

What Can Large Language Models Tell Us About Time Series Analysis

Ming Jin

Monash University



May 16th, 2024

Outline



Part 1. Introduction & Background

- Time series and language modeling
- Multimodal large language models
- Large language models for time series data
- Model reprogramming

What are time series? •



Time Series

['tīm 'sir-(,)ēz]

A sequence of data points that occur in successive order over some period





• What are time series?



• What is language modeling?



• What is language modeling?



I	saw a cat
I	saw a cat on the chair
I	saw a cat running after a dog
I	saw a cat in my dream
I	saw a cat book



• What is language modeling?





• What are large language models?



• Multimodal large language models



https://huyenchip.com/2023/10/10/multimodal.html

• Multimodal large language models



How time series analysis benefits from the recent advances of LLMs?



Example of TS forecasting



1950 - 2000s

Monash University



1950 - 2000s

2010s

Monash University





Monash University

Jin, M., Zhang, Y., Chen, W., Zhang, K., Liang, Y., Yang, B., ... & Wen, Q. (2024). Position Paper: What Can Large Language Models Tell Us about Time Series Analysis. arXiv preprint arXiv:2402.02713.









 Illustration of different adaptation techniques when repurposing/developing large language and other foundation models for time series data Time series foundation models, e.g., moirai and timesFM



 Illustration of different adaptation techniques when repurposing/developing large language and other foundation models for time series data

Monash University

e.g., LLMTime



 Illustration of different adaptation techniques when repurposing/developing large language and other foundation models for time series data

e.g., Time-LLM



 Illustration of different adaptation techniques when repurposing/developing large language and other foundation models for time series data

• A broad picture



Jin, M., Wen, Q., Liang, Y., Zhang, C., Xue, S., Wang, X., ... & Xiong, H. (2023). Large models for time series and spatio-temporal data: A survey and outlook. arXiv preprint arXiv:2310.10196. Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., ... & Wen, Q. (2024). Foundation models for time series analysis: A tutorial and survey. arXiv preprint arXiv:2403.14735.

Part 2. Time-LLM

- Motivation (Conceptual designs)
- Model architecture & Highlights
- Our main results

Motivation

• Reprogramming makes LLMs instantly ready for time series tasks



Motivation

• Reprogramming makes LLMs more powerful for time series tasks



Motivation



Figure: Schematic illustration of reprogramming LLMs in comparison of (a) task-specific learning and (b) model fine-tuning

- Task-specific learning: Most time series forecasting models are crafted for specific tasks and domains (e.g., traffic prediction), and trained end-to-end on small-scale data.
- **In-modality adaptation:** A typical example is the time series pre-trained models (TSPTMs)
- Cross-modality adaptation: Transferring the knowledge from powerful pre-trained source foundation models to perform target tasks via model fine-tuning or reprogramming



TL;DR Domain expert knowledge & Task instructions + Reprogrammed input time series = Significantly better forecasts





TL;DR Domain expert knowledge & Task instructions + Reprogrammed input time series = Significantly better forecasts

Unlocking the LLM's ability for time series

Cross-modal Adaptation: 1 2 5

Cross-modal Alignment: 3 4

 Patch Reprogramming: we reprogram TS patch embeddings into the source data representation space to align the modalities of time series and natural language to activate the backbone's time series understanding and reasoning capabilities.

Monash University



TL;DR Domain expert knowledge & Task instructions + Reprogrammed input time series = Significantly better forecasts



 Prompt-as-Prefix: natural language-based prompts (e.g., domain knowledge & task instructions) can act as prefixes to enrich the input context and guide the transformation of reprogrammed TS patches



 Patch reprogramming: Text prototypes learn connecting language cues, e.g., "short up" (red lines) and "steady down" (blue lines), which are then combined to represent the local patch information (e.g., "short up then down steadily" for characterizing Patch 5)



- **Prompt-as-Prefix** is proposed to enrich the input context and guide the transformation of reprogrammed time series patches
- **Prompt-as-Prefix** is more desired in time series forecasting compared to **Patch-as-Prefix**

