Exploring Large Models for Time Series

School of Software, Tsinghua University

THUML @ Tsinghua University
Machine Learning Group, School of Software, Tsinghua University

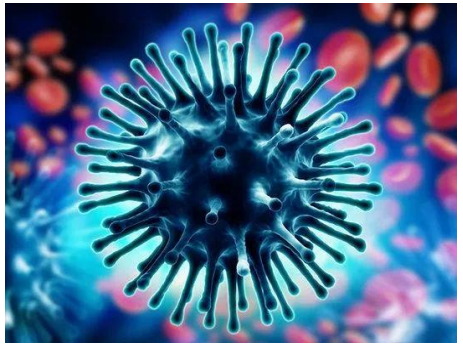
Content

- **Introduction**
- **Native Pre-trained Time Series Models**
- **Large Language Models for Time Series**
- **Limitations**
- **Resources**
  - **LTSM: Pre-Trained Checkpoint and Adaptation**
  - **OpenLTM: Open Codebase for Model Developing**

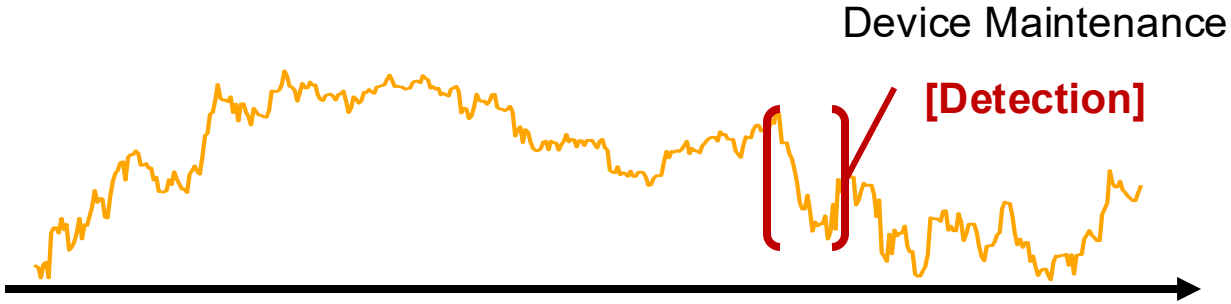
Time Series Applications

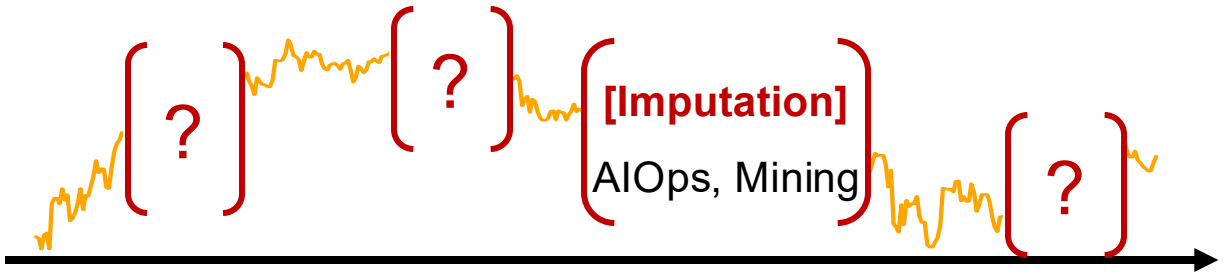






Time series is ubiquitous in the real world

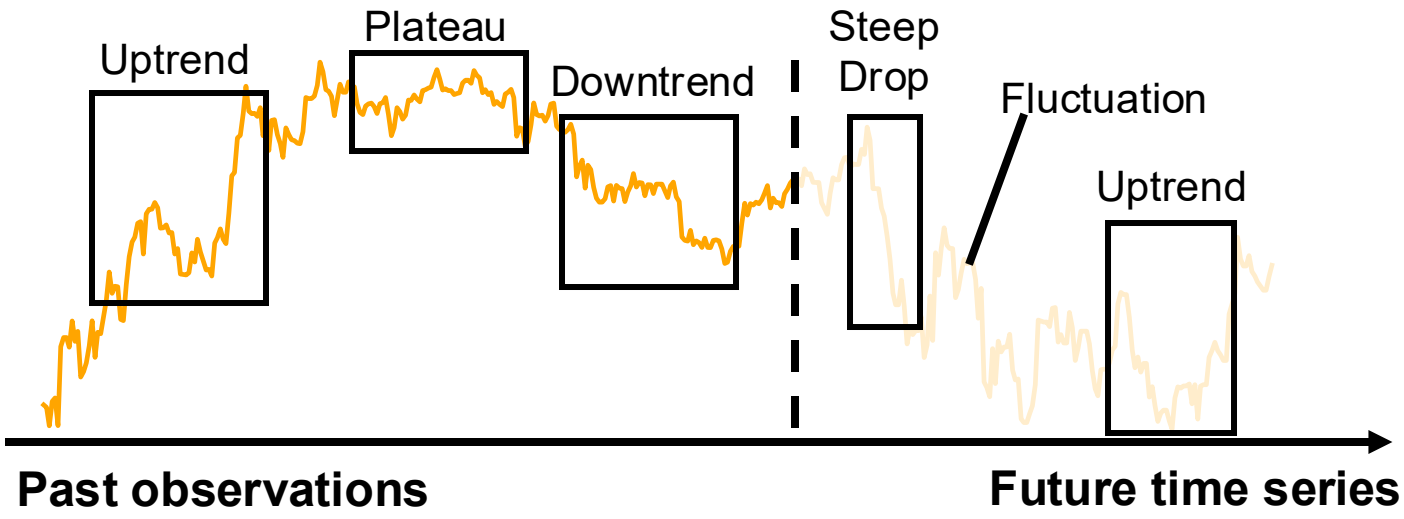


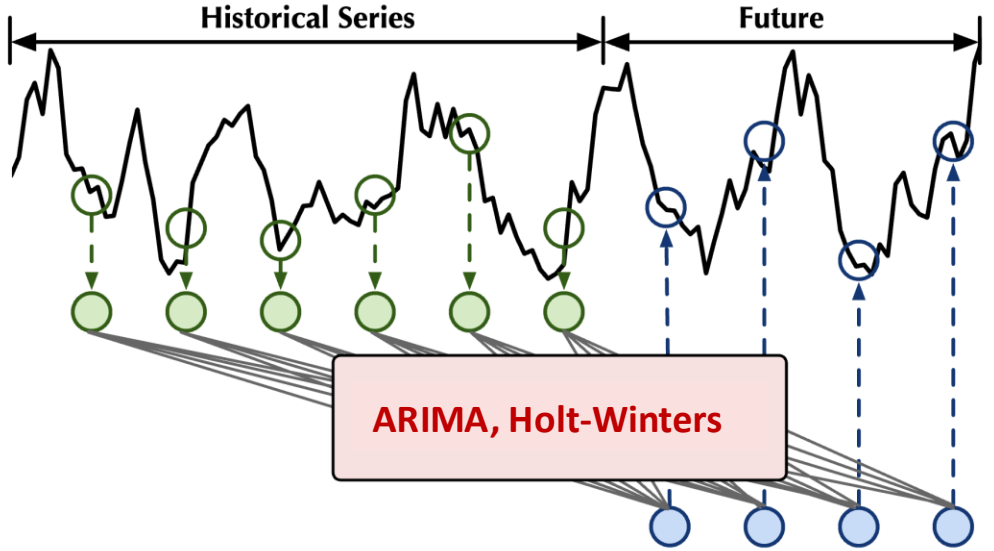


Time Series Analysis: Challenges

Increasing challenges in modern time series analysis

Complex variations (Nonlinearity)



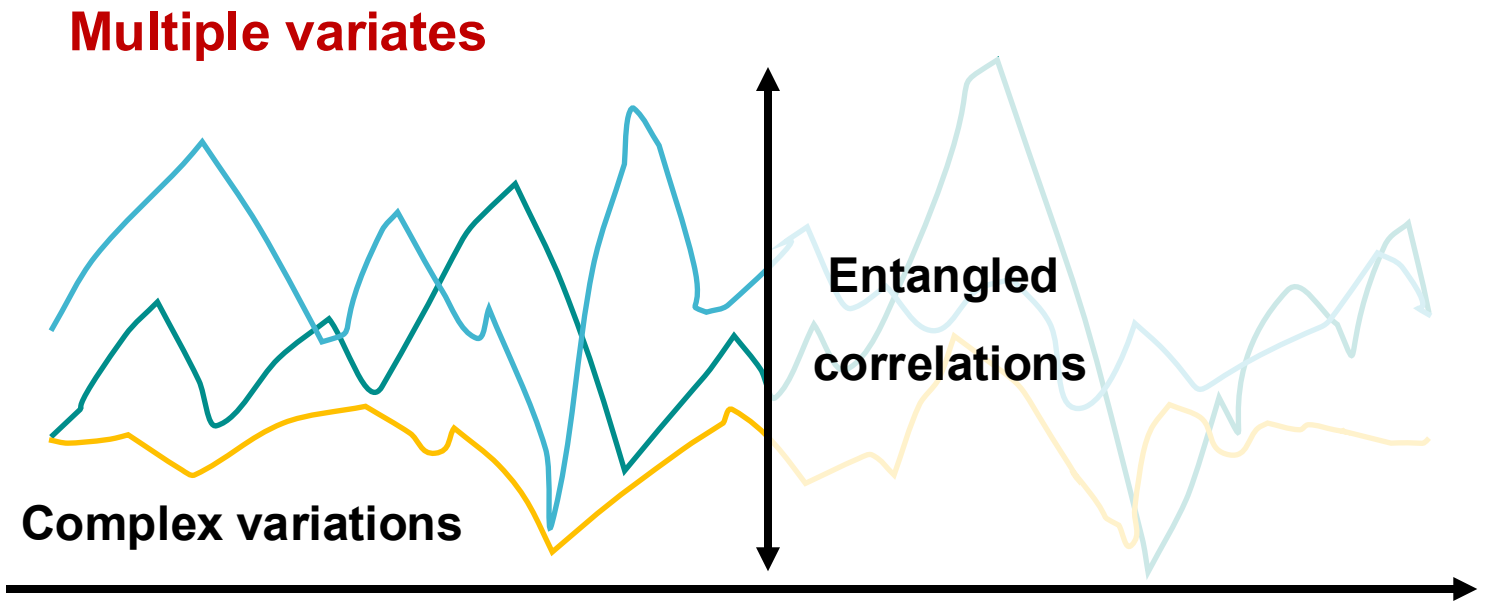


Statistical methods: may fail to capture nonlinear dependencies

Wu et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. NeurIPS 2021.

Time Series Analysis: Challenges

Increasing challenges in modern time series analysis



Clive W.J. Granger

Granger causality & Cointegration

Nobel Prize in Economics

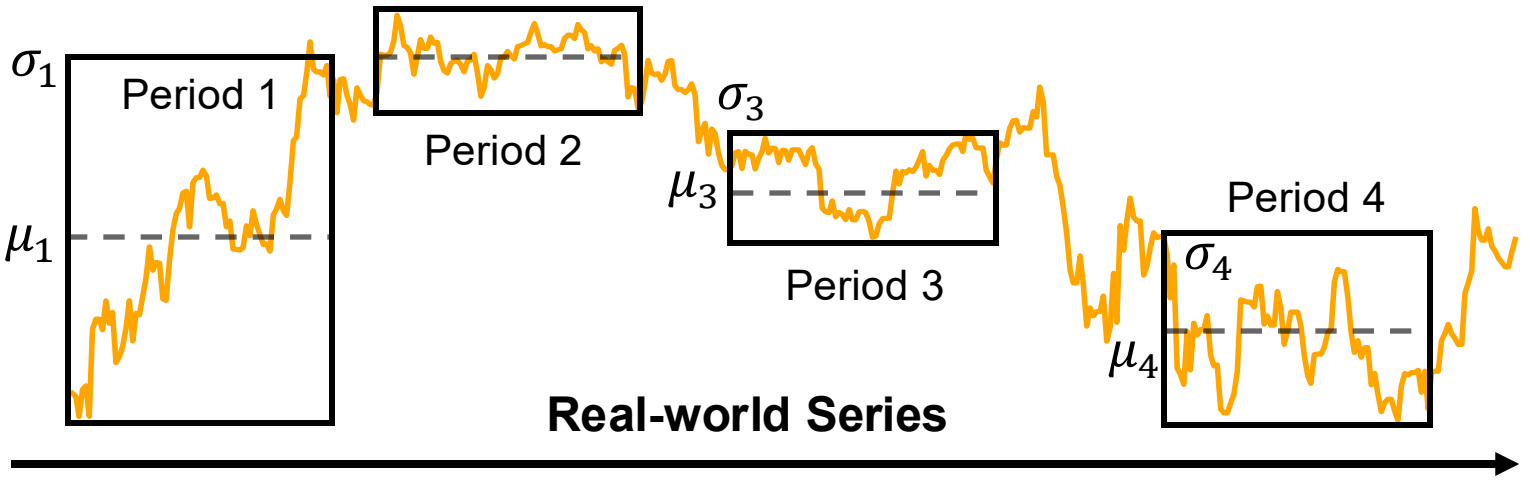
✓ **Towards powerful modeling of both time points and variates**

Liu et al. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. ICLR 2024.

Time Series Analysis: Challenges

Increasing challenges in modern time series analysis

Time-variant distribution (Non-stationarity)



Clive W.J. Granger

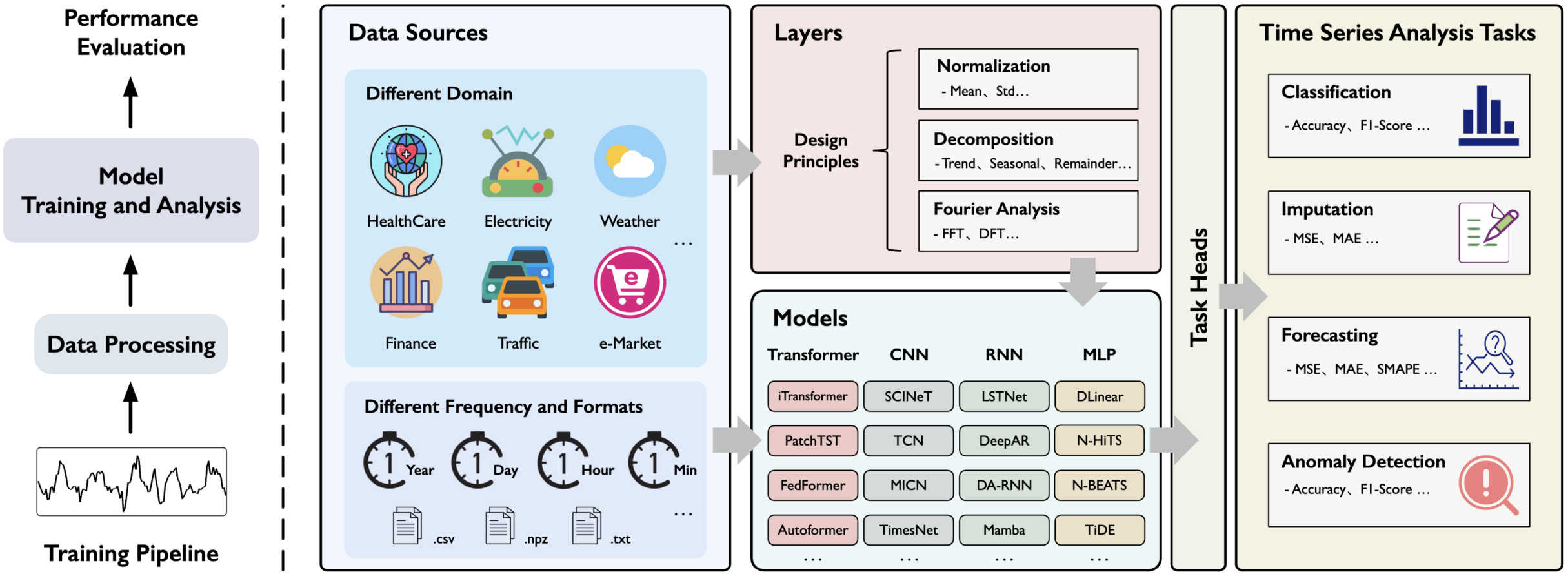
Granger causality & Cointegration

Nobel Prize in Economics

✓ **Theory-inspired / architecture-oriented non-stationary time series modeling**

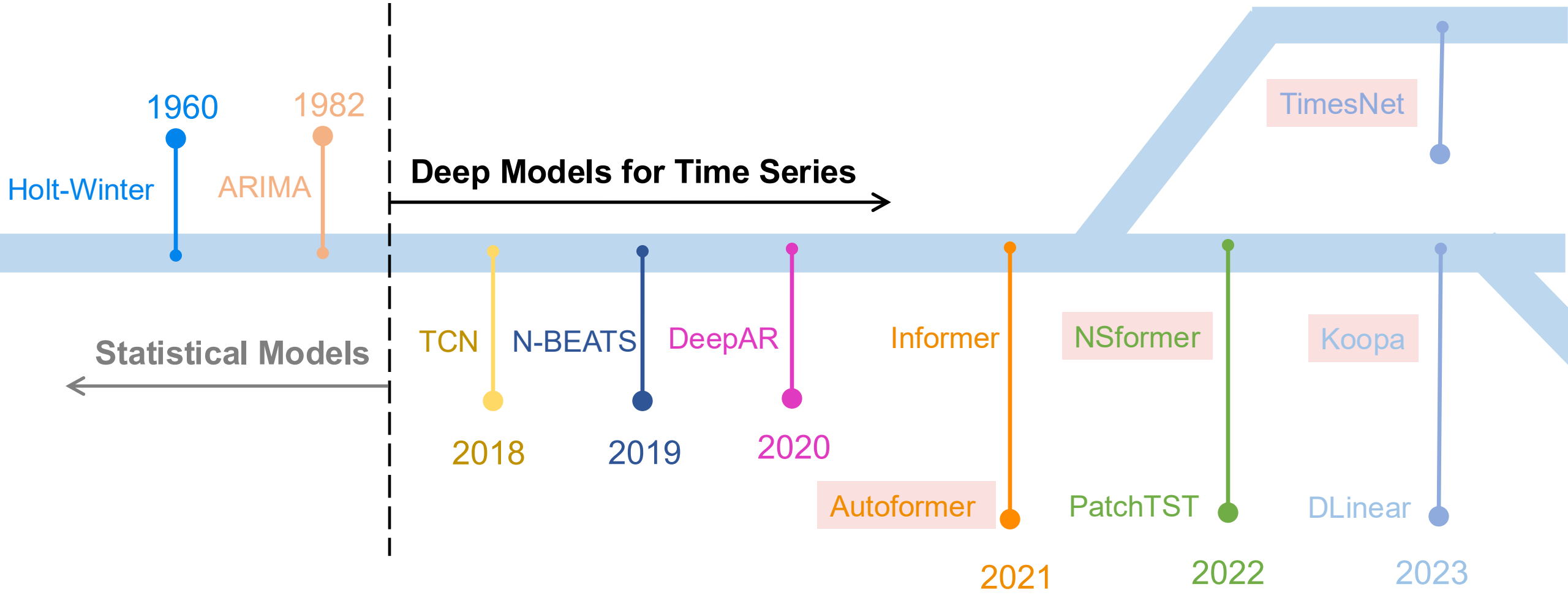
Liu et al. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. NeurIPS 2022.

Deep Models for Time Series: Pipeline



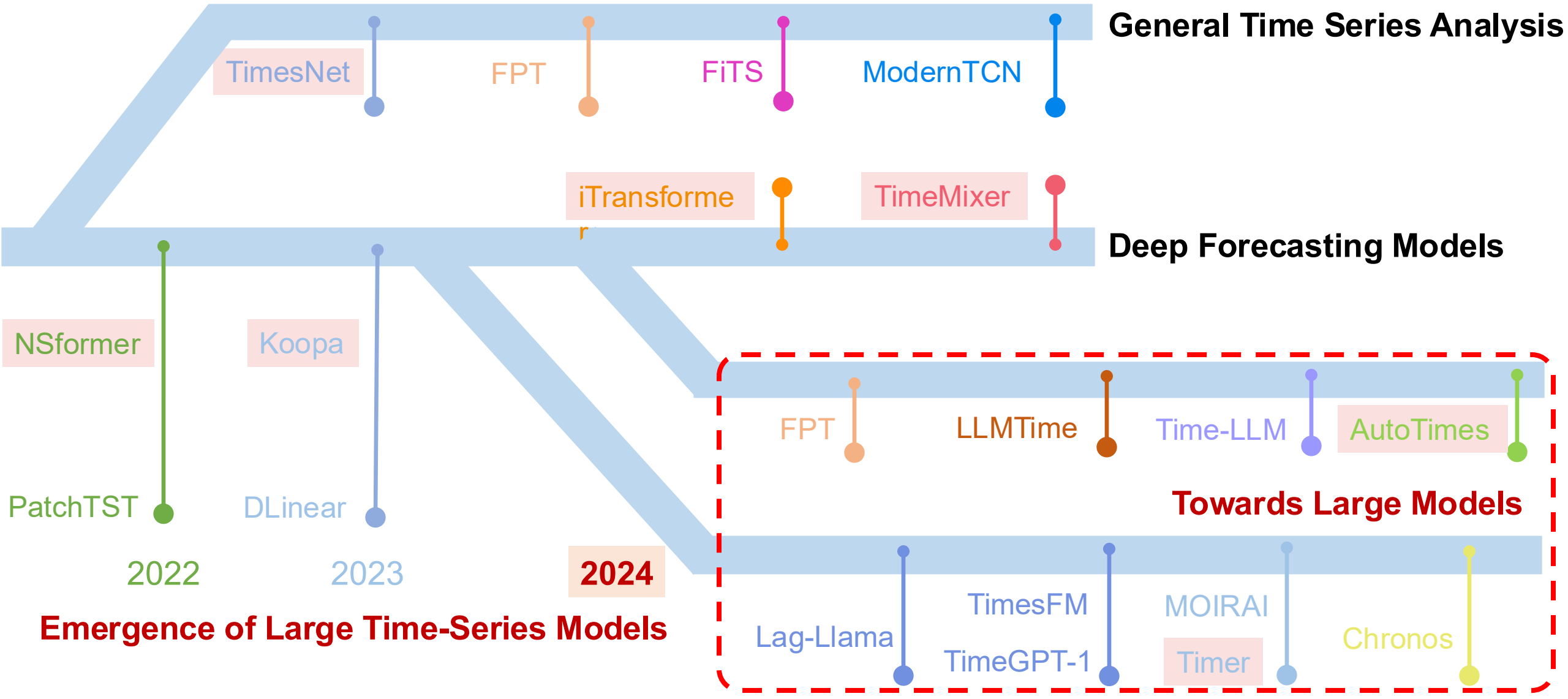
Extensively applied based on classical methodology, structural design, and end-to-end training

Deep Models for Time Series: Timeline



Deftly designed foundation backbones have advanced time series analysis

Deep Models for Time Series: Timeline



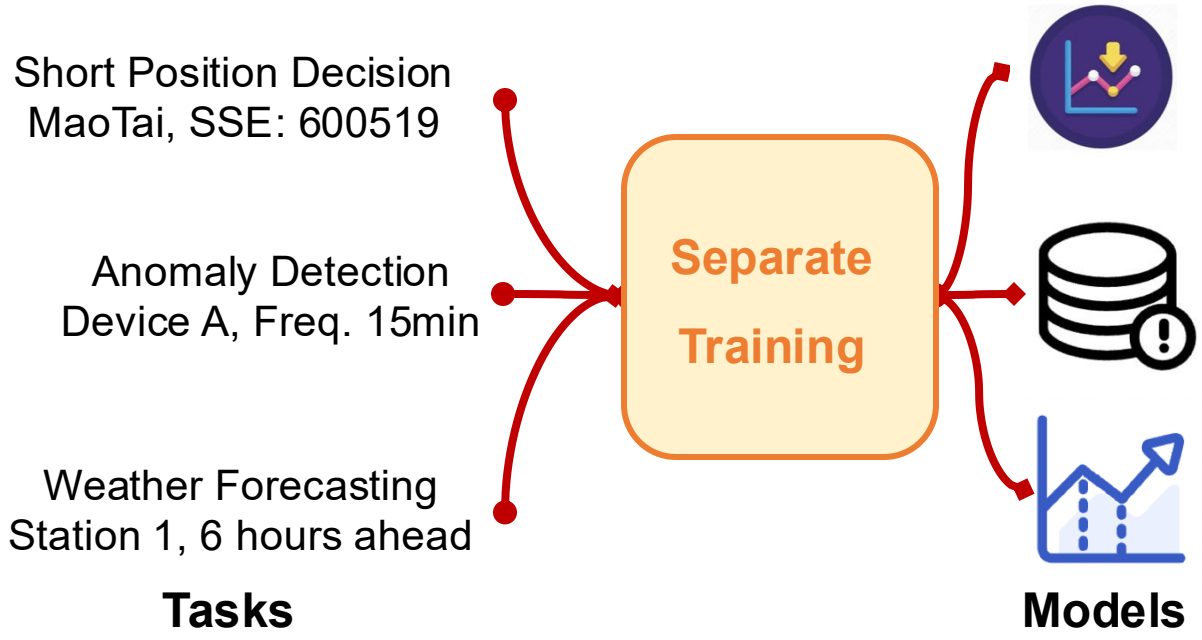
Large Time-Series Models: Motivations

☹ **Status quo:** Training models separately in specific scenarios (datasets, tasks, applications)

☹ **Data scarcity is common and challenging in real-world applications**

- Training samples are expensive and sometimes inaccessible
- Performance degrades greatly with limited samples

Nontransferable!

