

# Spatial-Temporal Large Language Model for Time Series Data

*Prof. Dr. Ziyue Li*

W2 Prof. @ Department of Operation & Technology

Department of Computer Science,

Munich Data Science Institute

Technical University of Munich



# About Me:



## Academia

Ph.D., 2021



The Hong Kong University of Science and Technology ([HKUST](#)), Dept of Industrial Engineering and Decision Analytics, Hong KongHK



Arizona State University ([ASU](#)), School of Computing, U.S.us



Visiting and Invited Lecture, 2023  
Georgia Tech, U.S.us



W1 Professor, 2022-2025  
University of Cologne, GermanyDE



W2 Professor, 2025-present  
Technical University of Munich, GermanyDE

## My Research Focus:

How to learn from the real-world, spatiotemporal, noisy data to make faithful, robust, and generalizable decision for smart mobility & cities

## Industry



Research Scientist, 2019

The Bell Labs, Stuttgart DE



Research Project Manager, 2021

HK Mass Transit Railway ([MTR](#))HK



Technology Leader of Tomorrow, 2021

HK Science & Tech ParkHK

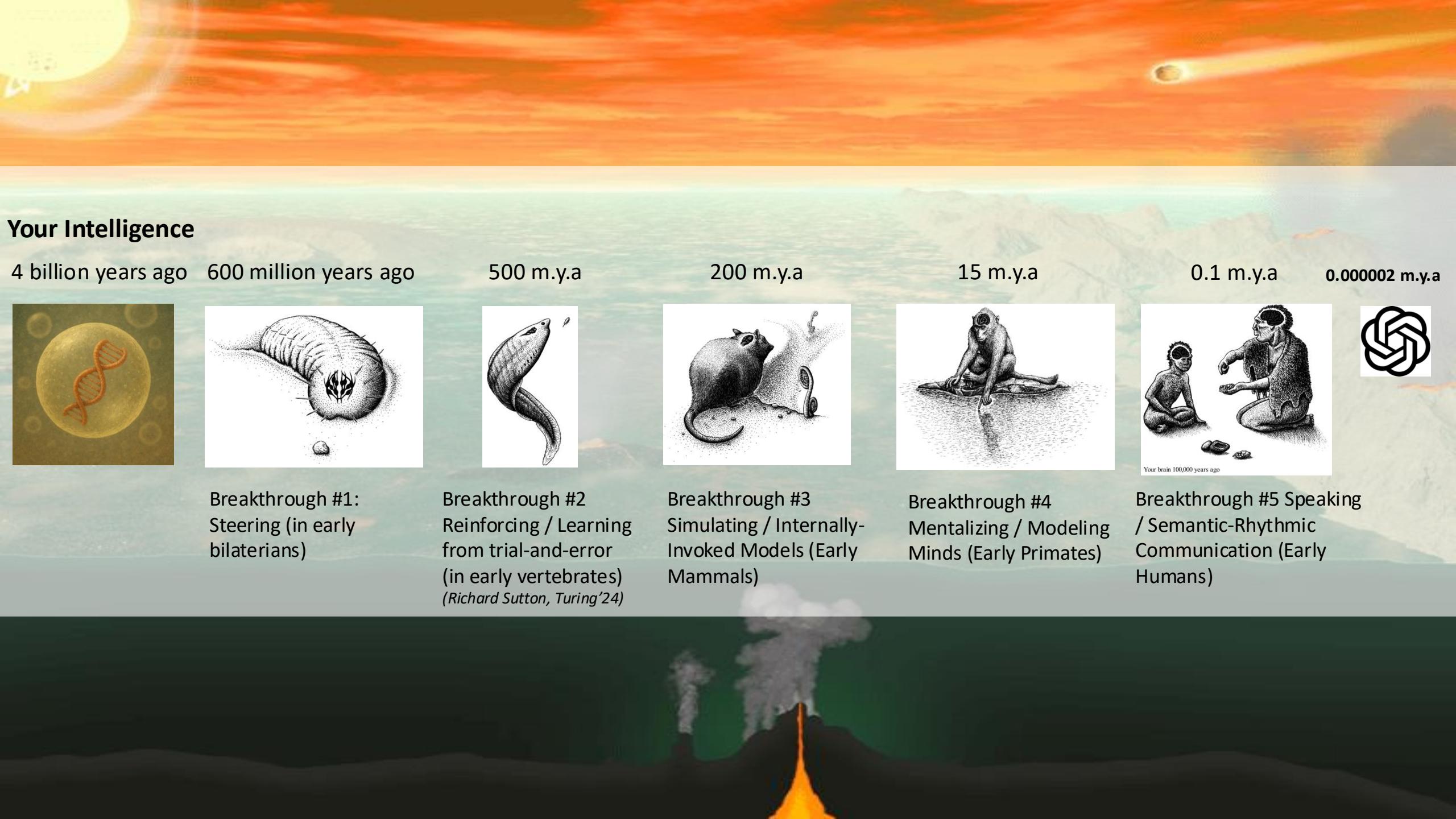
Tech Lead, 2022, of 10-ppl team

CityBrain in S.T. ResearchHKSGUS

I understand deeply both:

- real industry question
- Impactful & meaningful research problem





## Your Intelligence

4 billion years ago 600 million years ago

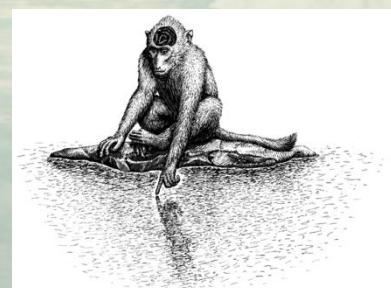
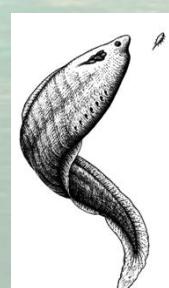
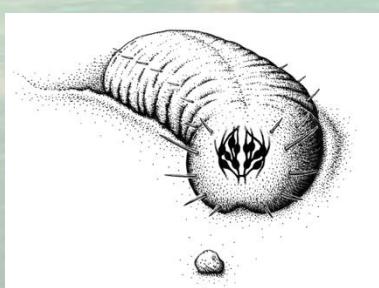
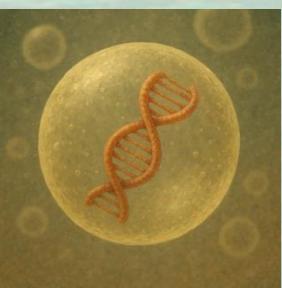
500 m.y.a

200 m.y.a

15 m.y.a

0.1 m.y.a

0.000002 m.y.a



Breakthrough #1:  
Steering (in early  
bilaterians)

Breakthrough #2  
Reinforcing / Learning  
from trial-and-error  
(in early vertebrates)  
*(Richard Sutton, Turing'24)*

Breakthrough #3  
Simulating / Internally-  
Invoked Models (Early  
Mammals)

Breakthrough #4  
Mentalizing / Modeling  
Minds (Early Primates)

Breakthrough #5 Speaking  
/ Semantic-Rhythmic  
Communication (Early  
Humans)

Your brain 100,000 years ago

# Is it possible?

ChatGPT ▾

do a prediction of the next 4 numbers based on the sequence:  
1, 2, 3, 4, 3, 2, 3, 4, 5, 4, 3,



# Is it possible?



To predict the next 4 numbers in the sequence, we need to identify the pattern or rule governing the sequence. Let's break down the sequence step-by-step:

1, 2, 3, 4, 3, 2, 3, 4, 5, 4, 3

Observations:

- The sequence increases from 1 to 4, then decreases to 2, then increases again to 5, and starts decreasing.

Looking at the sequence in parts:

- 1, 2, 3, 4 (increasing)
- 3, 2 (decreasing)
- 3, 4, 5 (increasing)
- 4, 3 (decreasing)

The pattern seems to involve alternating increasing and decreasing segments. To verify, let's list out the segments clearly:

- First increasing: 1, 2, 3, 4
- First decreasing: 4, 3, 2
- Second increasing: 2, 3, 4, 5
- Second decreasing: 5, 4, 3

Following this pattern, after a decreasing segment, we expect an increasing segment starting from the number after the last one in the decreasing segment (which is 3).

Thus, the next segment should increase starting from 3:

- Next number:  $3 + 1 = 4$
- Next number:  $4 + 1 = 5$
- Next number:  $5 + 1 = 6$

So the sequence should be:

1, 2, 3, 4, 3, 2, 3, 4, 5, 4, 3, 4, 5, 6

To add 4 more numbers:

- Continuing the increasing segment: 4, 5, 6
- Next should be decreasing:  $6 - 1 = 5$

Thus, the next 4 numbers are: 4, 5, 6, 5.

To add 4 more numbers:

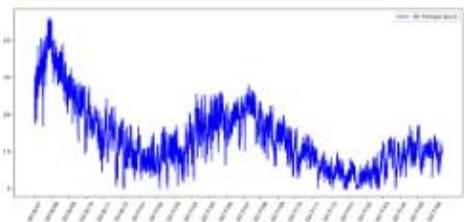
- Continuing the increasing segment: 4, 5, 6
- Next should be decreasing:  $6 - 1 = 5$

Thus, the next 4 numbers are: 4, 5, 6, 5.

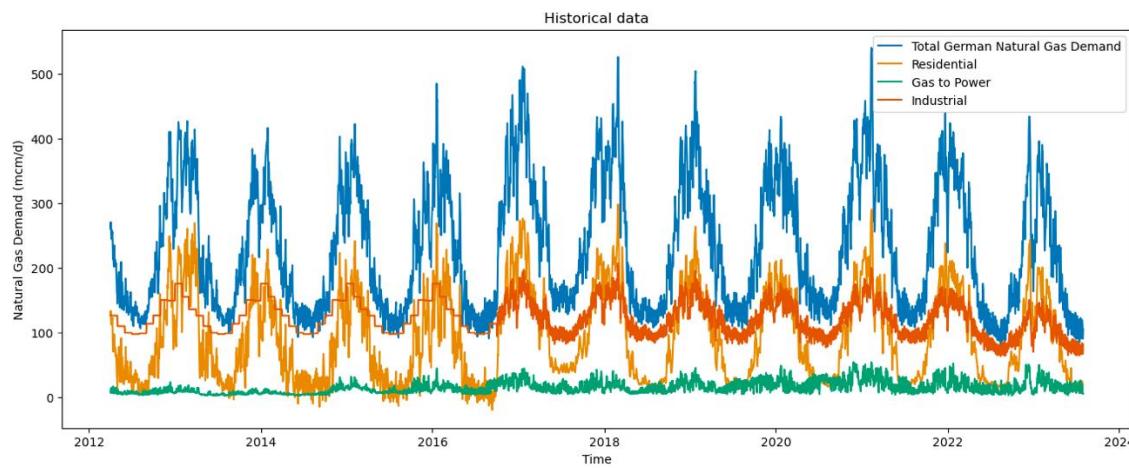


# When the time series data get more complicated

Electricity Transformer Temperature



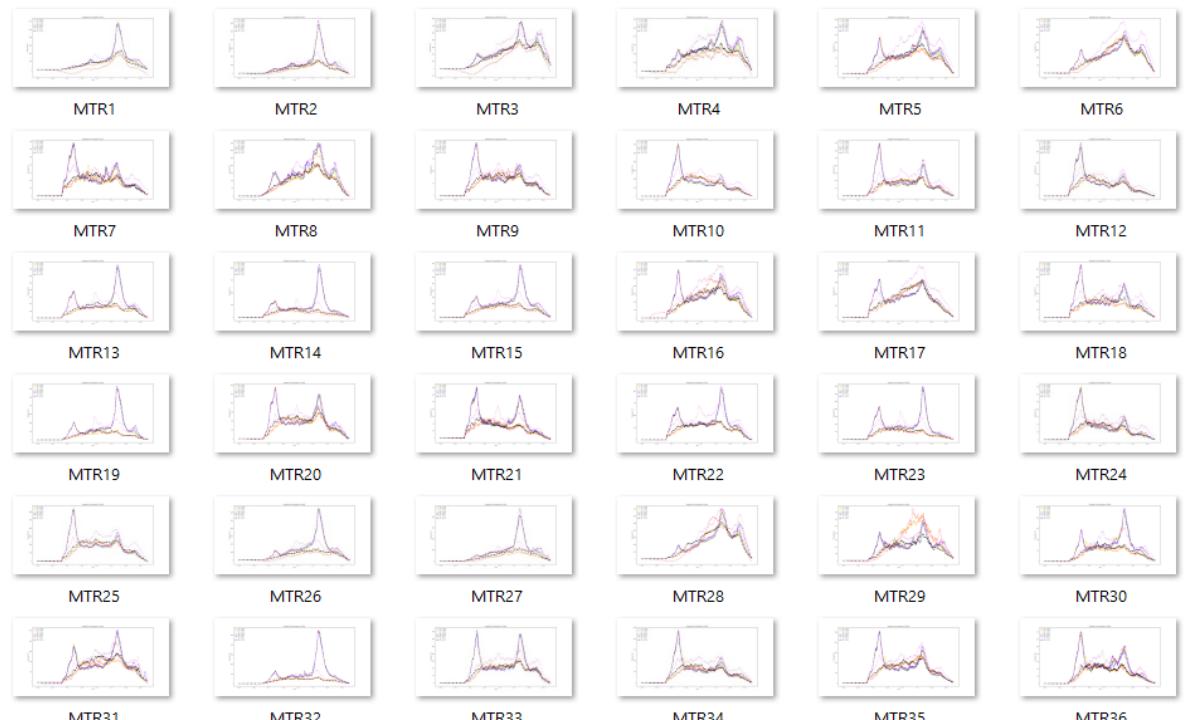
German Energy Data



+ seasonality

+ noise

Hong Kong Metro Data

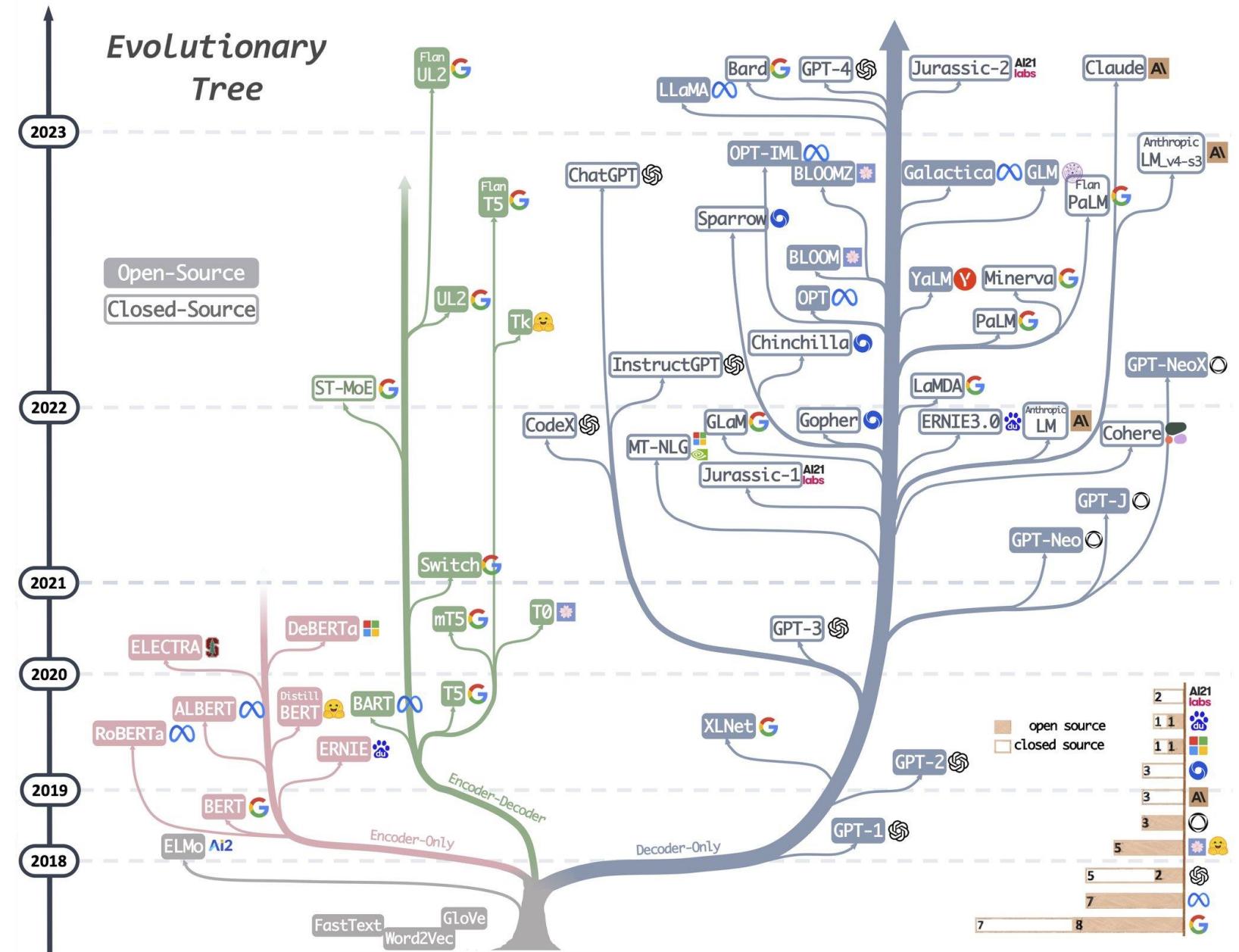


+ spatial relations



# What is LLMs?

Large  
Language  
Model



# Where does the story start?

## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

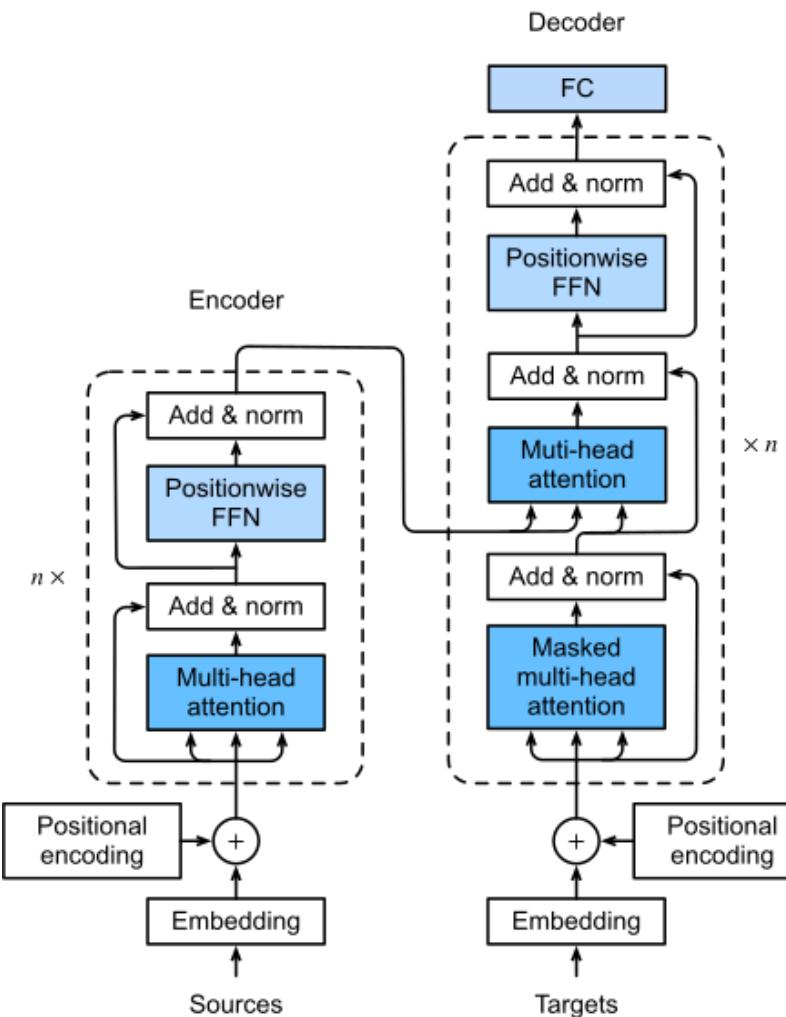
Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Lukasz Kaiser\*  
Google Brain  
lukaszkaiser@google.com

Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

### Abstract

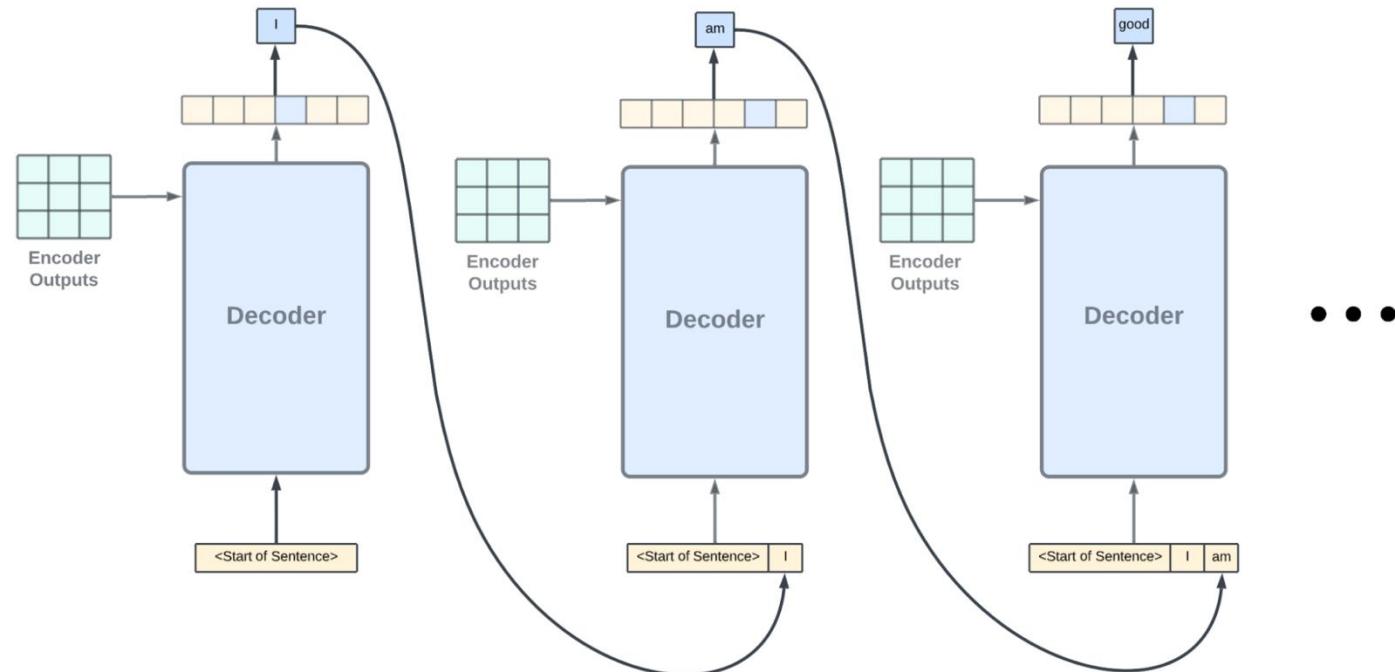
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



"I like Li's presentation"

我喜歡黎的演講."

# If we stack all the Transformer layers together

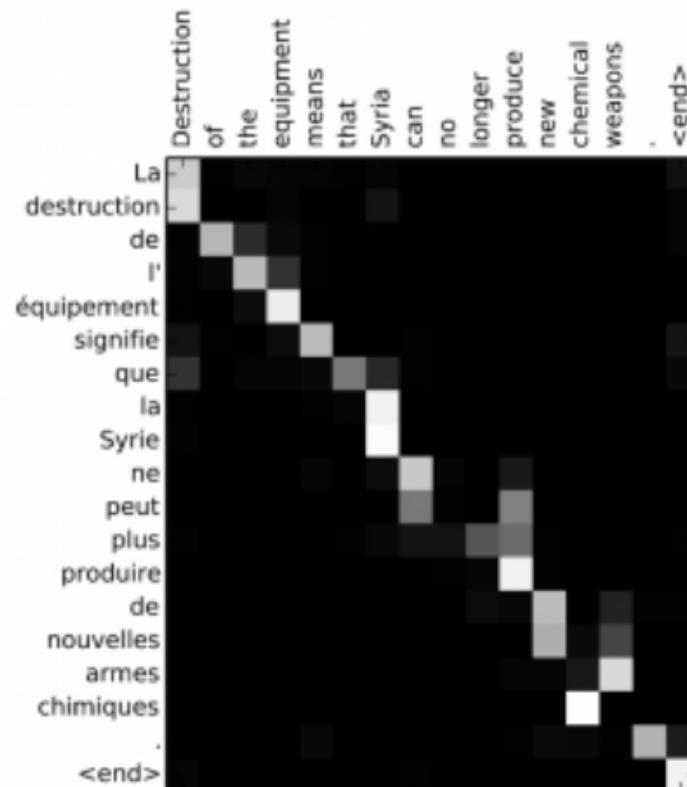


x 48 Transformers: GPT-2 (1.5B)

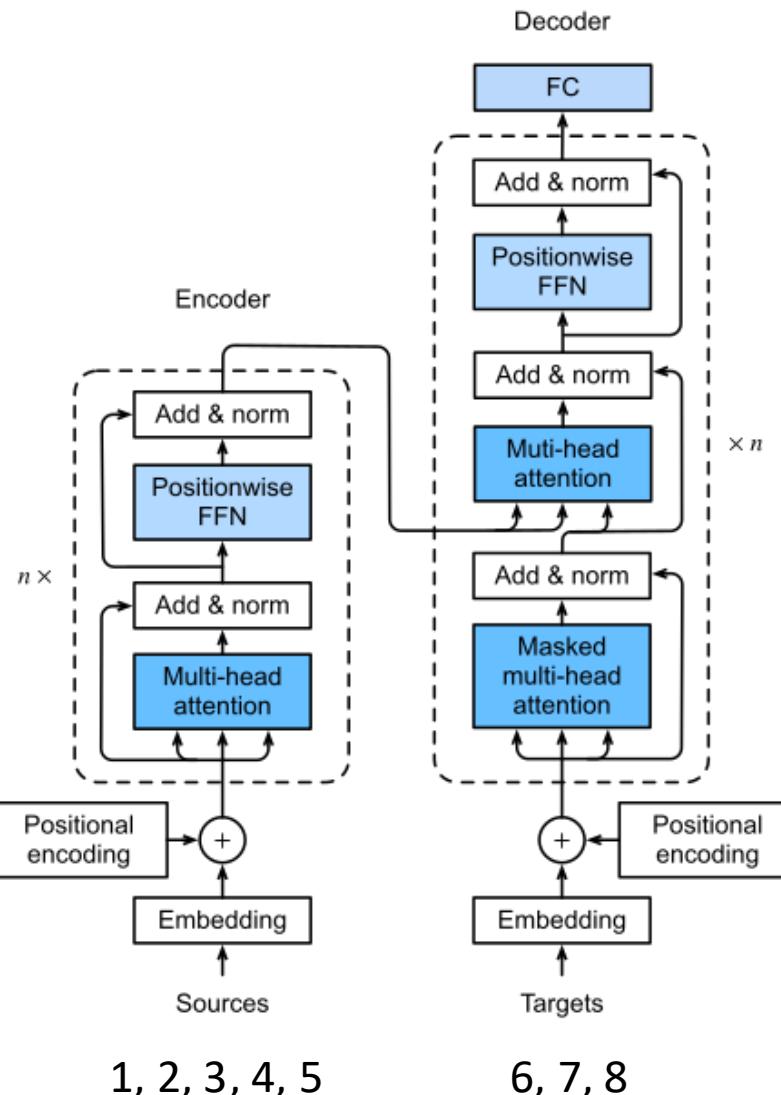
x 96 Transformers: GPT-3.5 (175B)

x ~120 Transformers: GPT-4 (1.8 Trillion)

What is the most important module?  
What is exactly the LLMs' knowledge?



# Can LLM be used in time-series data?



## One Fits All: Power General Time Series Analysis by Pretrained LM

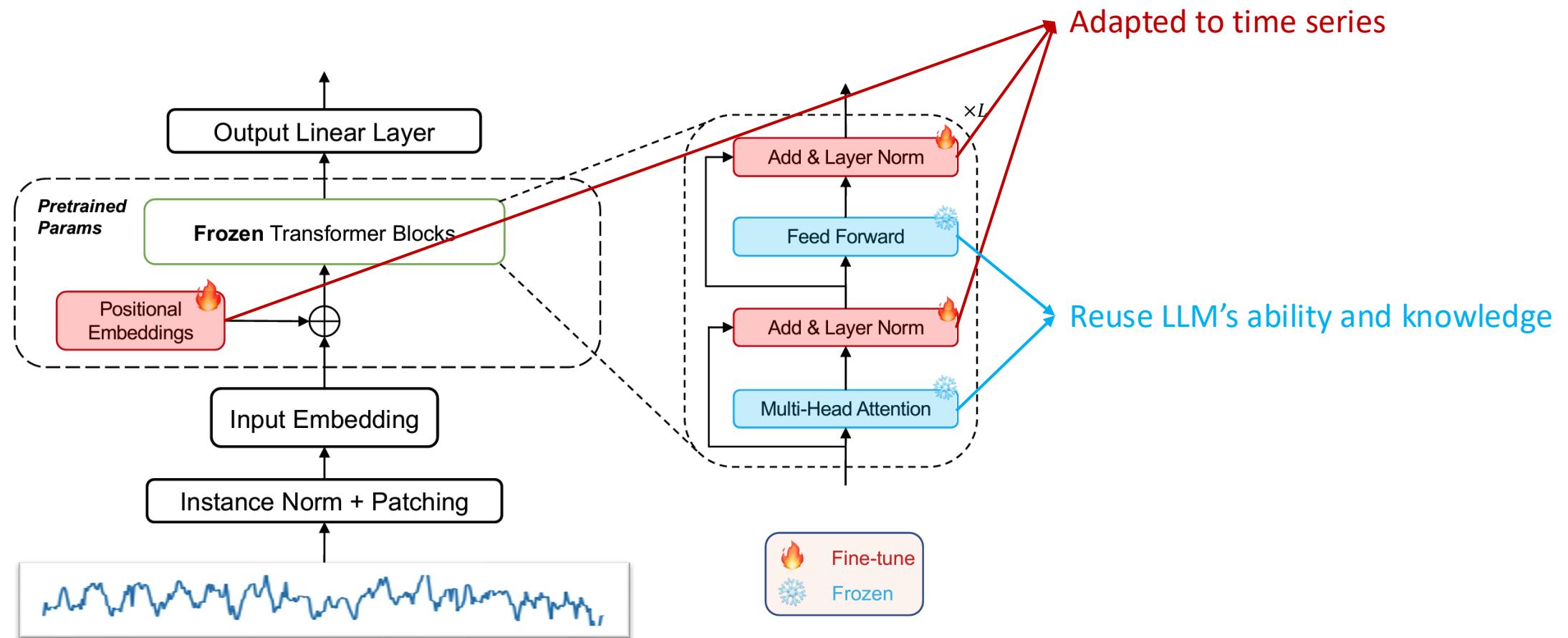
Tian Zhou\* Peisong Niu\* Xue Wang\* Liang Sun Rong Jin<sup>†</sup>  
`{tian.zt,niupeisong.nps,xue.w,liang.sun,jinrong.jr}@alibaba-inc.com`

### Abstract

Although we have witnessed great success of pre-trained models in natural language processing (NLP) and computer vision (CV), limited progress has been made for general time series analysis. Unlike NLP and CV where a unified model can be used to perform different tasks, specially designed approach still dominates in each time series analysis task such as classification, anomaly detection, forecasting, and few-shot learning. The main challenge that blocks the development of pre-trained model for time series analysis is the lack of a large amount of data for training. In this work, we address this challenge by leveraging language or CV models, pre-trained from billions of tokens, for time series analysis. Specifically, we refrain from altering the self-attention and feedforward layers of the residual blocks in the pre-trained language or image model. This model, known as the Frozen Pretrained Transformer (FPT), is evaluated through fine-tuning on all major types of tasks involving time series. Our results demonstrate that pre-trained models on natural language or images can lead to a comparable or state-of-the-art performance in all main time series analysis tasks, as illustrated in Figure 1. We also found both theoretically and empirically that the self-attention module behaviors similarly to principle component analysis (PCA), an observation that helps explains how transformer bridges the domain gap and a crucial step towards understanding the universality of a pre-trained transformer. The code is publicly available at [https://github.com/DAMO-DI-ML/One\\_Fits\\_All](https://github.com/DAMO-DI-ML/One_Fits_All).



# First Try



# Does it work well?

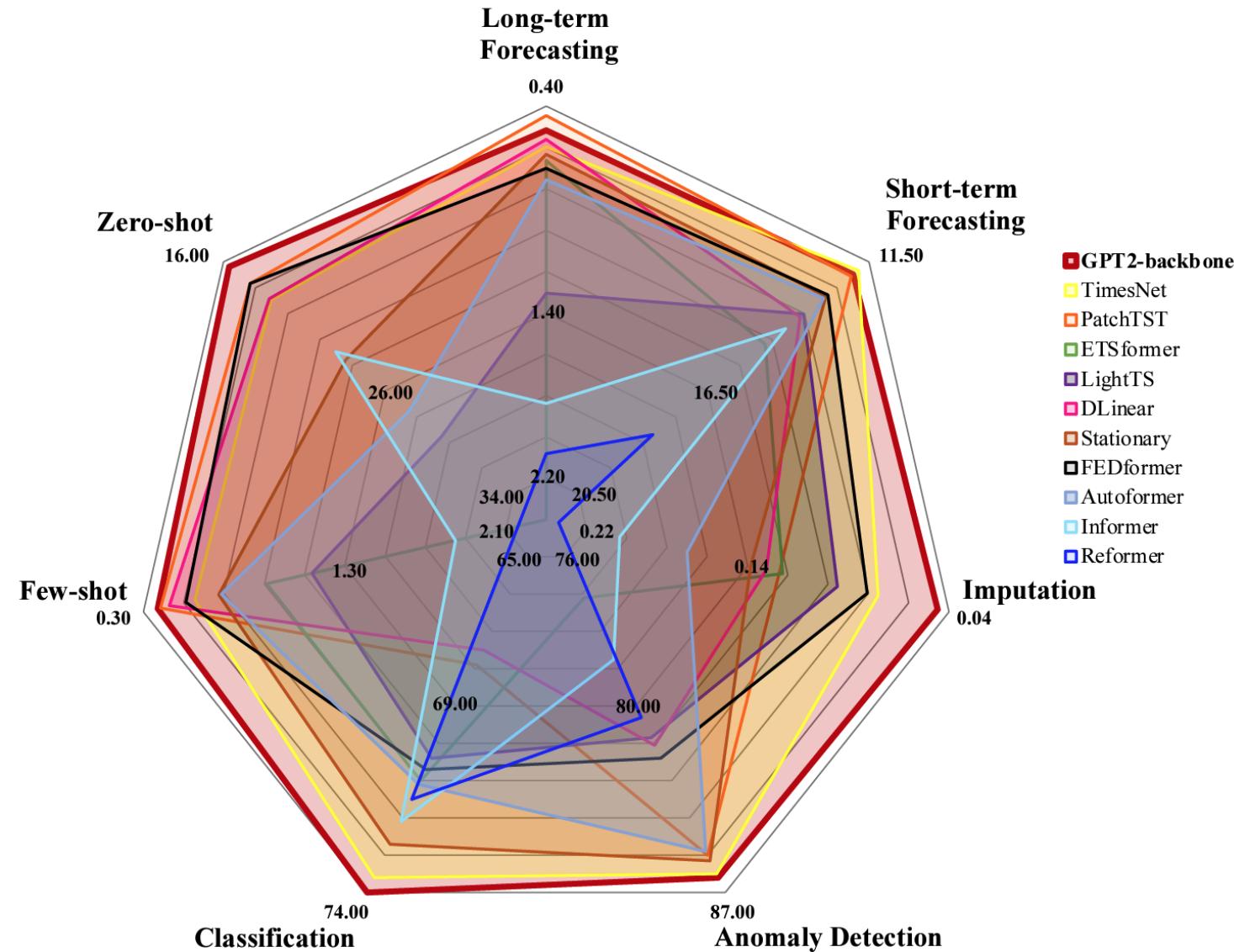


Figure 1: Model performance comparison on various tasks.