The Electricity Transformer Temperature (ETT) indicates the electric power long-term deployment. Each data point consists of the target oil temperature and 6 power load features ... Below is the information about the input time series: [BEGIN DATA] *** [Domain]: We usually observe that electricity consumption peaks at noon, with a significant increase in transformer load *** [Instruction]: Predict the next <*H*> steps given the previous <T> steps information attached *** [Statistics]: The input has a minimum of <min_val>, a maximum of <max_val>, and a median of <median_val>. The overall trend is <upward or downward>. The top five lags are <lag_val>. <input context> [END DATA]

ion ioad ious haximum rall trend al>.

0.6

- In this prompt template, < > and < > are task-specific configurations and calculated input statistics
- In practice, there are **much more possibilities** in bringing time series and the related textual contexts (that are prompts) together

Table 1: Long-term forecasting results. All results are averaged from four different forecasting horizons: $H \in \{24, 36, 48, 60\}$ for ILI and $\{96, 192, 336, 720\}$ for the others. A lower value indicates better performance. Red: the best, <u>Blue</u>: the second best. Our full results are in Appendix D.

Methods	TIME	-LLM urs)	GPT (202	74TS 23a)	DLi (20	near 23)	Patch (20	nTST 23)	Time (20	esNet 23)	FEDf	ormer 22)	Autof (20	ormer 21)	Statio	onary 22)	ETSfe (20)	ormer 22)	Ligh (202	ntTS 22a)	Infor (202	rmer 21)	Refo (20	rmer 20)
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.408	0.423	0.465	0.455	0.422	0.437	0.413	<u>0.430</u>	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	0.491	0.479	1.040	0.795	1.029	0.805
ETTh2	0.334	<u>0.383</u>	0.381	0.412	0.431	0.446	0.330	0.379	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	0.602	0.543	4.431	1.729	6.736	2.191
ETTm1	0.329	0.372	0.388	0.403	0.357	<u>0.378</u>	<u>0.351</u>	0.380	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.435	0.437	0.961	0.734	0.799	0.671
ETTm2	0.251	0.313	0.284	0.339	0.267	0.333	0.255	<u>0.315</u>	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	0.409	0.436	1.410	0.810	1.479	0.915
Weather	0.225	0.257	0.237	0.270	0.248	0.300	0.225	<u>0.264</u>	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.261	0.312	0.634	0.548	0.803	0.656
ECL	0.158	0.252	0.167	0.263	0.166	0.263	<u>0.161</u>	0.252	0.192	0.295	0.214	0.327	0.227	0.338	0.193	0.296	0.208	0.323	0.229	0.329	0.311	0.397	0.338	0.422
Traffic	0.388	<u>0.264</u>	0.414	0.294	0.433	0.295	<u>0.390</u>	0.263	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.622	0.392	0.764	0.416	0.741	0.422
ILI	1.435	<u>0.801</u>	1.925	0.903	2.169	1.041	<u>1.443</u>	0.797	2.139	0.931	2.847	1.144	3.006	1.161	2.077	0.914	2.497	1.004	7.382	2.003	5.137	1.544	4.724	1.445
1 st Count		7	()	()		5	()	()	()	(0	()	()	C)	(0

- We note average performance gains of 12% and 20% over GPT4TS (OFA) and TimesNet, respectively
- When compared with the SOTA task-specific Transformer model PatchTST, by reprogramming <u>the</u> <u>smallest Llama</u>, Time-LLM realizes an average MSE reduction of **1.4%**
- Relative to the other models, e.g., DLinear, our improvements are also pronounced, exceeding 12%

Table 2: Short-term time series forecasting results on M4. The forecasting horizons are in [6, 48] and the three rows provided are weighted averaged from all datasets under different sampling intervals. A lower value indicates better performance. **Red**: the best, <u>Blue</u>: the second best. More results are in Appendix D.