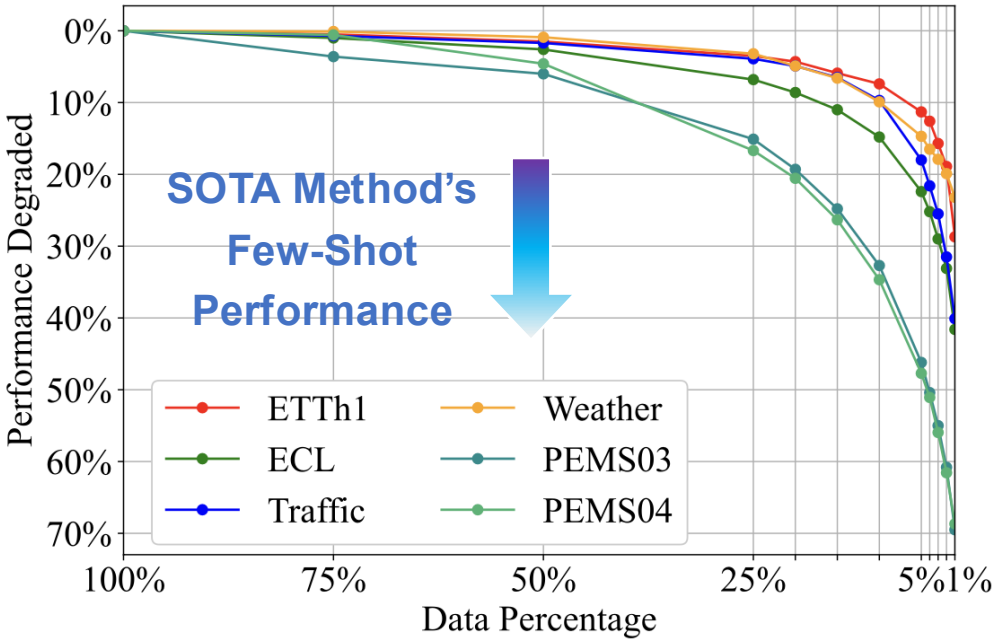


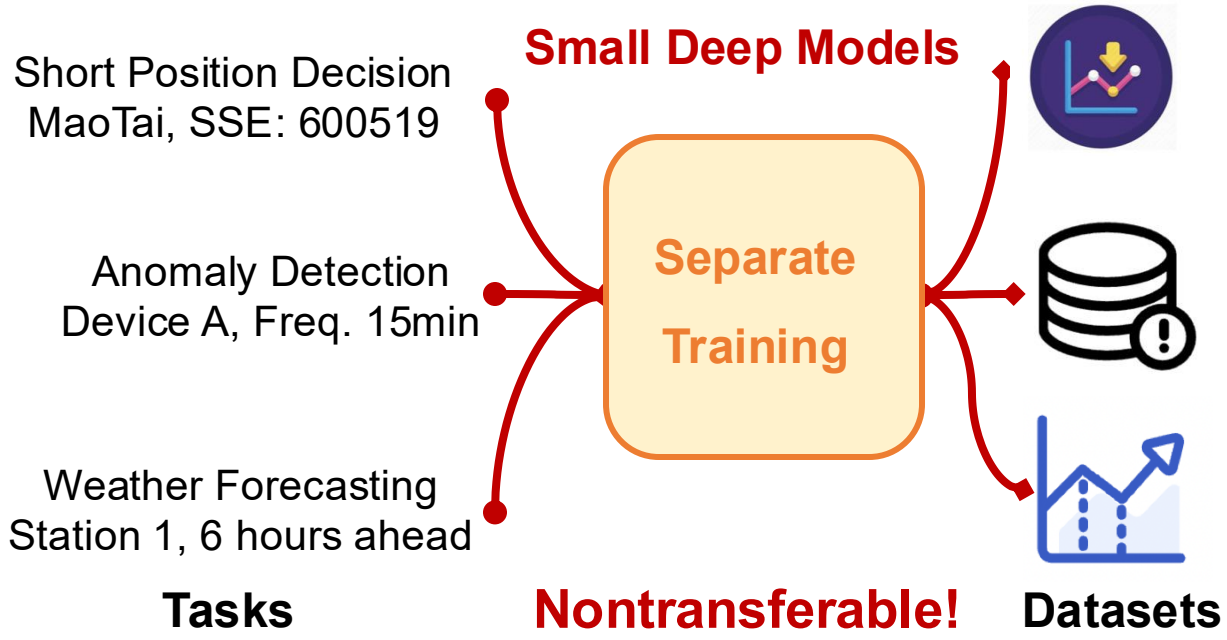
Large Time-Series Models: Motivations

☹ **Status quo:** Training models separately in specific scenarios (datasets, tasks, applications)

☹ **Data scarcity is common and challenging in real-world applications**

- Training samples are expensive and sometimes inaccessible
- Performance degrades greatly with limited samples

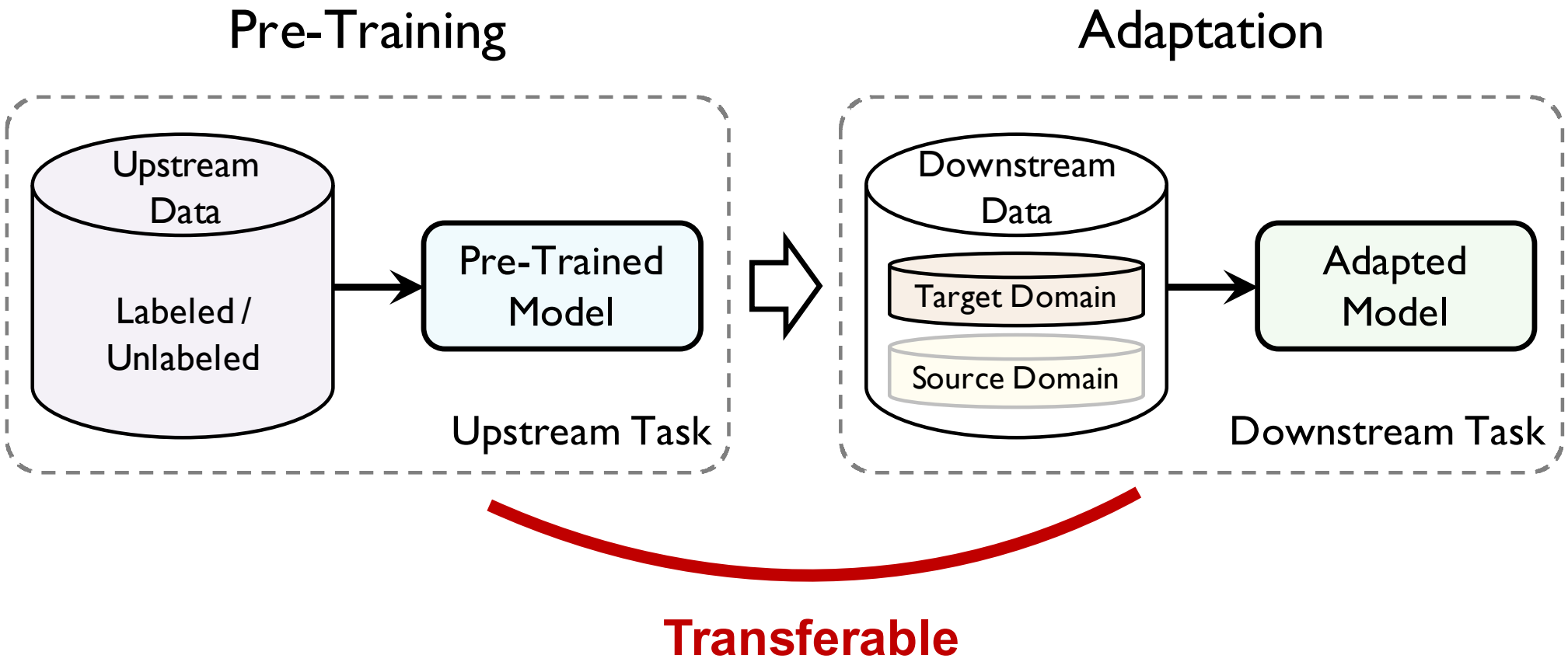




Large Time-Series Models: Capabilities

What is a Large Model

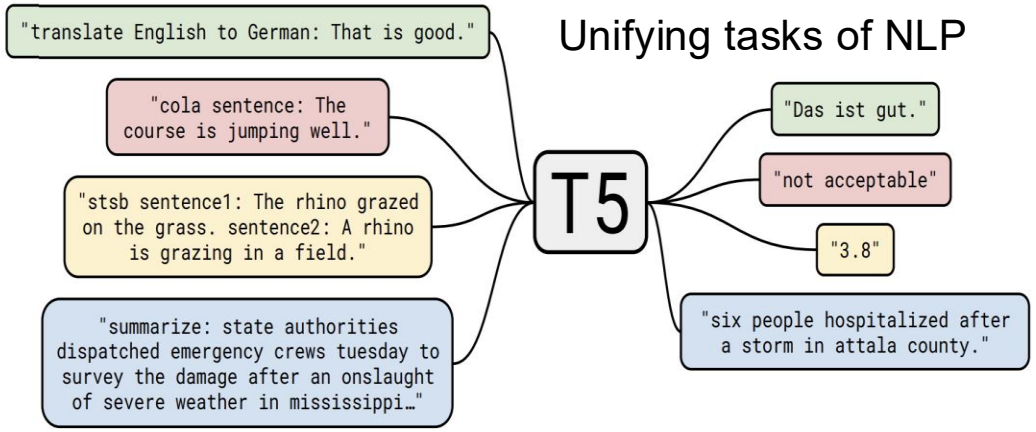
✓ **Generalizability:** One model fits different domains

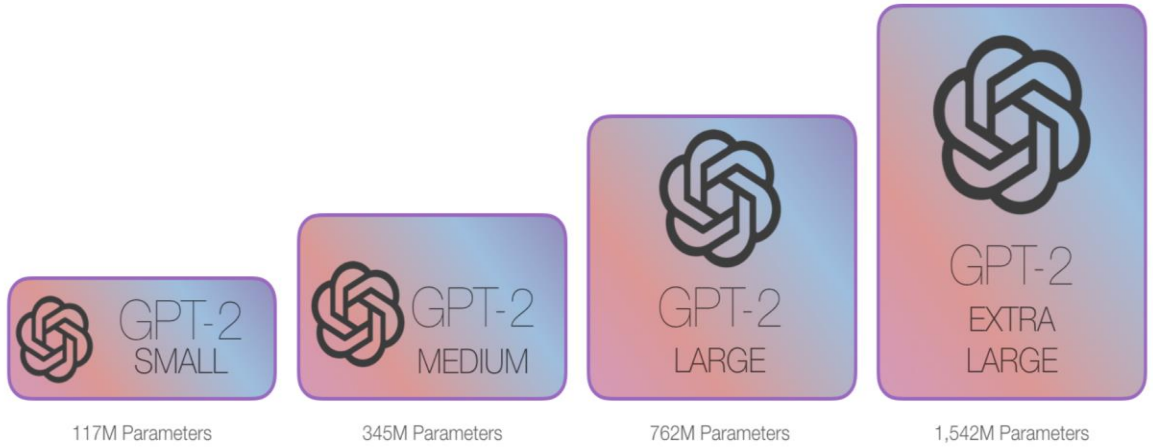


Large Time-Series Models: Capabilities

What is a Large Model

- ✓ **Generalizability:** One model fits different domains
- ✓ **Task Generality:** Versatility to tackle various scenarios / tasks
- ✓ **Scalability:** Performance improves with the scaling



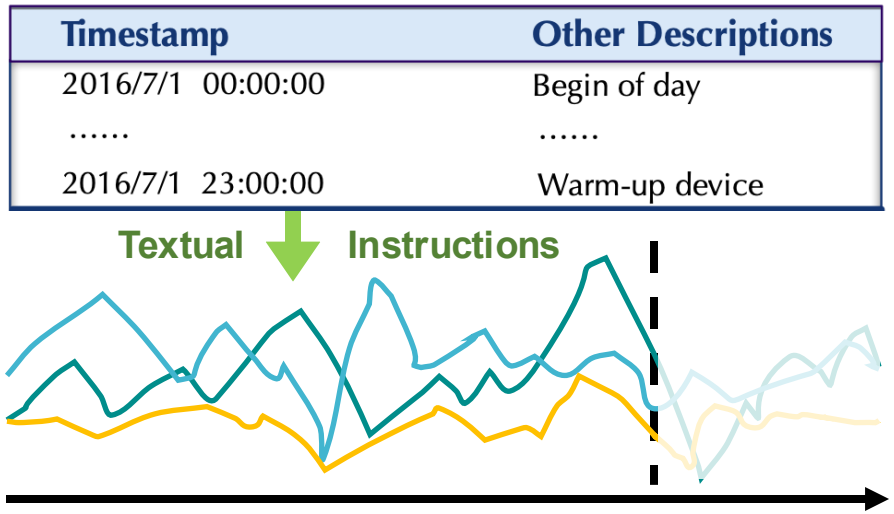


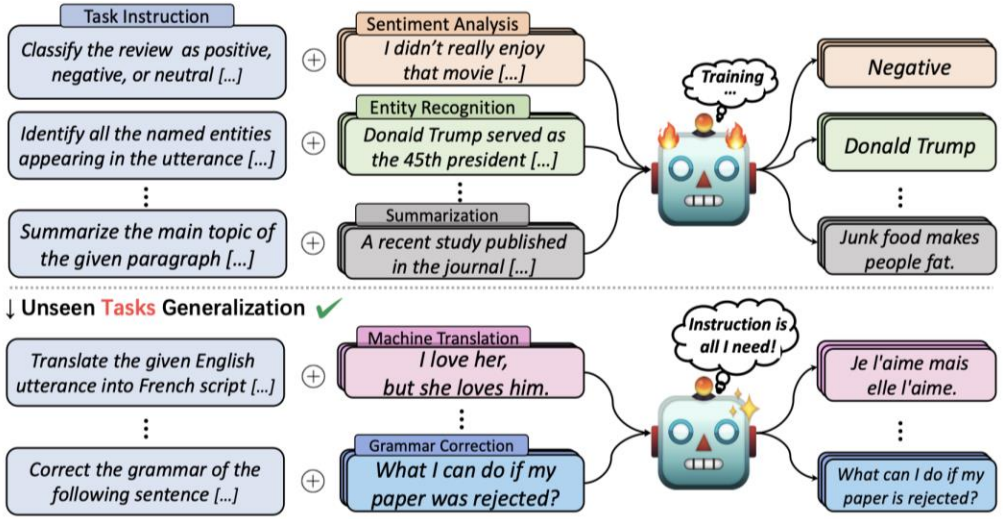
Raffle et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR 2020.

Large Time-Series Models: Capabilities

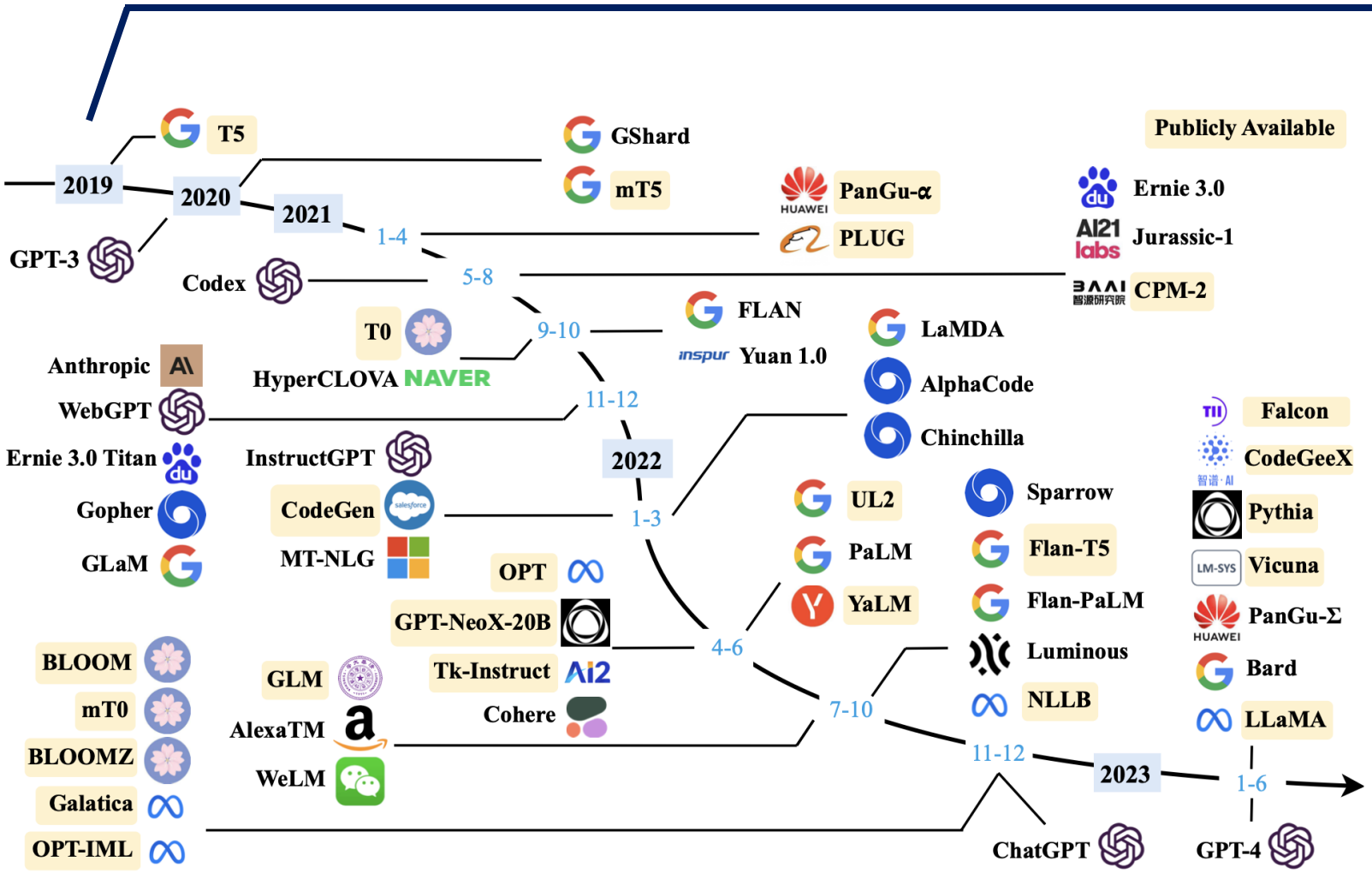
What is a Large Model

- ✓ **Generalizability:** One model fits different domains
- ✓ **Task Generality:** Versatility to tackle various scenarios/tasks
- ✓ **Scalability:** Performance improves with the scaling
- ✓ **Emergence Abilities:** Multimodality, instruction following…





Large Language Models: Timeline



Large Model for Time Series Are Still in Early Stages

Challenges

- ☐ Data Infrastructure
- ☐ Scalable Architecture
- ☐ Model Versatility

Zhao et al. A Survey of Large Language Models. arXiv 2023.

Large Time-Series Models: Basic Approaches

Two Approach to Develop Large Model for Time Series

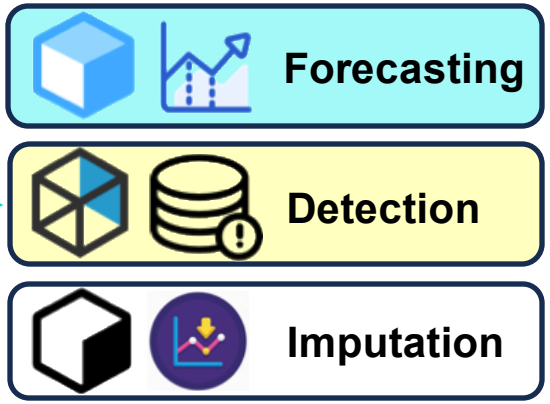
Domain-Specific / Domain-Universal Time-Series Data

Diverse Time Series

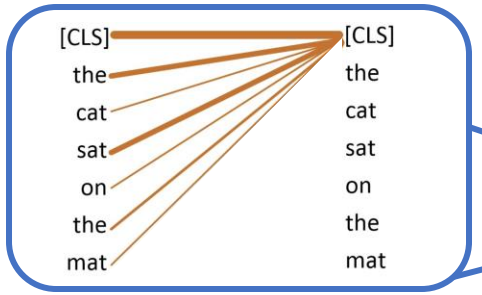
Large-Scale Pre-Training

Model Adaption

Various Applications

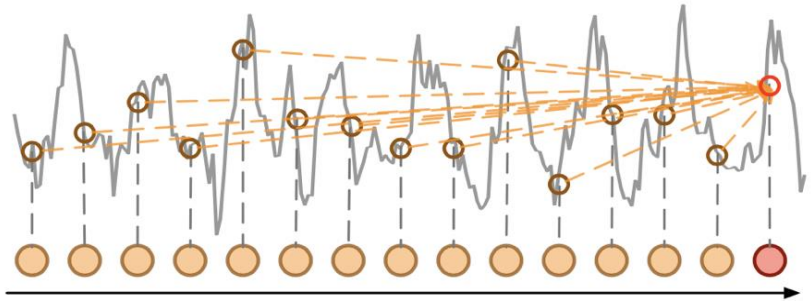


Large Time-Series Model



LLM for Time Series

Adaptation

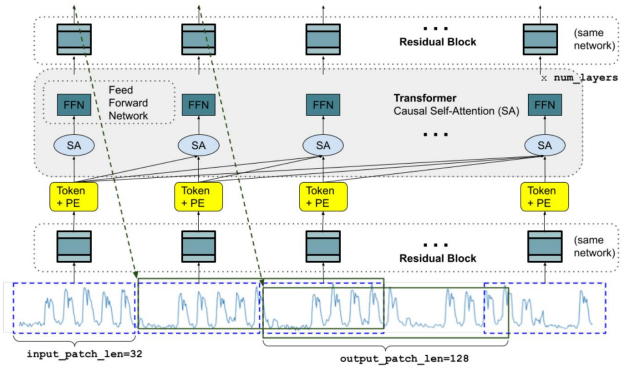


- **Language Dependencies**
- **Token Semantics**

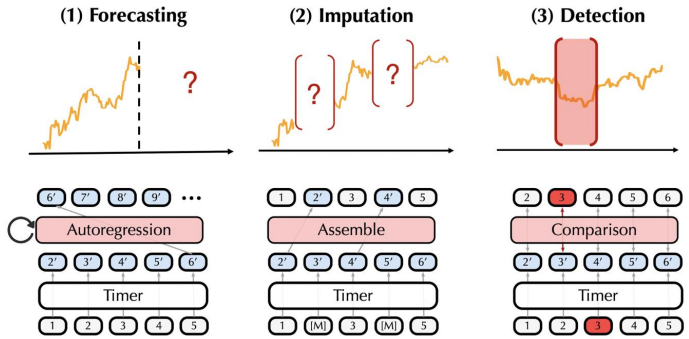
- **Time Points Dependencies**
- **Series Semantics**

Native Pre-Trained LTM

Decoder-Only

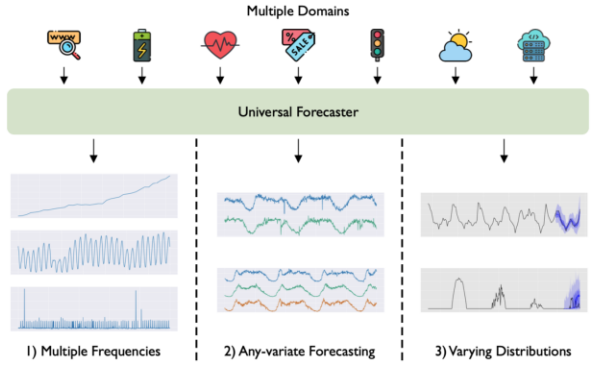


TimesFM (Google)

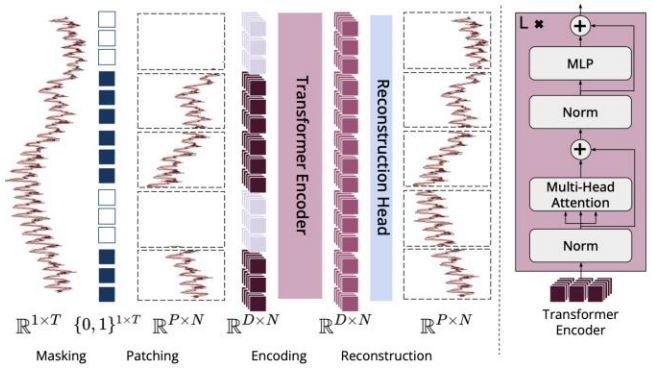


Timer (Tsinghua)

Encoder-Only

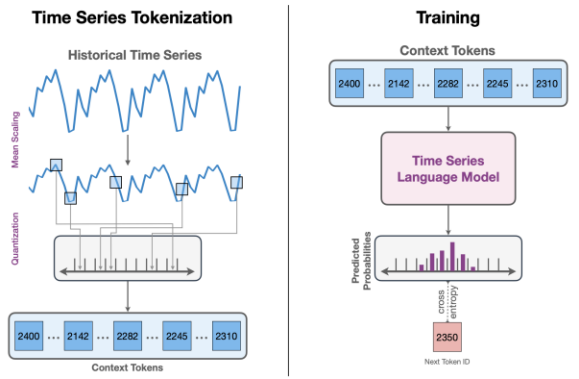


MOIRIA (SalesForce)

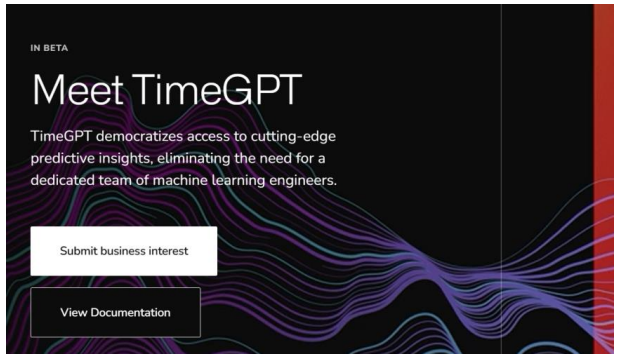


Moment (CMU)

Encoder-Decoder



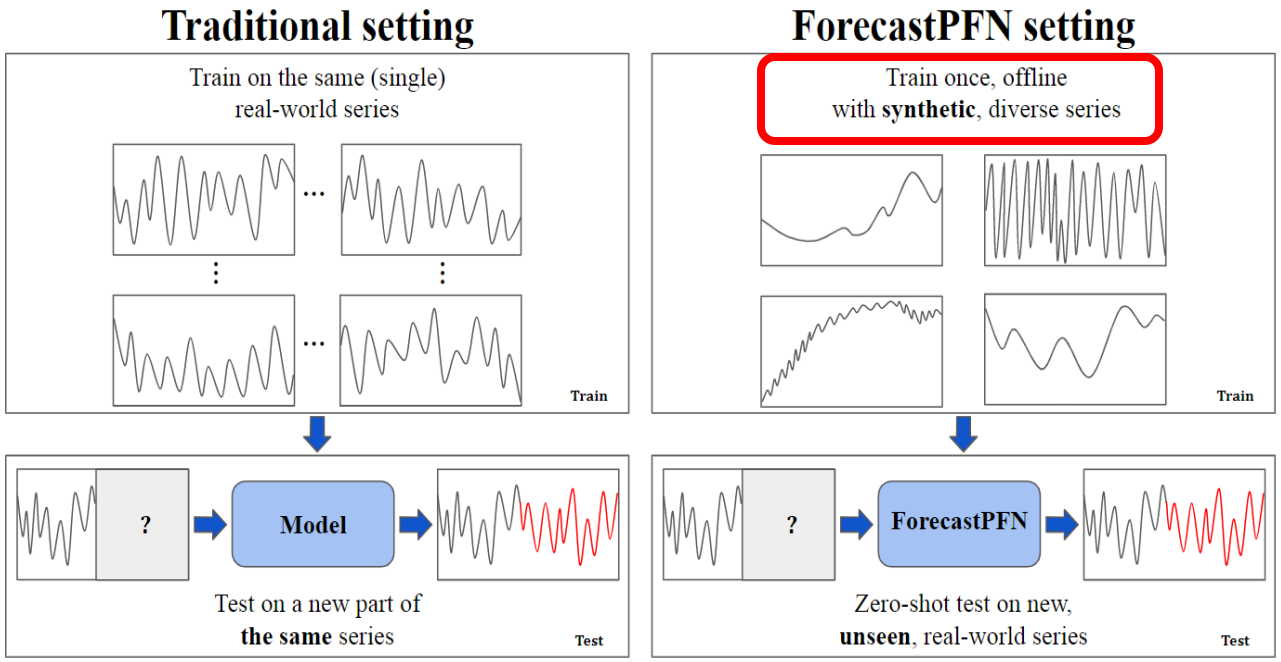
Chronos (Amazon)



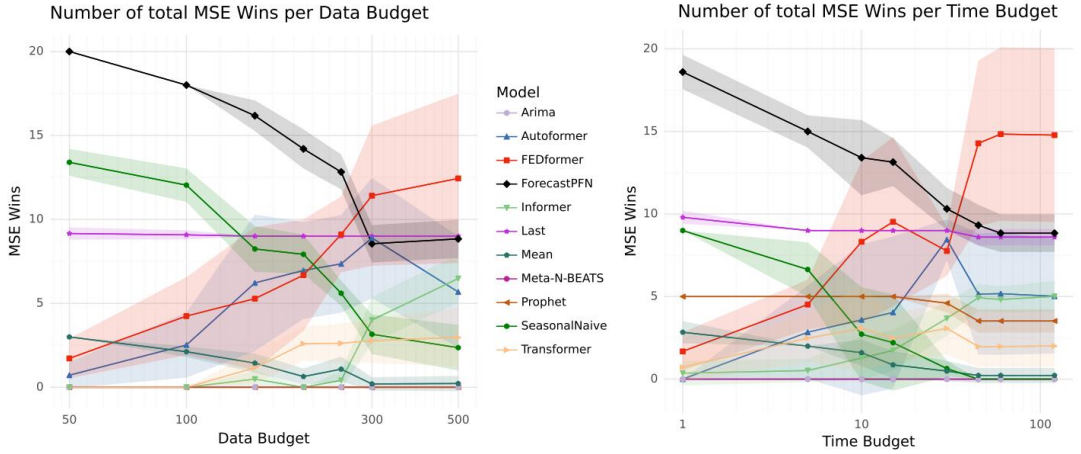
TimeGPT-1 (Nixtla)

ForecastPFN: Pre-trained on Synthetic Time Series

- **Lack of high-quality time series corpora**
- Completely pre-trained on **synthetic data** generated by prior distributions and mixups



- ✓ Support **zero-shot forecasting without downstream training**
- ✓ Give **probability predictions**
- ✗ Human-like / Earth-like time series might be out of scope



Dooley et al. ForecastPFN: Synthetically-Trained Zero-Shot Forecasting. NeurIPS 2023.