# Close look

## Imputation

Table 1: Imputation task. We randomly mask {12.5%, 25%, 37.5%, 50%} time points of 96-length time series. The results are averaged from 4 different mask ratios. **Black**: best, **Red**: second best. Appendix H.8 shows the full results.

Methods	GPT2(3)		TimesNet		PatchTST		ETSformer		LightTS		DLinear		FEDformer		Stationary		Autoformer		Informer		Reformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	<b>0.028</b>	<b>0.105</b>	<b>0.027</b>	<b>0.107</b>	0.047	0.140	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150	0.071	0.188	0.055	0.166
ETTm2	<b>0.021</b>	<b>0.084</b>	<b>0.022</b>	<b>0.088</b>	0.029	0.102	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105	0.156	0.292	0.157	0.280
ETTh1	<b>0.069</b>	<b>0.173</b>	<b>0.078</b>	<b>0.187</b>	0.115	0.224	0.202	0.329	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214	0.161	0.279	0.122	0.245
ETTh2	<b>0.048</b>	<b>0.141</b>	<b>0.049</b>	<b>0.146</b>	0.065	0.163	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156	0.337	0.452	0.234	0.352
ECL	<b>0.090</b>	<b>0.207</b>	0.092	0.210	<b>0.072</b>	<b>0.183</b>	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225	0.222	0.328	0.200	0.313
Weather	<b>0.031</b>	<b>0.056</b>	<b>0.030</b>	<b>0.054</b>	0.034	0.055	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057	0.045	0.104	0.038	0.087
Average	<b>0.047</b>	<b>0.127</b>	<b>0.049</b>	<b>0.132</b>	0.060	0.144	0.197	0.309	0.123	0.228	0.119	0.224	0.112	0.229	0.056	0.142	0.061	0.151	0.165	0.273	0.134	0.240

## Long-term Forecasting

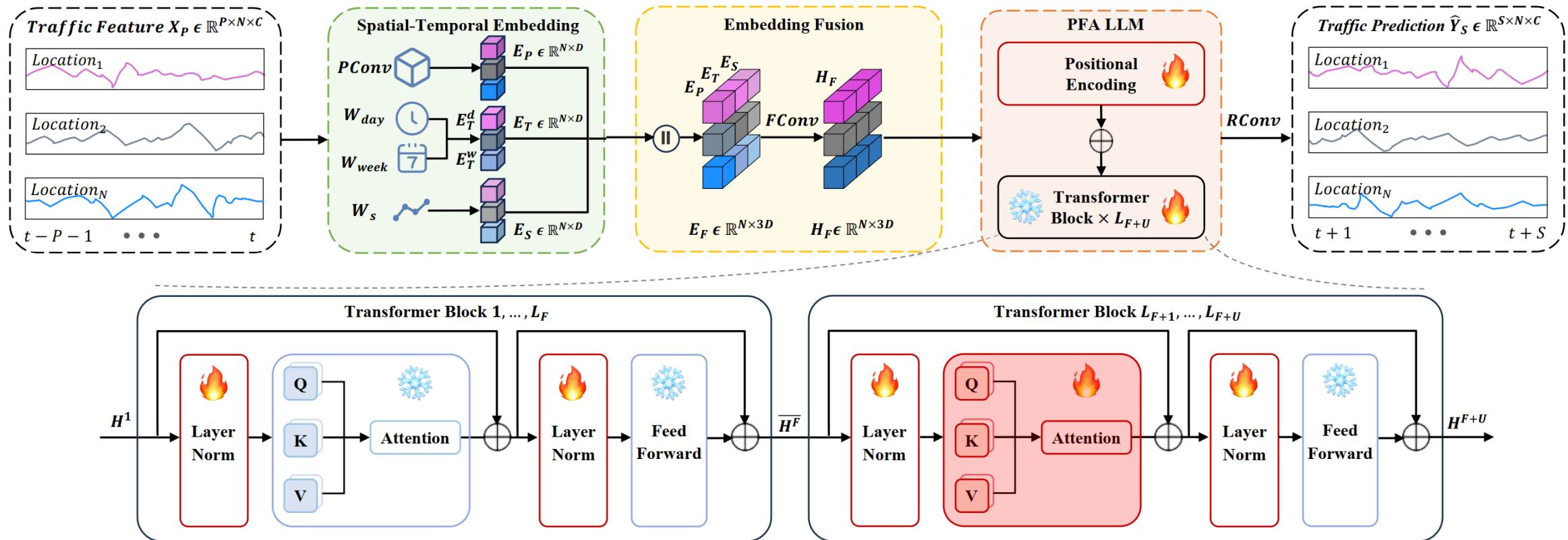
Table 3: Long-term forecasting task. All the results are averaged from 4 different prediction lengths, that is {24, 36, 48, 60} for ILI and {96, 192, 336, 720} for the others. **Black**: best, **Red**: second best. Appendix D.3 shows the full results.

Methods	GPT2(6)		TimesNet		ETSformer		LightTS		DLinear		FEDformer		PatchTST		Stationary		Autoformer		Informer		Reformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	<b>0.237</b>	<b>0.270</b>	0.259	0.287	0.271	0.334	0.261	0.312	0.249	0.300	0.309	0.360	<b>0.225</b>	<b>0.264</b>	0.288	0.314	0.338	0.382	0.634	0.548	0.803	0.656
ETTh1	<b>0.427</b>	<b>0.426</b>	0.458	0.450	0.542	0.510	0.491	0.479	0.423	0.437	0.440	0.460	<b>0.413</b>	<b>0.430</b>	0.570	0.537	0.496	0.487	1.040	0.795	1.029	0.915
ETTh2	<b>0.346</b>	<b>0.394</b>	0.414	0.427	0.439	0.452	0.602	0.543	0.431	0.447	0.437	0.449	<b>0.330</b>	<b>0.379</b>	0.526	0.516	0.450	0.459	4.431	1.729	6.736	2.191
ETTm1	<b>0.352</b>	<b>0.383</b>	0.400	0.406	0.429	0.425	0.435	0.437	0.357	<b>0.378</b>	0.448	0.452	<b>0.351</b>	0.387	0.481	0.456	0.588	0.517	0.961	0.734	0.799	0.671
ETTm2	<b>0.266</b>	<b>0.326</b>	0.291	0.333	0.293	0.342	0.409	0.436	0.267	0.334	0.305	0.349	<b>0.255</b>	<b>0.315</b>	0.306	0.347	0.327	0.371	1.410	0.810	1.479	0.915
ILI	<b>1.925</b>	<b>0.903</b>	2.139	0.931	2.497	1.004	7.382	2.003	2.169	1.041	2.847	1.144	<b>1.443</b>	<b>0.798</b>	2.077	0.914	3.006	1.161	5.137	1.544	4.724	1.445
ECL	0.167	<b>0.263</b>	0.192	0.295	0.208	0.323	0.229	0.329	<b>0.166</b>	<b>0.263</b>	0.214	0.327	<b>0.161</b>	<b>0.253</b>	0.193	0.296	0.227	0.338	0.311	0.397	0.338	0.422
Traffic	<b>0.414</b>	<b>0.294</b>	0.620	0.336	0.621	0.396	0.622	0.392	0.434	0.295	0.610	0.376	<b>0.390</b>	<b>0.264</b>	0.624	0.340	0.628	0.379	0.764	0.416	0.741	0.422
Average	<b>0.516</b>	<b>0.407</b>	0.596	0.433	0.662	0.473	1.303	0.616	0.562	0.436	0.701	0.489	<b>0.446</b>	<b>0.386</b>	0.633	0.465	0.757	0.511	1.836	0.871	2.081	0.954



# How about the spatial time-series (Spatial-Temporal Data)

An easy solution is: add the spatial embedding as the external information

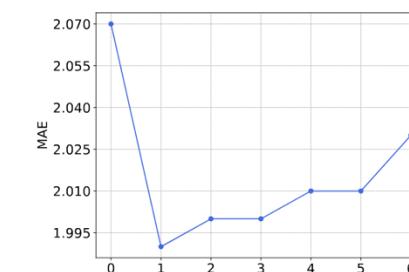
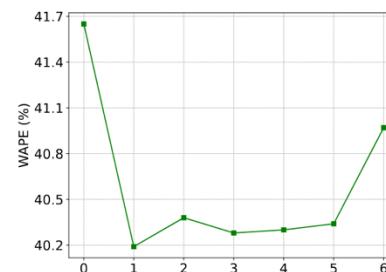
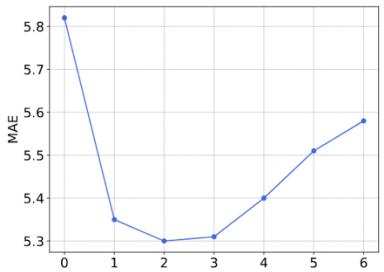
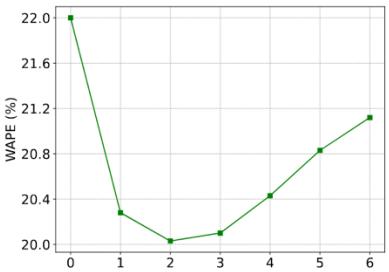


# Results

Table 1: Model comparison on NYCTaxi and CHBike datasets in terms of MAE, RMSE, MAPE (%), and WAPE (%).

Dataset	NYCTaxi Pick-up				NYCTaxi Drop-off				CHBike Pick-up				CHBike Drop-off			
Metric	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE	MAE	RMSE	MAPE	WAPE
DCRNN	5.40	9.71	35.09%	20.43%	5.19	9.63	37.78%	19.82%	2.09	3.30	54.22%	42.26%	1.96	2.94	51.42%	39.61%
STGCN	5.71	10.22	36.51%	21.62%	5.38	9.60	39.12%	20.55%	2.08	3.31	53.63%	42.08%	2.01	3.07	50.45%	40.62%
ASTGCN	7.43	13.84	47.96%	28.04%	6.98	14.70	45.48%	26.60%	2.76	4.45	64.23%	55.71%	2.79	4.20	69.88%	56.49%
GWN	5.43	<b>9.39</b>	37.79%	20.55%	<b>5.03</b>	<b>8.78</b>	35.63%	19.21%	<b>2.04</b>	<b>3.20</b>	<b>53.08%</b>	<b>40.95%</b>	<b>1.95</b>	2.98	50.30%	39.43%
AGCRN	5.79	10.11	40.40%	21.93%	5.45	9.56	40.67%	20.81%	2.16	3.46	56.35%	43.69%	2.06	3.19	51.91%	41.78%
GMAN	5.43	9.47	34.39%	20.42%	5.09	<b>8.95</b>	35.00%	19.33%	2.20	3.35	57.34%	44.06%	2.09	3.00	54.82%	42.00%
ASTGNN	5.90	10.71	40.15%	22.32%	6.28	12.00	49.78%	23.97%	2.37	3.67	60.08%	47.81%	2.24	3.35	57.21%	45.27%
STG-NCDE	6.24	11.25	43.20%	23.46%	5.38	9.74	40.45%	21.37%	2.15	3.97	55.49%	61.38%	2.28	3.42	60.96%	46.06%
DGCRN	5.44	9.82	35.78%	20.58%	5.14	9.39	35.09%	19.64%	2.06	3.21	54.06%	41.51%	1.96	<b>2.93</b>	51.99%	39.70%
OFA	5.82	10.42	36.67%	22.00%	<b>5.60</b>	10.14	37.39%	21.36%	2.06	3.21	53.55%	41.70%	1.96	<b>2.97</b>	49.64%	<b>39.68%</b>
GATGPT	5.92	10.55	37.83%	22.39%	5.66	10.39	37.36%	21.60%	2.07	3.23	52.54%	41.70%	1.95	2.94	49.26%	39.43%
GCNGPT	6.58	12.23	40.19%	24.88%	6.64	12.24	42.46%	25.32%	2.37	3.80	56.24%	47.66%	2.24	3.48	51.05%	45.37%
LLAMA2	5.35	9.48	41.32%	20.27%	5.66	10.74	47.47%	21.63%	2.10	3.37	56.63%	42.49%	1.99	3.03	55.23%	40.28%
ST-LLM	<b>5.29</b>	<b>9.42</b>	<b>33.55%</b>	<b>20.03%</b>	<b>5.07</b>	9.07	<b>33.34%</b>	<b>19.18%</b>	<b>1.99</b>	<b>3.08</b>	53.54%	<b>40.19%</b>	<b>1.89</b>	<b>2.81</b>	<b>49.50%</b>	<b>38.27%</b>

How many layers we need to fine-tune?



It all went well, until...

---

## Are Language Models Actually Useful for Time Series Forecasting?

---

**Mingtian Tan**

University of Virginia

wtd3gz@virginia.edu

**Mike A. Merrill**

University of Washington

mikeam@cs.washington.edu

**Vinayak Gupta**

University of Washington

vinayak@cs.washington.edu

**Tim Althoff**

University of Washington

althoff@cs.washington

**Thomas Hartvigsen**

University of Virginia

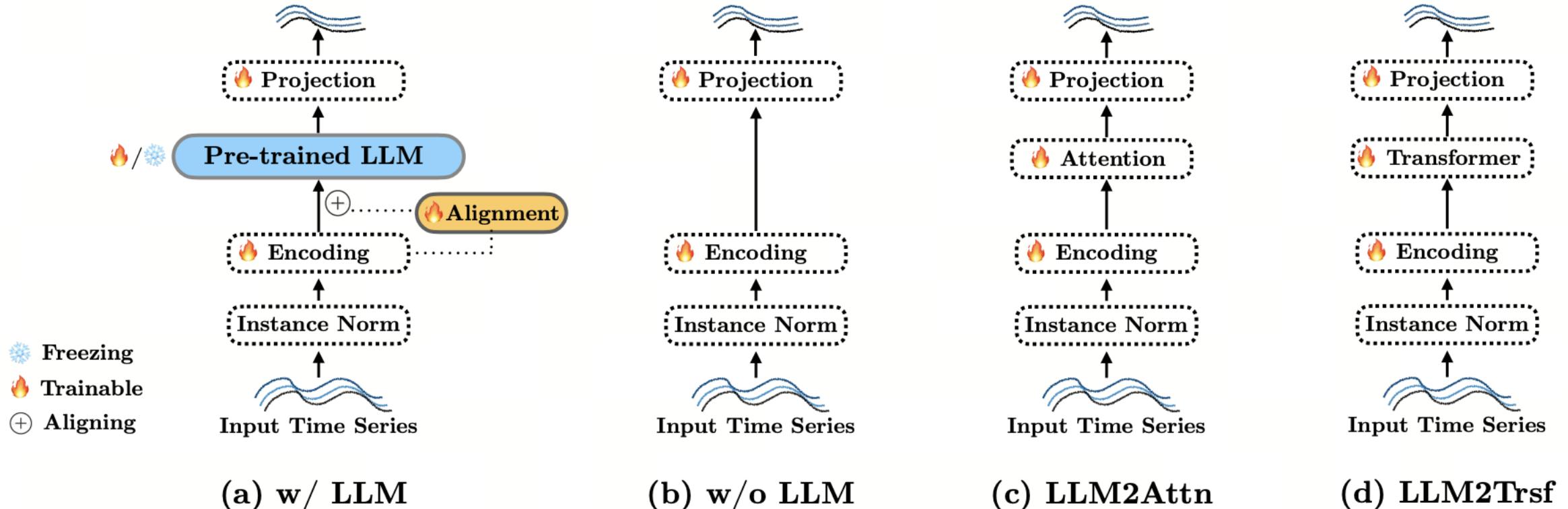
hartvigsen@virginia.edu

### Abstract

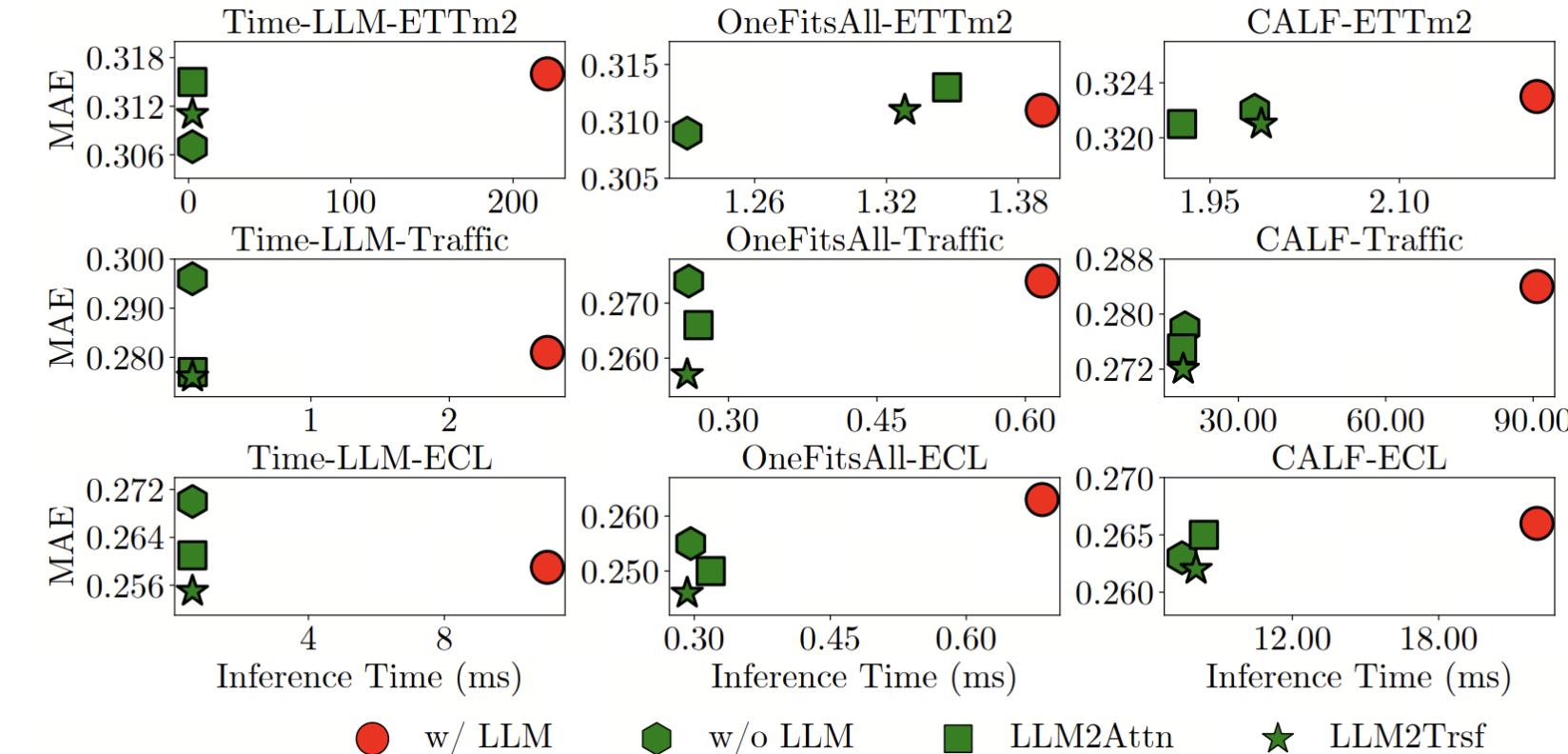
Large language models (LLMs) are being applied to time series forecasting. But are language models actually useful for time series? In a series of ablation studies on three recent and popular LLM-based time series forecasting methods, we find that removing the LLM component or replacing it with a basic attention layer does not degrade forecasting performance—in most cases, the results even improve! We also find that despite their significant computational cost, pretrained LLMs do no better than models trained from scratch, do not represent the sequential dependencies in time series, and do not assist in few-shot settings. Additionally, we explore time series encoders and find that patching and attention structures perform similarly to LLM-based forecasters.<sup>1</sup>



