratch?

II. Our Planning

a) Open source of time series

b) Flowchart: Time-Series VS Sequential models

a) Open source of time series

Introduction to the collection of time series

b) Flowchart: Time-series VS sequential models

Introduction to the flowchart

Cooperative Students with LAMDA-1

Jin-Hui Wu

Qin-Cheng Zheng

En-Hao Gao

Wen-Chao Hu

Hao-Yi Lei

Xin-Hao Zhu

Jia-Yang Zhou

Zi-Chen Zhao

Qi-Jie Li

Shy boys with no photo available

Current Students in NJU

Shu-Hao Zhang

Jia-Wei Huang

Shuang Liang

Qian Sun

Mao-Hua Li

Jia-Lei Niu

Thanks!