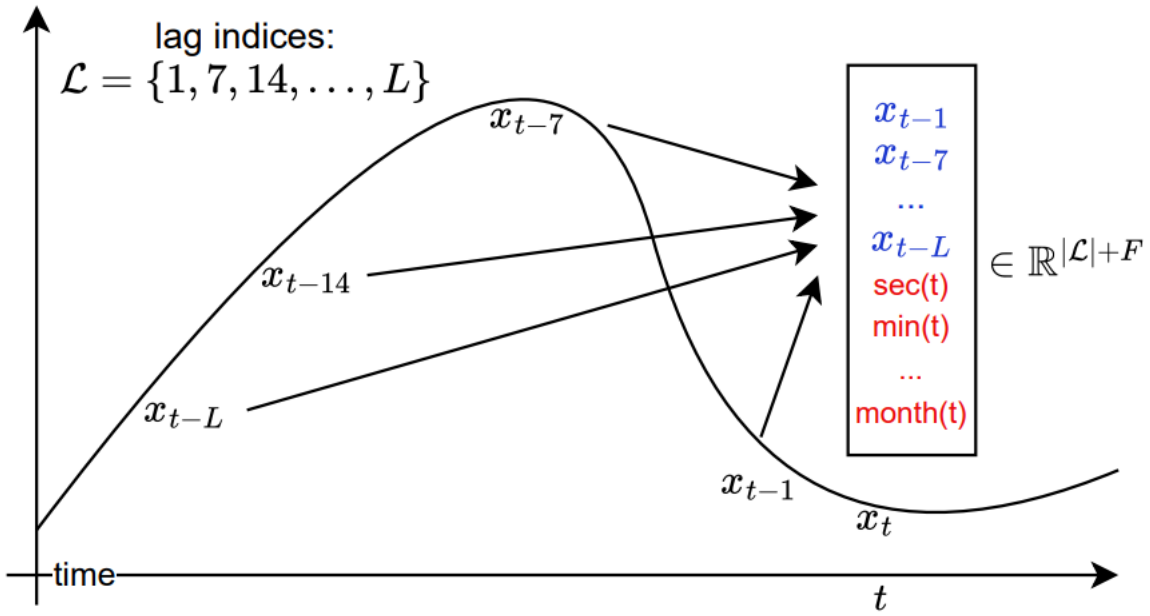
Lag-Llama: Probabilistic Univariate Forecaster

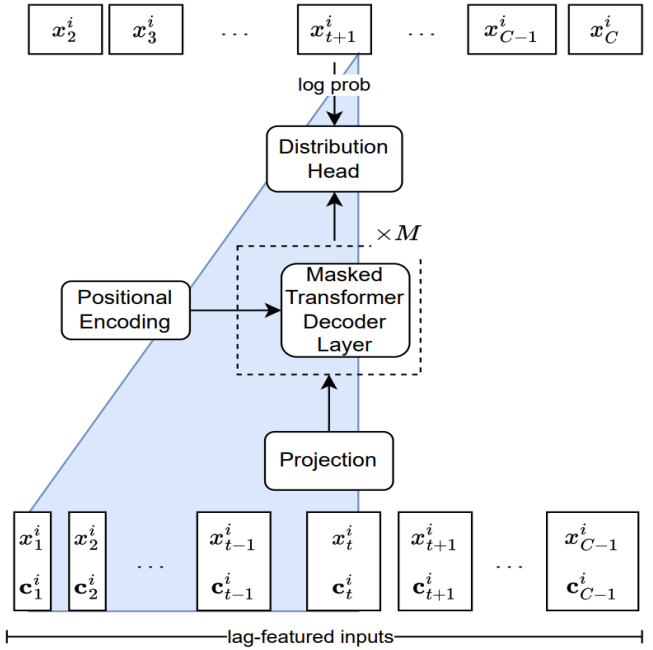
- Training on **real-world** time series (360M)
- Based on LLaMA, **encoding lagged values**
- Generate only one time point at one step

✓ Support **zero-shot forecasting** on univariate time series

✓ Better performance with **fine-tuning**

✗ Inference with error accumulations

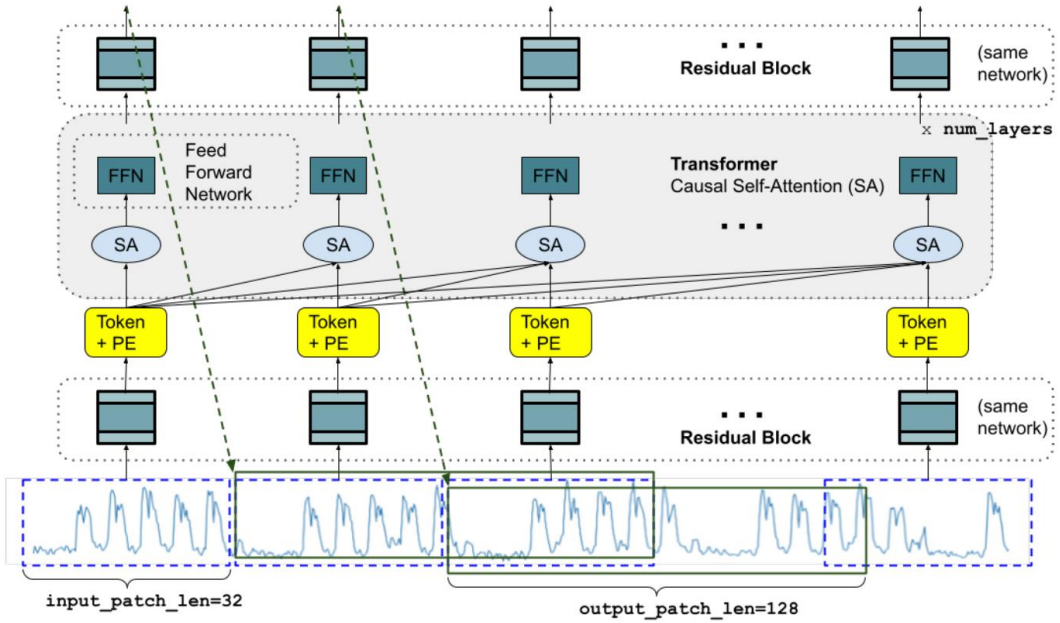




Rasul et al. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. arXiv 2023.

TimesFM: LTM Developed by Google

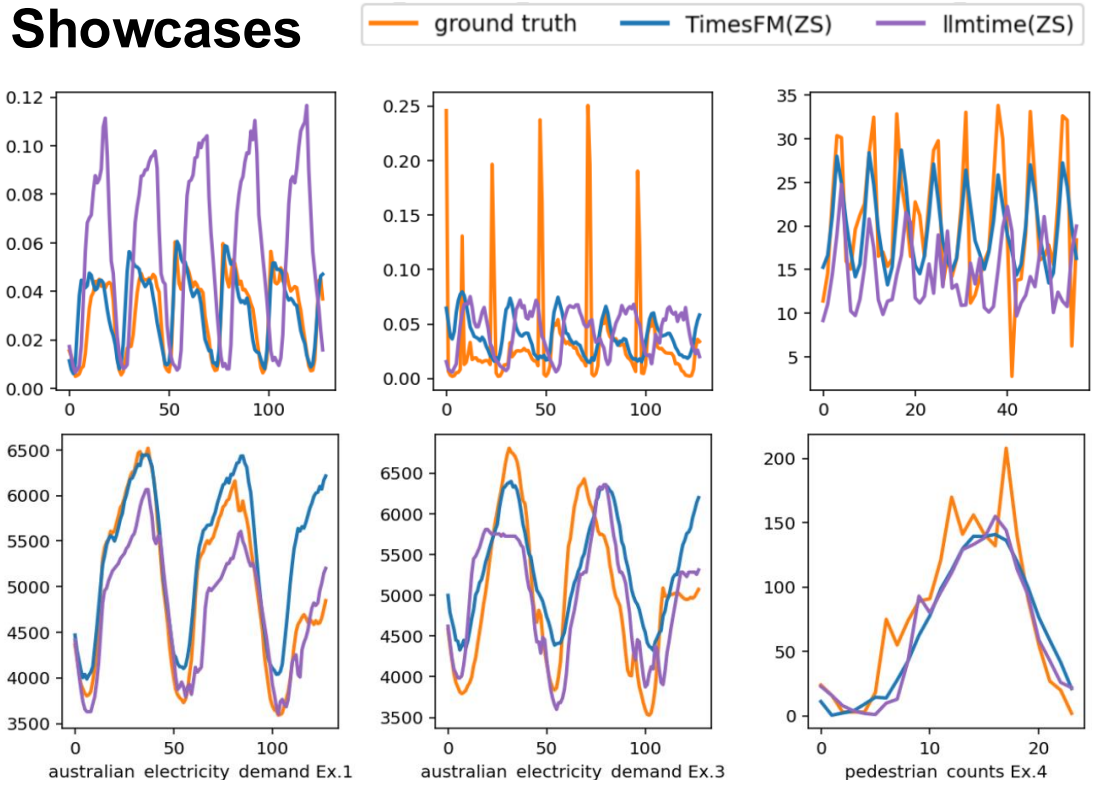
- Architecture: **Decoder-only Transformer**
- Dataset: **100 billion time points,** using Google Trends and Wiki Page View
- Parameter Counts: **200M**



✓ Greatly **enlarged training scale**

✓ **Zero-shot** and **patch-level** predictions

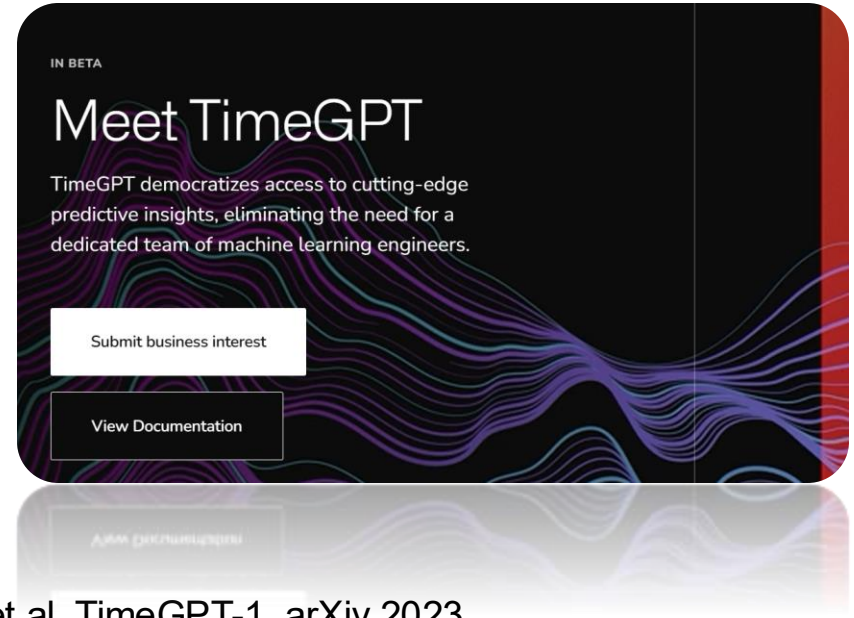
Showcases



Das et al. A Decoder-Only Foundation Model for Time-Series Forecasting. ICML 2024.

TimeGPT: First Commercialized LTM

- **Production:** a large time-series model launched by Nixtla
- Pre-training on **100B time points** from diverse datasets and domains
- **Release API call for inference**

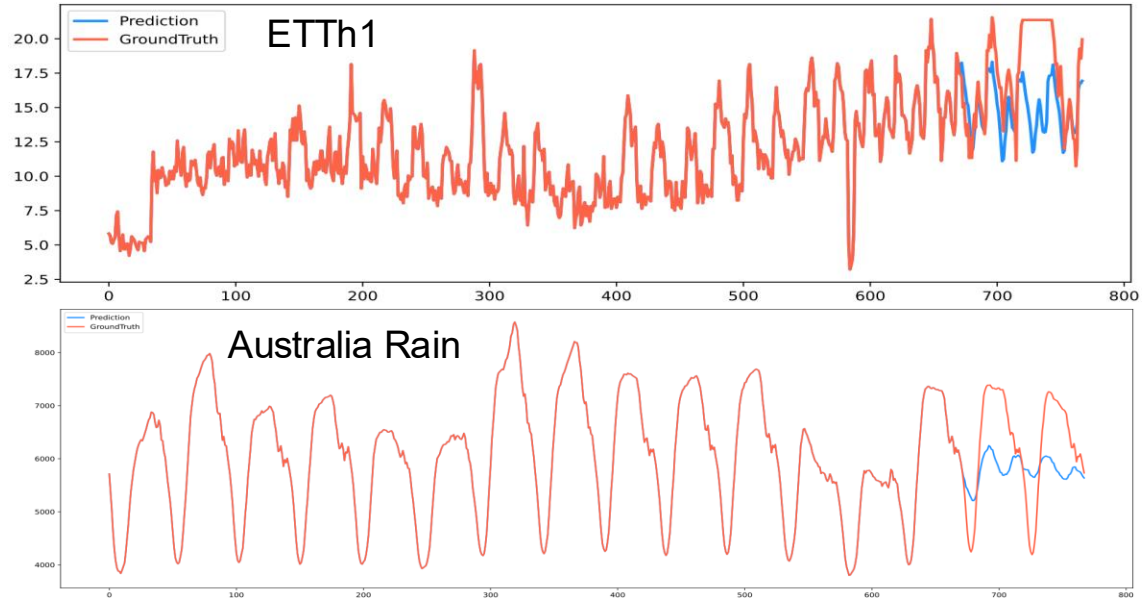


✓ Support anomaly detection, forecasting with **covariates and probabilities**

✓ **Zero-shot / Finetune** on given data

✗ Unrevealed architecture and training details

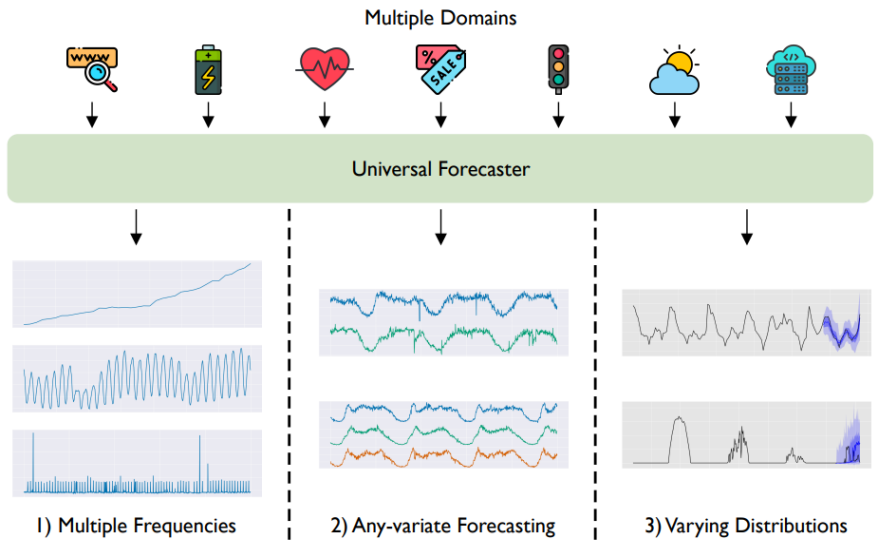
Showcases



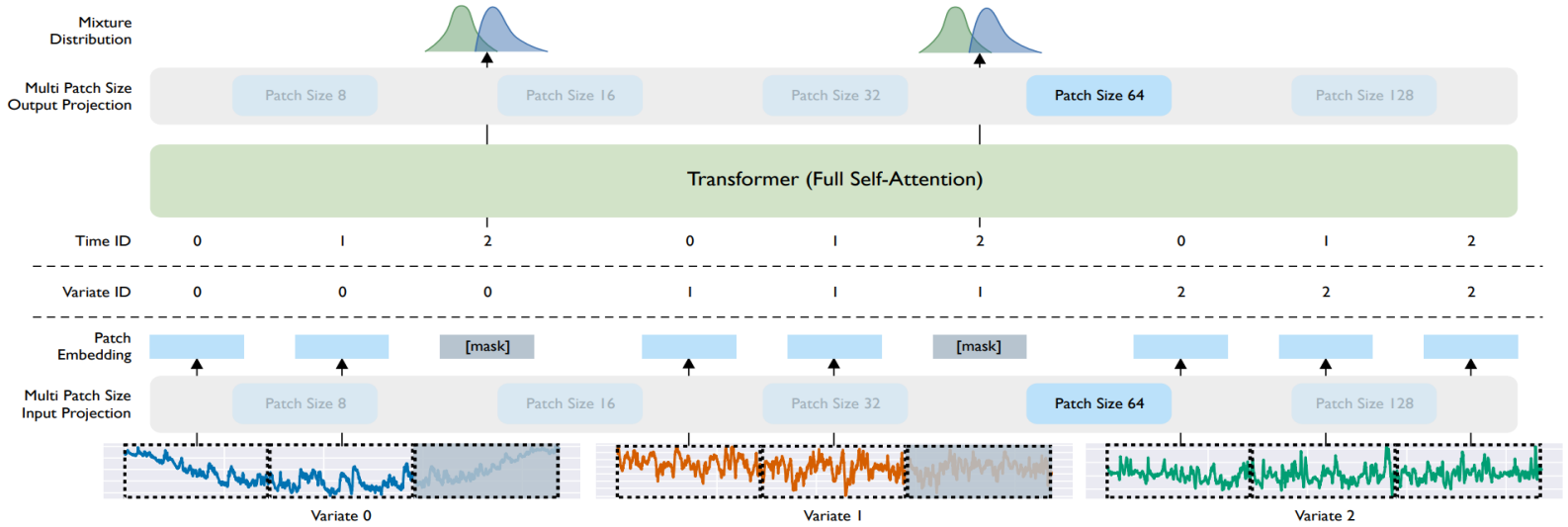
Graza et al. TimeGPT-1. arXiv 2023.

MOIRAI: LTM for Multiple Variates

- ✓ Accommodate **vary-frequency** time series and suitable for **multivariate time series**
- ✓ Able to generalize on a variety of time series based on pre-defined mixture distributions



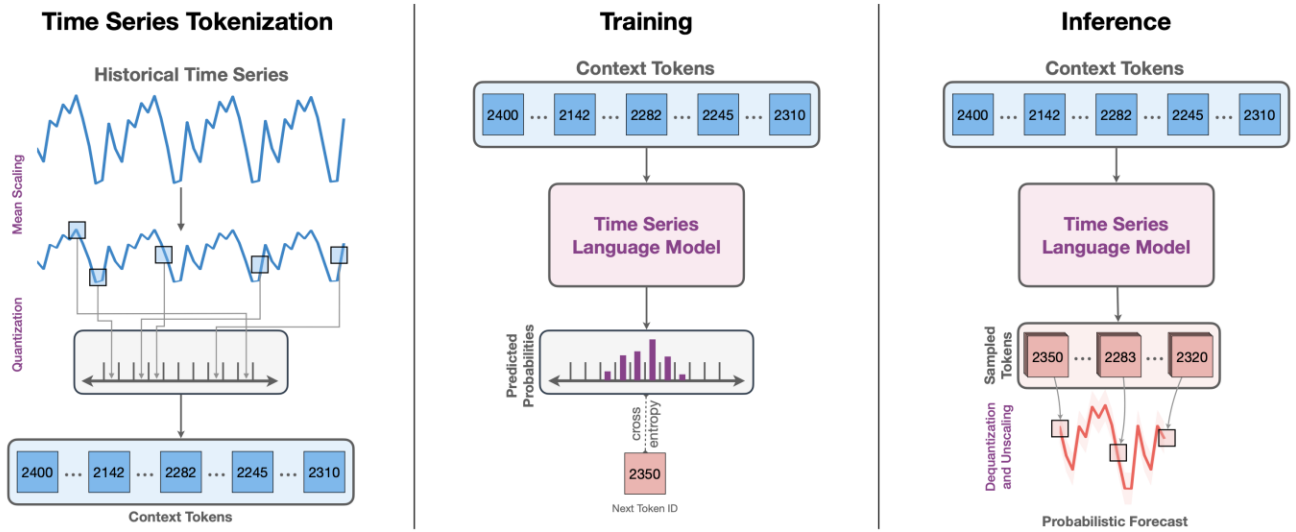
- Parameter Counts: **14M ~ 311M**
- Dataset: **27B**
- Method: **MLM on Encoder-only Trm.**

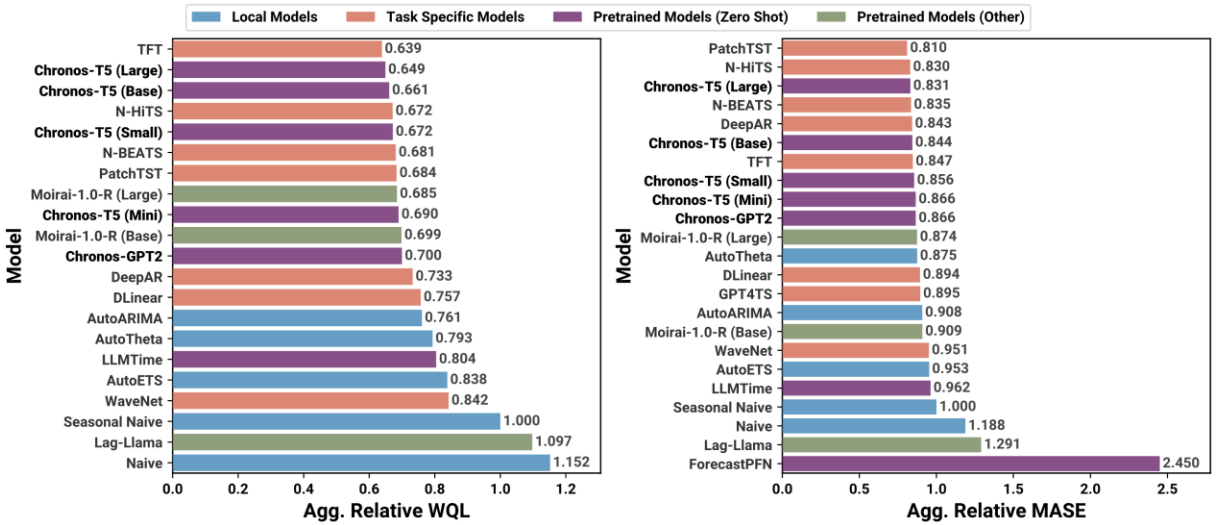


Woo et al. Unified Training of Universal Time Series Forecasting Transformers. ICML 2024.

Chronos: Learning the Language of Time Series

- Quantizing continuous time points to **discrete words** based on T5
- Training on mixup augmentation and Gaussian mixture synthesis





- ✓ Good performance on zero-shot forecasting for **short-term outputs**
- ✓ Give **probability predictions**
- ✗ **Point-level autoregression:** suffer from error accumulation

Woo et al. Chronos: Learning the Language of Time Series. ICML 2024.

Timer (Ours): Task-General LTM

ICML International Conference On Machine Learning

Timer: Generative Pre-trained Transformers Are Large Time Series Models

Yong Liu [*1] **Haoran Zhang** [*1] **Chenyu Li** [*1] **Xiangdong Huang** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Yong Liu

Haoran Zhang

Chenyu Li

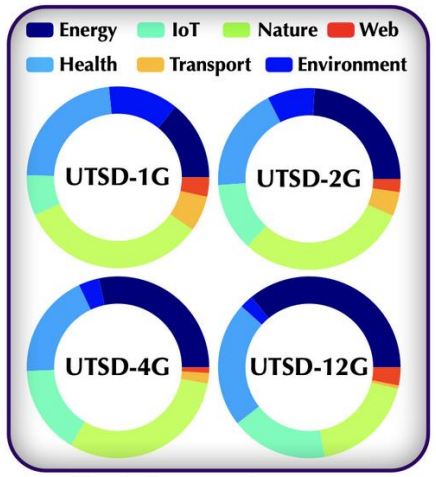
Xiangdong Huang

Jianmin Wang

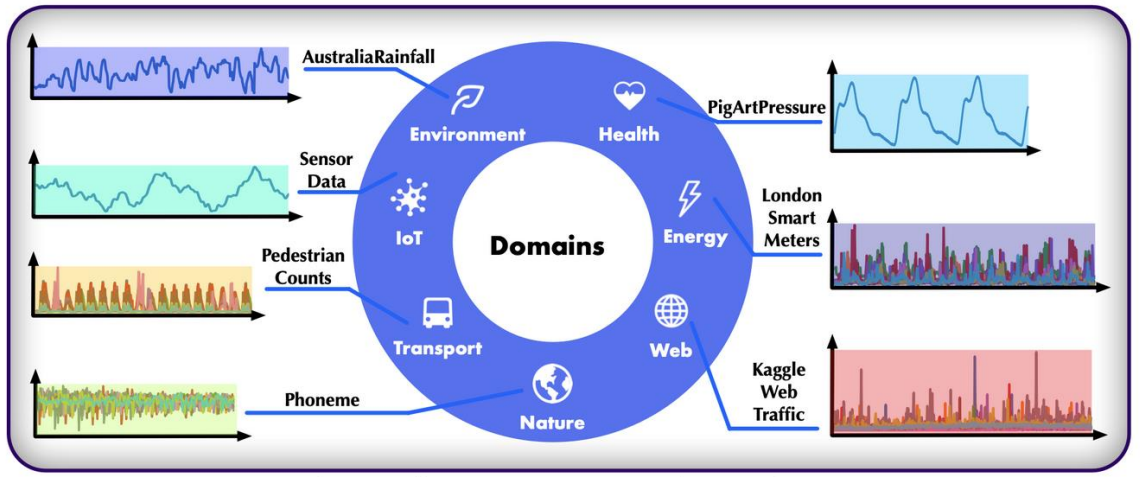
Mingsheng Long

Timer: Well-curated Datasets

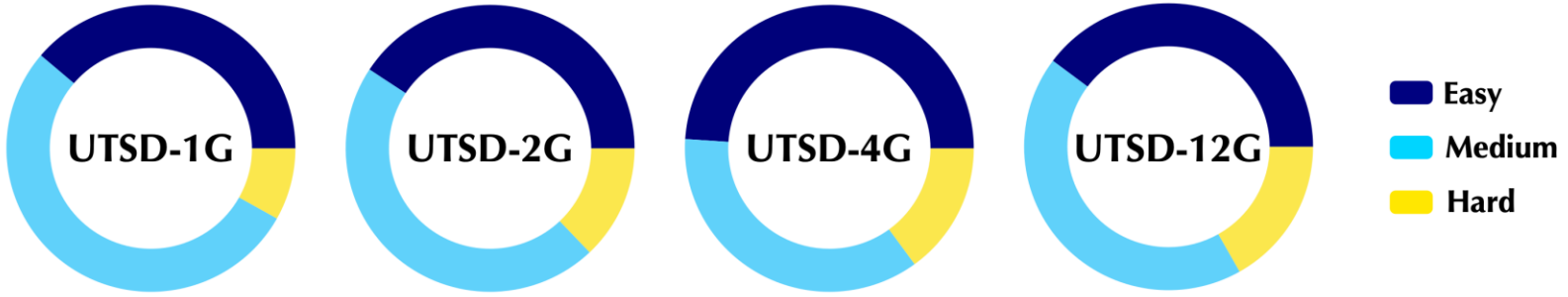
☐ Aspect 1: Unified Time Series Dataset



Unified Time Series Dataset



Diverse Shapes, Patterns and Domains



Dataset: https://huggingface.co/datasets/thuml/UTSD

Data quality is also important!