N	fethods	TIME-LLM (Ours)	GPT4TS (2023a)	TimesNet (2023)	PatchTST (2023)	N-HiTS (2023b)	N-BEATS (2020)	ETSformer (2022)	LightTS (2022a)	DLinear (2023)	FEDformer (2022)	Stationary (2022)	Autoformer (2021)	Informer (2021)	Reformer (2020)
se	SMAPE	11.983	12.69	12.88	12.059	<u>12.035</u>	12.25	14.718	13.525	13.639	13.16	12.780	12.909	14.086	18.200
/era	MASE	1.595	1.808	1.836	1.623	1.625	1.698	2.408	2.111	2.095	1.775	1.756	1.771	2.718	4.223
¥	OWA	0.859	0.94	0.955	<u>0.869</u>	<u>0.869</u>	0.896	1.172	1.051	1.051	0.949	0.930	0.939	1.230	1.775

- Time-LLM consistently surpasses all baselines, outperforming GPT4TS (OFA) by 8.7%
- Time-LLM remains competitive even when compared with the SOTA model, N-HiTS, w.r.t. MASE and OWA

Table 3: Few-shot learning on 10% training data. We use the same protocol in Tab. 1. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$. Our full results are in Appendix E.

Methods	TIME	-LLM	GPT	TATS	DLi	near	Patch	nTST	Time	esNet	FEDf	ormer	Autof	ormer	Statio	onary	ETSf	ormer	Ligh	ntTS	Info	rmer	Refo	ormer
	0	urs)	(202	23a)	(20	(23)	(20	23)	(20	23)	(20	22)	(20	21)	(20	22)	(20	22)	(202	22a)	(20	21)	(20	(20)
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.556	0.522	<u>0.590</u>	<u>0.525</u>	0.691	0.600	0.633	0.542	0.869	0.628	0.639	0.561	0.702	0.596	0.915	0.639	1.180	0.834	1.375	0.877	1.199	0.809	1.249	0.833
ETTh2	0.370	0.394	<u>0.397</u>	<u>0.421</u>	0.605	0.538	0.415	0.431	0.479	0.465	0.466	0.475	0.488	0.499	0.462	0.455	0.894	0.713	2.655	1.160	3.872	1.513	3.485	1.486
ETTm1	0.404	0.427	0.464	0.441	<u>0.411</u>	<u>0.429</u>	0.501	0.466	0.677	0.537	0.722	0.605	0.802	0.628	0.797	0.578	0.980	0.714	0.971	0.705	1.192	0.821	1.426	0.856
ETTm2	0.277	0.323	<u>0.293</u>	<u>0.335</u>	0.316	0.368	0.296	0.343	0.320	0.353	0.463	0.488	1.342	0.930	0.332	0.366	0.447	0.487	0.987	0.756	3.370	1.440	3.978	1.587
Weather	0.234	0.273	<u>0.238</u>	<u>0.275</u>	0.241	0.283	0.242	0.279	0.279	0.301	0.284	0.324	0.300	0.342	0.318	0.323	0.318	0.360	0.289	0.322	0.597	0.495	0.546	0.469
ECL	0.175	<u>0.270</u>	<u>0.176</u>	0.269	0.180	0.280	0.180	0.273	0.323	0.392	0.346	0.427	0.431	0.478	0.444	0.480	0.660	0.617	0.441	0.489	1.195	0.891	0.965	0.768
Traffic	0.429	<u>0.306</u>	0.440	0.310	0.447	0.313	<u>0.430</u>	0.305	0.951	0.535	0.663	0.425	0.749	0.446	1.453	0.815	1.914	0.936	1.248	0.684	1.534	0.811	1.551	0.821
1 st Count	:	8	1	1	(0		1	()		D	(D	(D	(0	(0	0)	(0

 In the realm of 10% few-shot learning, our methodology realizes a 5% MSE reduction in comparison to GPT4TS (OFA), without necessitating any fine-tuning on the LLM. In relation to recent SOTA models such as PatchTST, DLinear, and TimesNet, our average enhancements surpass 8%, 12%, and 33% w.r.t. MSE

Table 4: Few-shot learning on 5% training data. We use the same protocol in Tab. 1. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$. Our full results are in Appendix E.