# What they are doing is ablation



# The results:



- **removing or replacing the LLM does not degrade forecasting performance — sometimes it even improves it.**
- The LLM-based methods incur **much higher computational cost** (training time, inference time, parameters) compared to the simpler ablated models.

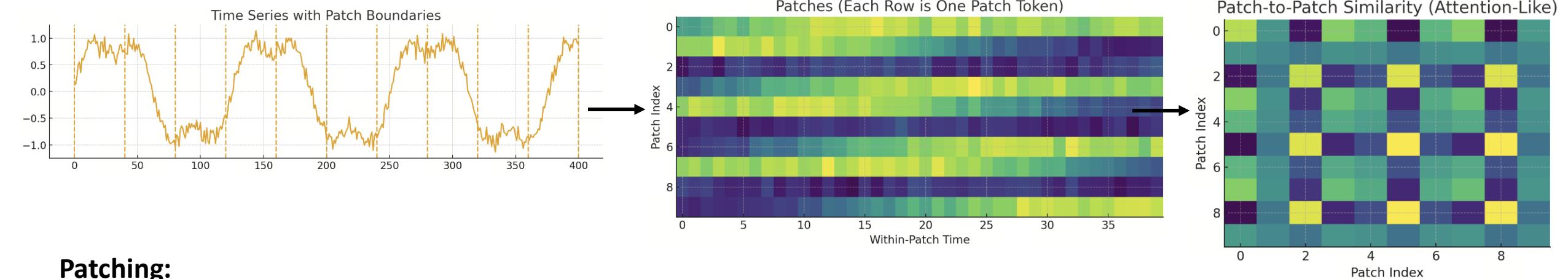
# Where the performance comes from?

Methods	CALF [22]		OneFitAll [50]		Time-LLM [15]		D-LTrsf		LTrsf		PAttn		D-PAtn		DLinear		MeanP		Seasonal		
Metric	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	
ETTh1	96	0.389	0.369	0.397	0.376	0.392	0.362	0.392	0.379	0.390	0.371	0.387	0.383	0.387	0.376	0.399	0.375	0.525	0.753	0.799	1.185
	192	0.423	0.427	0.418	0.416	0.418	0.398	0.421	0.436	0.418	0.428	0.416	0.439	0.415	0.430	0.416	0.405	0.544	0.775	0.788	1.130
	336	0.436	0.456	0.433	0.442	0.427	0.430	0.427	0.447	0.426	0.449	0.419	0.447	0.419	0.440	0.443	0.439	0.548	0.762	0.785	1.099
	720	0.467	0.479	0.456	0.477	0.457	0.442	0.460	0.470	0.465	0.457	0.474	0.455	0.469	0.490	0.472	0.549	0.720	0.789	1.083	
ETTh2	96	0.331	0.279	0.342	0.285	0.328	0.268	0.332	0.281	0.333	0.285	0.327	0.277	0.327	0.276	0.353	0.289	0.378	0.368	1.231	2.777
	192	0.380	0.353	0.389	0.354	0.375	0.329	0.379	0.353	0.376	0.351	0.373	0.345	0.373	0.347	0.418	0.383	0.422	0.444	1.294	2.987
	336	0.394	0.362	0.407	0.373	0.409	0.368	0.382	0.350	0.379	0.346	0.375	0.338	0.376	0.341	0.465	0.448	0.450	0.476	1.313	3.049
	720	0.426	0.404	0.441	0.406	0.420	0.372	0.431	0.411	0.426	0.406	0.425	0.403	0.429	0.408	0.551	0.605	0.456	0.469	1.324	3.073
ETTm1	96	0.349	0.323	0.346	0.292	0.334	0.272	0.347	0.300	0.351	0.300	0.342	0.301	0.346	0.300	0.343	0.299	0.512	0.738	0.752	1.056
	192	0.375	0.374	0.372	0.332	0.358	0.310	0.367	0.335	0.366	0.327	0.366	0.344	0.370	0.343	0.365	0.335	0.518	0.745	0.769	1.075
	336	0.399	0.409	0.394	0.366	0.384	0.352	0.383	0.359	0.386	0.356	0.383	0.369	0.388	0.370	0.386	0.369	0.524	0.752	0.778	1.086
	720	0.438	0.477	0.421	0.417	0.411	0.383	0.414	0.418	0.416	0.421	0.413	0.426	0.418	0.431	0.421	0.425	0.538	0.764	0.782	1.089
ETTm2	96	0.256	0.178	0.262	0.173	0.253	0.161	0.256	0.168	0.258	0.172	0.249	0.164	0.252	0.165	0.260	0.167	0.345	0.312	1.208	2.730
	192	0.297	0.242	0.301	0.229	0.293	0.219	0.294	0.225	0.295	0.226	0.292	0.224	0.291	0.220	0.303	0.224	0.358	0.335	1.285	2.943
	336	0.339	0.307	0.341	0.286	0.392	0.271	0.322	0.268	0.323	0.269	0.319	0.266	0.319	0.263	0.343	0.281	0.376	0.364	1.318	3.036
	720	0.393	0.397	0.401	0.378	0.379	0.352	0.372	0.344	0.372	0.344	0.370	0.345	0.369	0.339	0.421	0.397	0.418	0.433	1.339	3.109
Illness	24	-	-	0.881	2.063	0.727	1.285	0.814	1.602	0.826	1.706	0.800	1.711	0.817	1.685	1.081	2.215	1.393	4.453	1.854	7.099
	36	-	-	0.892	1.868	0.814	1.404	0.895	1.738	0.927	2.300	0.794	1.636	0.853	1.739	0.963	1.963	1.410	4.524	1.897	7.231
	48	-	-	0.884	1.790	0.807	1.523	0.931	1.897	0.992	2.534	0.853	1.767	0.874	1.775	1.024	2.130	1.444	4.819	1.882	7.205
	60	-	-	0.957	1.979	0.857	1.531	0.979	2.049	0.992	2.490	0.829	1.658	0.853	1.695	1.096	2.368	1.492	5.058	1.892	7.096
Weather	96	0.204	0.164	0.212	0.162	0.201	0.147	0.203	0.160	0.204	0.161	0.191	0.151	0.195	0.152	0.237	0.176	0.312	0.297	0.550	0.576
	192	0.250	0.214	0.248	0.204	0.234	0.189	0.242	0.202	0.246	0.205	0.232	0.193	0.236	0.196	0.282	0.220	0.323	0.316	0.577	0.605
	336	0.291	0.269	0.286	0.254	0.279	0.262	0.283	0.253	0.283	0.253	0.273	0.244	0.277	0.247	0.319	0.265	0.341	0.343	0.594	0.625
	720	0.352	0.355	0.337	0.326	0.316	0.304	0.333	0.320	0.332	0.322	0.323	0.311	0.330	0.316	0.362	0.323	0.369	0.384	0.605	0.633
<b>1<sup>st</sup> and 2<sup>nd</sup> Wins</b>		<b>33</b>		<b>3</b>		<b>2</b>		<b>34</b>		<b>21</b>		<b>3</b>		<b>0</b>		<b>0</b>		<b>0</b>			
Traffic	96	0.268	0.407	0.282	0.388	0.248	0.362	0.264	0.383	0.234	0.357	0.262	0.394	0.262	0.394	0.282	0.410	0.371	1.441	0.884	1.765
	192	0.278	0.430	0.290	0.407	0.247	0.374	0.275	0.399	0.243	0.377	0.265	0.404	0.266	0.404	0.287	0.423	0.374	1.455	0.844	1.628
	336	0.281	0.444	0.294	0.412	0.271	0.385	0.284	0.411	0.250	0.388	0.270	0.413	0.269	0.412	0.296	0.436	0.376	1.469	0.827	1.574
	720	0.300	0.477	0.312	0.450	0.288	0.430	0.309	0.453	0.273	0.428	0.289	0.449	0.315	0.466	0.346	0.466	0.378	1.488	0.816	1.543
Electricity	96	0.238	0.145	0.238	0.139	0.224	0.131	0.226	0.132	0.220	0.130	0.226	0.134	0.227	0.134	0.237	0.140	0.738	0.944	0.891	1.201
	192	0.252	0.161	0.251	0.153	0.241	0.152	0.247	0.154	0.239	0.149	0.239	0.149	0.241	0.149	0.249	0.153	0.742	0.953	0.861	1.107
	336	0.267	0.175	0.266	0.169	0.248	0.160	0.265	0.171	0.254	0.164	0.257	0.166	0.257	0.166	0.267	0.169	0.747	0.966	0.847	1.064
	720	0.303	0.222	0.297	0.206	0.298	0.192	0.299	0.211	0.278	0.193	0.288	0.204	0.289	0.204	0.301	0.203	0.762	0.998	0.838	1.031
<b>1<sup>st</sup> and 2<sup>nd</sup> Wins</b>		<b>11</b>		<b>0</b>		<b>16</b>		<b>4</b>		<b>1</b>		<b>0</b>		<b>0</b>		<b>0</b>		<b>0</b>			

Table 18: Comparision between simple methods and original results from reference paper [22, 50, 15]. Red: Best performance. Blue: Second Best.



# What is so special?



## Patching:

- Captures local temporal structure efficiently
- Reduces sequence length → attention becomes feasible
- Channel independence (treating each variable separately)
- Patching + normalization stabilizes learning

- one-layer attention is enough**
- the model only needs to weigh these patch relationships, not understand deep language structure



# My Words?

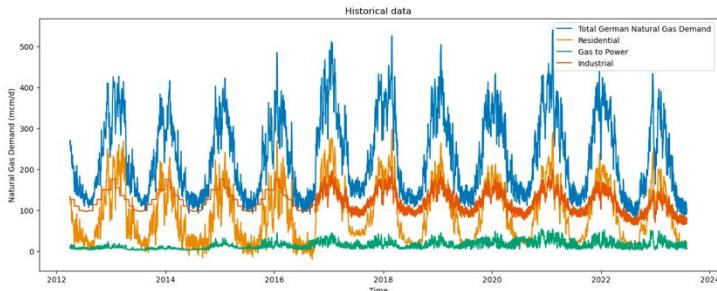
## Don't be sad!

They note that their findings should not be over-interpreted to mean “LLMs are useless for time series” — rather: “(Vanilla) LLMs are *not obviously better* for standard forecasting tasks and thus their use in that domain should be more critically considered.”

**The community should focus LLMs more on tasks that leverage their strengths, e.g., time series + language** (multimodal), reasoning about time series (rather than pure forecasting), anomaly detection with textual context, etc.



# + Text



Energy Price



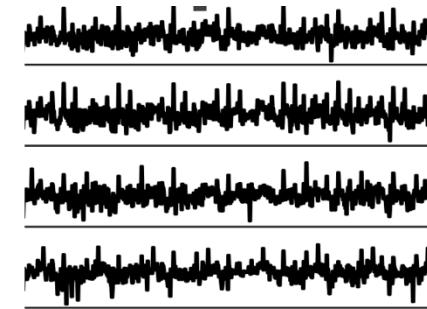
Clean Energy Wire @cleanenergywire · 2h  
Germany wants EU push to end last 20% of Russia energy imports

- Berlin, Czech Republic call for new high-level working group
- EU has seen imports of Russian LNG increase since Ukraine war

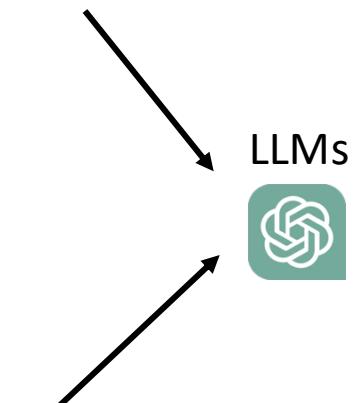


From bloomberg.com

News



ECG signals



Clinical History:  
.....

Report

# How to align the two modalities?

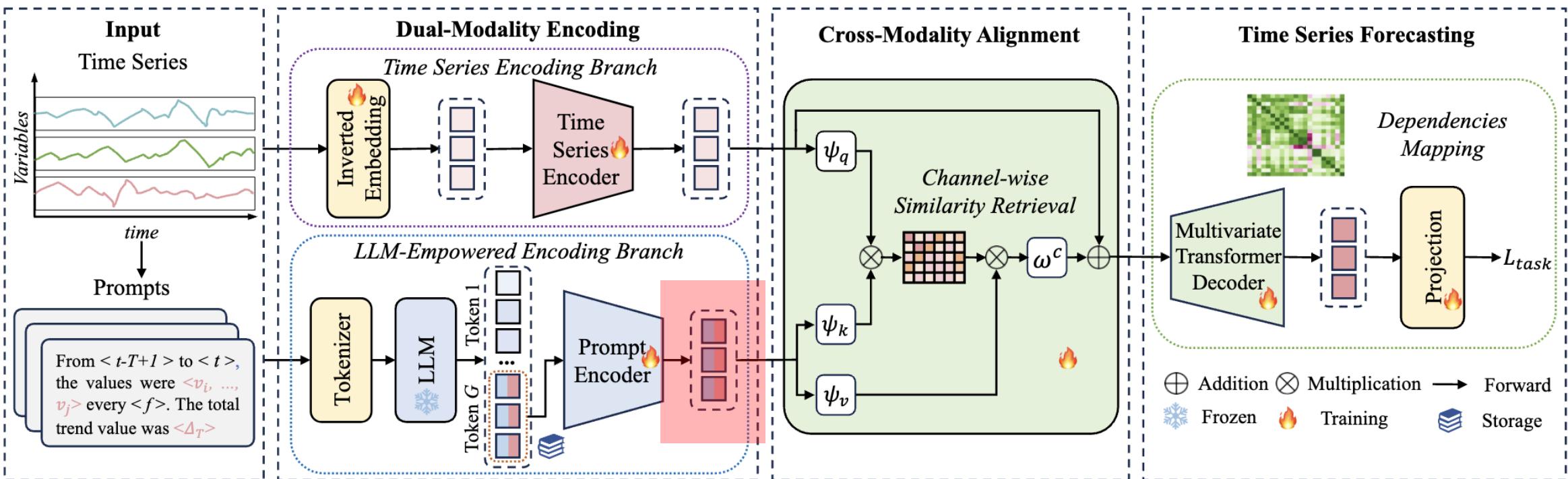


Figure 2: TimeCMA framework. Given time series data and corresponding prompts, we embed them through **Dual-Modality Encoding**. Last token  $G$  of the prompts is tailored, and its embeddings are stored for efficient inference. During **Cross-Modality Alignment**, LLM-empowered time series embeddings are retrieved from the prompt embeddings based on channel-wise similarity. **Time Series Forecasting** decodes the aligned embeddings while capturing multivariate dependencies for robust forecasting.