- **Aggregation & Filter**
- **Preprocess & Evaluate**
- **Stacking up with a hierarchy**

- **1 Billion Time Points**
- **7 Typical Domain**
- **4 Scalable Volums**
- **Continuous Expansion…**

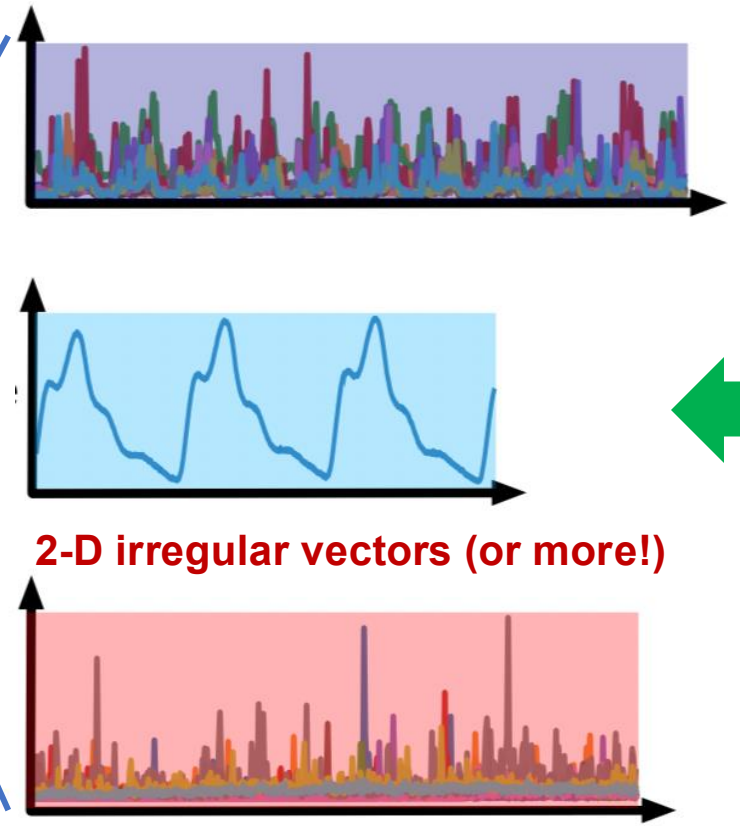
Timer: Issue of Data Heterogenity

- **Aspect 2: Unified format to address data heterogeneity**

Distinct in Shape/Freq/Scale!

Dataset	Dim	Frequency
ETTh1, ETTh2	7	Hourly
ETTm1, ETTm2	7	15min
Exchange	8	Daily
Weather	21	10min
ECL	321	Hourly
Traffic	862	Hourly
Solar-Energy	137	10min
PEMS03	358	5min
PEMS04	307	5min
PEMS07	883	5min
PEMS08	170	5min

Intractable for scalable training!



2-D irregular vectors (or more!)

No Neat as Natural Language

The patient is a frail 88-year-old caucasian male was admitted to our hospital for complaints of nausea and vomiting and suspected urinary tract infection.

He has a past medical history of hypertension, atrial fibrillation and chronic right hip pain after total hip replacement in 2012.

The patient was started on antibiotics. Urine culture confirmed an E. coli urinary tract infection sensitive to trimethoprim.

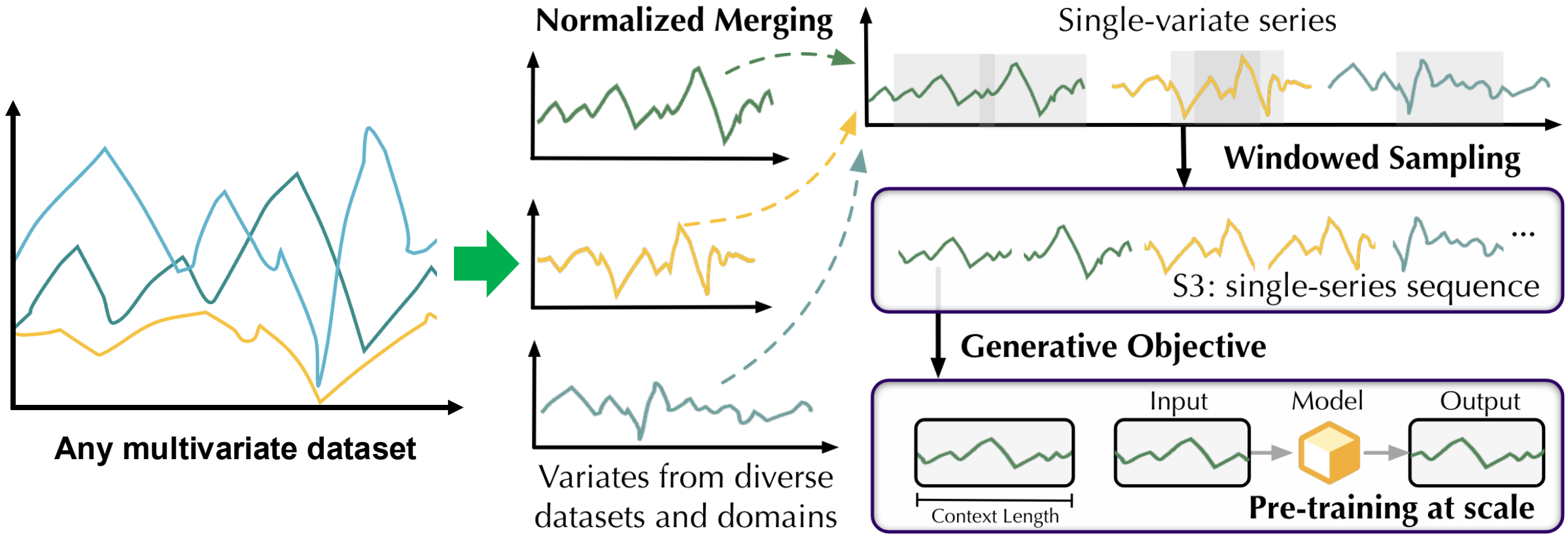
During admission an episode of possible coffee ground vomiting coupled with his non-steroidal inflammatory drug use prompted an upper GI endoscopy at which no abnormality was detected. Fecal occult blood was negative.

The patient was also provided with physiotherapy and fully remobilised.

Simple 1-D discrete tokens

Timer: Single-Series Sequence

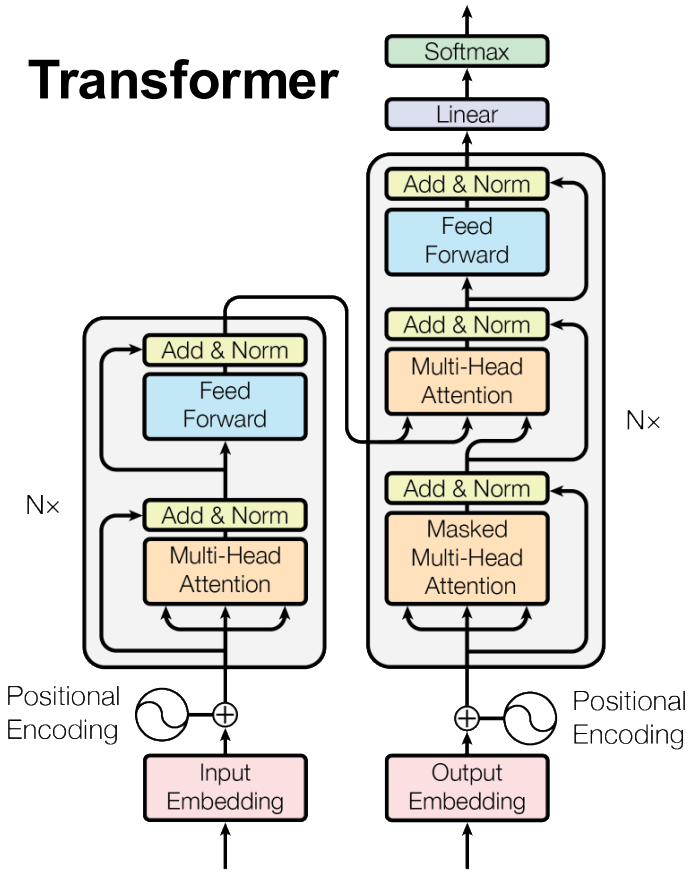
- **Aspect 2: Unified format to address data heterogeneity: Single-Series Sentence**



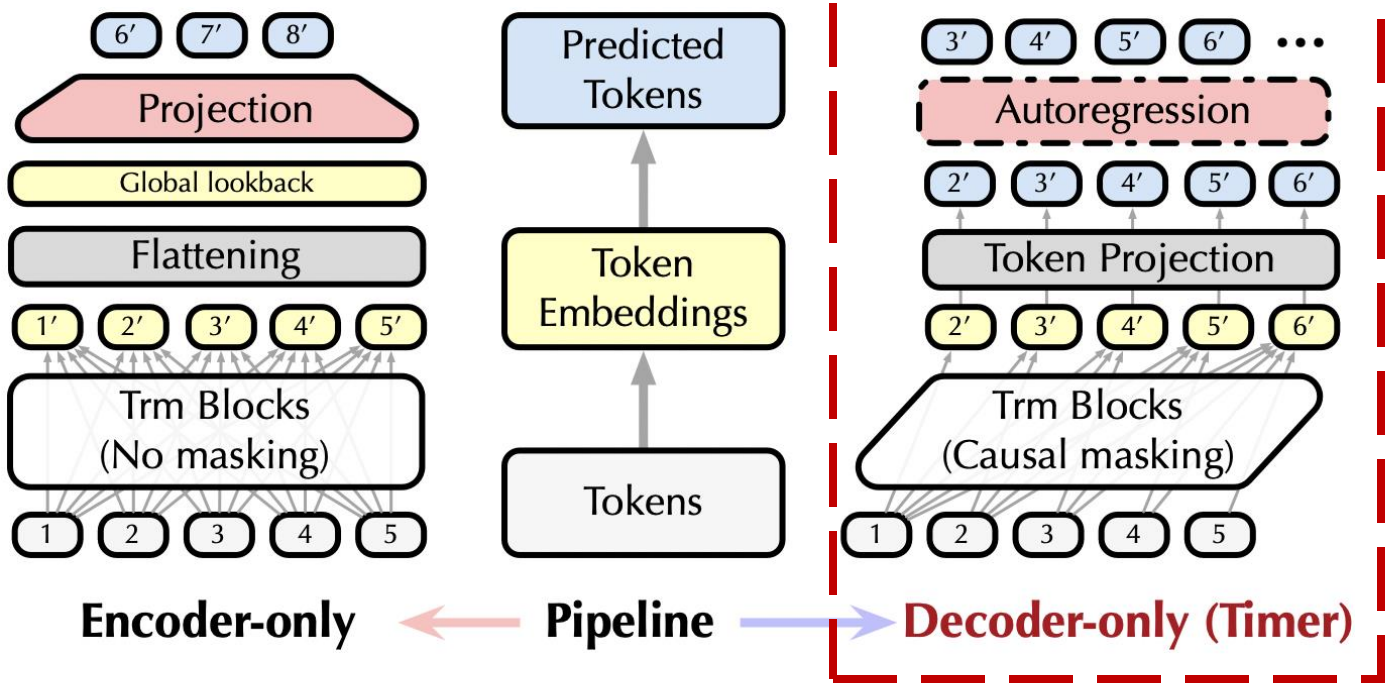
Define the basic "sentence" of multivariate time series

Timer: Backbones for Large Model

- **Aspect 3: Decoder-only Transformer with autoregression**



We make the initial exploration on architectures for LTMs



Popular in small models

✓ **Prevalent in LMs**

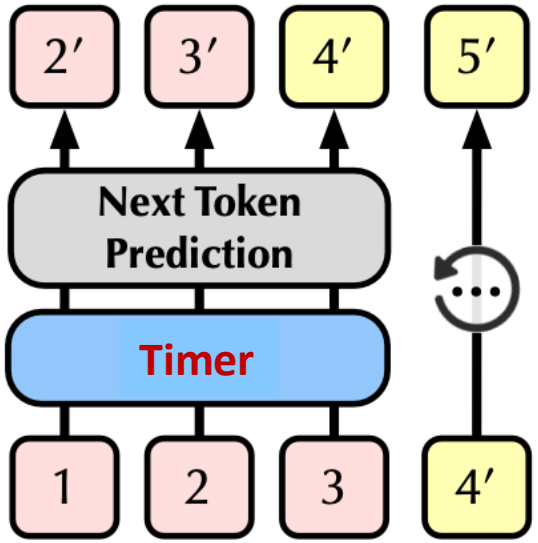
Timer: Generative Pre-training

- **Aspect 3: Next Token Prediction (Both training and inference)**

$$\text{Tokenize}: \ \mathbf{s}_i = \{x_{(i-1)S+1}, \ldots, x_{iS}\} \in \mathbb{R}^S.$$

$$\begin{aligned} \mathbf{h}_i^0 &= \mathbf{W}_e \mathbf{s}_i + \mathbf{TE}_i, \ i = 1, \ldots, N, \\ \text{Forwarding}: \quad \mathbf{H}^l &= \text{TrmBlock}(\mathbf{H}^{l-1}), \ l = 1, \ldots, L, \\ \{\hat{\mathbf{s}}_{i+1}\} &= \mathbf{H}^L \mathbf{W}_d, \ i = 1, \ldots, N, \end{aligned}$$

$$\text{NTP}: \ \mathcal{L}_{\text{MSE}} = \frac{1}{NS} \sum ||\mathbf{s}_i - \hat{\mathbf{s}}_i||_2^2, \ i = 2, \ldots, N+1.$$

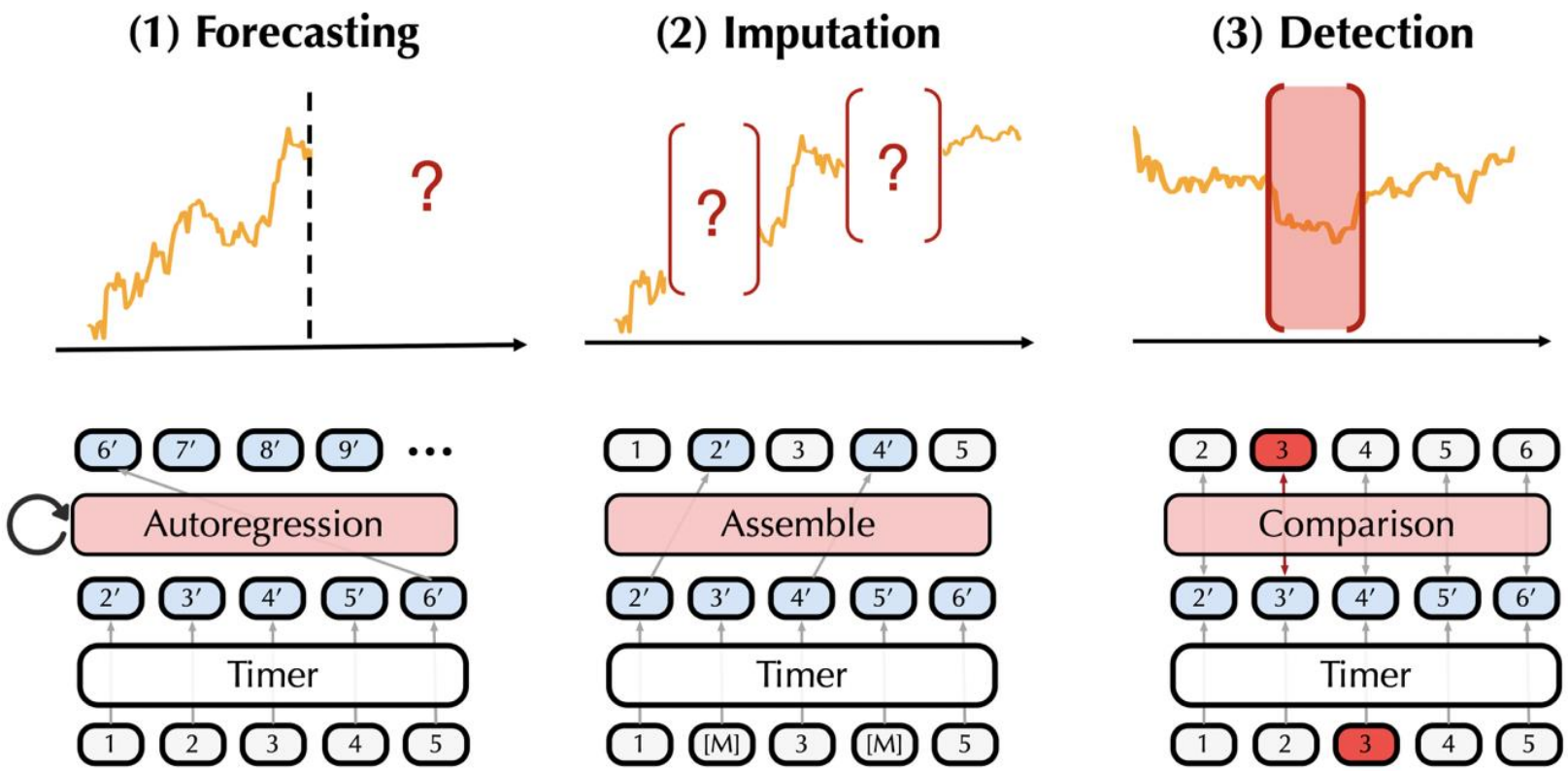


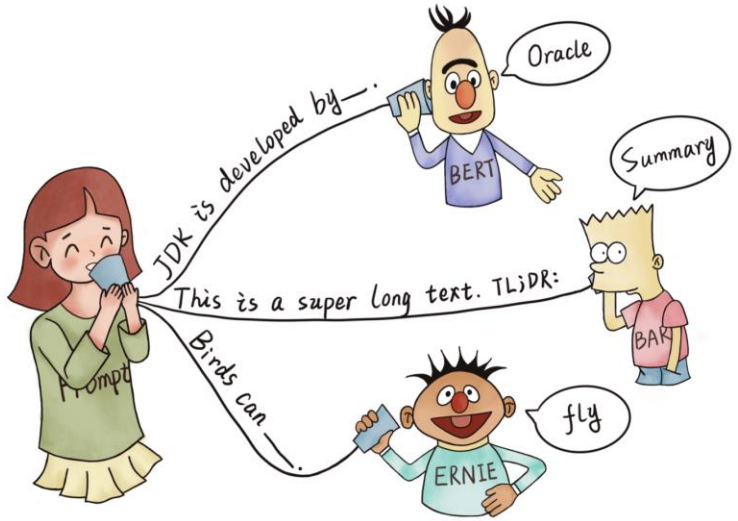
- **Token-wise supervision: the token of each position is independently supervised**

✓ **Enables flexible input-output lengths to address a variety of real-world scenarios**

Timer: Unified Task Formulation

- **Aspect 4: Unify Time Series Analysis into Generative Tasks**





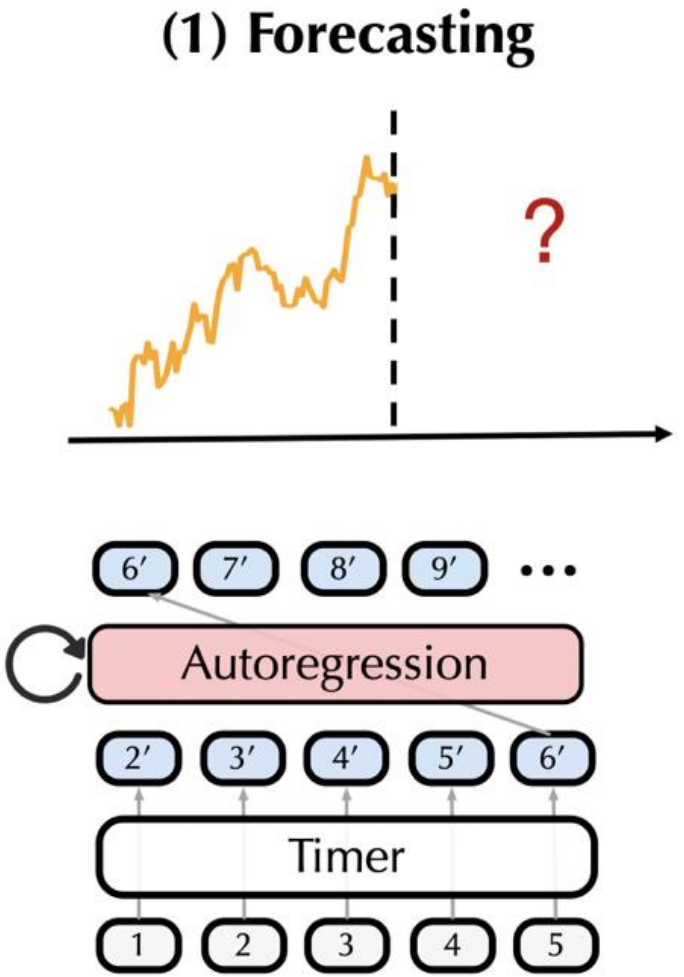
Towards the initial effort in prompting diverse time series analysis tasks

Liu et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM 2023.

Task Generality: Forecasting

Time Series Forecasting

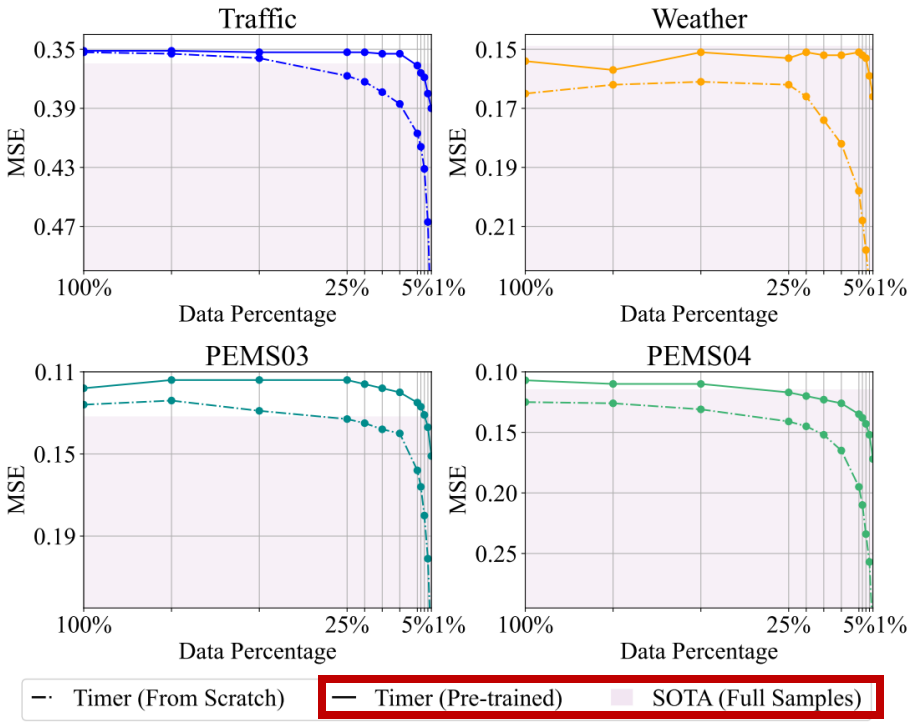
- Predict any next tokens by autoregression
- Timer trained with only **1~5% samples** outperforms SOTA with **100% samples**



Task Generality: Forecasting

Time Series Forecasting

- Predict any next tokens by autoregression
- Timer trained with only **1~5% samples** outperforms SOTA with **100% samples**



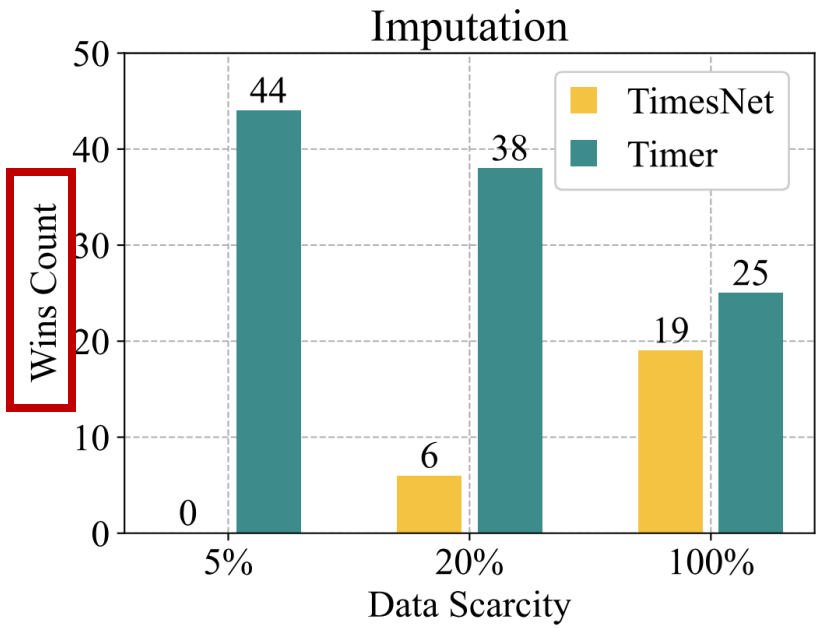
Takeaways:

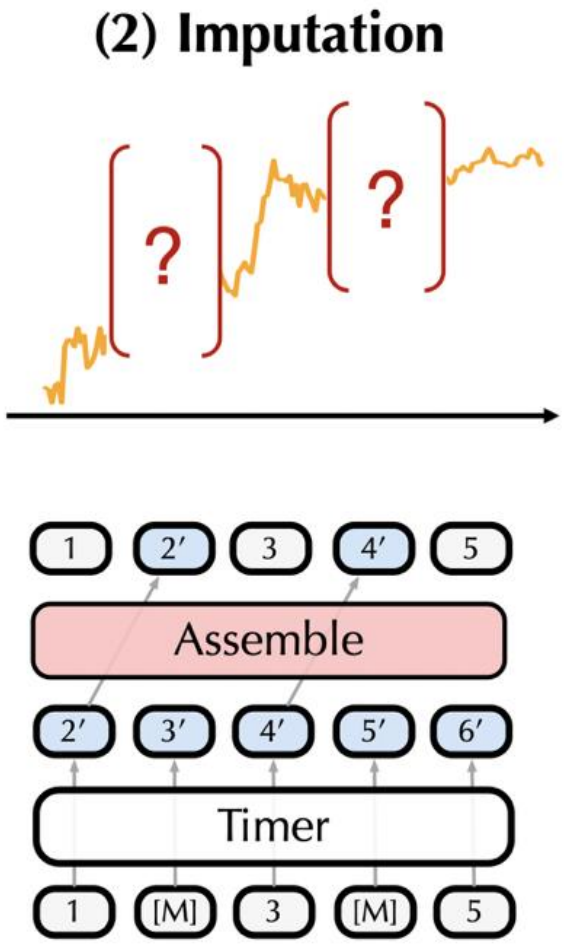
1. Timer fine-tuned on few samples achieves better results than advanced deep models
2. For widespread data-scarce scenarios, the performance degradation can be alleviated by the few-shot ability of Timer

Task Generality: Imputation

Time Series Imputation

- Imputation is performed by generating masked tokens with the previous context
- Surpass previous **SOTA TimesNet** in **44** imputation cases and data scarcities