Methods	TIME (O	-LLM urs)	GPT (202	24TS 23a)	DLi (20	near 23)	Patch (20	hTST 123)	Time (20	esNet 23)	FEDf	former (22)	Autof (20	former 21)	Static (20	onary (22)	ETSf (20	ormer 22)	Ligh (202	ntTS 22a)	Info (20	rmer 21)	Refo	ormer 20)
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.627	0.543	0.681	<u>0.560</u>	0.750	0.611	0.694	0.569	0.925	0.647	0.658	0.562	0.722	0.598	0.943	0.646	1.189	0.839	1.451	0.903	1.225	0.817	1.241	0.835
ETTh2	0.382	0.418	<u>0.400</u>	<u>0.433</u>	0.694	0.577	0.827	0.615	0.439	0.448	0.463	0.454	0.441	0.457	0.470	0.489	0.809	0.681	3.206	1.268	3.922	1.653	3.527	1.472
ETTm1	0.425	<u>0.434</u>	0.472	0.450	0.400	0.417	0.526	0.476	0.717	0.561	0.730	0.592	0.796	0.620	0.857	0.598	1.125	0.782	1.123	0.765	1.163	0.791	1.264	0.826
ETTm2	0.274	0.323	<u>0.308</u>	<u>0.346</u>	0.399	0.426	0.314	0.352	0.344	0.372	0.381	0.404	0.388	0.433	0.341	0.372	0.534	0.547	1.415	0.871	3.658	1.489	3.581	1.487
Weather	0.260	0.309	<u>0.263</u>	0.301	0.263	0.308	0.269	<u>0.303</u>	0.298	0.318	0.309	0.353	0.310	0.353	0.327	0.328	0.333	0.371	0.305	0.345	0.584	0.527	0.447	0.453
ECL	<u>0.179</u>	0.268	0.178	<u>0.273</u>	0.176	0.275	0.181	0.277	0.402	0.453	0.266	0.353	0.346	0.404	0.627	0.603	0.800	0.685	0.878	0.725	1.281	0.929	1.289	0.904
Traffic	0.423	<u>0.298</u>	0.434	0.305	0.450	0.317	0.418	0.296	0.867	0.493	0.676	0.423	0.833	0.502	1.526	0.839	1.859	0.927	1.557	0.795	1.591	0.832	1.618	0.851
1 st Count	:	5		2		1		1	()		0	(D	(D	()	()	()	(0

 Analogous trends are discernible in the 5% few-shot learning scenarios, where our average advancement over GPT4TS exceeds 5%. When compared with PatchTST, DLinear, and TimesNet, TIME-LLM manifests a striking average improvement of over 20%

Table 5: Zero-shot learning results. **Red**: the best, <u>Blue</u>: the second best. Appendix E shows our detailed results.

Methods	TIME-LLM (Ours)	GPT4TS (2023a)	LLMTime (2023)	DLinear (2023)	PatchTST (2023)	TimesNet (2023)
Metric	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE	MSE MAE
$ETTh1 \rightarrow ETTh2$	0.353 0.387	0.406 0.422	0.992 0.708	0.493 0.488	0.380 0.405	0.421 0.431
$ETTh1 \rightarrow ETTm2$	0.273 0.340	0.325 0.363	1.867 0.869	0.415 0.452	0.314 0.360	0.327 0.361
$ETTh2 \rightarrow ETTh1$	0.479 0.474	0.757 0.578	1.961 0.981	0.703 0.574	0.565 0.513	0.865 0.621
$ETTh2 \rightarrow ETTm2$	0.272 0.341	0.335 0.370	1.867 0.869	0.328 0.386	0.325 0.365	0.342 0.376
$ETTm1 \rightarrow ETTh2$	0.381 0.412	0.433 0.439	0.992 0.708	0.464 0.475	0.439 0.438	0.457 0.454
$ETTm1 \rightarrow ETTm2$	0.268 0.320	0.313 0.348	1.867 0.869	0.335 0.389	0.296 0.334	0.322 0.354
$ETTm2 \rightarrow ETTh2$	0.354 0.400	0.435 0.443	0.992 0.708	0.455 0.471	0.409 0.425	0.435 0.443
$ETTm2 \rightarrow ETTm1$	0.414 0.438	0.769 0.567	1.933 0.984	0.649 0.537	0.568 0.492	0.769 0.567