# How to design the prompt?

From  $t-T+1$  to  $t$ , the values were  $\langle v_i, \dots, v_j \rangle$  every  $f$

**Prompt 1:** capture frequency, e.g. “hour”

From  $t-T+1$  to  $t$ , the values were  $\langle v_i, \dots, v_j \rangle$  every  $f$ . Predict the next few  $M$

**Prompt 2:** give prediction steps, “days”

From  $t-T+1$  to  $t$ , the values were  $\langle v_i, \dots, v_j \rangle$  every  $f$ . The average value was  $\bar{v}_{:n}$

**Prompt 3:** calculate average, e.g., 23

From  $t-T+1$  to  $t$ , the values were  $\langle v_i, \dots, v_j \rangle$  every  $f$ . The total number of historical hours:  $T$

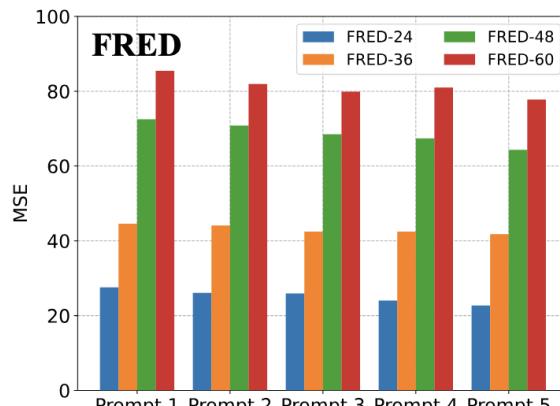
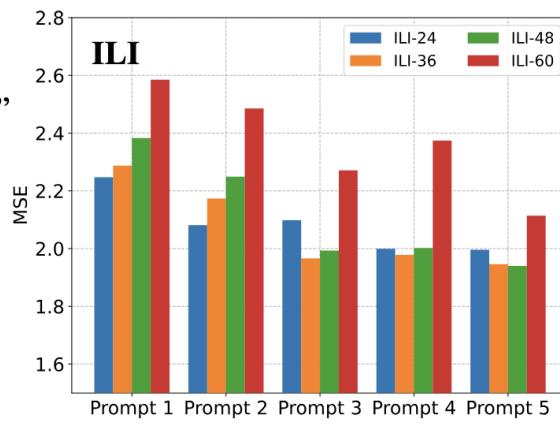
**Prompt 4:** count history length, e.g., 96

From  $t_1$  to  $t_r$ , the values were  $\langle v_i, \dots, v_j \rangle$  every  $<\text{day}>$ .

The total trend value was  $\Delta_T$

**Prompt 5:** summarize the trend, e.g. 2

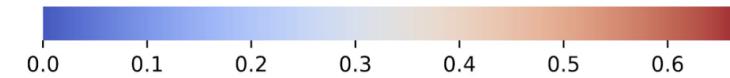
(a) Five prompt designs



(b) Results in ILI & FRED

Figure 4: Five prompts with different purposes to trigger the last token, time-series related token in pink: numerical-ended prompts (3,4,5) are better; prompt 5 abstracting time series trends is the best.

From  $t_1$  to  $t_L$ , the values were  $\langle v_i, \dots, v_j \rangle$  every hour. The total trend value was  $\Delta_L$



(a) ETTh1.

From  $t_1$  to  $t_L$ , the values were  $\langle v_i, \dots, v_j \rangle$  every hour. The total trend value was  $\Delta_L$



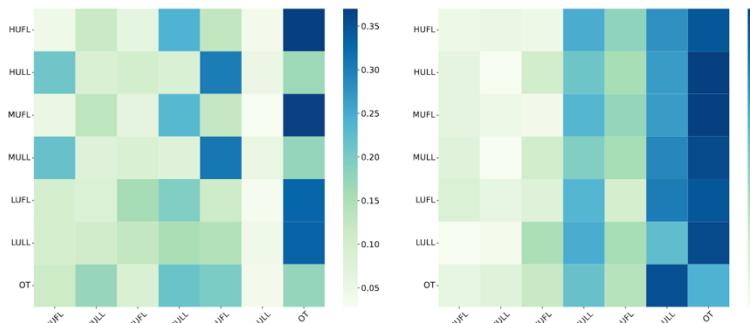
(b) ETTh2.

Figure 5: Last token attention: last token effectively encapsulate temporal information in the prompt.

Dataset	ETTm1 - 96			ETTm2 - 96			
	Metric	Param.	Mem.	Speed	Param.	Mem.	Speed
Time-LLM	44.66	28,882	1.08	44.95	29,140	1.08	
UniTime	108.54	4,168	0.39	108.54	4,168	0.39	
OFA	<b>1.75</b>	914	0.18	<b>1.74</b>	914	0.17	
TimeCMA	17.99	<b>821</b>	<b>0.09</b>	17.99	<b>818</b>	<b>0.08</b>	

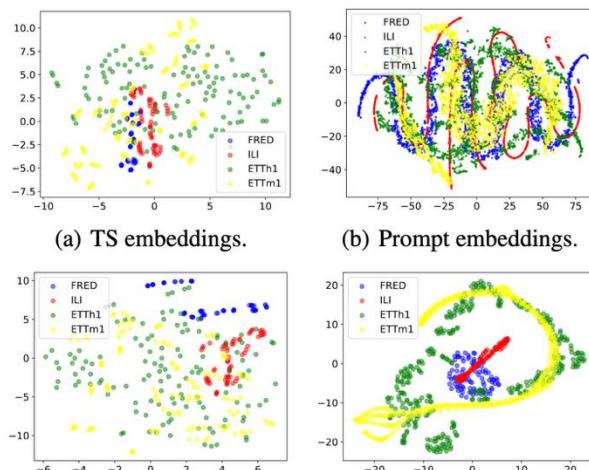
Table 2: Efficiency analysis of LLM-based baselines.

# The Best Performance



(a) Time series Transf. encoder (b) Prompt encoder from LLM

Figure 6: Attention map of Transformer and LLM encoders, capturing complementary variables' pairwise relations: (a) Transformer captures local and individualized relations for each variable; (b) LLM captures global and shared relations for all multi-variables.



(c) Retrieved embeddings. (d) Forecasted TS embeddings.

Figure 7: T-SNE visualization.

Models	TimeCMA		Time-LLM		UniTime		OFA		iTransformer		PatchTST		TimesNet		Dlinear		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTm1	96	<b>0.312</b>	<b>0.351</b>	0.359	0.381	0.322	0.363	0.335	0.369	0.334	0.368	0.344	0.373	0.338	0.375	0.345	0.372
	192	<b>0.361</b>	<b>0.378</b>	0.383	0.393	<b>0.366</b>	<b>0.387</b>	0.374	0.385	0.377	0.391	0.367	0.386	0.374	0.387	0.380	0.389
	336	<b>0.392</b>	<b>0.401</b>	0.416	0.414	<b>0.398</b>	<b>0.407</b>	0.407	0.406	0.426	0.420	<b>0.392</b>	0.407	0.410	0.411	0.413	0.413
	720	0.453	<b>0.438</b>	0.483	0.449	<b>0.454</b>	<b>0.440</b>	0.469	0.442	0.491	0.459	0.464	0.442	0.478	0.450	0.474	0.453
	Avg	<b>0.380</b>	<b>0.392</b>	0.410	0.409	<b>0.385</b>	<b>0.399</b>	0.396	0.401	0.407	0.410	0.392	0.402	0.400	0.406	0.403	0.407
ETTm2	96	<b>0.173</b>	<b>0.258</b>	0.193	0.280	0.183	0.266	0.190	0.275	0.180	0.264	<b>0.177</b>	<b>0.260</b>	0.187	0.267	0.193	0.292
	192	<b>0.238</b>	<b>0.301</b>	0.257	0.318	0.251	0.310	0.253	0.313	0.250	0.309	<b>0.246</b>	<b>0.305</b>	0.249	0.309	0.284	0.362
	336	<b>0.297</b>	<b>0.338</b>	0.317	0.353	0.319	0.351	0.321	0.360	0.311	0.348	<b>0.305</b>	<b>0.343</b>	0.321	0.351	0.369	0.427
	720	<b>0.393</b>	<b>0.394</b>	0.419	0.411	0.420	0.410	0.411	0.406	0.412	0.407	<b>0.410</b>	<b>0.405</b>	0.408	0.403	0.554	0.522
	Avg	<b>0.275</b>	<b>0.323</b>	0.296	0.340	0.293	0.334	0.294	0.339	0.288	0.332	<b>0.285</b>	<b>0.328</b>	0.291	0.333	0.350	0.401
ETTh1	96	<b>0.373</b>	<b>0.391</b>	0.398	0.410	0.397	0.418	0.398	0.424	0.386	0.405	0.404	0.413	0.384	0.402	0.386	0.400
	192	<b>0.427</b>	<b>0.421</b>	0.451	0.440	0.434	0.439	0.449	0.427	0.441	0.436	0.454	0.430	0.434	0.429	0.437	0.432
	336	<b>0.458</b>	<b>0.448</b>	0.473	0.451	<b>0.468</b>	<b>0.457</b>	0.492	<b>0.466</b>	0.487	0.458	0.497	0.462	0.491	0.469	0.481	0.459
	720	<b>0.449</b>	<b>0.460</b>	0.469	0.470	0.469	0.477	0.487	0.483	0.503	0.491	0.496	0.481	0.521	0.500	0.519	0.516
	Avg	<b>0.423</b>	<b>0.431</b>	0.448	0.443	0.442	0.448	0.457	0.450	0.454	0.447	0.463	0.449	0.458	0.450	0.456	0.452
ETTh2	96	<b>0.286</b>	<b>0.336</b>	0.295	0.345	0.296	0.345	0.312	0.360	0.297	0.349	0.312	0.358	0.340	0.374	0.333	0.387
	192	<b>0.363</b>	<b>0.387</b>	0.386	0.399	0.374	<b>0.394</b>	0.387	0.405	0.380	0.400	0.397	0.408	0.402	0.414	0.477	0.476
	336	<b>0.406</b>	<b>0.421</b>	0.419	0.429	<b>0.415</b>	<b>0.427</b>	0.424	0.437	0.428	0.432	0.435	0.440	0.452	0.452	0.594	0.541
	720	<b>0.417</b>	<b>0.438</b>	0.425	0.442	<b>0.425</b>	<b>0.444</b>	0.433	0.453	0.427	0.445	0.436	0.449	0.462	0.468	0.831	0.657
	Avg	<b>0.372</b>	<b>0.397</b>	0.381	0.404	<b>0.378</b>	<b>0.403</b>	0.389	0.414	0.383	0.407	0.395	0.414	0.414	0.427	0.559	0.515
ECL	96	<b>0.143</b>	<b>0.238</b>	0.172	0.265	0.196	0.287	0.197	0.290	<b>0.148</b>	<b>0.240</b>	0.186	0.269	0.168	0.272	0.197	0.282
	192	<b>0.161</b>	<b>0.259</b>	0.182	0.279	0.199	0.291	0.201	0.292	<b>0.162</b>	<b>0.253</b>	0.190	0.273	0.184	0.289	0.196	0.285
	336	<b>0.169</b>	<b>0.261</b>	0.195	0.288	0.214	0.305	0.217	0.309	<b>0.178</b>	<b>0.269</b>	0.206	0.290	0.198	0.300	0.209	0.301
	720	<b>0.219</b>	<b>0.315</b>	0.233	0.320	0.254	0.335	0.253	0.339	<b>0.225</b>	<b>0.317</b>	0.247	0.322	0.220	0.320	0.245	0.333
	Avg	<b>0.174</b>	<b>0.269</b>	0.195	0.288	0.216	0.306	0.217	0.308	<b>0.178</b>	<b>0.270</b>	0.207	0.289	0.192	0.295	0.212	0.300
FRED	24	<b>22.702</b>	<b>0.864</b>	27.285	0.875	31.178	0.931	28.317	0.947	28.017	0.893	35.777	1.014	43.268	1.266	37.898	1.070
	36	<b>40.880</b>	<b>1.157</b>	48.730	1.172	54.172	1.223	59.520	1.306	50.837	1.274	61.034	1.345	69.514	1.533	71.047	1.477
	48	<b>60.045</b>	<b>1.352</b>	73.494	1.460	83.836	1.518	74.808	1.516	78.018	1.793	93.482	1.667	89.913	1.742	118.579	2.002
	60	<b>65.015</b>	<b>1.509</b>	108.221	1.758	118.429	1.830	83.613	1.641	90.212	1.693	133.444	2.011	116.187	1.976	156.844	2.221
	Avg	<b>48.161</b>	<b>1.221</b>	64.433	1.316	71.901	1.376	61.565	1.353	61.771	1.413	80.934	1.509	79.721	1.629	96.092	1.693
ILI	24	<b>1.996</b>	0.998	2.383	1.004	2.346	0.954	2.732	1.100	2.347	1.731	2.335	0.989	<b>2.317</b>	<b>0.934</b>	2.398	1.040
	36	<b>1.906</b>	<b>0.915</b>	2.390	0.993	1.998	0.912	2.664	1.063	2.468	0.998	2.561	1.035	<b>1.972</b>	0.920	2.646	1.088
	48	<b>1.867</b>	<b>0.868</b>	2.394	1.003	1.979	<b>0.912</b>	2.617	1.041	2.489	1.016	2.465	1.022	<b>2.238</b>	0.913	2.614	1.086
	60	<b>1.920</b>	<b>0.904</b>	2.562	1.049	2.109	0.938	2.478	1.035	2.471	1.065	2.189	0.997	<b>2.027</b>	0.928	2.804	1.146
	Avg	<b>1.922</b>	<b>0.921</b>	2.432	1.012	2.108	<b>0.929</b>	2.623	1.060	2.444	1.203	2.388	1.011	<b>2.139</b>	0.931	2.616	1.090
Weather	96	<b>0.167</b>	<b>0.211</b>	0.198	0.235	0.171	0.214	0.203	0.244	0.174	0.214	0.177	0.218	0.172	0.220	0.196	0.255
	192	<b>0.212</b>	<b>0.253</b>	0.240	0.269	0.217	<b>0.254</b>	0.247	0.277	0.221	<b>0.254</b>	0.222	0.259	0.219	0.261	0.237	0.296
	336	<b>0.270</b>	<b>0.292</b>	0.295	0.308	<b>0.274</b>	<b>0.293</b>	0.297	0.311	0.278	<b>0.296</b>	0.277	0.297	0.280	0.306	0.283	0.335
	720	0.350	<b>0.348</b>	0.368	0.353	0.351	0.343	0.368	0.356	0.358	0.349	0.352	0.347	0.365	0.359	<b>0.345</b>	0.381
	Avg	<b>0.250</b>	<b>0.276</b>	0.275	0.291	0.253	<b>0.276</b>	0.279	0.297	0.258	0.278	0.257	0.280	0.259	0.287	0.265	0.317

Table 1: Forecasting performance comparisons. The input sequence length is 36 for the Illness and FRED datasets and 96 for others.

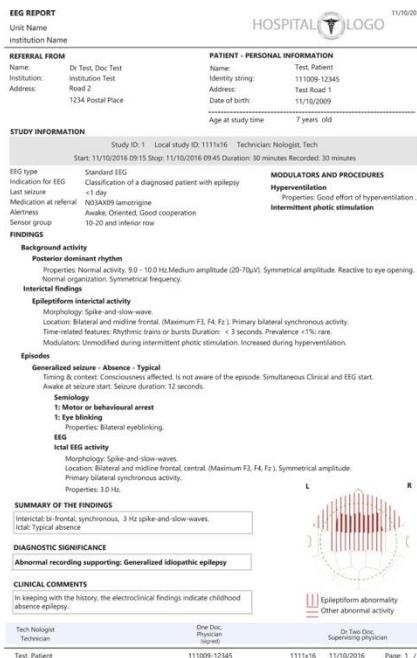
# The Text Was Still Self-Designed, What if the real-world text

seizure detection and classification

ECG data

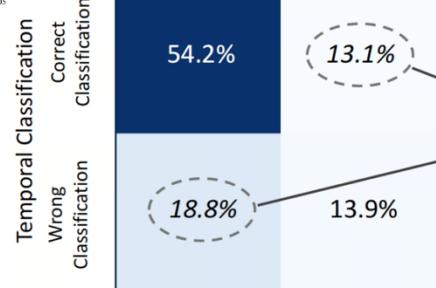


Doctor Report

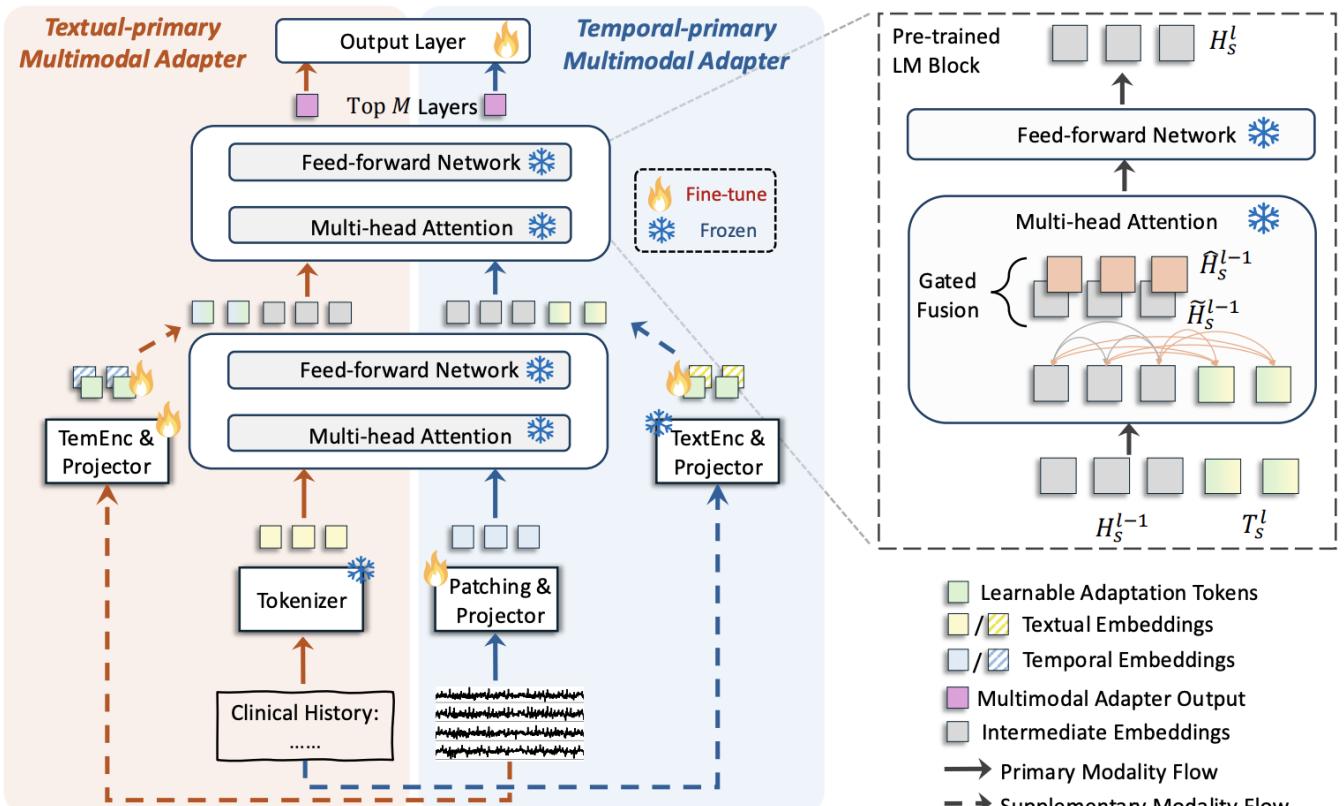


Textual Classification

Classification	Percentage
Correct Classification	54.2%
Wrong Classification	45.8%