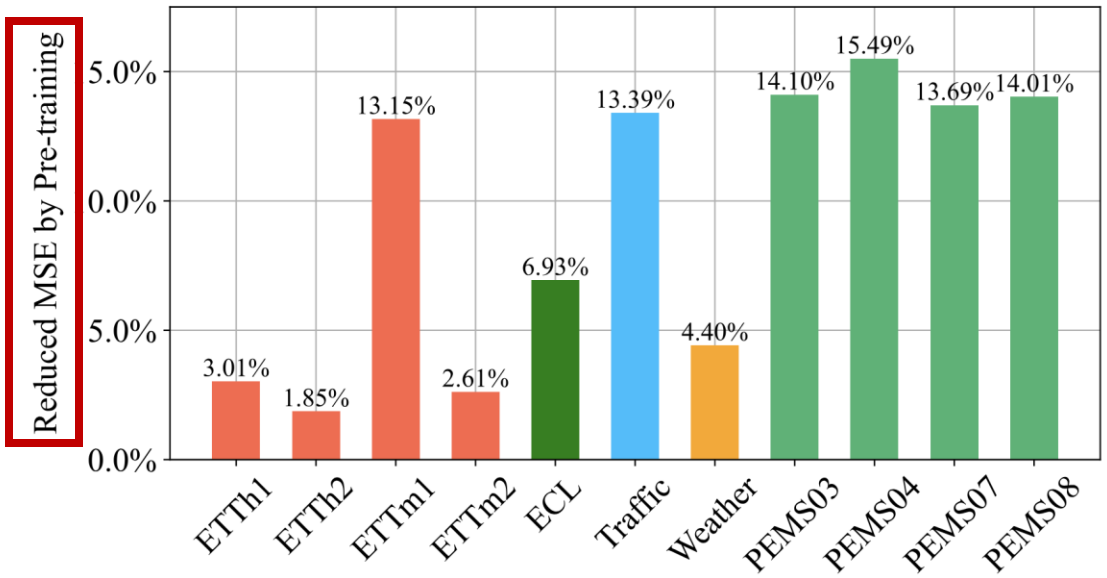




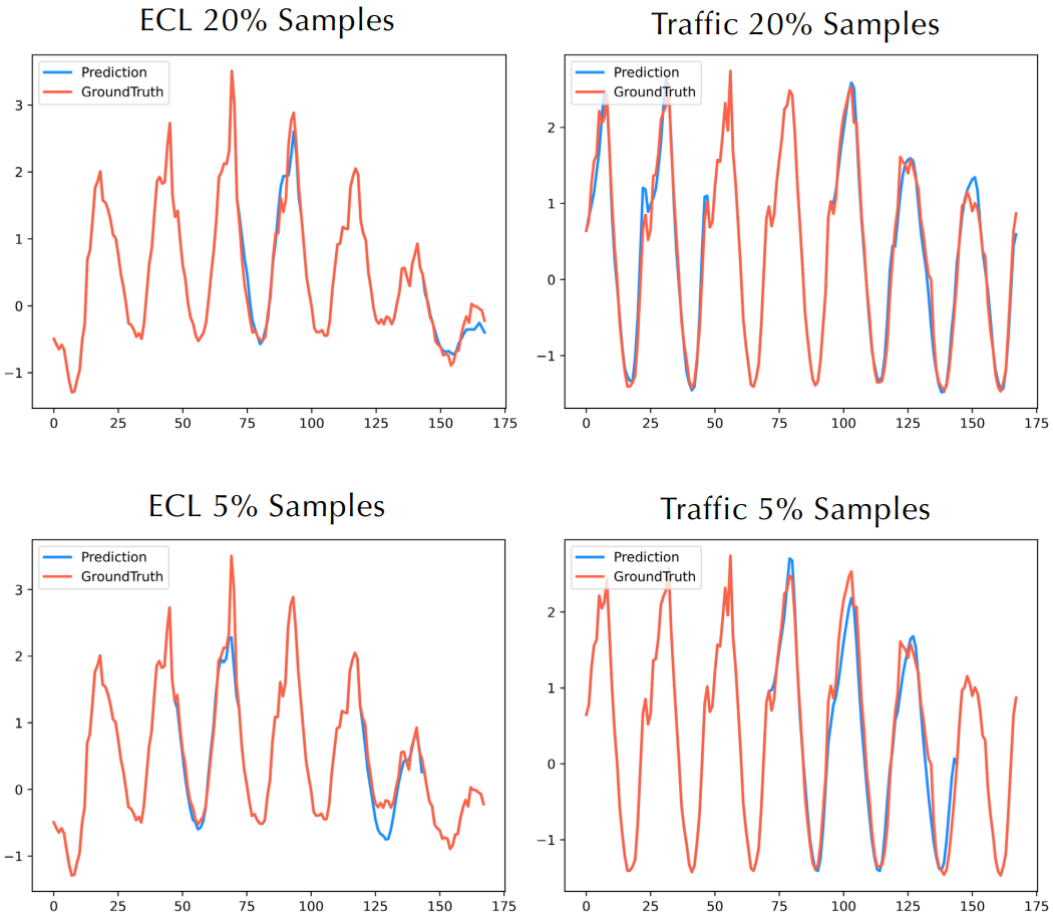
Task Generality: Imputation

Time Series Imputation

- Imputation is performed by generating masked tokens with previous context
- **Stable improvement** exhibited in imputation by large-scale pre-training



Showcases

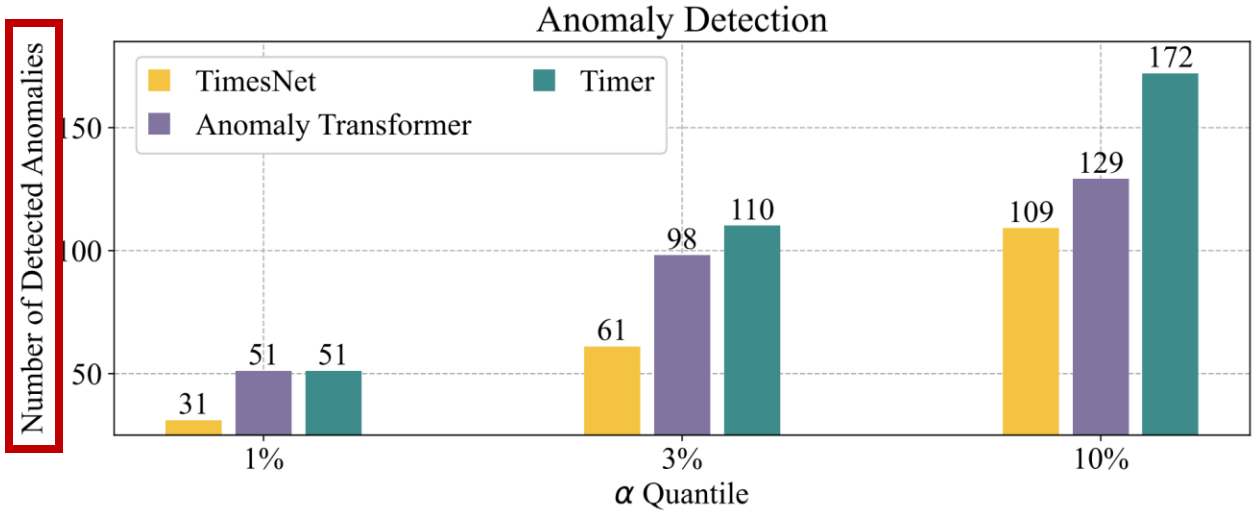


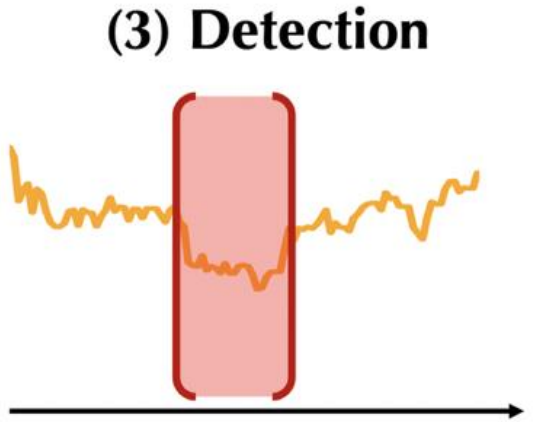
Imputing 50% missing time points

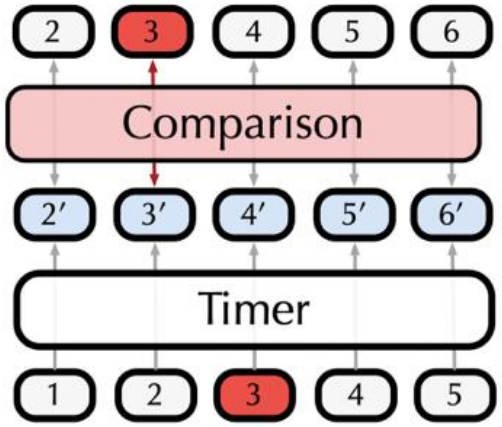
Task Generality: Anomaly Detection

Anomaly Detection

- Conducted in a **predictive** approach by generating normal time series
- **Quantile** MSE as the abnormal confidence
- Surpass **task-specific SOTA models** in **256 tasks** of UCR Anomaly Archive





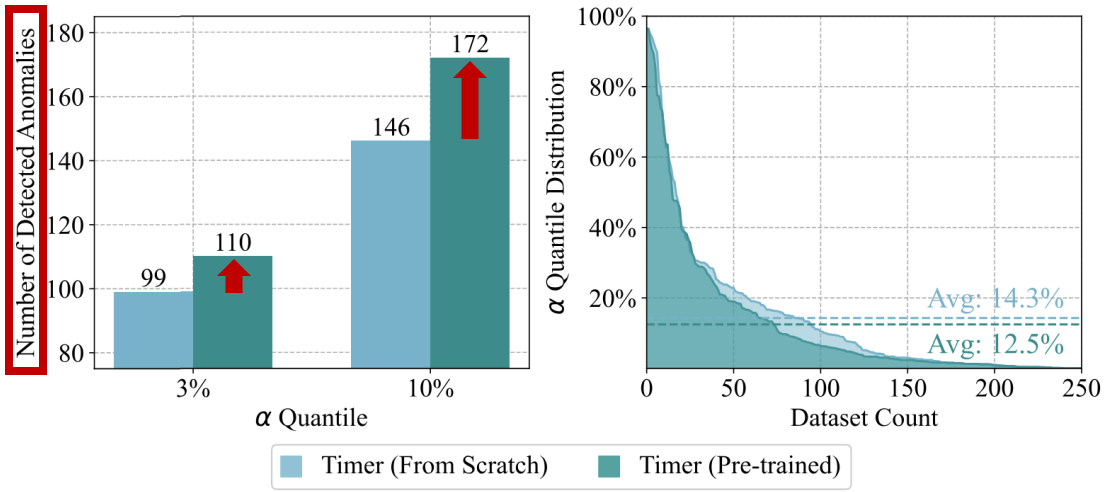


Task Generality: Anomaly Detection

Anomaly Detection

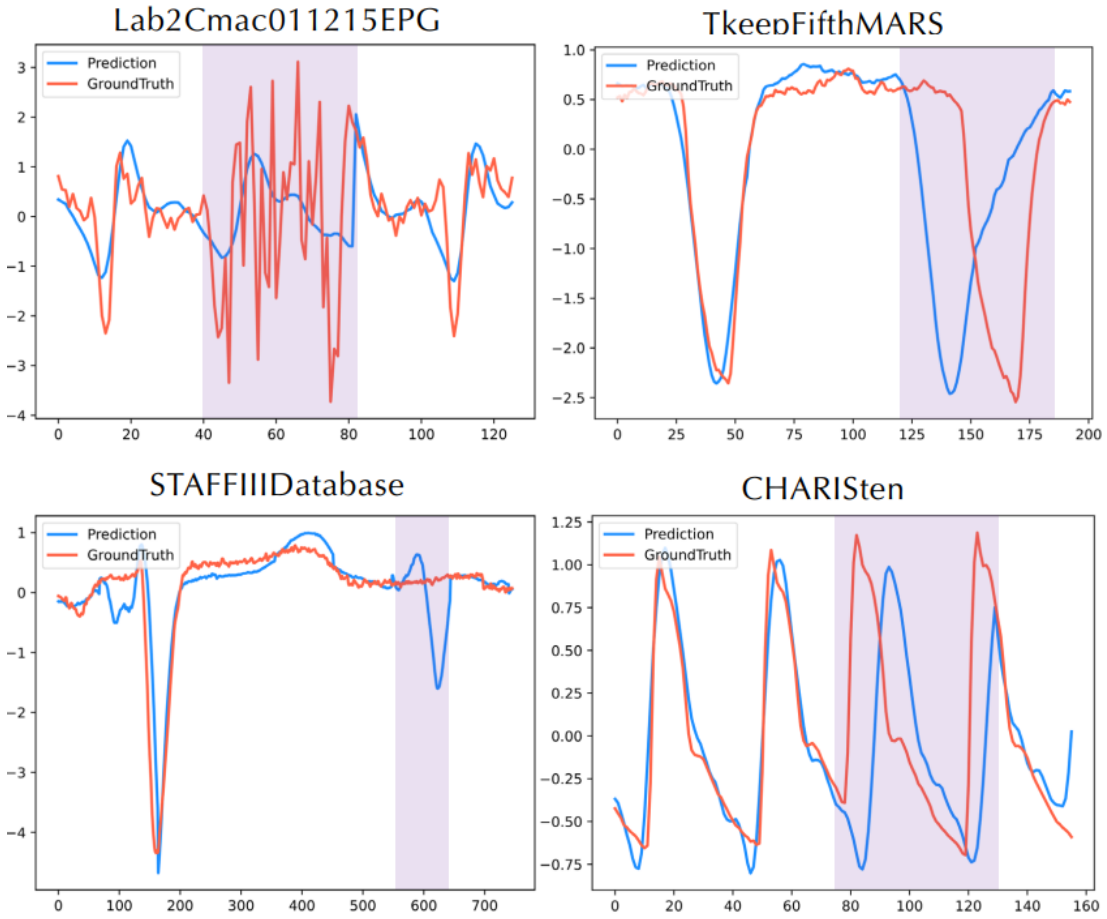
- Conducted in a **predictive** approach by generating normal time series

- **Stable improvement** exhibited in anomaly detection by large-scale pre-training



A smaller α indicates better performance

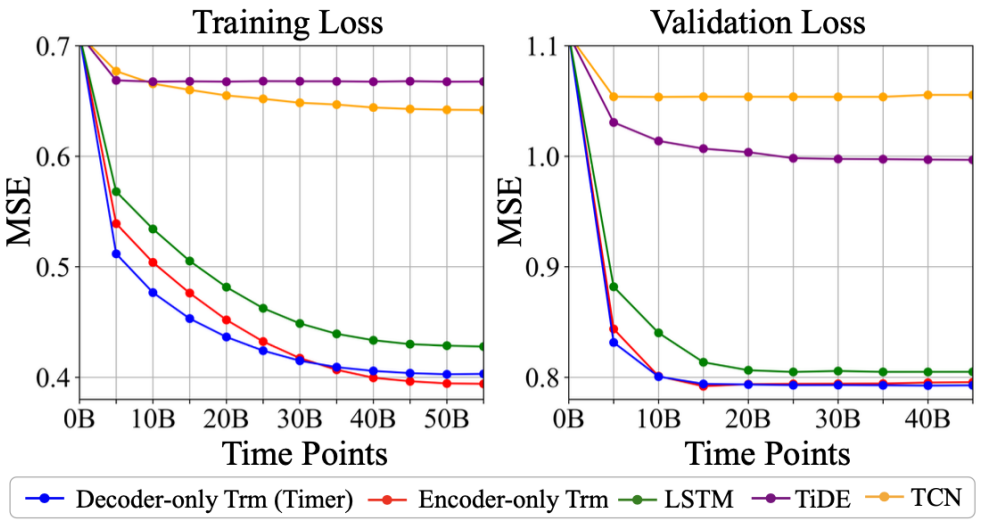
Showcases



Anomalies detected

Explore the Backbone for Large Model

Loss Curve of Sequence Models



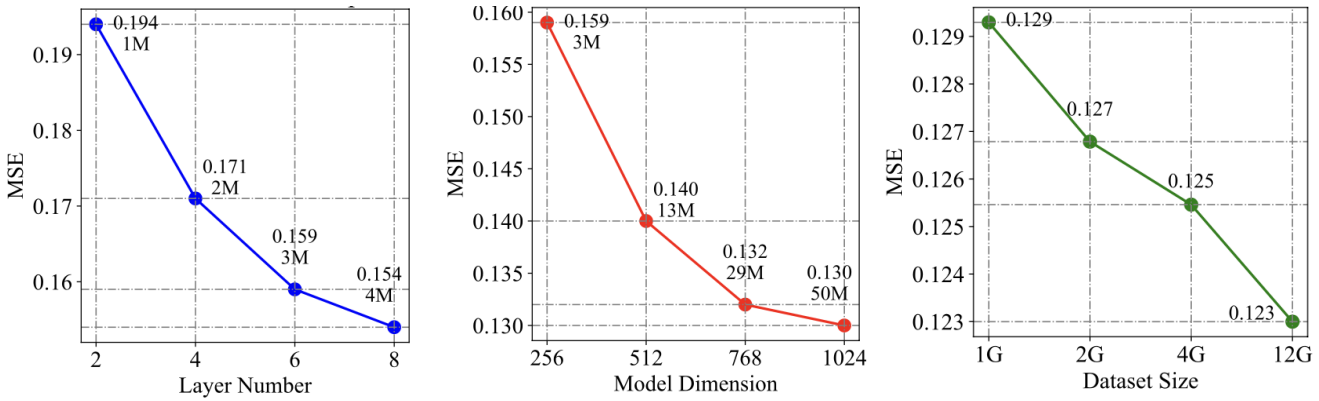
Scalable backbone remains underexplored in the time series community

Takeaways:

- Transformer exhibits great model capacity to accommodate diverse time series
- The finding is surprising since lots of deep time series models focus on much smaller backbones

Scalability: Essence of Large Models

Scaling Model/Data Improves Performance



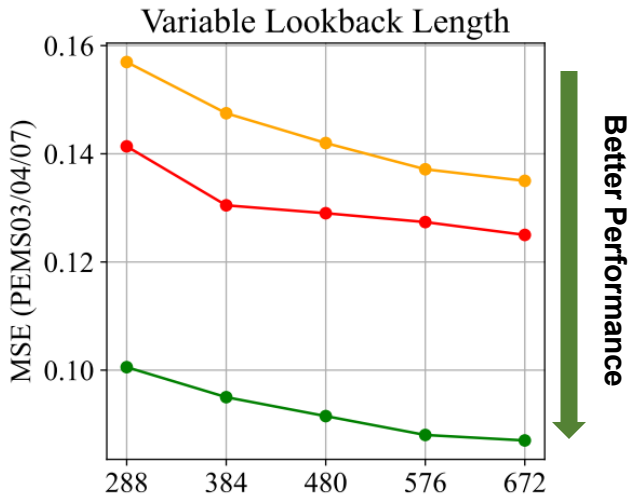
Transformer exhibits model capacity as the scalable architecture for LTM

Scaling Timer achieves MSE: 0.194 → 0.123 (−36.6%) *under data scarcity*, surpassing the SOTA (0.129) *trained on full samples*

Architecture Analysis: Flexible Output Length

Variable Lookback Length

- Small models are constrained to fixed input/output lengths
- Similar to LLMs, Timer is flexible on the input length
- Increasing the input window leads to stable accuracy growth



Iterative Multi-step Prediction

- Token-wise supervision can alleviate error accumulation

$$\mathcal{L}_{\text{MSE}} = \frac{1}{NS} \sum ||\mathbf{s}_i - \hat{\mathbf{s}}_i||_2^2, \ i = 2, ..., N+1.$$

Architecture Analysis: Flexible Output Length

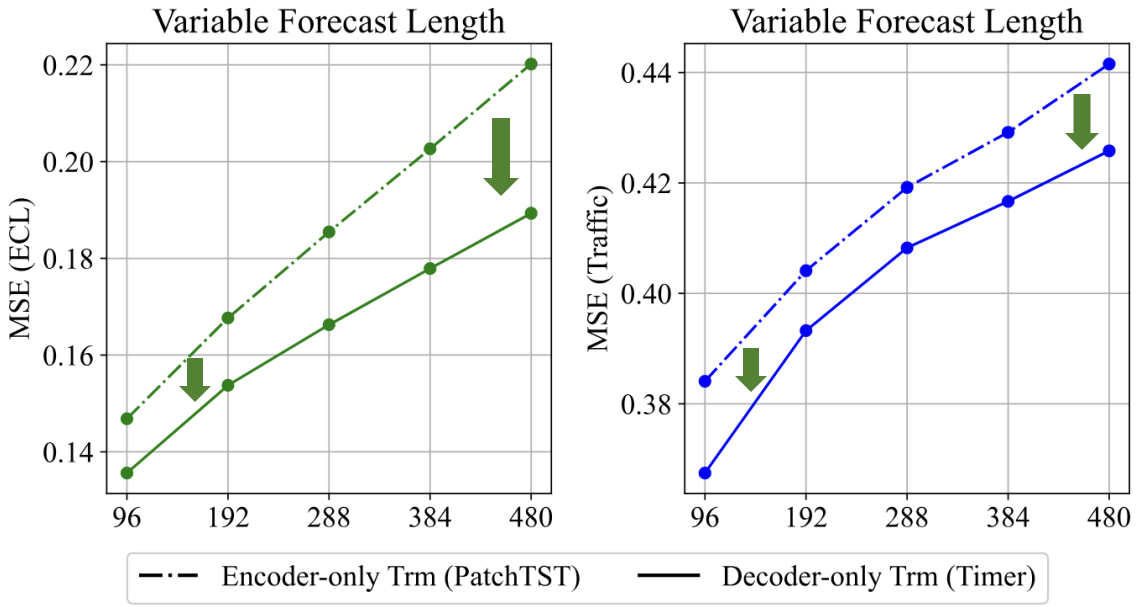
Variable Lookback Length

- Small models are constrained to fixed input/output lengths
- Similar to LLMs, Timer is flexible on the input length
- Increasing the input window leads to stable accuracy growth

Iterative Multi-step Prediction

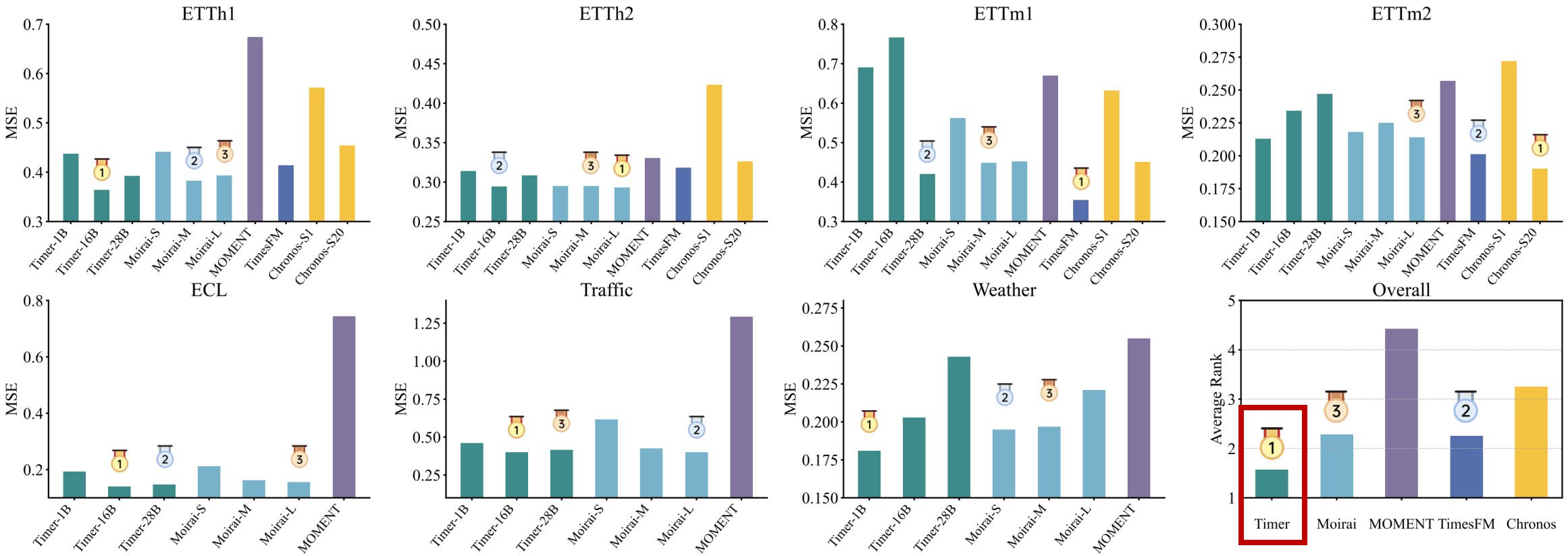
- Token-wise supervision can alleviate error accumulation

$$\mathcal{L}_{\text{MSE}} = \frac{1}{NS} \sum ||\mathbf{s}_i - \hat{\mathbf{s}}_i||_2^2, \ i = 2, ..., N+1.$$



Benchmarks of LTMs

Quantitative Evaluations (Zero-shot Forecasting)



We provided the average rank, where the first is the best,

to measure LTMs as a *general-purpose zero-shot* forecaster

Evaluations of LTMs

Quality Assessments

Method	**Timer (Ours)**	Moirai (2024)	MOMENT (2024)	Chronos (2024)	Lag-Llama (2023)	TimesFM (2023b)	TimeGPT-1 (2023)
Architecture	Decoder	Encoder	Encoder Decoder	Encoder Decoder	Decoder	Decoder	Encoder Decoder
Model size	29M, 50M, 67M	14M, 91M, 311M	40M, 125M 385M	20M, 46M, 200M, 710M	200M	17M, 70M, 200M	Unknown
Supported tasks	Forecast Imputation Detection	Forecast	Forecast Imputation Classification Detection	Forecast	Forecast	Forecast	Forecast Detection
Pre-training Scale	28B	27.65B	1.13B	84B	0.36B	100B	100B
Token type	Segment	Segment	Segment	Point	Point	Segment	Segment
Context length	$\leq$1440	$\leq$5000	= 512	$\leq$512	$\leq$1024	$\leq$512	Unknown
Variable length	True	True	False	True	True	True	True
Probabilistic	False	True	False	True	True	True	True

[2]https://huggingface.co/AutonLab/MOMENT-1-large
[3]https://huggingface.co/amazon/chronos-t5-large
[4]https://huggingface.co/google/timesfm-1.0-200m
[5]https://huggingface.co/collections/Salesforce/moirai-10-r-models-65c8d3a94c51428c300e0742

Future Directions

- **Larger Dataset**
- **Longer Context**
- **Probabilistic**
- **Complex Tasks**
- **……**

TIMER-XL: LONG-CONTEXT TRANSFORMERS FOR UNIFIED TIME SERIES FORECASTING

Yong Liu[*], **Guo Qin**[*], **Xiangdong Huang, Jianmin Wang, Mingsheng Long**[✉]
School of Software, BNRist, Tsinghua University, Beijing 100084, China
{liuyong21,qinguo24}@mails.tsinghua.edu.cn
{huangxdong,jimwang,mingsheng}@tsinghua.edu.cn

Yong Liu

Guo Qin

Xiangdong Huang

Jianmin Wang

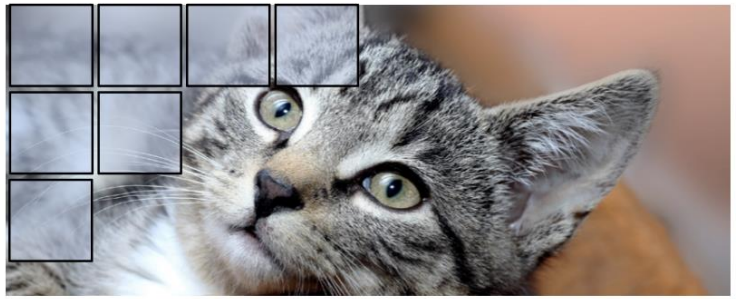
Mingsheng Long

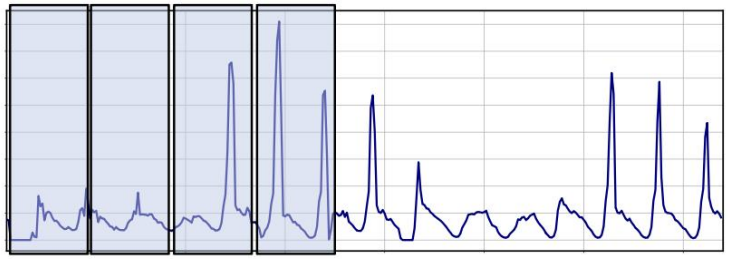
Context Length Matters

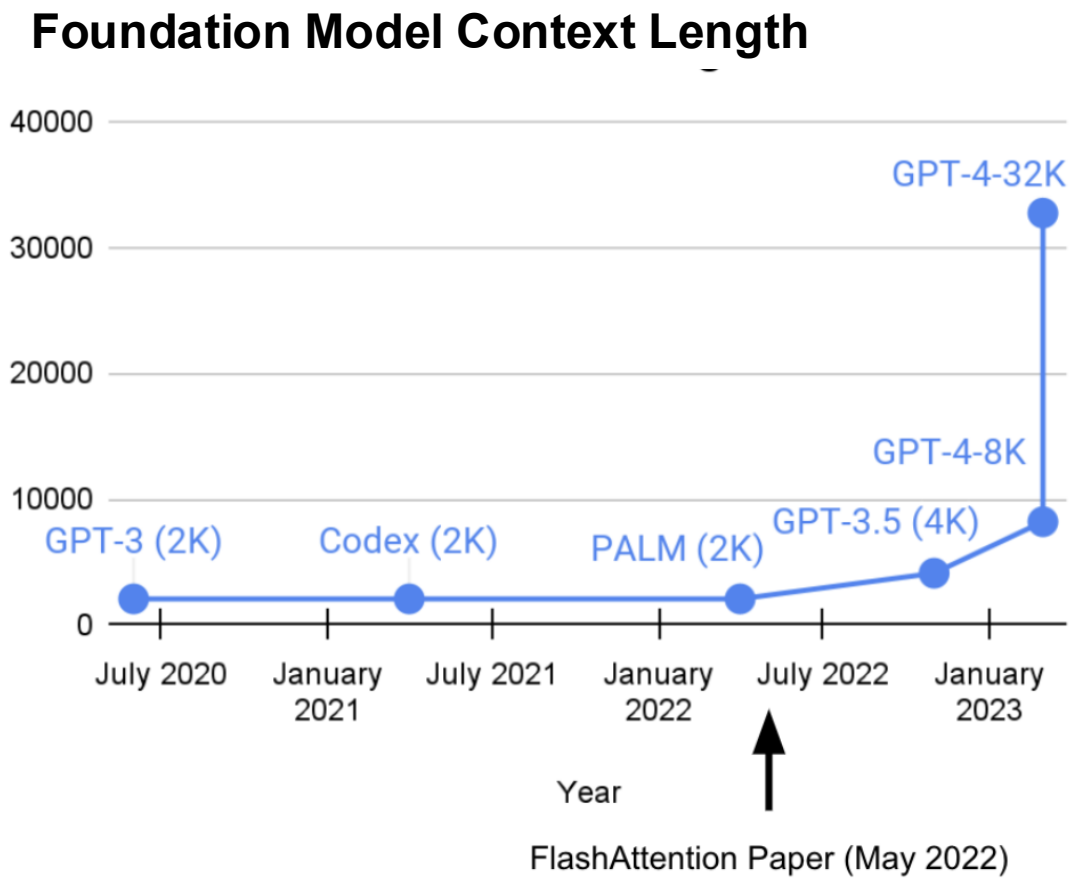
Context Length of Foundation Models is Scaling

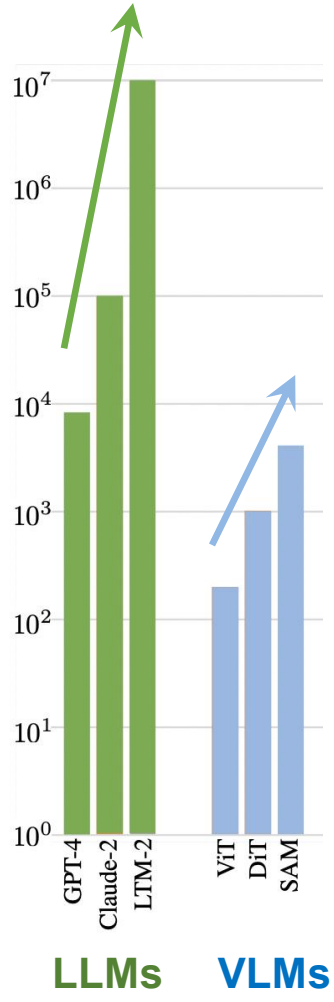
Answer the following mathematical questions:

Q: If you have 12 apples and you give 5 to your friend, how many apples do you have now?
A: The answer is 7.