- Time-LLM consistently outperforms the most competitive baselines by a large margin, over 14.2% w.r.t. the second-best in MSE reduction.
- Considering the few-shot results, we observe that reprogramming an LLM tends to yield significantly better results in data scarcity scenarios

Variant		Long-tern	n Forecasting			Few-shot	Forecasting	
	ETTh1-96	ETTh1-192	ETTm1-96	ETThm1-192	ETTh1-96	ETTh1-192	ETTm1-96	ETThm1-192
A.1 Llama (Default; 32)	0.362	0.398	0.272	0.310	0.448	0.484	0.346	0.373
A.2 Llama (8)	0.389	0.412	0.297	0.329	0.567	0.632	0.451	0.490
A.3 GPT-2 (12)	0.385	0.419	0.306	0.332	0.548	0.617	0.447	0.509
A.4 GPT-2 (6)	0.394	0.427	0.311	0.342	0.571	0.640	0.468	0.512
B.1 w/o Patch Reprogramming	0.410	0.412	0.310	0.342	0.498	0.570	0.445	0.487
B.2 w/o Prompt-as-Prefix	0.398	0.423	0.298	0.339	0.521	0.617	0.432	0.481
C.1 w/o Dataset Context	0.402	0.417	0.298	0.331	0.491	0.538	0.392	0.447
C.2 w/o Task Instruction	0.388	0.420	0.285	0.327	0.476	0.529	0.387	0.439
C.3 w/o Statistical Context	0.391	0.419	0.279	0.347	0.483	0.547	0.421	0.461

Table 6: Ablations on ETTh1 and ETTm1 in predicting 96 and 192 steps ahead (MSE reported). Red: the best.

- Language model variants: The scaling law retains after the LLM reprogramming. We adopt Llama-7B by default and it indeed surpasses its 1/4 capacity variant (A.2) by 14.5%. Also, an average MSE reduction of 14,7% is observed over GPT-2 (A.3), which slightly outperforms its 1/2 capacity variant (A.4).
- **Cross-modality alignment: (1)** we find that the alignment is crucial (see B.1 and B.2); **(2)** domain knowledge and task instructions are both valuable (C.1-C.3) and can be integrated via Prompt-as-Prefix (PaP)

Length		ETTh1-96		ETTh1-192			ETTh1-336			ETTh1-512	
Metric	Param. (M)	Mem. (MiB)	Speed(s/iter) Param. (M) Mem. (MiB)	Speed(s/iter)	Param. (M)	Mem. (MiB)	Speed(s/iter)	Param. (M)	Mem.(MiB)	Speed(s/iter)
D.1 LLama (32)	3404.53	32136	0.517 3404.57	33762	0.582	3404.62	37988	0.632	3404.69	39004	0.697
D.2 LLama (8)	975.83	11370	0.184 975.87	12392	0.192	975.92	13188	0.203	976.11	13616	0.217
D.3 w/o LLM	6.39	3678	0.046 6.42	3812	0.087	6.48	3960	0.093	6.55	4176	0.129

Table 7: Efficiency analysis of TIME-LLM on ETTh1 in forecasting different steps ahead.

Table 17: Efficiency comparison between model reprogramming and parameter-efficient fine-tuning (PEFT) with QLoRA (Dettmers et al., 2023) on ETTh1 dataset in forecasting two different steps ahead.