*More than 30% of the samples are misclassified by one modality but can be corrected by another.*



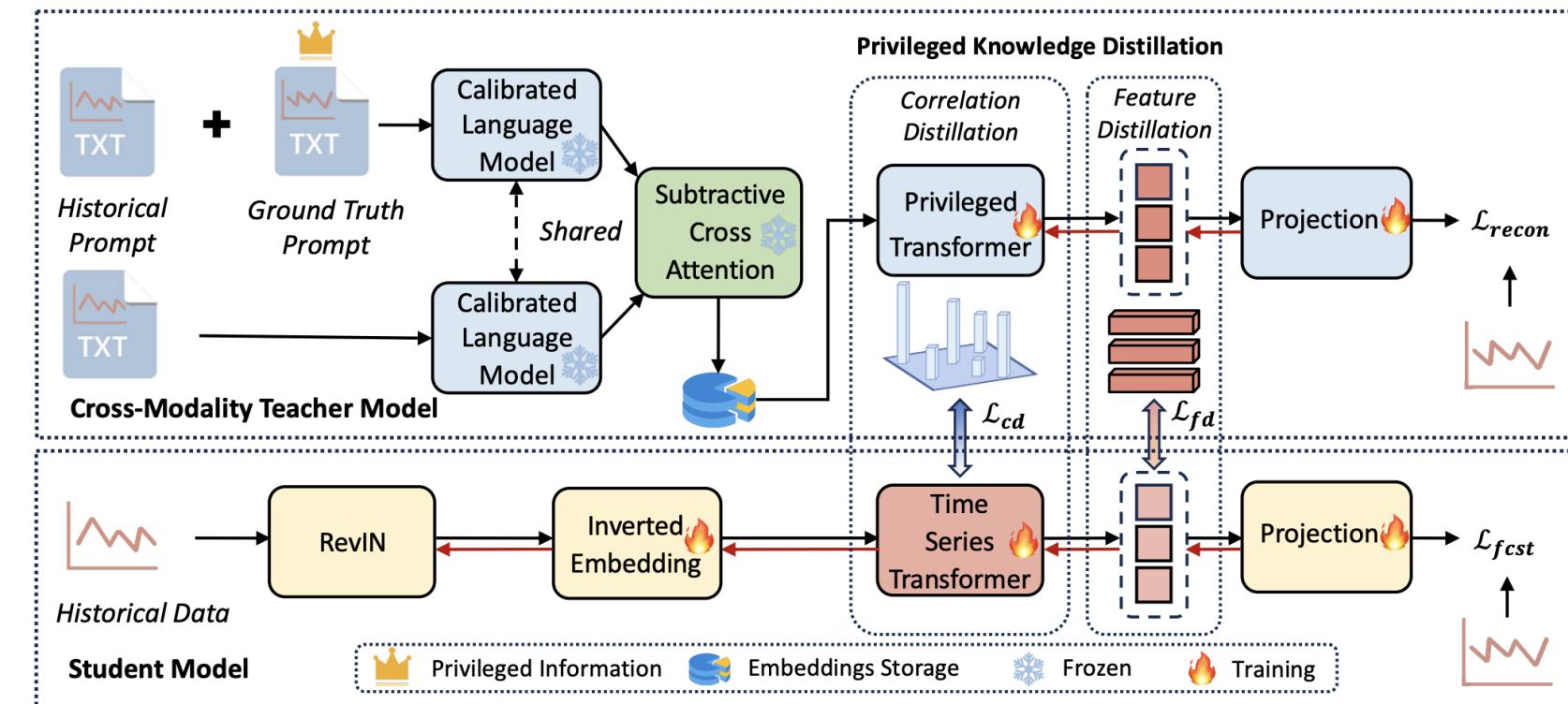
# Performance

Modality	Model	PTB-XL								TUSZ								Average	
		4 Classes				5 Classes				2 Classes				4 Classes					
		Acc.	Pre.	Rec.	F1	Acc.	F1												
Time	LSTM	0.68	0.60	0.48	0.48	0.67	0.63	0.50	0.52	0.76	0.53	0.54	0.54	0.58	0.44	0.27	0.26	0.67	0.45
	TimesNet	0.68	0.46	0.46	0.45	0.67	0.59	0.48	0.50	0.74	0.59	0.63	0.59	0.76	0.75	<u>0.72</u>	<u>0.71</u>	0.71	0.56
	LightTS	0.68	0.59	0.53	0.54	0.59	0.46	0.44	0.45	0.74	0.53	0.53	0.54	0.71	0.72	0.58	0.58	0.68	0.53
	Dlinear	0.68	0.58	0.50	0.49	0.61	0.46	0.41	0.41	0.78	0.52	0.52	0.52	0.71	0.62	0.60	0.59	0.70	0.50
	Pyraformer	0.76	0.66	0.59	0.58	0.66	0.56	0.49	0.51	<u>0.84</u>	0.47	0.50	0.47	<u>0.75</u>	<u>0.77</u>	0.67	<u>0.72</u>	<u>0.75</u>	0.57
	ETSformer	0.72	0.63	0.57	0.55	0.54	0.45	0.38	0.40	0.79	0.53	0.53	0.53	0.73	0.70	0.66	0.66	0.70	0.54
	Autoformer	0.72	0.56	0.56	0.54	0.62	0.47	0.44	0.44	0.79	0.52	0.51	0.51	0.70	0.64	0.64	0.61	0.71	0.53
	Crossformer	0.66	0.58	0.51	0.53	0.65	0.55	0.48	0.50	0.79	0.50	0.51	0.50	0.72	0.71	0.58	0.58	0.71	0.53
	FEDformer	0.67	0.57	0.50	0.51	0.65	0.53	0.47	0.49	0.76	0.57	0.58	0.57	0.68	0.48	0.54	0.48	0.69	0.51
	Informer	0.67	0.59	0.51	0.52	0.67	0.59	0.51	0.52	0.82	0.57	0.55	0.55	<u>0.77</u>	0.74	0.69	0.71	0.73	0.58
	Reformer	0.69	0.56	0.53	0.54	0.65	0.53	0.48	0.49	<u>0.84</u>	0.52	0.50	0.48	0.74	0.75	0.61	0.66	0.73	0.54
	iTransformer	0.56	0.42	0.36	0.37	0.54	0.39	0.31	0.29	0.80	0.50	0.50	0.49	0.73	0.75	0.59	0.61	0.66	0.44
	PatchTST	0.78	0.76	0.62	0.62	<u>0.74</u>	0.69	0.59	0.62	0.73	0.54	0.55	0.54	0.70	0.65	0.59	0.57	0.74	0.59
Time	GPT4TS	0.71	0.58	0.52	0.53	0.59	0.46	0.45	0.45	0.78	0.48	0.48	0.48	0.71	0.73	0.60	0.64	0.70	0.53
Text	GPT2	0.72	0.65	0.56	0.58	0.73	0.65	0.61	0.62	0.72	0.49	0.49	0.50	0.64	0.69	0.53	0.58	0.70	0.57
	BERT	0.70	0.64	0.51	0.53	0.73	0.65	0.59	0.62	0.72	0.49	0.49	0.49	0.59	0.45	0.39	0.40	0.69	0.51
	Llama 3	0.73	0.60	0.60	0.60	<u>0.74</u>	<u>0.69</u>	0.56	0.65	0.72	0.53	0.53	0.55	0.66	0.62	0.47	0.47	0.71	0.57
	ClinicalBERT	0.73	0.57	0.54	0.53	<u>0.74</u>	0.63	0.58	<u>0.66</u>	0.72	0.55	0.63	0.56	0.67	0.36	0.64	0.43	0.72	0.55
Time + Text	TimeLLM	0.69	0.60	0.48	0.47	0.67	0.59	0.46	0.48	0.75	0.51	0.51	0.51	0.69	0.70	0.50	0.47	0.70	0.48
	UniTime	0.67	0.33	0.42	0.37	0.64	0.54	0.43	0.44	0.79	0.54	0.53	0.53	<u>0.77</u>	<u>0.78</u>	<u>0.71</u>	0.71	0.72	0.51
	GPT4MTS	0.72	0.59	0.60	0.59	0.65	0.48	0.50	0.48	0.82	0.64	0.63	<u>0.63</u>	0.70	0.72	0.60	0.53	0.72	0.56
	MedTsLLM	0.74	0.64	0.63	<u>0.64</u>	0.68	0.64	0.60	0.56	0.81	0.66	<u>0.64</u>	<u>0.64</u>	0.72	0.74	0.59	0.60	0.74	<u>0.61</u>
Text	MedualTime (Time)	0.73	0.65	<u>0.66</u>	0.62	0.71	0.62	<u>0.64</u>	0.59	0.83	<u>0.68</u>	0.63	0.60	0.74	0.75	0.61	0.62	<u>0.75</u>	<u>0.61</u>
	MedualTime (Text)	0.79	<u>0.73</u>	0.71	0.70	<u>0.75</u>	0.66	0.67	<u>0.66</u>	0.82	0.67	0.64	<u>0.63</u>	<u>0.75</u>	0.76	0.68	0.65	<u>0.78</u>	<u>0.66</u>
	MedualTime	<b>0.83</b>	<b>0.78</b>	<b>0.75</b>	<b>0.76</b>	<b>0.81</b>	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>	<b>0.87</b>	<b>0.75</b>	<b>0.65</b>	<b>0.68</b>	<b>0.79</b>	<b>0.82</b>	<b>0.74</b>	<b>0.75</b>	<b>0.83</b>	<b>0.73</b>

Table 1: **Supervised Learning**. MedualTime achieves an average improvement of **8%** in Acc. and **12%** in F1 compared to baselines. The best results are in **bold** while the second and third best are in underlined. "Acc.", "Pre.", and "Rec." represent accuracy, precision and recall respectively. All LM-based models are highlighted in grey.



# Your Own Local Small Model



Models	ETTm1			
	Trainabl. Param.	Train. Time	Mem.	Infer. Speed
iTransformer	<b>0.22</b>	<b>12.97</b>	1,722	0.08
Time-LLM	44.66	4799.64	28,882	1.08
UniTime	108.54	2472.15	4,168	0.39
OFA	1.75	425.12	910	0.18
TimeCMA	17.99	50.13	821	0.09
TimeKD	1.72	49.78	<b>730</b>	<b>0.06</b>

Fig. 3: TimeKD Framework. **Cross-Modality Teacher Model** processes textual prompts to reconstruct time series during the training stage. **Student Model** learned from the teacher model via privileged knowledge distillation for efficient forecasting.

# Strong Generalizability

TABLE V: Few-shot forecasting on 10% training data. The input length and forecasting horizon are set to 96 time steps.

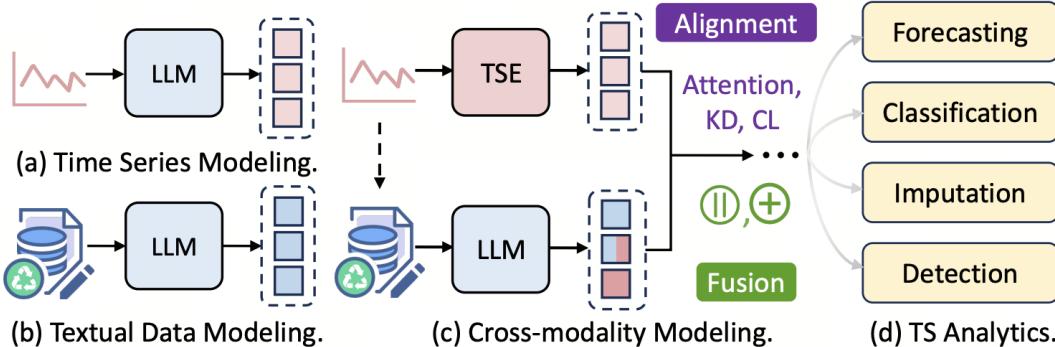
Datasets	TimeKD		TimeCMA		Time-LLM		UniTime		OFA		iTransformer		PatchTST	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	<b>0.429</b>	<b>0.396</b>	<u>0.442</u>	<u>0.439</u>	0.587	0.491	<u>0.559</u>	<u>0.486</u>	0.615	0.497	<u>0.565</u>	<u>0.484</u>	<u>0.558</u>	<u>0.478</u>
ETTm2	<b>0.183</b>	<b>0.261</b>	<u>0.185</u>	<u>0.265</u>	0.189	0.270	<u>0.186</u>	<u>0.267</u>	0.187	0.266	<u>0.194</u>	<u>0.277</u>	<u>0.189</u>	<u>0.268</u>
ETTh1	<b>0.421</b>	<b>0.415</b>	<u>0.431</u>	<u>0.422</u>	0.498	0.462	<u>0.502</u>	<u>0.467</u>	0.462	0.449	<u>0.537</u>	<u>0.493</u>	<u>0.433</u>	<u>0.428</u>
ETTh2	<b>0.304</b>	<b>0.341</b>	<u>0.314</u>	<u>0.355</u>	0.329	0.367	<u>0.331</u>	<u>0.368</u>	0.327	0.359	<u>0.341</u>	<u>0.378</u>	<u>0.314</u>	<u>0.354</u>

TABLE VI: Zero-shot Forecasting Results on ETT. The notation “trained dataset” → “test dataset” indicates the dataset used for training and the one used for testing, respectively. The forecasting horizon is set to 96 time steps.

Model	TimeKD		TimeCMA		Time-LLM		UniTime		OFA		iTransformer		PatchTST	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1 → ETTm2	<b>0.189</b>	<b>0.263</b>	<u>0.192</u>	<u>0.266</u>	0.194	0.267	<u>0.197</u>	<u>0.273</u>	0.196	<u>0.267</u>	0.202	<u>0.279</u>	<u>0.193</u>	<u>0.269</u>
ETTm2 → ETTm1	<b>0.487</b>	<b>0.439</b>	<u>0.495</u>	<u>0.443</u>	0.514	0.441	<u>0.529</u>	<u>0.449</u>	0.519	<u>0.457</u>	0.679	<u>0.52</u>	<u>0.554</u>	<u>0.460</u>
ETTh1 → ETTh2	<b>0.268</b>	<b>0.336</b>	<u>0.295</u>	<u>0.341</u>	0.300	0.346	<u>0.308</u>	<u>0.349</u>	0.297	<u>0.344</u>	0.296	<u>0.344</u>	<u>0.296</u>	<u>0.343</u>
ETTh2 → ETTh1	<b>0.375</b>	<b>0.391</b>	<u>0.389</u>	<u>0.440</u>	0.522	0.479	<u>0.537</u>	<u>0.490</u>	0.512	<u>0.472</u>	0.575	<u>0.515</u>	<u>0.485</u>	<u>0.461</u>



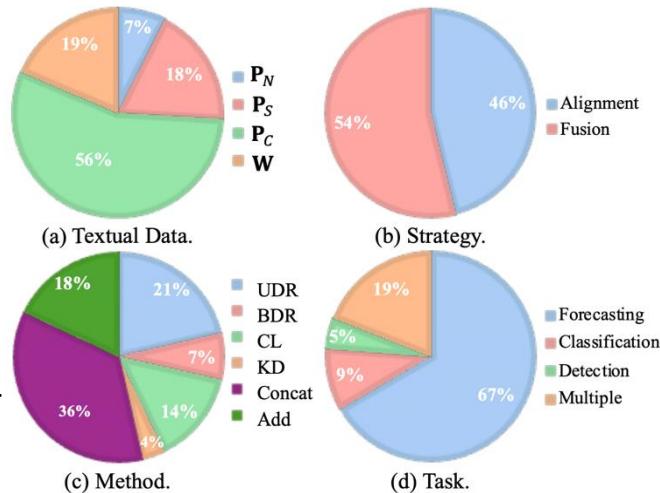
# In Which Case Multi-Modality LLM Works? Does it always work?