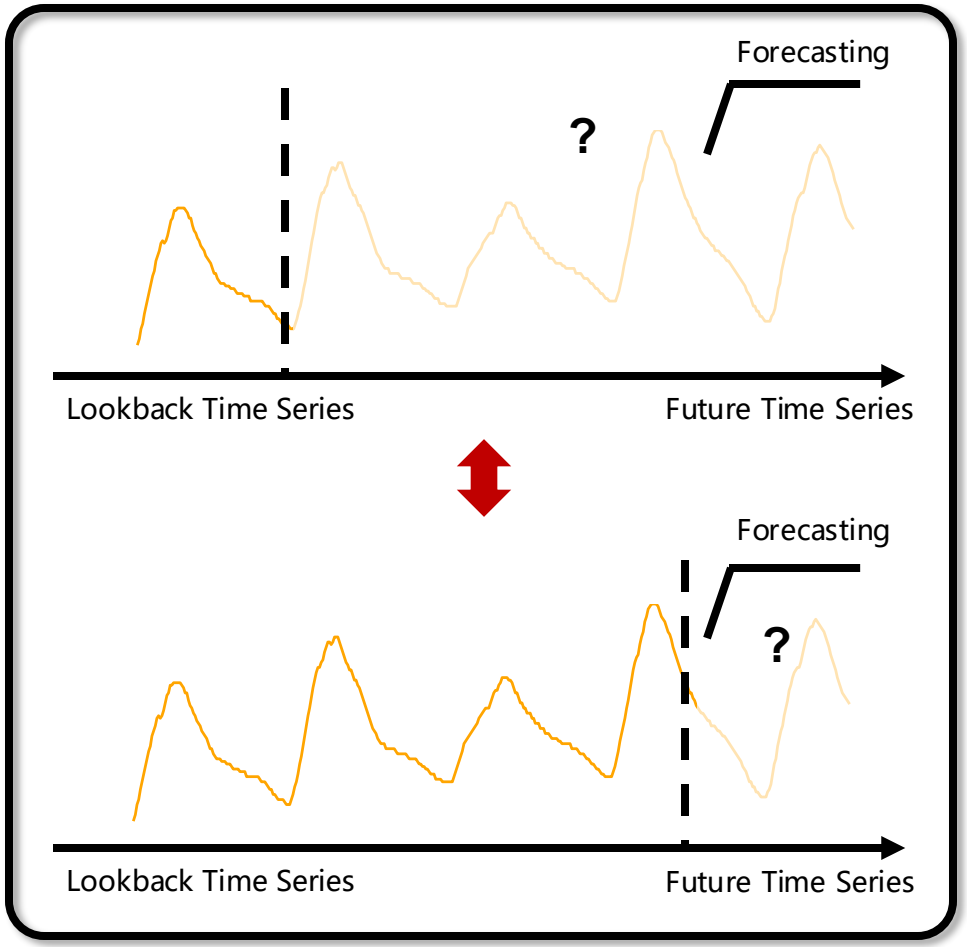






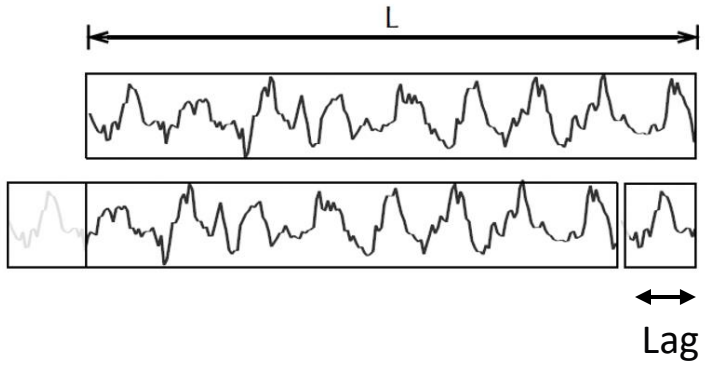
Long-Context Forecasting

Long-Term Forecasting -> Long-Context Forecasting

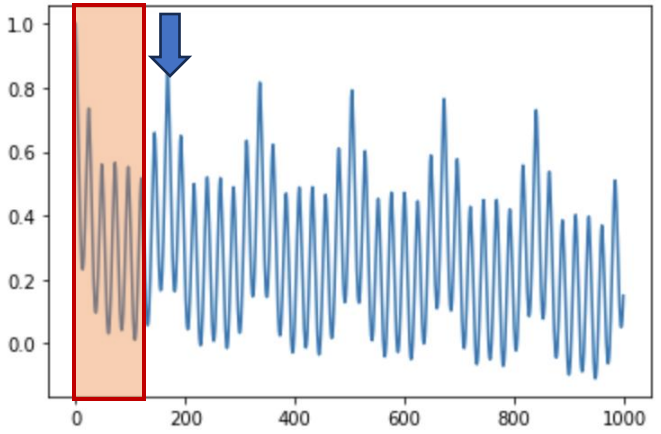


$$\mathcal{R}_{\mathcal{X}\mathcal{X}}(\tau) = \lim_{L\to\infty} \frac{1}{L} \sum_{t=1}^{L} \mathcal{X}_t \mathcal{X}_{t-\tau}.$$

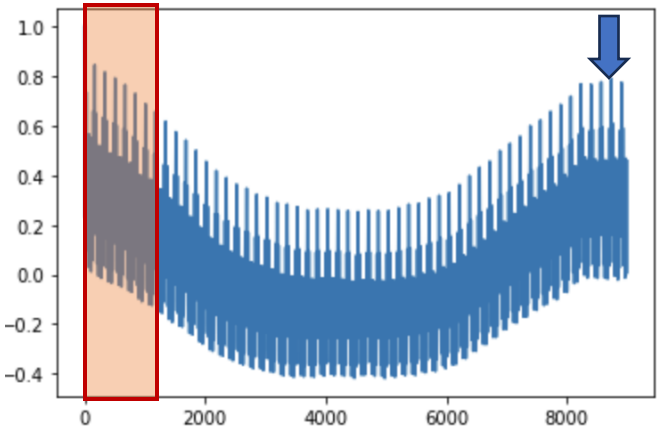
ACF indicates Periodicity



Weekly Period



Yearly Period

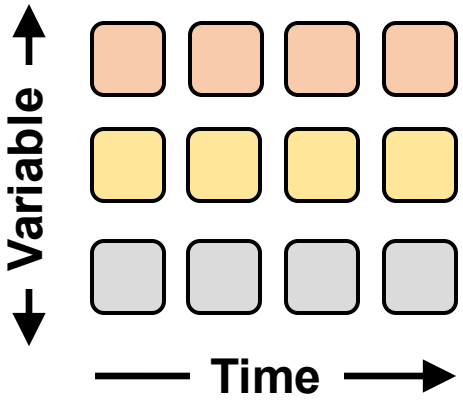


INFORMATION INCOMPLETE

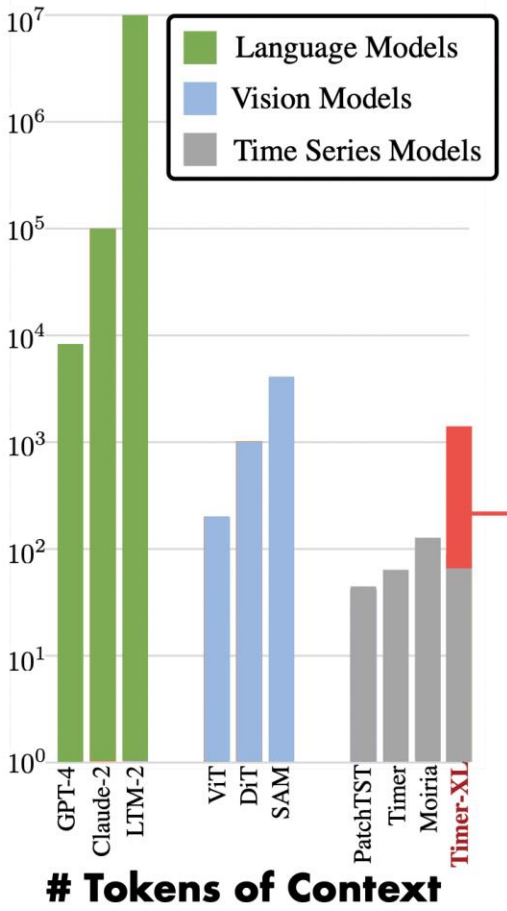
Unified Time Series Forecasting

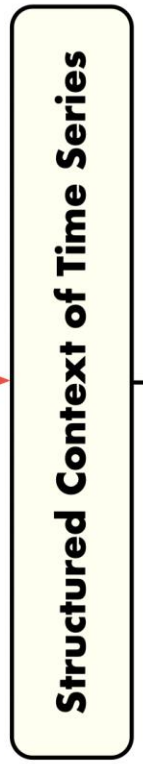
Long-Context Forecasting -> Unified Time Series Forecasting

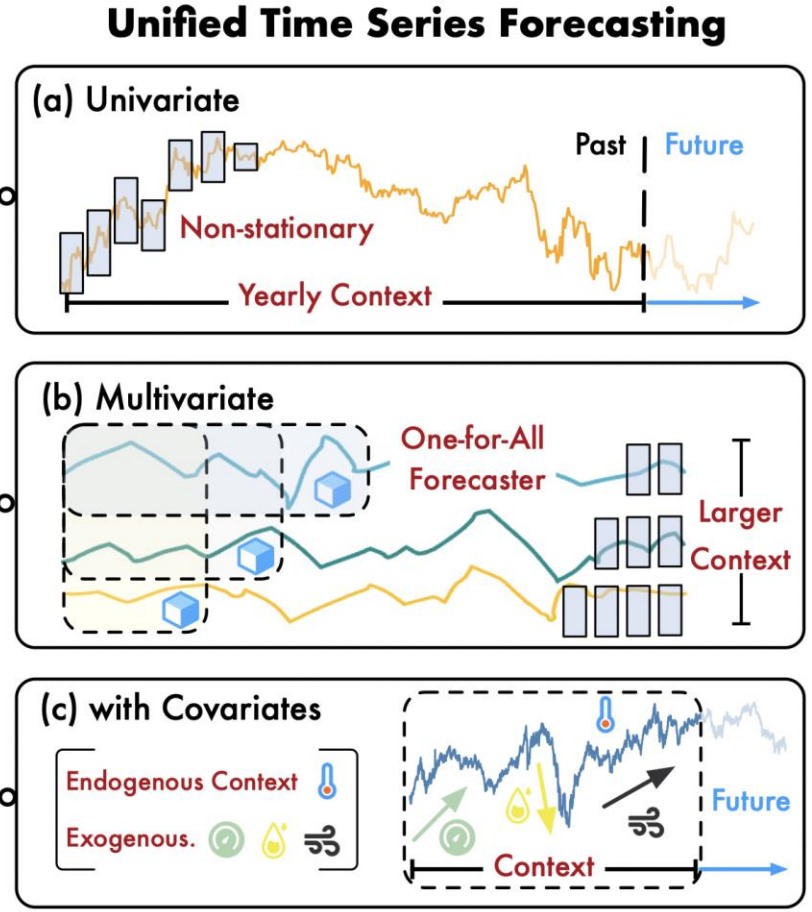
2D Time Series



Overlength Context

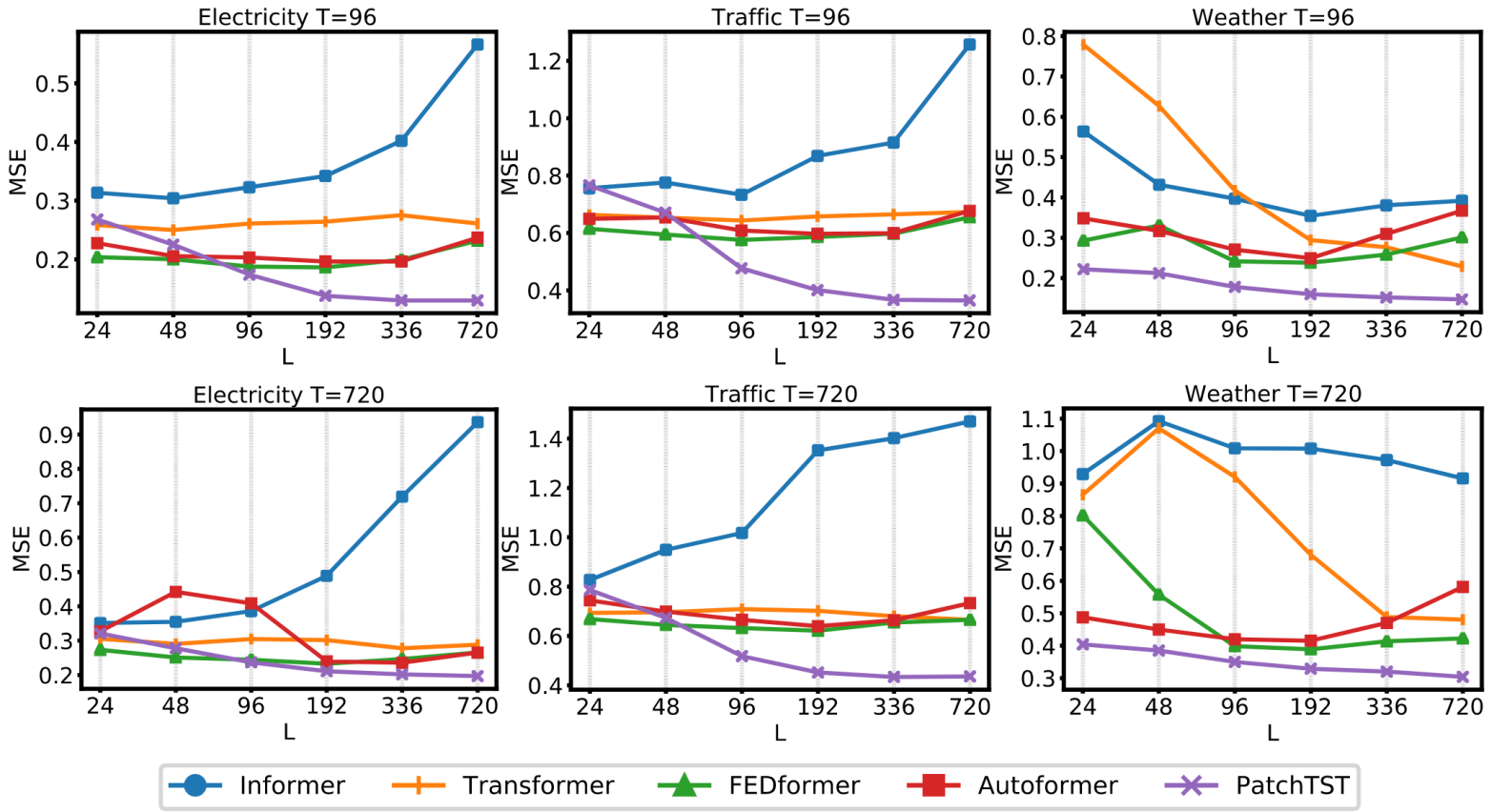






Rethinking Long-Context Transformers

How Long Should be Inputted? Is Longer Context Better?



Performance (MSE) - Context Length (L)

Tokenization

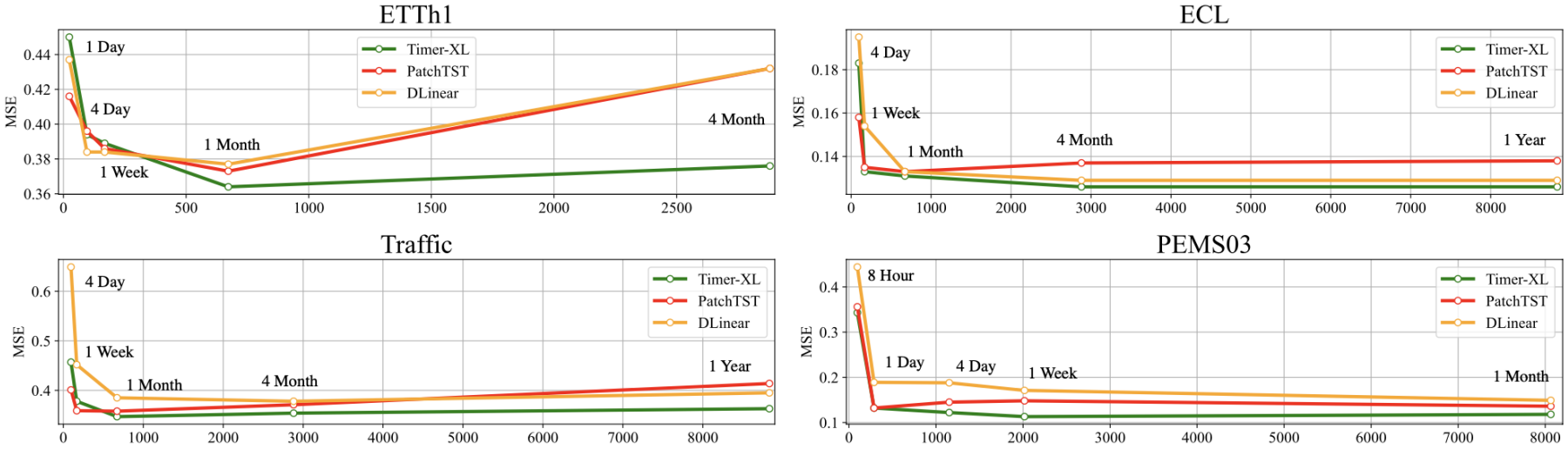
- Point-Level
- ✓ **Patch-Level**

Prediction Length

- Long-Term
- ✓ **Short-Term**

A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *ICLR 2023.*

Rethinking Long-Context Transformers

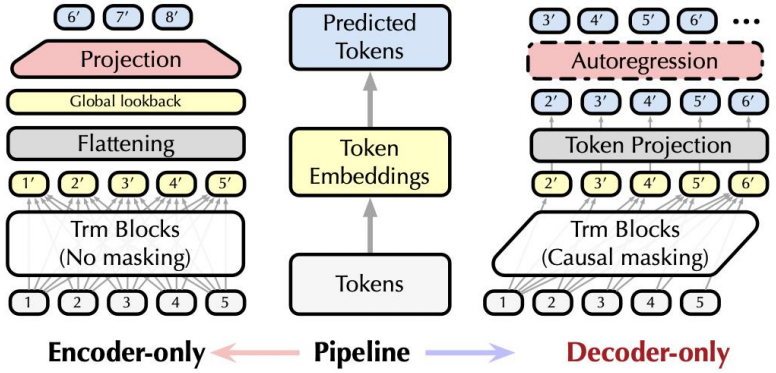


Architecture

- Encoder-Only
- ✓ **Decoder-Only**

Performance - Context Length

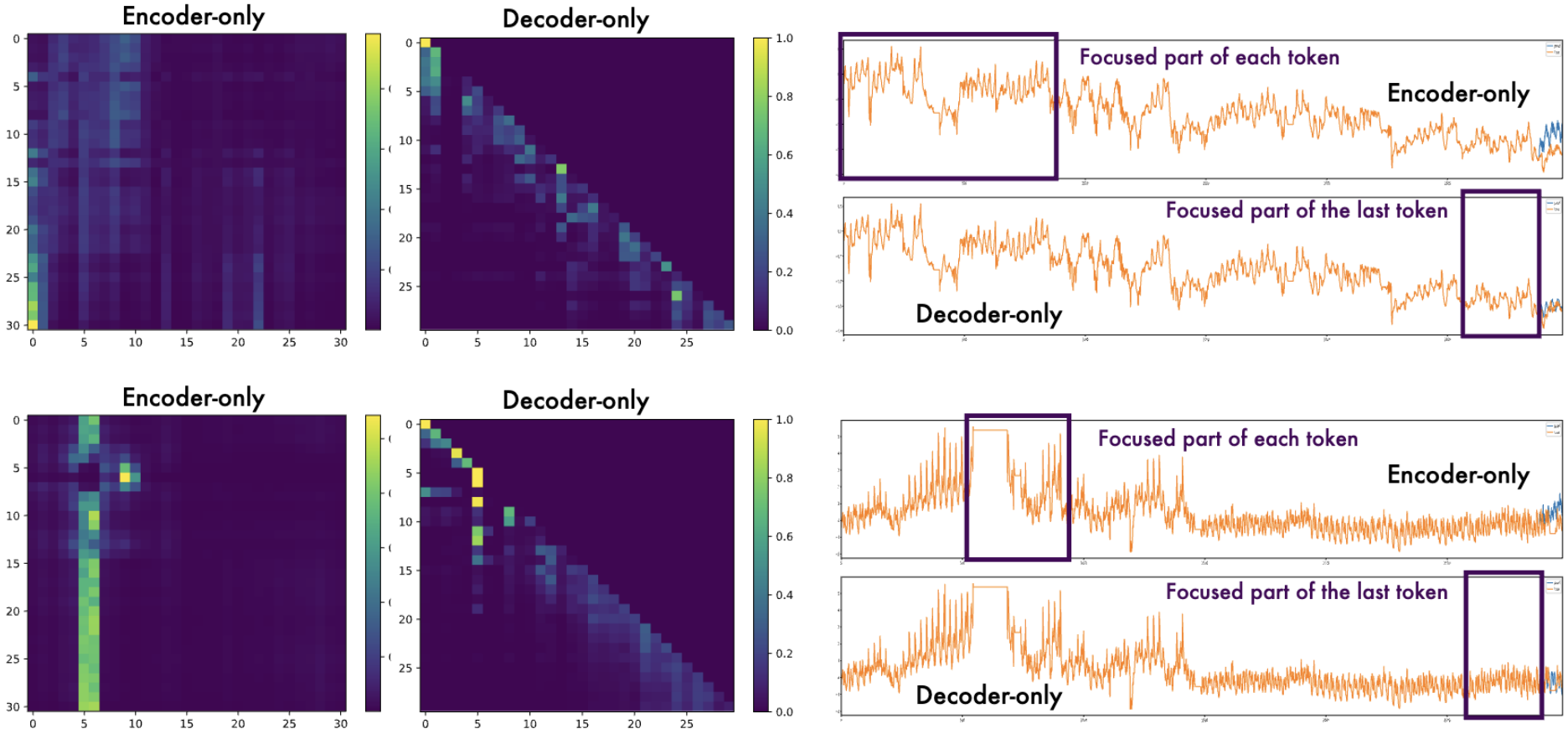
Models	Timer-XL		PatchTST		DLinear	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
Lookback-8 (1 Day)	0.0847	0.2100	0.0897	0.2196	0.0970	0.2276
Lookback-32 (4 Day)	0.0713	0.1928	0.0778	0.2080	0.0841	0.2113
Lookback-56 (1 Week)	0.0688	0.1891	0.0785	0.2082	0.0814	0.2081
Lookback-224 (1 Month)	0.0675	0.1868	0.0745	0.2042	0.0788	0.2048
Lookback-960 (4 Month)	0.0667	0.1863	0.1194	0.2696	0.0773	0.2031
Lookback-2944 (1 Year)	0.0663	0.1857	0.1109	0.2638	0.0763	0.2024



Decoder-Only Transformers Outperform Encoder-Only Models on Long-Context Sequences

Rethinking Long-Context Transformers

Attention - Raw Time Series



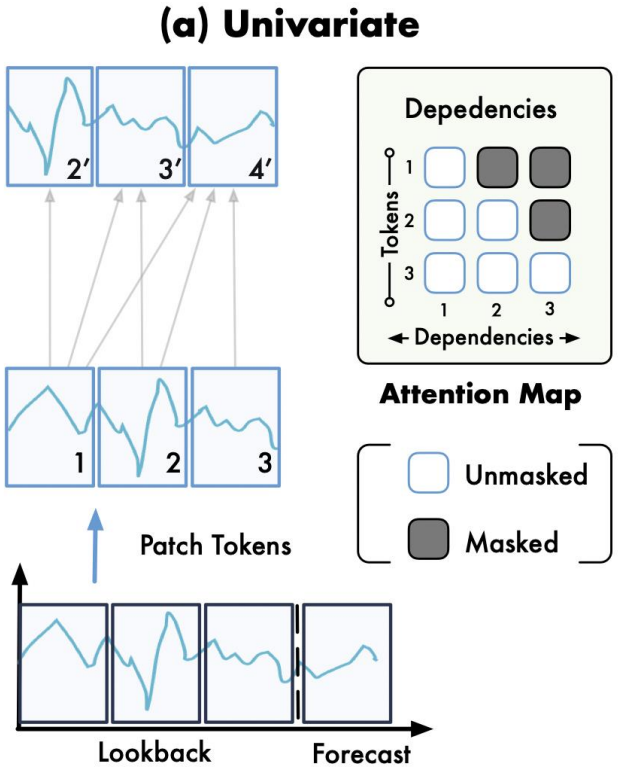
Decoder-Only Transformers Can Selectively Focus on Long-Context Sequences