Length		E'	TTh1-96		EI	Th1-336	
Me	tric	Trainable Param. (M)	Mem. (MiB)	Speed(s/iter)	Trainable Param. (M)	Mem. (MiB)	Speed(s/iter)
Llama (8)	QLoRA	12.60	14767	0.237	12.69	15982	0.335
	Reprogram	5.62	11370	0.184	5.71	13188	0.203
Llama (32)	QLoRA	50.29	45226	0.697	50.37	49374	0.732
	Reprogram	6.39	32136	0.517	6.48	37988	0.632

 Reprogramming efficiency: (1) our reprogramming network is lightweight in activating the LLM's ability for time series forecasting (see D.3 -- i.e., fewer than 6.6M trainable parameters; only around 0.2% of the parameters in Llama-7B) (2) this is favorable even compared to parameter-efficient fine-tuning (PEFT; Tab. 17)



• **Reprogramming interpretation:** Here we provide a showcase on ETTh1 of reprogramming 48 TS patches with 100 text prototypes

The top 4 subplots visualize the optimization of reprogramming space from (a) randomlyinitialized to (d) well-optimized



We find only a small set of prototypes (columns) participated in reprogramming the input patches (rows), see subplot (e)

Also, TS patches undergo different representations through varying combinations of prototypes



Text prototypes learn to summarize language cues, and a select few are highly relevant for representing information in local TS patches, which we visualize by randomly selecting 10 in subplot (f)

TS patches usually have different underlying semantics, necessitating different prototypes to represent

Time-LLM: Summary

- Time-LLM shows promise in adapting frozen LLMs for time series forecasting by reprogramming time series data into natural language representation space more natural for LLMs and providing natural language guidance via Prompt-as-Prefix to augment reasoning
- Our evaluations demonstrate the adapted LLMs can significantly outperform many specialized expert models, indicating their potential as effective time series machines
- We provide a novel insight that time series forecasting can be cast as yet another "language" task that can be tackled by an off-the-shelf LLM to simply achieve or match SOTA performance
- We are the first to achieve "multimodal augmented time series forecasting" We can even do more with the Prompt-as-Prefix!

Part 3. Related Work & Prospects

- Other representative related works

- What's next?

LLM-related Work: GPT4TS (OFA)



LLM-related Work : LLMTime



Figure 1: We propose LLMTIME, a method for time series forecasting with large language models (LLMs) by encoding numbers as text and sampling possible extrapolations as text completions. LLMTIME can outperform many popular time series methods without any training on the target dataset (i.e. zero-shot). The performance of LLMTIME also scales with the power of the underlying base model. Notably, models that undergo alignment (e.g. RLHF) do not follow the scaling trend. For example, GPT-4 demonstrates inferior performance to GPT-3 (Section 6).

LLM-related Work : LLMTime



Figure 2: Careful tokenization is important for good forecasting with LLMs. Using the Australian Wine dataset from Darts [23], with values [151, 167, ..., 267], we show the tokenization used by GPT-3 [9] and LLaMA-2 [44] and the corresponding effect on forecasting performance. Added spaces allow GPT-3 to create one token per digit, leading to good performance. LLaMA-2, on the other hand, tokenizes digits individually, and adding spaces hurts performance.

LLM-related Work: LLMTime



Figure 5: Extended experiments on the Darts datasets. (left): Example probabilistic forecasts with baseline negative log likelihood per dimension (NLL/D). LLMs easily extrapolate trends (e.g. AirPassengers) and reproduce local patterns when data is noisy (e.g. GasRateCO2). (center): When using probabilistic metrics like NLL and CRPS, LLMTIME outperforms all baselines, including PromptCast [50], a competing LLM method. Error bars show standard errors over datasets with Darts. (right): LLMTIME is much more sample efficient than competing methods. While the performance of other methods degrades rapidly when we restrict them to a fraction of the original training set, LLMTIME can assign high likelihood with only a few examples.

LLM-related Work: LLMTime



Figure 5: Extended experiments on the Darts datasets. (left): Example probabilistic forecasts with baseline negative log likelihood per dimension (NLL/D). LLMs easily extrapolate trends (e.g. AirPassengers) and reproduce local patterns when data is noisy (e.g. GasRateCO2). (center): When using probabilistic metrics like NLL and CRPS, LLMTIME outperforms all baselines, including PromptCast [50], a competing LLM method. Error bars show standard errors over datasets with Darts. (right): LLMTIME is much more sample efficient than competing methods. While the performance of other methods degrades rapidly when we restrict them to a fraction of the original training set, LLMTIME can assign high likelihood with only a few examples.