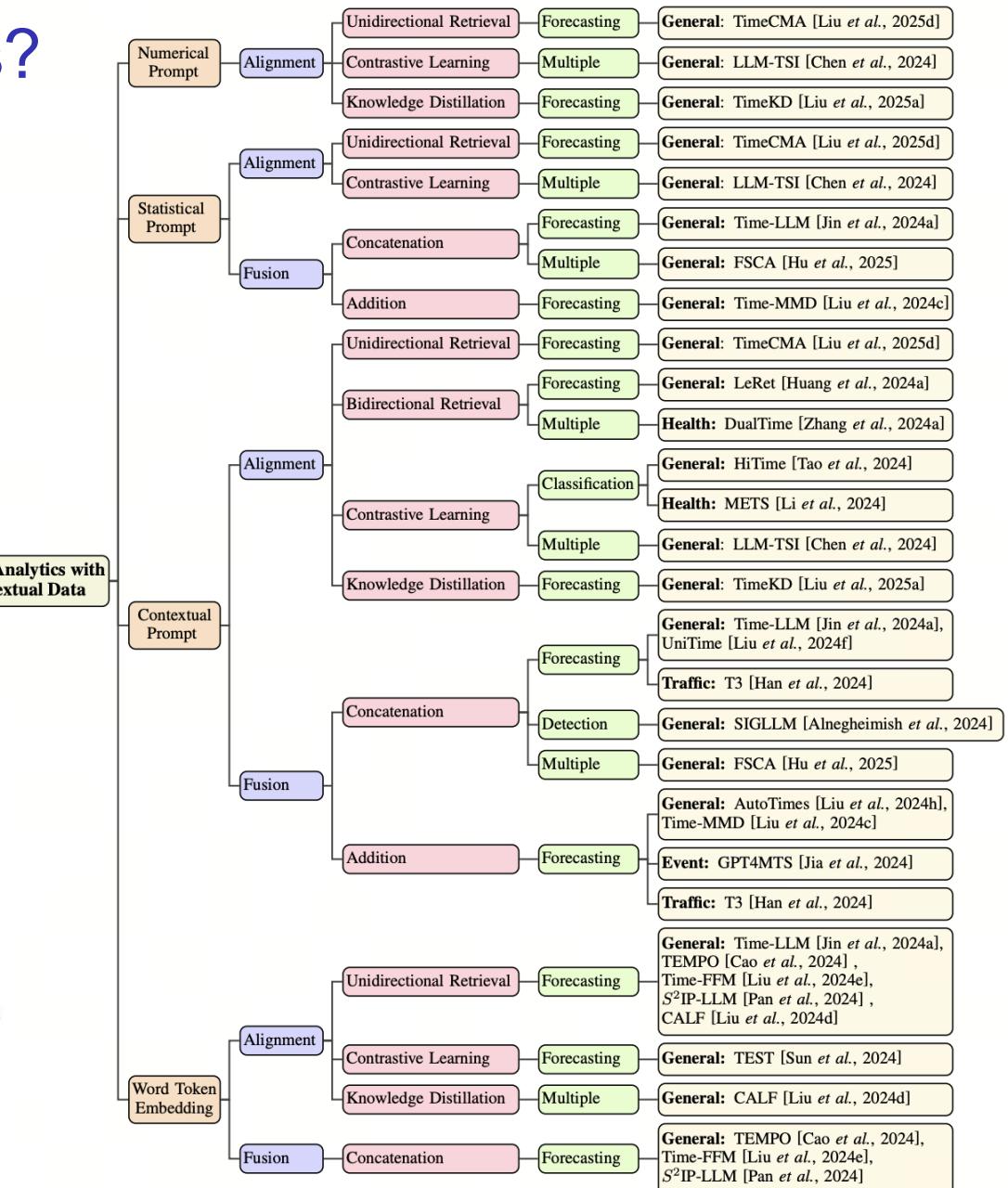
**(1) What kind of text (2) how to deal with two modalities**

**(3) which tasks**

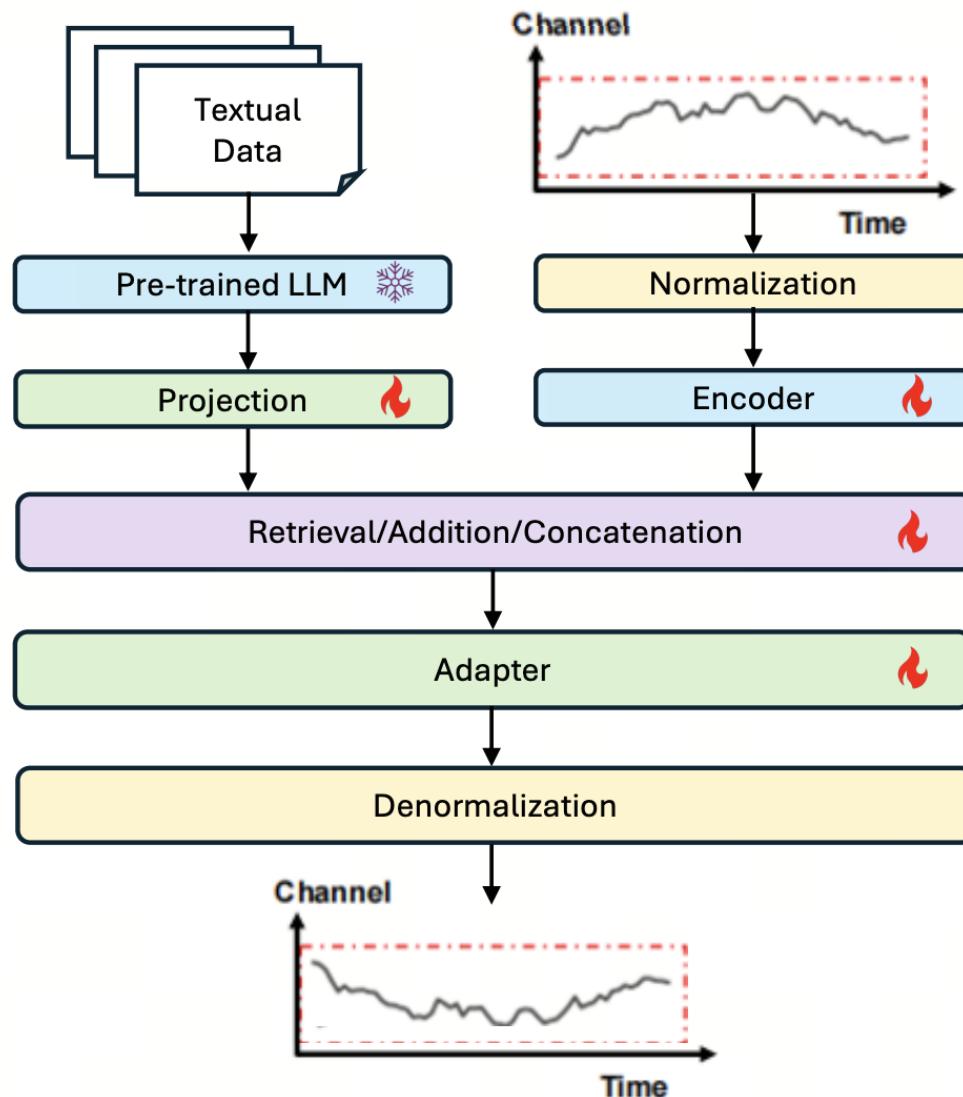
70 papers 2023-2025



[IJCAI'25<sup>A\*</sup>] LZXMLLZ, Towards Cross-  
Modality Modeling for Time Series  
Analytics: A Survey in the LLM Era



# Testing Protocol



Domain	Dim	Frequency	Samples	Timespan
Agriculture	1	Monthly	496	1980 - 2024
Climate	5	Monthly	496	2000 - 2024
Economy	3	Monthly	423	1987 - 2024
Energy	9	Weekly	1479	1993 - 2024
Health	11	Weekly	1389	2002 - 2024

# Insights

1. Textual Data Enhances Forecasting (overall)

Data	Method	Agriculture		Climate		Energy		Economy		Health	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Single Modality											
Time Series	-	2.68	1.35	0.371	0.473	0.183	0.315	0.0244	0.128	1.12	0.774
Cross Modality											
Time Series & Numerical Prompt	Retrieval	2.83	1.38	<b>0.287</b>	<b>0.425</b>	0.186	0.315	0.0247	0.129	<b>0.96</b>	<b>0.665</b>
	Addition	2.77	1.34	0.297	0.434	<b>0.180</b>	<b>0.309</b>	0.0252	0.132	1.05	0.754
	Concatenation	2.87	1.41	<u>0.296</u>	<u>0.425</u>	0.224	0.350	0.0267	0.132	1.16	0.797
Time Series & Statistical Prompt	Retrieval	<b>2.67</b>	<b>1.33</b>	0.386	0.483	0.196	0.333	<b>0.0232</b>	<b>0.125</b>	0.97	0.667
	Addition	<b>2.64</b>	<b>1.29</b>	0.380	0.478	0.183	0.313	0.0244	0.126	1.08	0.771
	Concatenation	2.75	1.38	0.386	0.488	0.190	0.325	0.0254	0.130	1.16	0.802
Time Series & Contextual Prompt	Retrieval	2.88	1.35	0.389	0.493	0.185	0.317	0.0261	0.131	1.11	0.661
	Addition	2.85	1.32	0.364	0.467	<u>0.182</u>	<u>0.313</u>	0.0263	0.132	1.17	0.713
	Concatenation	2.93	1.41	0.387	0.485	<u>0.193</u>	<u>0.327</u>	0.0272	0.134	1.28	0.776
Time Series & Word Token Embedding	Retrieval	2.90	1.37	0.388	0.490	0.181	0.312	0.0271	0.135	1.18	0.721
	Addition	2.95	1.45	0.393	0.479	0.187	0.319	0.0265	0.133	1.22	0.768
	Concatenation	2.92	1.42	0.389	0.484	0.188	0.320	0.0274	0.138	1.29	0.779

3. Alignment Outperforms Fusion.

4. Economy and health forecasting benefit considerably from numerical and statistical prompts, Climate and agriculture forecasting show more moderate improvements

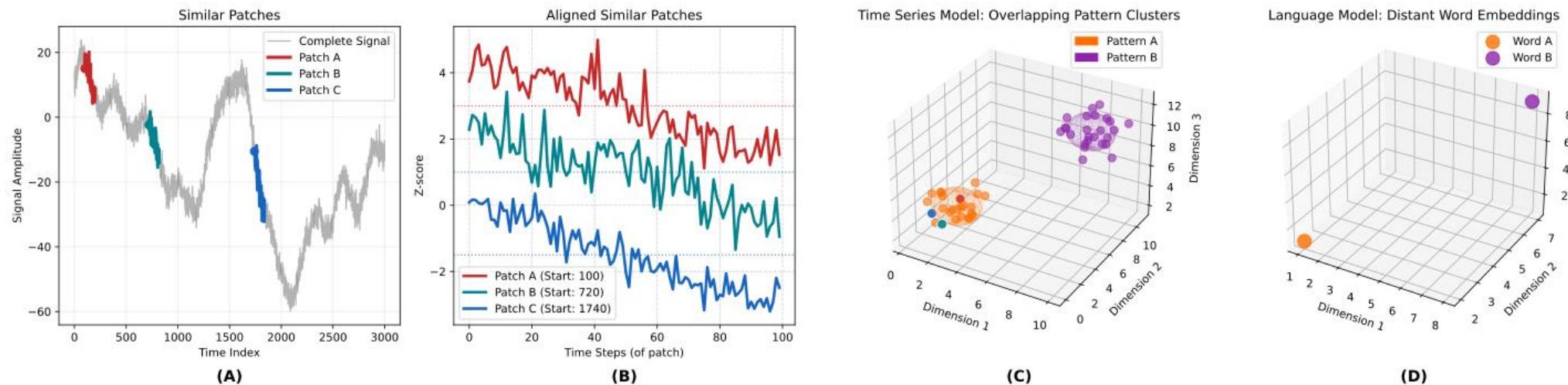


# In the language of time

**Key hypothesis:** Time series foundation models: *moving from deterministic vector tokens to latent probabilistic distributional tokens.*

**Intuition:** Patches with similar temporal trends share similar semantics, forming a token cluster, where each numerical variation corresponds to a specific embedding, analogous to how words in language models map to word vectors.

**Implication:** This reveals that time series exhibit clear **quasi-language characteristics**, thereby inheriting the expressive and generalization capabilities of language models.



**The illustration of probabilistic distributional tokens (time series) and deterministic vector tokens (language)**

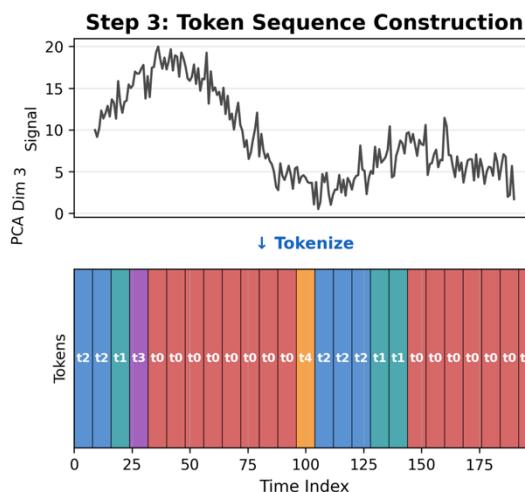
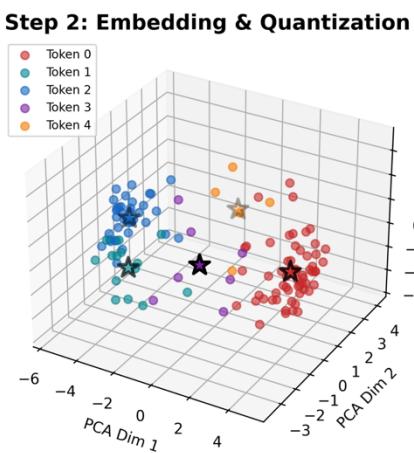
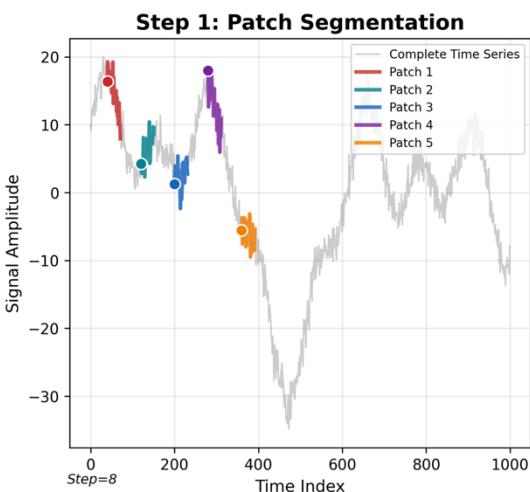
[under review, Arxiv] The language of time: a language model perspective on time-series foundation models

# Our Empirical Study

**Objective:** To verify the quasi-language statistical properties of time series after patching and quantization.

**Pipeline:** (1) Patch Segmentation (Split “words” or “phrases”); (2) Embedding & Quantization (Construct the codebook, K-Means herein); (3) Token Sequences Construction; (4) Statistical Analysis.

**Datasets:** 37 cross-domain time series datasets (covering six major categories including meteorology, energy, transportation, and finance), containing approximately 187 million time points and generating about 790 million patch tokens.



## Step 4: Statistical Analysis

- 1 Zipf Law Confirmation
- 2 Time Vocabulary Construction
- 3 Grammar Analysis
- ...

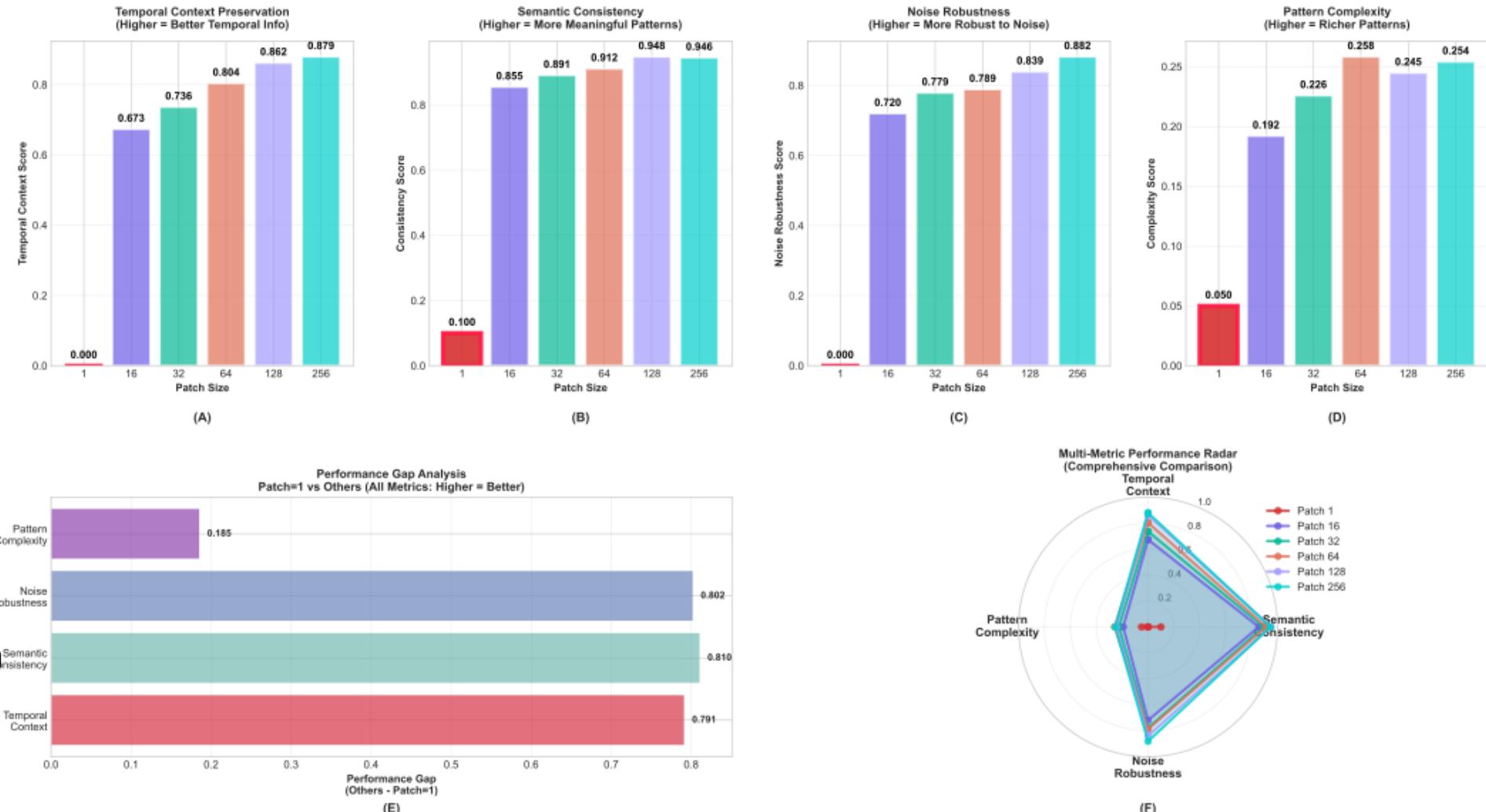
Table 3: Overview of Time Series Datasets