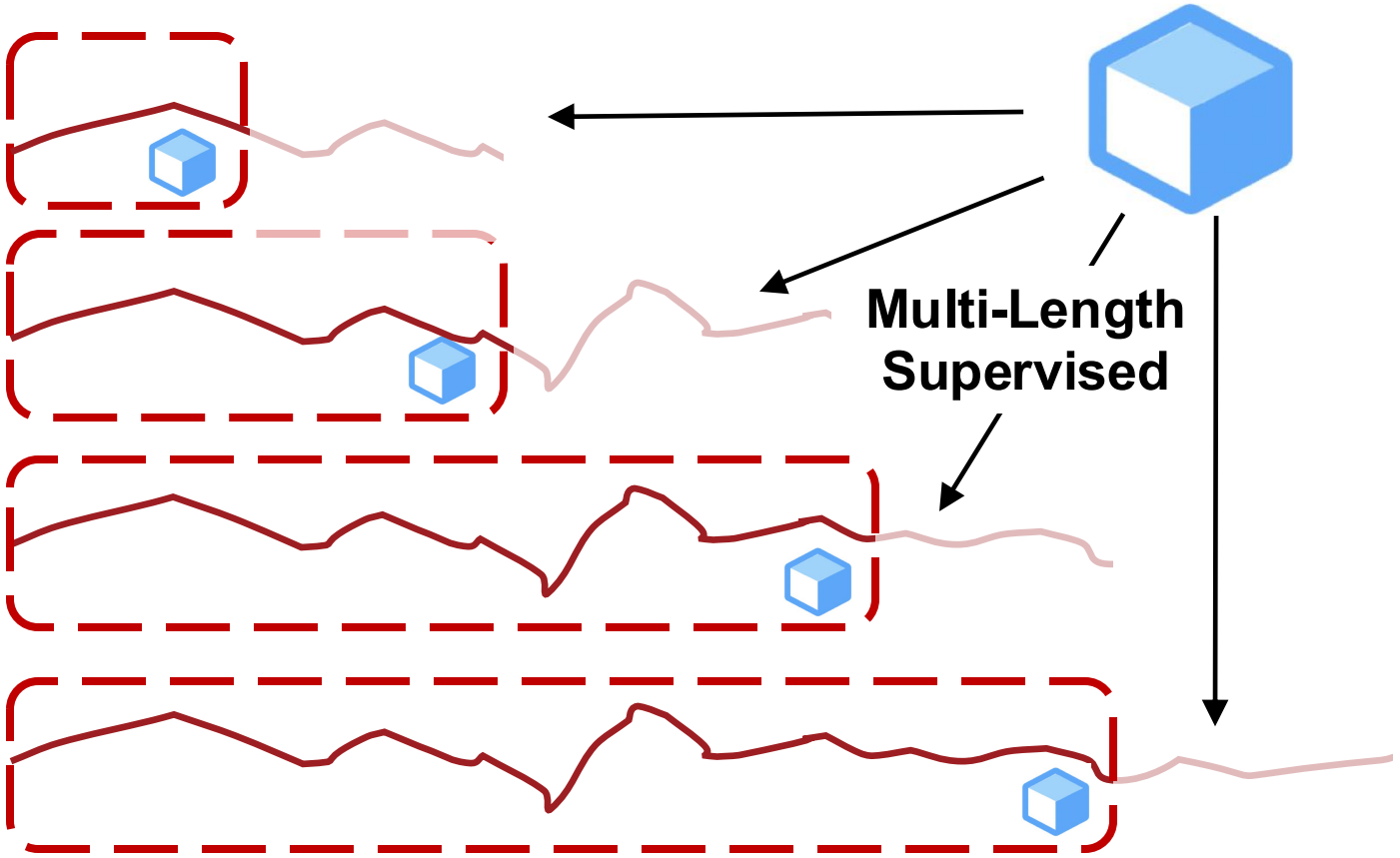
Extending 1D Sequences to 2D Time Series

Next Token Prediction (Patch Tokenization)

$$\mathbf{x}_i = \{x_{(i-1)P+1}, \ldots, x_{iP}\}$$

$$P(\mathbf{X}) = \prod_{i=1}^{T} p(\mathbf{x}_{i+1} | \mathbf{x}_{\leq i})$$





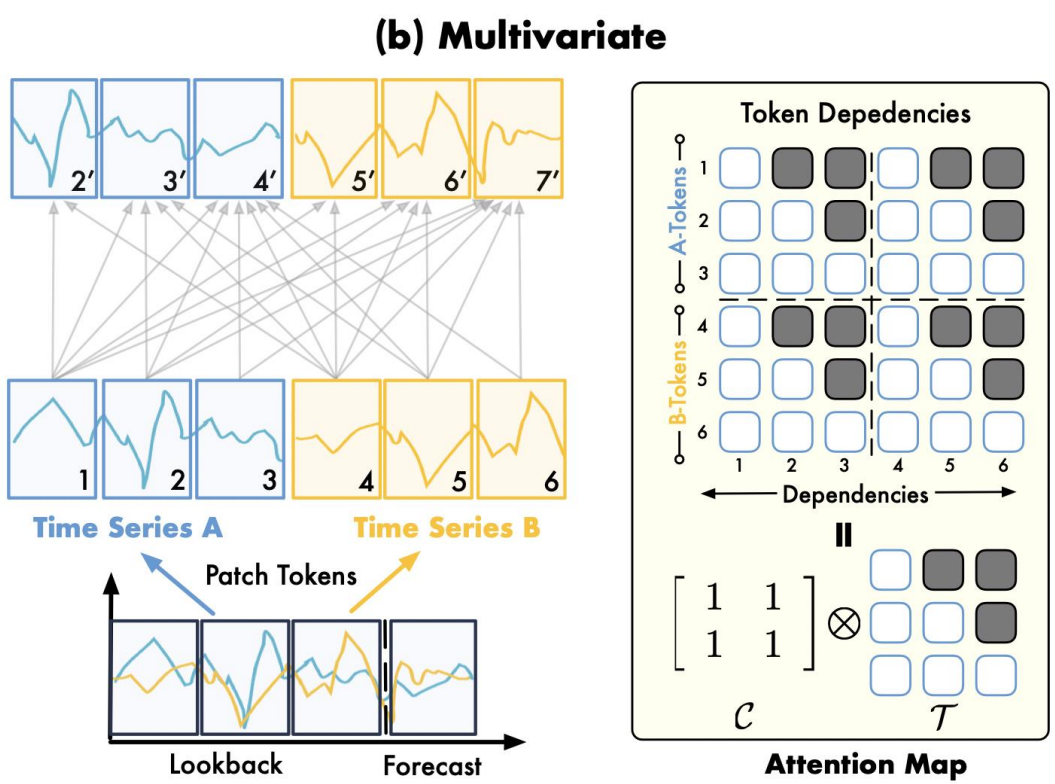
Decoder-Only Transformers Are One-For-All-Length Models

Extending 1D Sequences to 2D Time Series

Next Token Prediction -> Multivariate Next Token Prediction

$$P(\mathbf{X}) = \prod_{i=1}^{T} p(\mathbf{x}_{i+1} | \mathbf{x}_{\leq i})$$

$$P(\mathbf{X}) = \prod_{m=1}^{N} \prod_{i=1}^{T} p(\mathbf{x}_{m,i+1} | \mathbf{x}_{:,\leq i}) \qquad \mathbf{x}_{m,i} = \{\mathbf{X}_{m,(i-1)P+1}, \ldots, \mathbf{X}_{m,iP}\}$$



Kronecker Product

- Temporal Causality

$$\mathcal{T}_{i,j} = \begin{cases} 1 & \text{if } j \leq i, \\ 0 & \text{otherwise.} \end{cases}$$

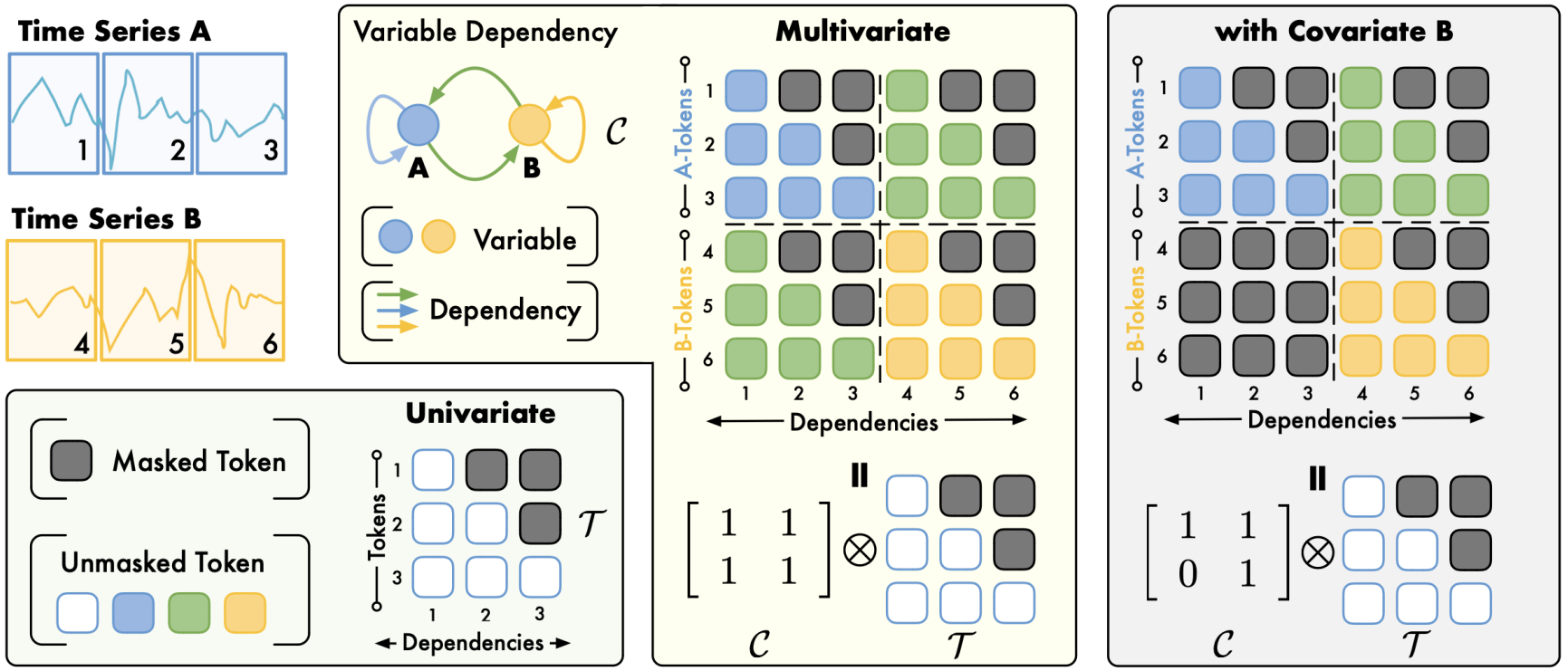
- Variable Dependence

$$\mathcal{C}_{m,n} = \begin{cases} 1 & \text{if variable } m \text{ is dependent on } n, \\ 0 & \text{otherwise.} \end{cases}$$

TimeAttention

A Versatile Masking Mechanism for Multidimensional Time Series

$$\text{TimeAttention}(\mathbf{H}) = \text{Softmax}\left(\frac{\text{Mask}(\mathcal{C} \otimes \mathcal{T}) + \mathcal{A}}{\sqrt{d_k}}\right)\mathbf{H}\mathbf{W}_v, \quad \text{Mask}(\mathcal{M}) = \begin{cases} 0 & \text{if } \mathcal{M}_{i,j} = 1, \\ -\infty & \text{if } \mathcal{M}_{i,j} = 0. \end{cases}$$



Kronecker Product

- Temporal Causality

$$\mathcal{T}_{i,j} = \begin{cases} 1 & \text{if } j \le i, \\ 0 & \text{otherwise.} \end{cases}$$

- Variable Dependence

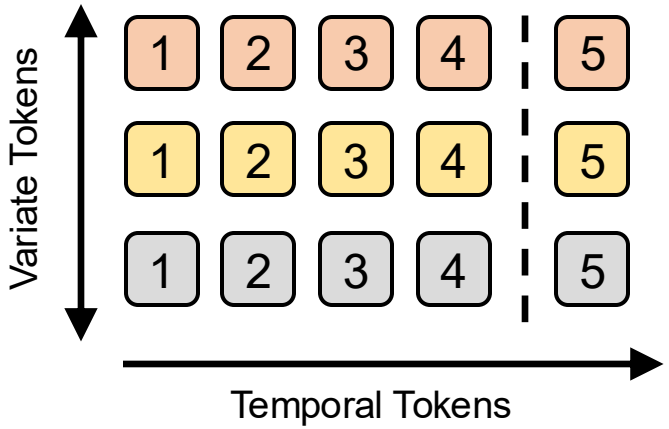
$$\mathcal{C}_{m,n} = \begin{cases} 1 & \text{if variable } m \text{ is dependent on } n, \\ 0 & \text{otherwise.} \end{cases}$$

Position Embedding in Self-Attention

*Tokens of multivariate time series are both **temporal tokens** and **variate tokens***

$$\mathcal{A}_{mn,ij} = \mathbf{h}_{m,i}^{\top}\mathbf{W}_q\mathbf{R}_{\theta,i-j}\mathbf{W}_k^{\top}\mathbf{h}_{n,j} + u \cdot \mathbb{1}(m=n) + v \cdot \mathbb{1}(m \neq n)$$

RoPE ***Alibi***



Permutation-Invariant

$\mathcal{H} : \mathbb{R}^T \to \mathbb{R}$

$\mathcal{H}(x_1, \ldots, x_T) = \mathcal{H}(\pi\{x_1, \ldots, x_T\})$

π : permutation of temporal tokens

RoPE: Avoid PI (inherent in self-attention) on the Temporal dimension

Learnable Alibi: Maintain PE on the Variate dimension (only distinguish endo-/exo-variates)

Permutation-Equivalent

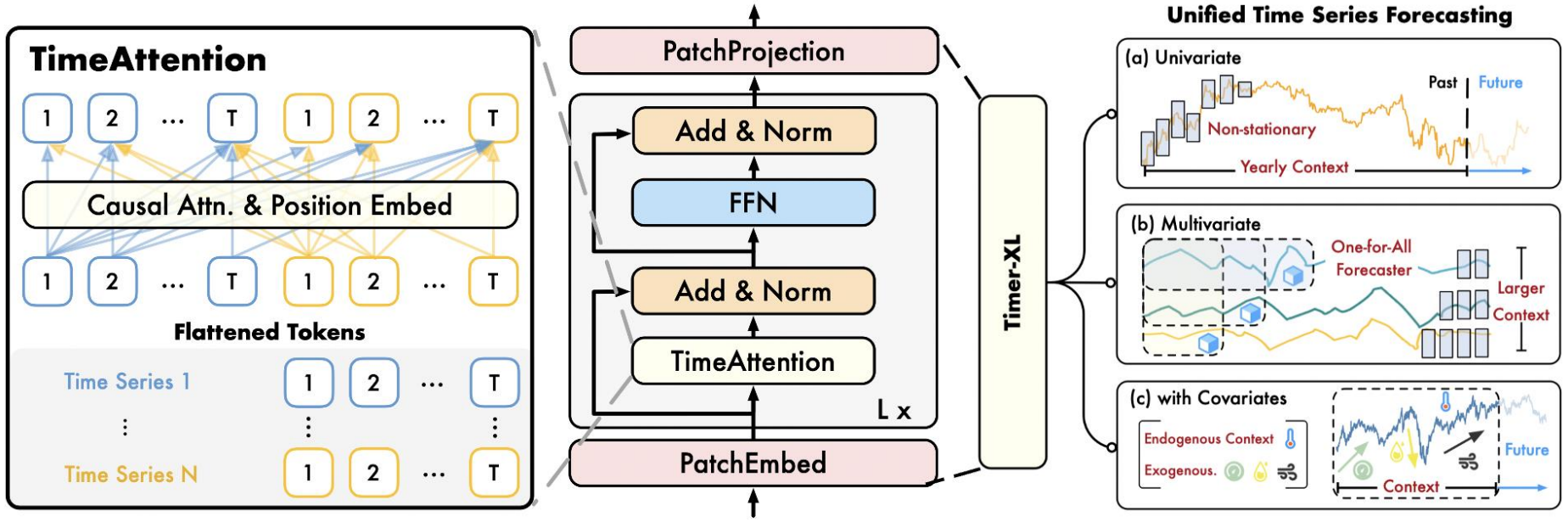
$\mathcal{H} : \mathbb{R}^N \to \mathbb{R}^N$

$\pi\{\mathcal{H}(x_1, \ldots, x_N)\} = \mathcal{H}(\pi\{x_1, \ldots, x_N\})$

π : permutation of variate tokens

Timer-XL

A Decoder-Only Long-Context Transformer for Unified Forecasting



Unified Context

Timer-XL can be used for (1) task-specific training and (2) scalable pre-training, handling arbitrary-length and any-variable time series

Timer-XL

A Decoder-Only Long-Context Transformer for Unified Forecasting

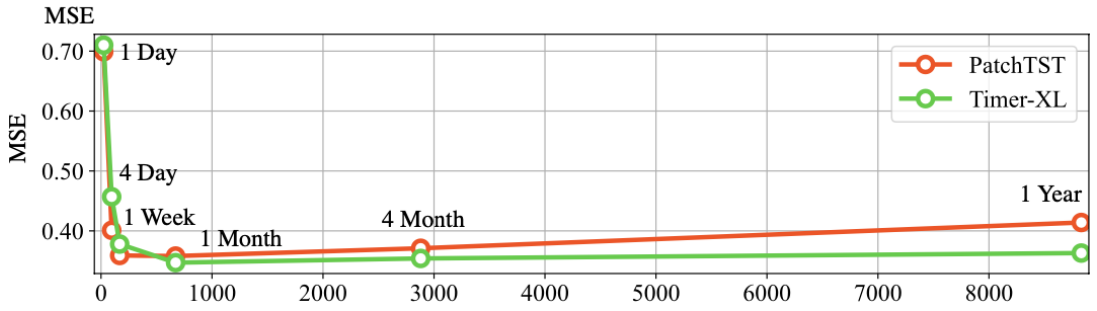
Table 1: Comparison among representative time-series Transformers.

Model	PatchTST (2022)	iTrans. (2023)	TimeXer (2024b)	UniTST (2024a)	Moirai (2024)	Timer (2024c)	**Timer-XL (Ours)**
Intra-Series	✓	✗	✓	✓	✓	✓	✓
Inter-Series	✗	✓	✓	✓	✓	✗	✓
Causal Trm.	✗	✗	✗	✗	✗	✓	✓
Pre-Trained	✗	✗	✗	✗	✓	✓	✓

Timer-XL can be used for (1) task-specific training and (2) scalable pre-training, handling arbitrary-length and any-variable time series

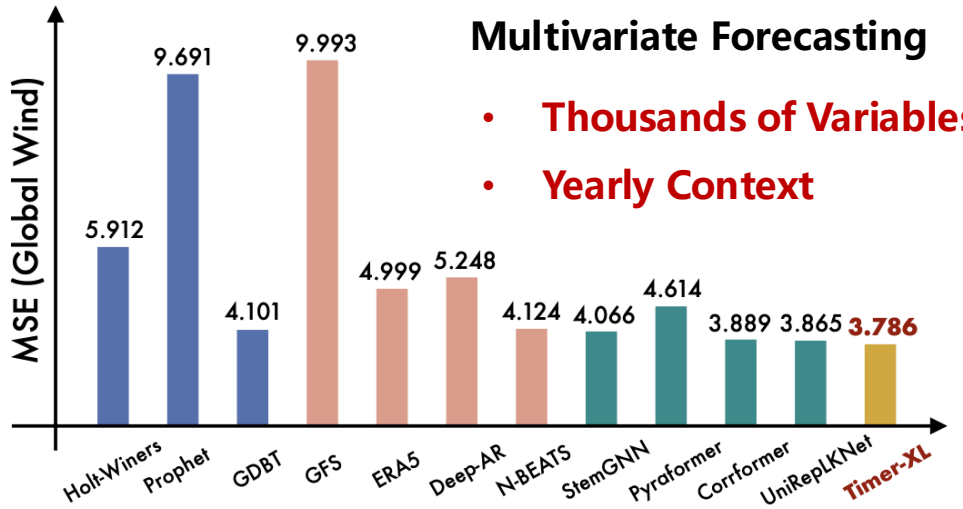
Supervised Training Performance

Univariate Forecasting (Non-Stationary)



Multivariate Forecasting

- Thousands of Variables
- Yearly Context



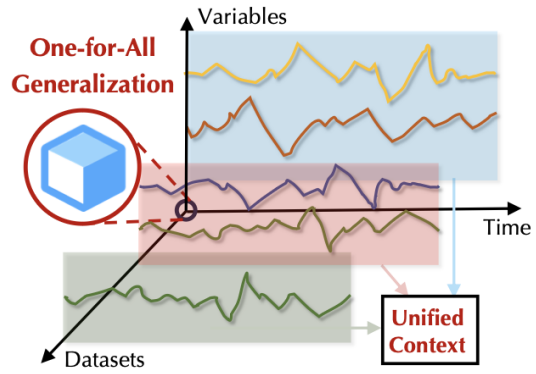
Timer-XL

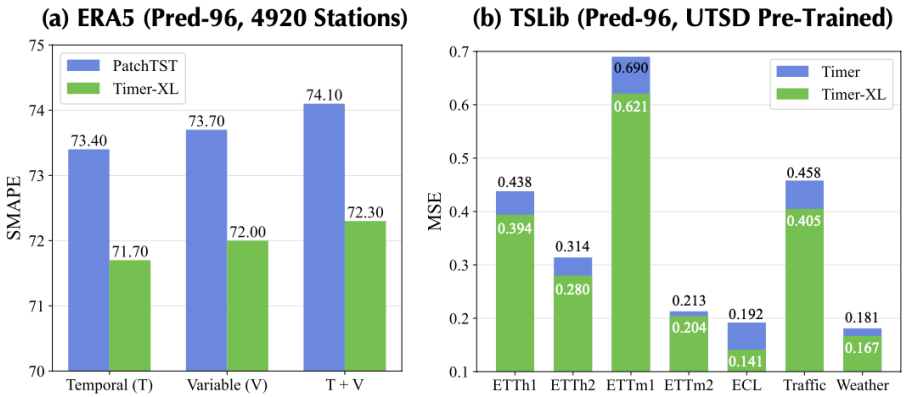
Forecasting with Covariates

Models	Timer-XL (Ours)		Timer-XL (Noncausal)		TimeXer (2024b)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
NP	**0.234**	**0.262**	0.237	0.265	0.238	0.268
PJM	0.089	0.187	0.092	0.188	**0.088**	0.188
BE	**0.371**	**0.243**	0.410	0.279	0.379	**0.243**
FR	**0.381**	**0.204**	0.406	0.220	0.384	0.208
DE	**0.434**	**0.415**	0.435	**0.415**	0.440	0.418
Average	**0.302**	**0.262**	0.316	0.273	0.306	0.265

Outperform Task-Specific Models

Large-Scale Pre-Training & Zero-Shot Forecasting





Pre-Training Large Time-Series Model

Zero-Shot Forecasting (Pre-trained on 260B Time Points)

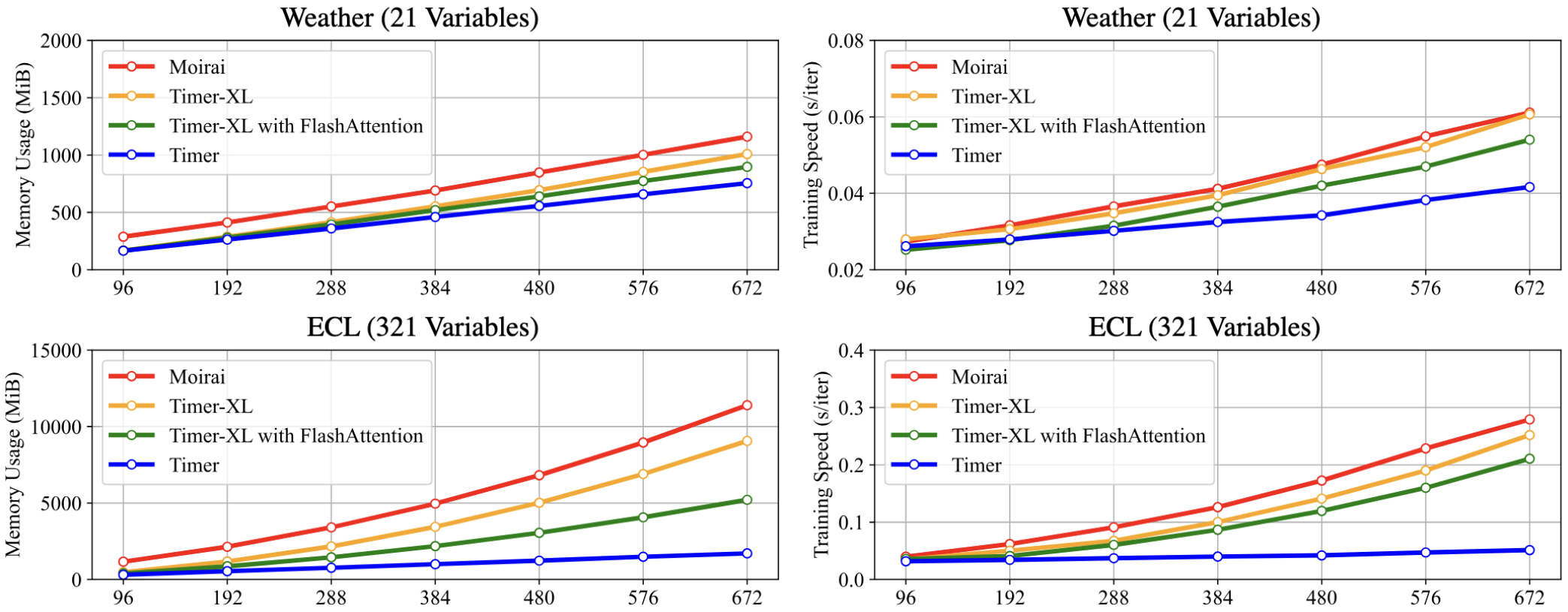
Table 7: Averaged results of zero-shot forecasting. Full results of all prediction lengths are provided in Table 13. 1st Count represents the number of wins achieved by a model under all prediction lengths and datasets. The configuration of **Timer-XL**$_{Base}$ shown in Table 11 is comparable with **Moirai**$_{Base}$, which is pre-trained on UTSD (Liu et al., 2024c) and LOTSA (Woo et al., 2024).

Models	**Timer-XL**$_{Base}$ (Ours)		**Time-MoE**$_{Base}$ (2024)		**Time-MoE**$_{Large}$ (2024)		**Time-MoE**$_{Ultra}$ (2024)		**Moirai**$_{Small}$ (2024)		**Moirai**$_{Base}$ (2024)		**Moirai**$_{Large}$ (2024)		**TimesFM** (2023)		**MOMENT** (2024)		**Chronos**$_{Base}$ (2024)		**Chronos**$_{Large}$ (2024)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.373	0.392	0.394	0.415	0.376	0.405	**0.356**	0.391	0.436	0.410	0.406	**0.385**	0.422	0.391	0.433	0.418	0.670	0.536	0.645	0.500	0.555	0.465
ETTm2	**0.273**	**0.336**	0.317	0.365	0.316	0.361	0.288	0.344	0.307	0.347	0.311	0.337	0.329	0.343	0.328	0.346	0.316	0.365	0.310	0.350	0.295	0.338
ETTh1	0.404	**0.417**	0.400	0.424	**0.394**	0.419	0.412	0.426	0.428	0.427	0.417	0.419	0.480	0.439	0.473	0.443	0.683	0.566	0.591	0.468	0.588	0.466
ETTh2	**0.347**	0.388	0.366	0.404	0.405	0.415	0.371	0.399	0.361	0.384	0.362	0.382	0.367	**0.377**	0.392	0.406	0.361	0.409	0.405	0.410	0.455	0.427
ECL	**0.174**	0.278	-	-	-	-	-	-	0.218	0.303	0.187	0.274	0.186	**0.270**	-	-	0.765	0.686	0.214	0.278	0.204	0.273
Weather	**0.256**	0.294	0.265	0.297	0.270	0.300	**0.256**	0.288	0.275	0.286	0.287	0.281	0.264	**0.273**	-	-	0.294	0.326	0.292	0.315	0.279	0.306
1st Count	**15**	**10**	2	1	3	0	10	7	0	0	0	5	1	**10**	0	1	2	0	0	0	0	2

The model checkpoint is available at: https://huggingface.co/thuml/timer-base-84m.

Model Efficiency

Evaluating Memory/FLOPS of Time-Series Transformers



Efficiency - Context Length

Model Efficiency

Computational Complexity of Time-Series Transformer

- **FFN**: Linear growth with the context length - $O(NT)$ ***Dominate Term in TS!***
- **Attention**: Quadratic growth with the context length - $O(N^2T^2)$

Table 8: Parameters count and computational complexity of Transformers for multivariate time series.