LLM-related Work: AutoTimes



- Next token prediction
- More precise tokenwise prompting
- Overall, this is another
 step forward compared
 to Time-LLM

Liu, Y., Qin, G., Huang, X., Wang, J., & Long, M. (2024). AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. arXiv preprint arXiv:2402.02370.

LLM-related Work: AutoTimes

Table 2. Long-term forecasting results averaged from four prediction lengths $S \in \{96, 192, 336, 720\}$. The context length of our proposed method is set as 672 for all prediction lengths. Deep forecasters are respectively trained on different prediction lengths and the lookback length is also searched from $\{96, 192, 336, 512, 672, 720\}$ to report the best results. Results of other LLM-based forecasters are reported based on their original paper and "-" indicates not reported. Full results can be found in Table 9.

			LLM	IS FOR	TIME	Serie	s Met	HOD					DE	ep Foi	RECAS	ΓER		
METHOD	Аито	TIMES	Тіме	LLM	LLN	14TS	F	РТ	Uni	Тіме	ITR	ANS.	DLI	NEAR	PATC	HTST	Тімі	ESNET
	(01	URS)	(20	23)	(20	23)	(20	023)	(20)	23в)	(20	23C)	(20	23)	(20	22)	(20)22)
METRIC	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTH1	0.389	<u>0.422</u>	0.408	0.423	<u> 0.404</u>	0.418	0.427	0.426	0.442	0.448	0.438	0.450	0.423	0.437	0.413	0.431	0.458	0.450
ECL	0.159	0.253	0.159	0.253	0.159	0.253	0.167	0.263	0.216	0.305	0.161	0.256	0.177	0.274	0.159	0.253	0.192	0.295
TRAFFIC	0.374	0.264	0.388	0.264	0.401	0.273	0.414	0.294	-	-	0.379	0.272	0.434	0.295	0.391	0.264	0.620	0.336
WEATHER	0.235	0.273	0.225	0.257	0.223	<u>0.260</u>	0.237	0.270	0.253	0.276	0.238	0.272	0.240	0.300	0.226	0.264	0.259	0.287
SOLAR.	0.197	0.242	-	-	-	-	-	-	-	-	0.202	0.269	0.217	0.278	0.189	0.257	0.200	0.268







Monash University

Liu, Y., Qin, G., Huang, X., Wang, J., & Long, M. (2024). AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. arXiv preprint arXiv:2402.02370.



What's Next?



Although we have witnessed great success of pretrained models in NLP and CV, limited progress has been made for powerful general time series analysis...

Q: Can we just simply prompting a reprogrammed LLM to perform various general time series analytical tasks?



Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook

Ming Jin, Qingsong Wen[†], Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, Shirui Pan[†], Vincent S. Tseng, *Fellow, IEEE*, Yu Zheng, *Fellow, IEEE*, Lei Chen, *Fellow, IEEE*, Hui Xiong, *Fellow, IEEE*

Foundation Models for Time Series Analysis: A Tutorial and Survey

Yuxuan Liang¹, Haomin Wen^{2,1}, Yuqi Nie³, Yushan Jiang⁴, Ming Jin⁵, Dongjin Song⁴, Shirui Pan⁶, Qingsong Wen^{7†} ¹The Hong Kong University of Science and Technology (Guangzhou) ²Beijing Jiaotong University ³Princeton University ⁴University of Connecticut ⁵Monash University ⁶Griffith University ⁷Squirrel AI, USA yuxliang@outlook.com,wenhaomin@bjtu.edu.cn,ynie@princeton.edu,{yushan.jiang,dongjin.song}@uconn.edu ming.jin@monash.edu,s.pan@griffith.edu.au,qingsongedu@gmail.com

Figure 3: A comprehensive taxonomy of TSFMs, categorized according to data and methodologies.



Thank you

🤐 https://mingjin.dev/