Dataset	Length	Features	Total Points	Size (MB)	Frequency	Description
ETTh1	17,420	7	121,940	0.93	1 hour	Power transformer temperature (2016.7-2018.7)
ETTh2	17,420	7	121,940	0.93	1 hour	Power transformer temperature (2016.7-2018.7)
ETTm1	69,680	7	487,760	3.72	15 min	Power transformer temperature (2016.7-2018.7)
ETTm2	69,680	7	487,760	3.72	15 min	Power transformer temperature (2016.7-2018.7)
NASDAQ	14,224	1	14,224	0.11	1 day	NASDAQ composite index historical data
S&P500	2,598	1	2,598	0.02	1 day	Standard & Poor's 500 index data
WTH	35,064	12	420,768	3.21	1 hour	Weather data (12 meteorological parameters)
Air Quality	9,357	13	121,641	0.93	1 hour	Air quality monitoring (13 environmental indicators)
COVID Deaths	212	233	49,396	0.38	1 day	COVID-19 death cases time series data
Electricity	26,304	321	8,443,584	64.42	1 hour	Electricity load (321 consumption points)
Electricity Demand	230,736	5	1,153,680	8.8	1 min	High-frequency electricity demand (4+ years)
Electricity Weekly	156	321	50,076	0.38	1 week	Electricity load (weekly aggregated)
Energy	19,735	26	513,110	3.91	1 day	Energy consumption (26 indicators)
Exchange Rate	7,588	8	60,704	0.46	1 day	Foreign exchange rates (8 currency pairs)
FRED-MD	728	107	77,896	0.59	1 month	Federal Reserve economic data (107 indicators)
Hospital	84	767	64,428	0.49	1 week	Hospital-related time series (767 indicators)
Web Traffic Daily	803	145,035	116,463,105	888.54	1 day	Wikipedia page views (145,035 pages)
Web Traffic Weekly	114	145,035	16,533,990	126.14	1 week	Wikipedia page views (weekly aggregated)
Metro	48,204	1	48,204	0.37	1 hour	Urban metro passenger traffic
National Illness	966	7	6,762	0.05	1 week	Disease surveillance (7 illness types)
NN5 Daily	791	111	87,801	0.67	1 day	NN5 competition dataset (111 series)
NN5 Weekly	113	111	12,543	0.1	1 week	NN5 competition dataset (weekly)
Oikolab Weather	100,057	8	800,456	6.11	1 hour	Oikolab weather data (8 parameters)
River Flow	23,741	1	23,741	0.18	1 day	River discharge monitoring
Saugeen Day	23,741	1	23,741	0.18	1 day	Saugeen river basin data
Solar 10min	52,560	137	7,200,720	54.94	10 min	Solar power (137 stations, Alabama 2006)
Solar 1min	493,149	1	493,149	3.76	1 min	High-frequency solar power generation
Solar 4sec	7,397,222	1	7,397,222	56.44	4 sec	Ultra-high-frequency solar power data
Solar Weekly	52	137	7,124	0.05	1 week	Solar power generation (weekly)
Sunspot	73,924	1	73,924	0.56	1 month	Solar activity data (1749-present, 270+ years)
TCPC	52,416	8	419,328	3.2	1 hour	Temperature-correlated power consumption
Traffic	17,544	862	15,122,928	115.38	1 hour	Highway traffic (862 monitoring points)
Traffic Weekly	104	862	89,648	0.68	1 week	Traffic flow (weekly aggregated)
US Births	7,305	1	7,305	0.06	1 day	US birth rate data
Weather	52,696	21	1,106,616	8.44	1 hour	Comprehensive weather (21 parameters)
Wind 1min	493,144	1	493,144	3.76	1 min	High-frequency wind power generation
Wind 4sec	7,397,147	1	7,397,147	56.44	4 sec	Ultra-high-frequency wind power

# Evidence 1: Patches v.s. Points

**Premise :** Individual points describe *what happens now*, patches describe *how things evolve*. In time series, meaning lies in evolution, not in isolation.

**Evidence:** Over 37 datasets, patches consistently yield richer, more stable, and more interpretable representations, proving that temporal meaning emerges at the patch level.



# Evidence 2: Vocabulary Construction

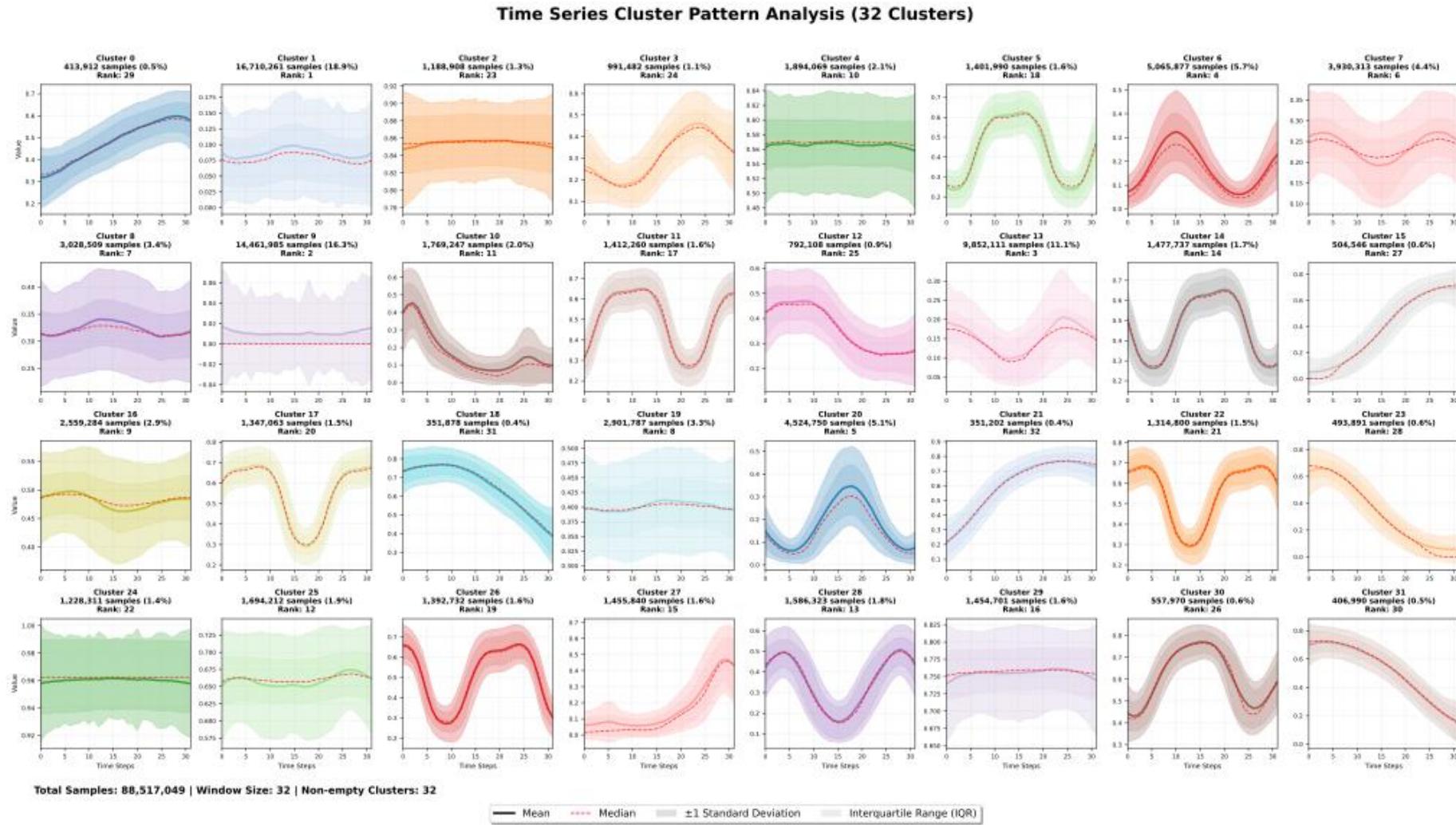
**Objective :** *From Patches to Time*

**Words.** To convert continuous temporal patches into discrete units that can serve as the vocabulary of time.

**Method:** *Quantizing Temporal Patches.* Quantize the embeddings of temporal patches ([discretization methods include K-Means, VQ-VAE, etc.](#)) Each entry in the vocabulary represents a typical temporal motif, or a “time word,” which characterizes a specific temporal pattern in the time series, forming a distribution due to variations in magnitude.

**Results: Structured Vocabulary**

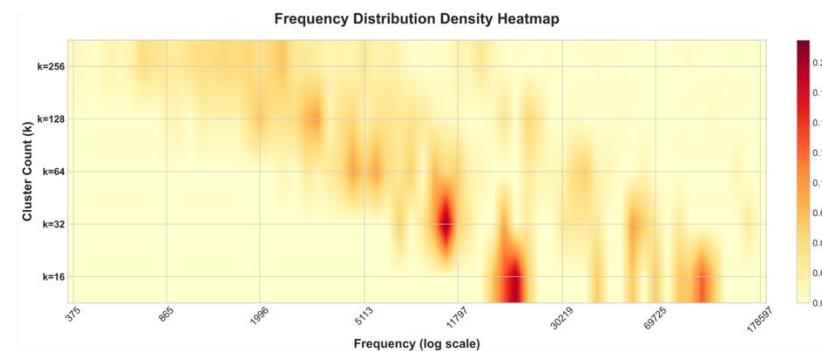
**Emerges.** Tokens capture recurring local dynamics, like trends, cycles, bursts, and form a compact dictionary of temporal behaviors.



# Evidence 3: Quasi-Linguistic Statistics

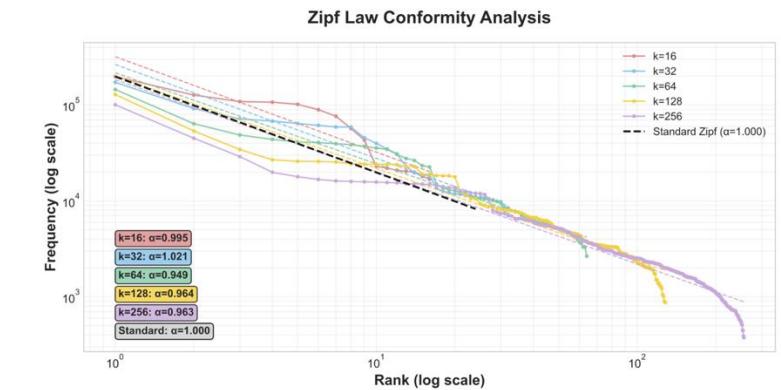
- **Findings 1: Zipf-like Frequency Hierarchy.**

Time tokens exhibit a power-law rank–frequency distribution, mirroring how words in language follow Zipf's law, a few "core tokens" dominate usage while many remain rare.

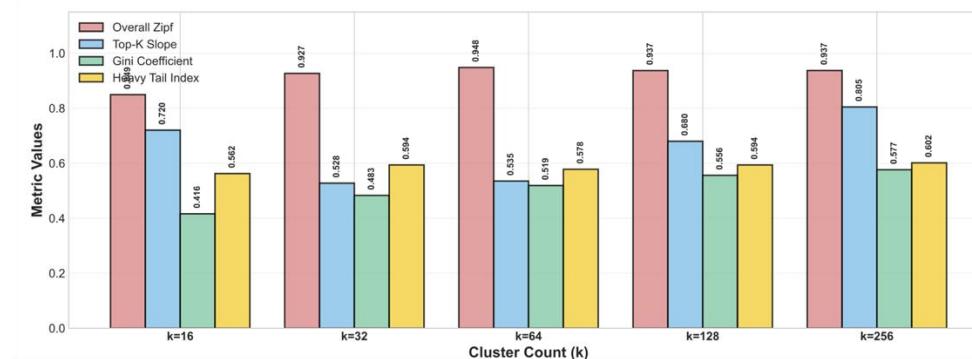


- **Finding 2: Hierarchical and Compositional Semantics.** The long-tail structure reflects semantic hierarchy and compositionality: frequent motifs encode generic temporal dynamics, while rare motifs capture specialized or context-dependent events.

Long-tailed and semantic hierarchy



Zipf's law conformity with Statistical Robustness

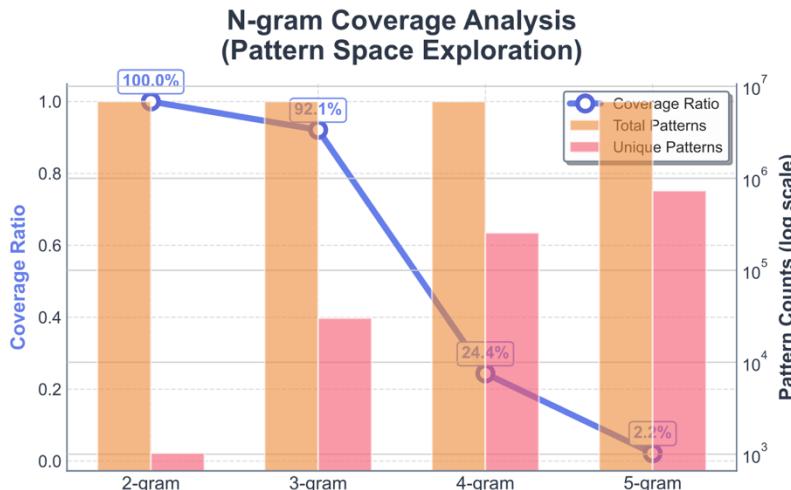


Scale-invariant statistical quasi-linguistic regularities

# Evidence 4: Time Series Grammar



N-gram transition probabilities



**Existence of Time Grammar.** N-gram analysis reveals clear **transition probabilities between temporal motifs**, indicating that time series follow **structured and rule-based organization** — what we call the “time grammar.”

## Basic Rules of Time Grammar:

**State Inertia:** Motifs tend to persist, dynamics are continuous.

**Sparse Syntax:** Only limited motif transitions are valid, forming structured grammar.

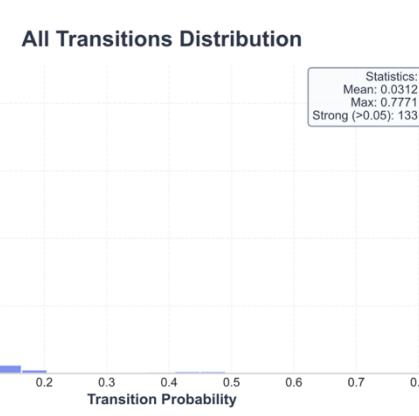
**Temporal Phrases:** Longer n-grams compose richer dynamic patterns.

**Chunked Diversity:** Complex sequences arise from stable local chunks.

**Long-tail Usage:** A few transitions dominate, most remain rare.

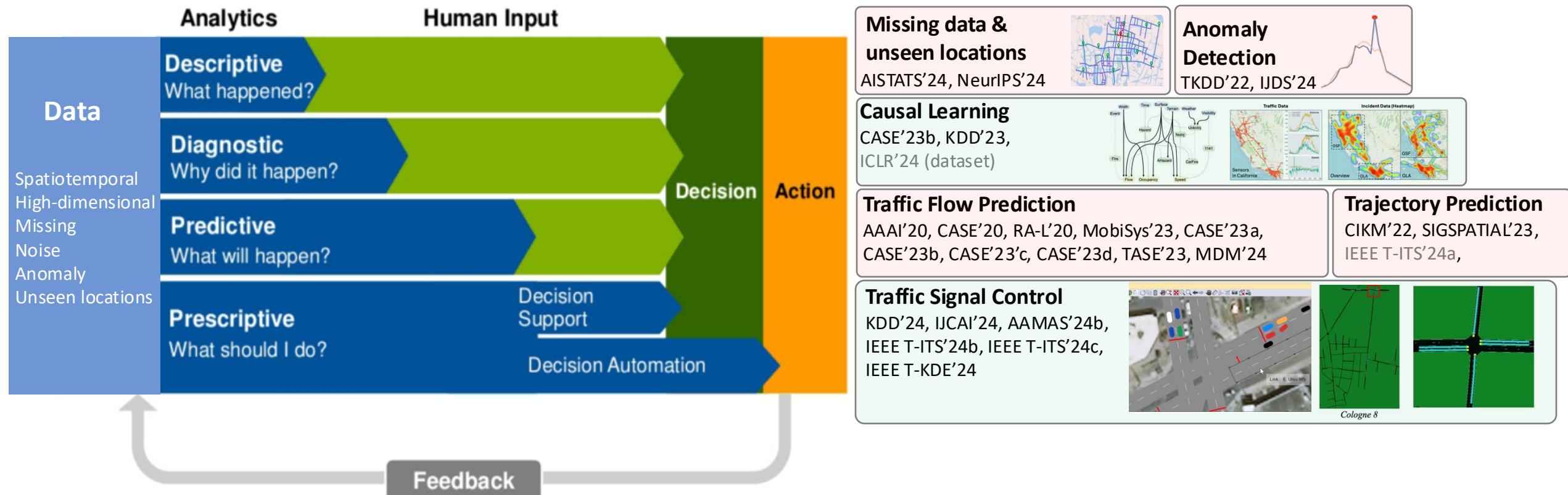


Complexity and repetition trade-off



Long-tailed transition probabilities distribution

# Four Analytical Pillars in Spatiotemporal Task



# Thank you!