Metric	Type	Count	Complexity
FLOPs (Training Speed)	Channel Independence	$12\big(PDNT + L(D+H)NT^2 + (2+\alpha)LD^2NT\big)$	$\mathcal{O}\big(LDNT(D+T)\big)$
	Channel Dependence	$12\big(PDNT + L(D+H)N^2T^2 + (2+\alpha)LD^2NT\big)$	$\mathcal{O}\big(LDNT(D+NT)\big)$
Parameters	Encoder-Only	$(4+2\alpha)LD^2 + 4LD + (1+T)PD$	$\mathcal{O}(LD^2)$
	Decoder-Only	$(4+2\alpha)LD^2 + 4LD + 2PD$	$\mathcal{O}(LD^2)$
Memory Footprint	Self-Attention	$4(D+P)NT + (32+8\alpha)LDNT + 4LHN^2T^2$	$\mathcal{O}\big(LHN^2T^2\big)$
	FlashAttention	$4(D+P)NT + (32+8\alpha)LDNT$	$\mathcal{O}\big(LDNT\big)$

$*$ L is the block number of Transformers. D is the dimension of embeddings (the hidden dimension of FFN D_{ff} is set as αD). H is the head number and the dimension of query, key, and value $d_k = D/H$. The overhead is to train on a multivariate time series (N-variables and TP time points) with patch token length P and context length T. Set $N = 1$ for training on univariate time series.

Model Analysis

Non-stationary Forecasting

Table 16: Evaluations (672-pred-96) on the effect of ReVIN (Kim et al., 2021) on Transformers.

Models	Timer-XL with ReVIN		Timer-XL w/o ReVIN		PatchTST with ReVIN		PatchTST w/o ReVIN	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.364	0.397	0.370	0.401	0.370	0.399	0.421	0.448
Weather	0.157	0.205	0.151	0.205	0.149	0.198	0.173	0.242
ECL	0.127	0.219	0.130	0.225	0.129	0.222	0.138	0.244

Small Gap

Big Gap

- Long-context Transformers do not rely on Stationarization

Ablation Study

Table 14: Embedding ablation in TimeAttention. For the temporal dimension, we compare prevalent relative and absolute position embeddings. As for the variable dimension, we explore the effectiveness of the variable embedding that distinguishes endogenous and exogenous variables.

Design	Temporal	Variable	Traffic		Weather		Solar-Energy		ERA5-MS	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Timer-XL	**RoPE** (2024)	**with**	**0.340**	**0.238**	**0.157**	**0.205**	**0.162**	0.221	**0.164**	0.307
Replace	ALiBi (2021)	with	0.351	0.246	0.162	0.212	0.188	0.210	0.167	0.308
	Relative (2020)	with	0.361	0.250	0.163	0.214	0.197	0.215	0.168	0.309
	Absolute (2017)	with	0.381	0.270	0.159	0.207	0.171	**0.204**	0.165	**0.306**
w/o	RoPE (2024)	w/o	0.361	0.254	0.171	0.217	0.181	0.221	0.235	0.373
	w/o	w/o	0.363	0.253	0.164	0.215	0.194	0.215	0.167	0.309

$$\mathcal{A}_{mn,ij} = \mathbf{h}_{m,i}^{\top} \mathbf{W}_q \mathbf{R}_{\theta,i-j} \mathbf{W}_k^{\top} \mathbf{h}_{n,j} + u \cdot \mathbb{1}(m=n) + v \cdot \mathbb{1}(m \neq n)$$

Temporal

Variable

- RoPE outperformers other counterparts
- It is helpful to distinguish endogenous and exogenous variables

Interpretability

Attention Map

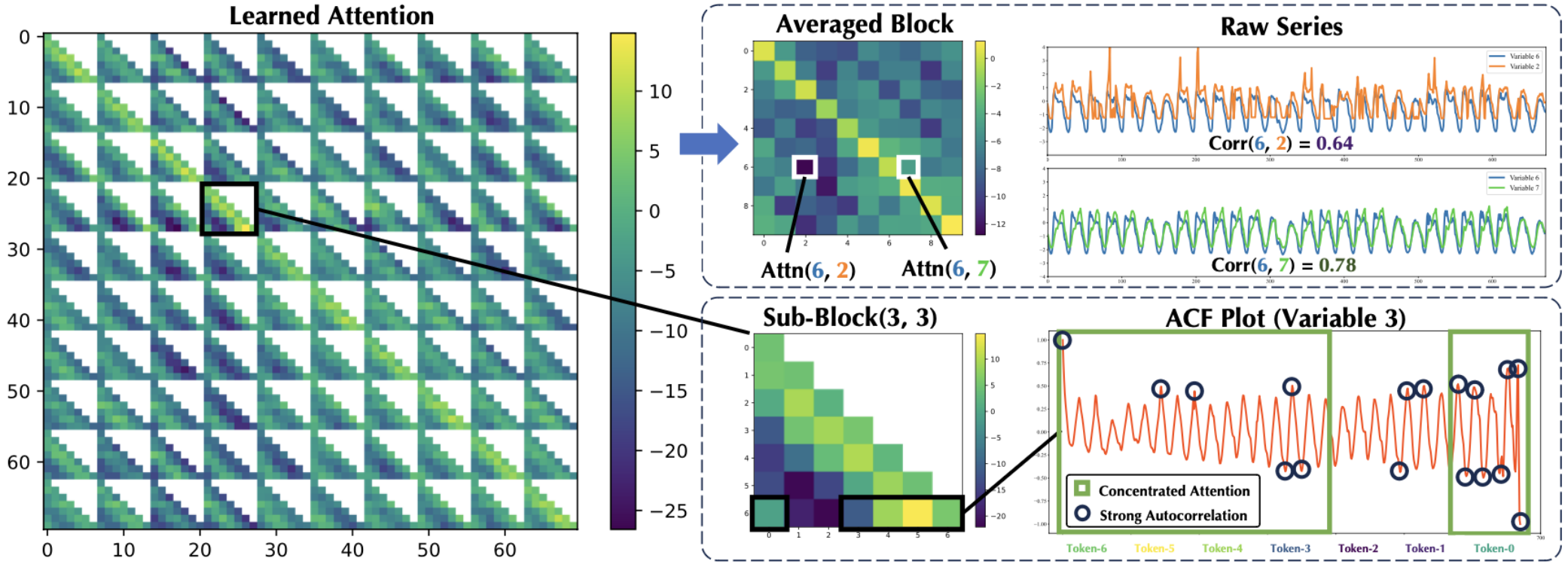


Figure 7: Visualization of TimeAttention. It is from the first sample of a length 672 in the test split of Traffic. We visualize the last 10 variables with each contains 7 tokens. We present auto-correlation function plot. Auto-correlation can be reflected by the distribution of attention scores (bottom right). We average TimeAttention across sub-blocks, which indicates Pearson correlations (upper right).

Open Source

Timer (Large Time Series Model)

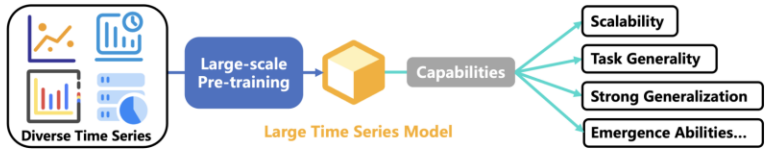
This repo provides official code, datasets and checkpoints for Timer: Generative Pre-trained Transformers Are Large Time Series Models. [Poster], [Slides].

Updates

- **News** (2024.6) Pre-training dataset (UTSD) is available in HuggingFace. Dataloader is also contained.
- **News** (2024.5) Accepted by ICML 2024, a camera-ready version of **31 pages**.
- **News** (2024.4) The pre-training scale has been extended, enabling zero-shot forecasting.
- **News** (2024.2) Releasing model checkpoints and code for adaptation.

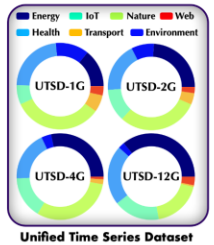
Introduction

Time Series Transfor**mer** (Timer) is a Generative Pre-trained Transformer for general time series analysis. You can visit our Homepage for a more detailed introduction.

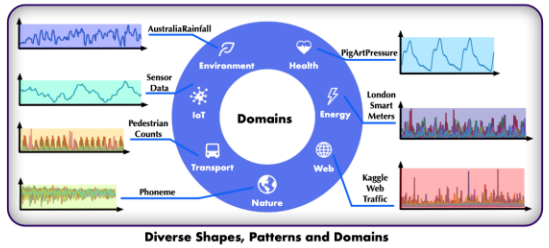


Datasets

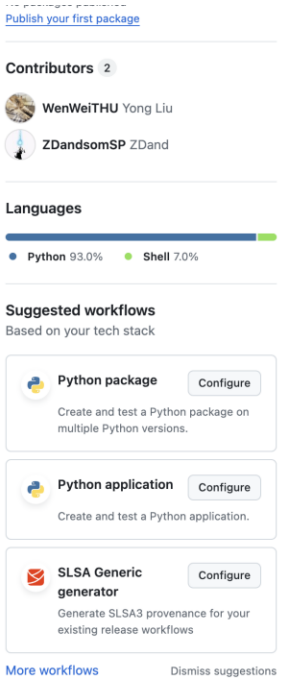
We curate Unified Time Series Datasets (UTSD) comprised of **1B time points** and **4 volumes** to facilitate the research on large time series models and pre-training.



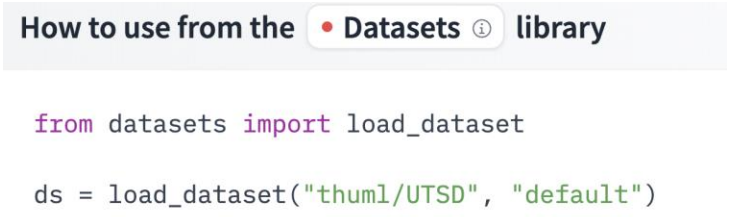
Unified Time Series Dataset



Diverse Shapes, Patterns and Domains



Timer-Base (Pre-Trained on 260B) is Released!



All in one for using/developing LTMs: Pre-trained checkpoint, dataset, and fine-tuning scripts

GitHub: https://github.com/thuml/Large-Time-Series-Model

Checkpoint: https://huggingface.co/thuml/timer-base-84m

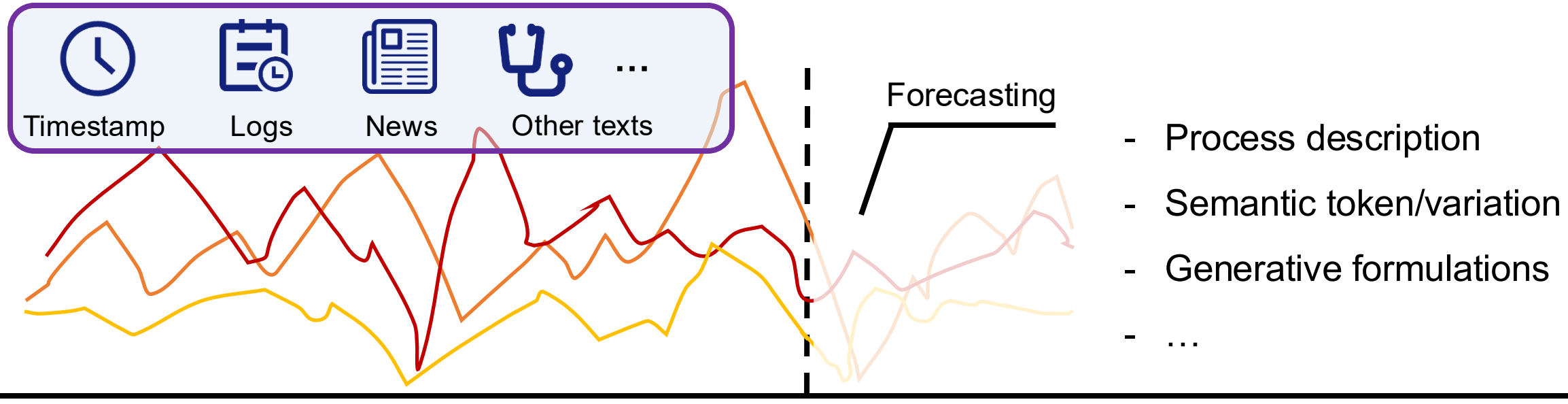
Text-Informed Time Series Forecasting

Industrial | **Finance** | **Climate** | **Health** | **IoT**





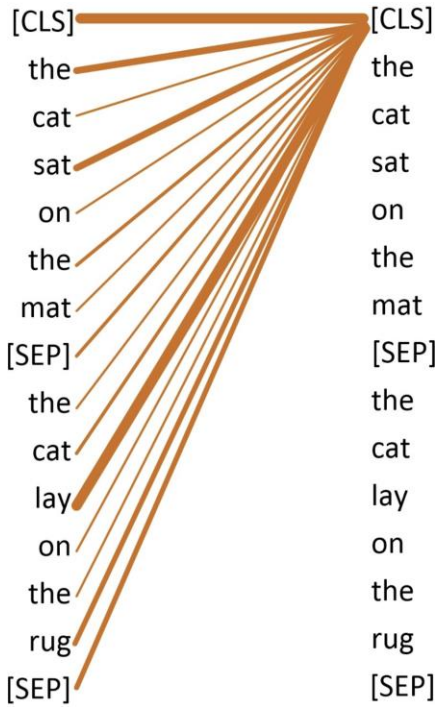
Time series and natural language always go together



- Process description
- Semantic token/variation
- Generative formulations
- …

LLMs for Time Series: Motivations

Align time series and natural language



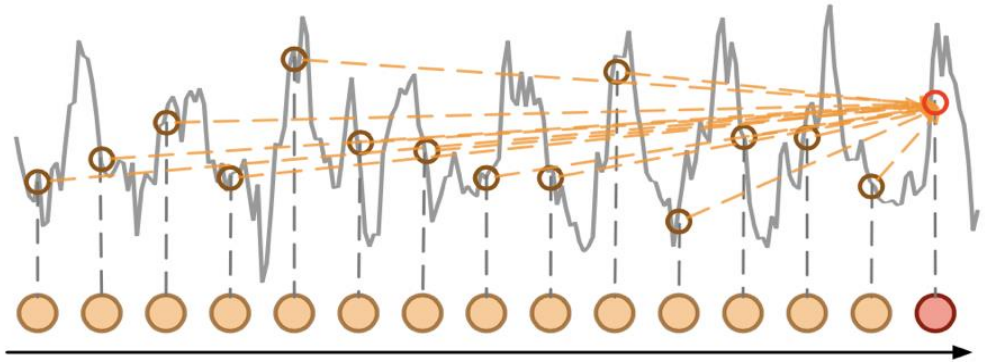
Dependencies of language tokens

Language modeling (Bengio et al., 2000):

$$P(\mathcal{U}) = \prod_{i=1}^{N} p(u_i | u_{<i})$$

Time series forecasting:

$$P(\mathbf{x}_{L+1:L+F} | \mathbf{x}_{1:L}), \mathbf{x} \in \mathbb{R}^C$$

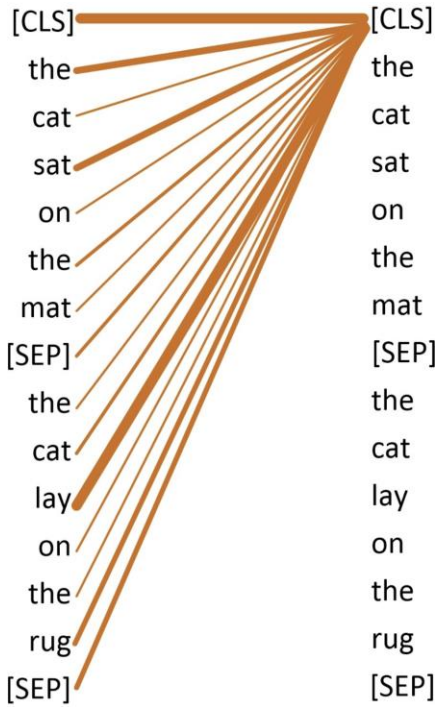


Dependencies of time points

Goal of LLM4TS: Leverage off-the-shelf LLMs as foundation models for time series

LLMs for Time Series: Motivations

Align time series and natural language



Large Language Models

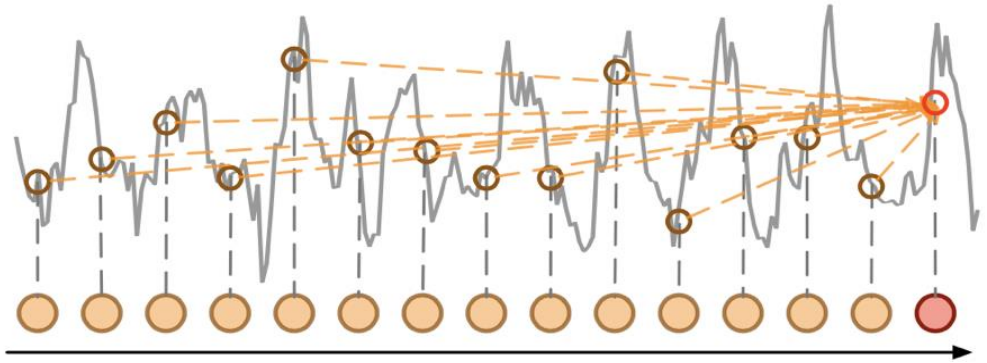
- Token Semantics
- Token Transitions
- …

Pre-train

Adaptation

AutoTimes

Large Time-Series Models



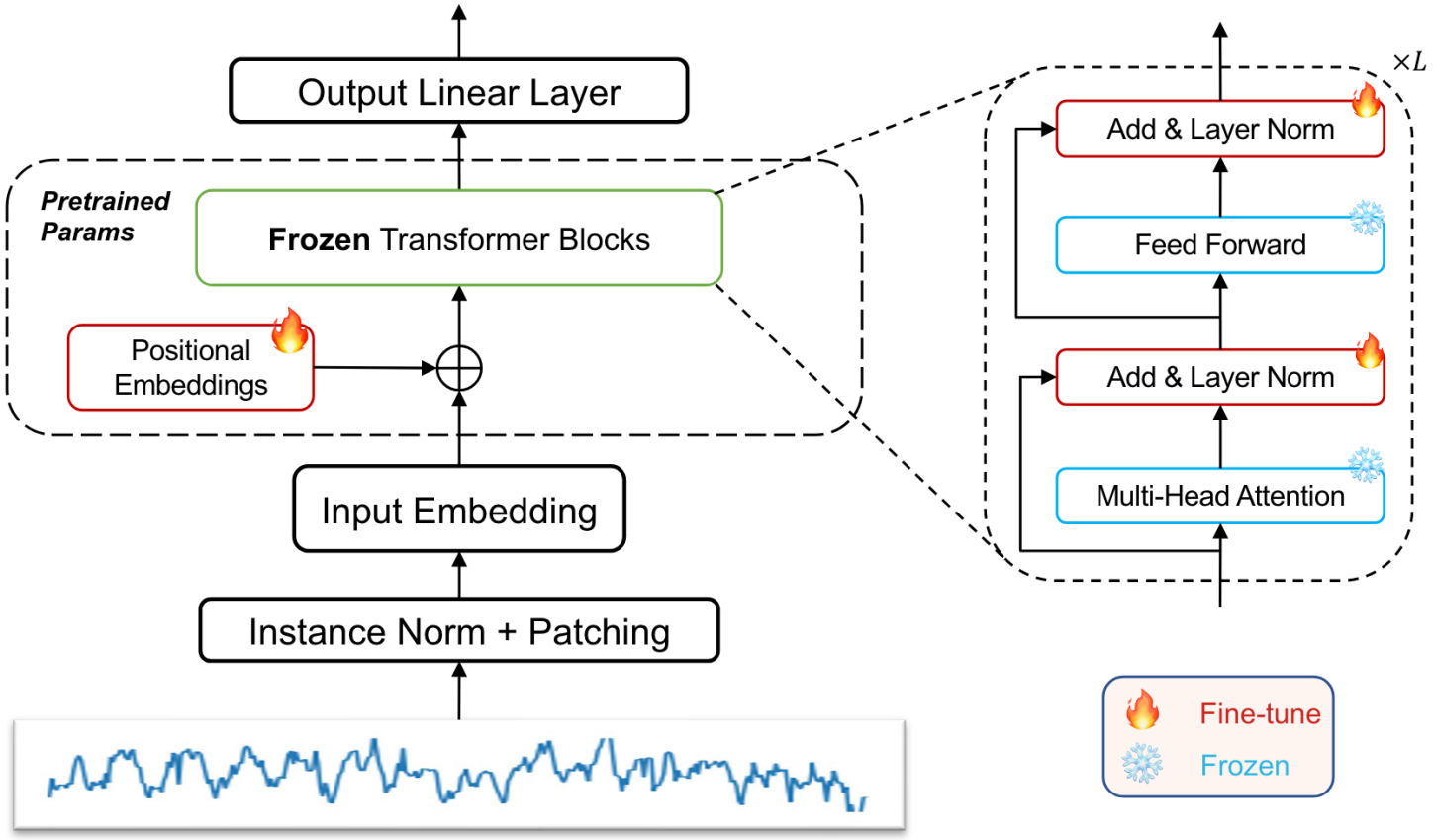
- Limited scale of datasets
- Avoid case-by-case training

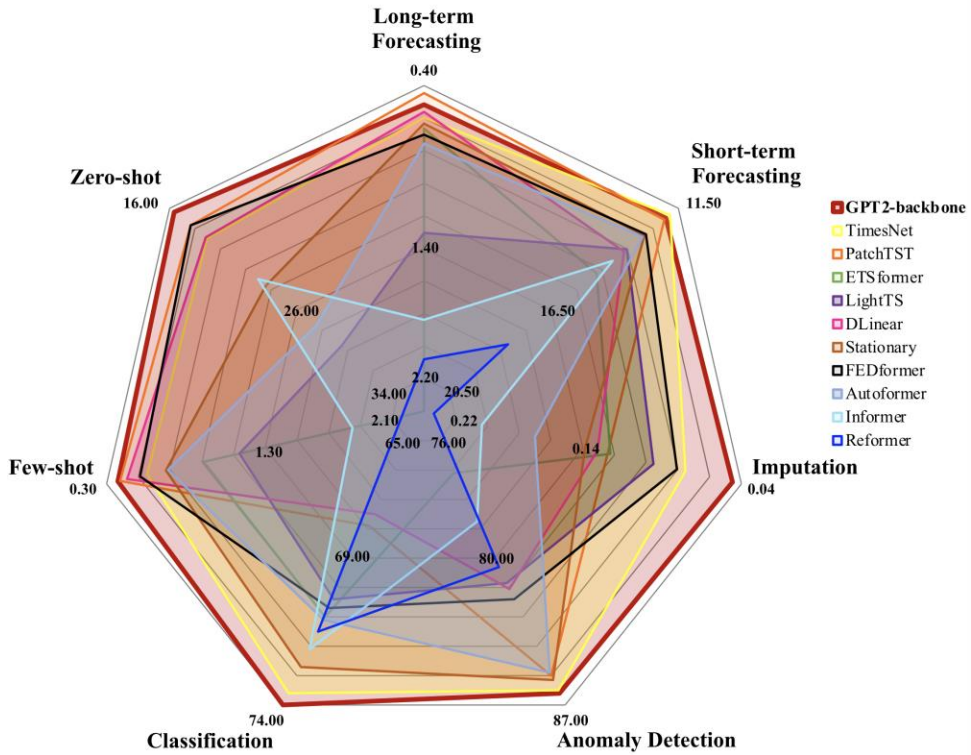
- **Large-scale text corpora**
- **Scalable and versatile architecture**

Goal of LLM4TS: Leverage off-the-shelf LLMs as foundation models for time series

FPT: Fine-tune LLM for Time Series

- Fine-tune **GPT-2** in a **BERT-style** on time series analysis tasks, following TimesNet

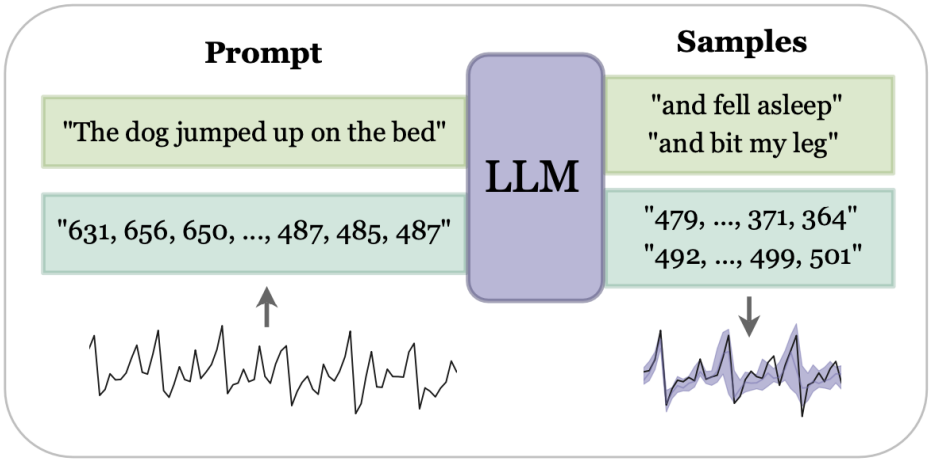


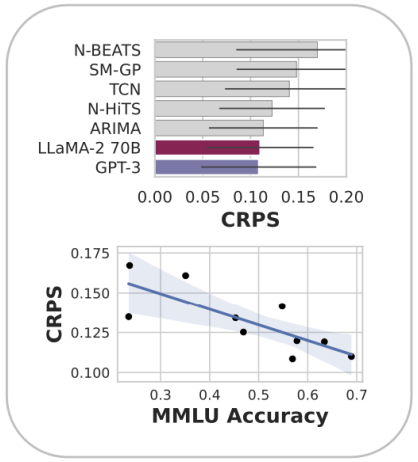


Zhou et al. One Fits All: Power General Time Series Analysis by Pretrained LM. NeurIPS 2023.

LLMTime: Directly Encoding Time Series As Words

$$0.123, 1.23, 12.3, 123.0 \rightarrow \text{" 1 2 , 1 2 3 , 1 2 3 0 , 1 2 3 0 0"}.$$



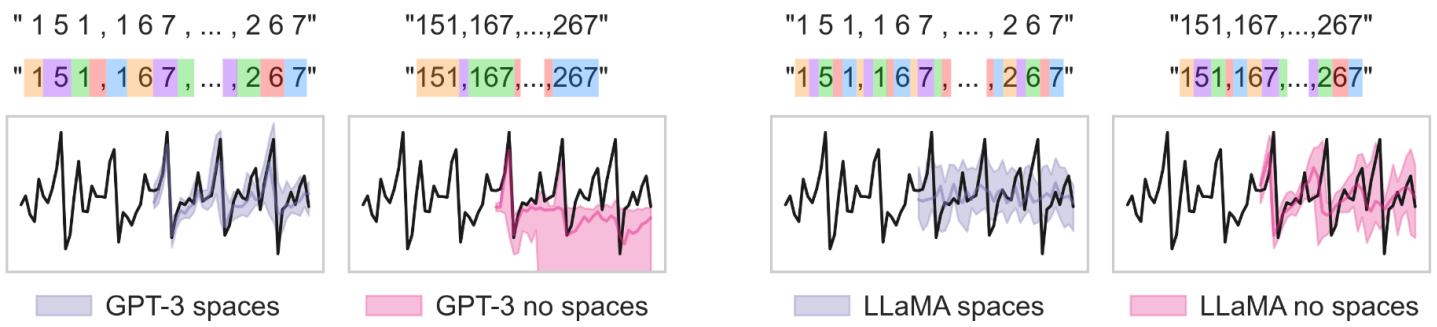


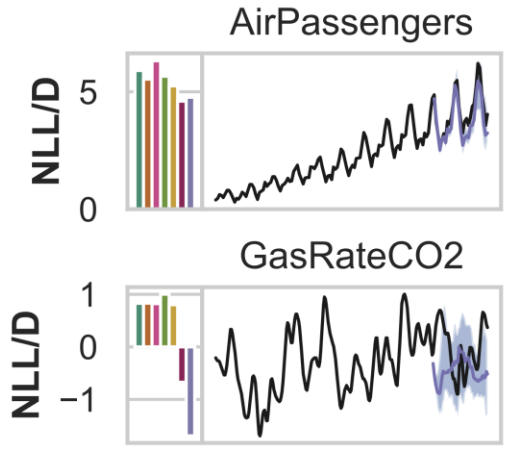
- **Encode TS as numerical tokens series**
- Applied on larger LM (GPT-3, LLaMA)

✓ Conduct **zero-shot forecasting**

✗ Fine-grained tokens: costly to produce multivariate and long predictions

✗ Applicable on simple time series





Gruver et al. Large Language Models Are Zero-Shot Time Series Forecasters. NeurIPS 2024.

Time-LLM: Prompting Time Series With Texts

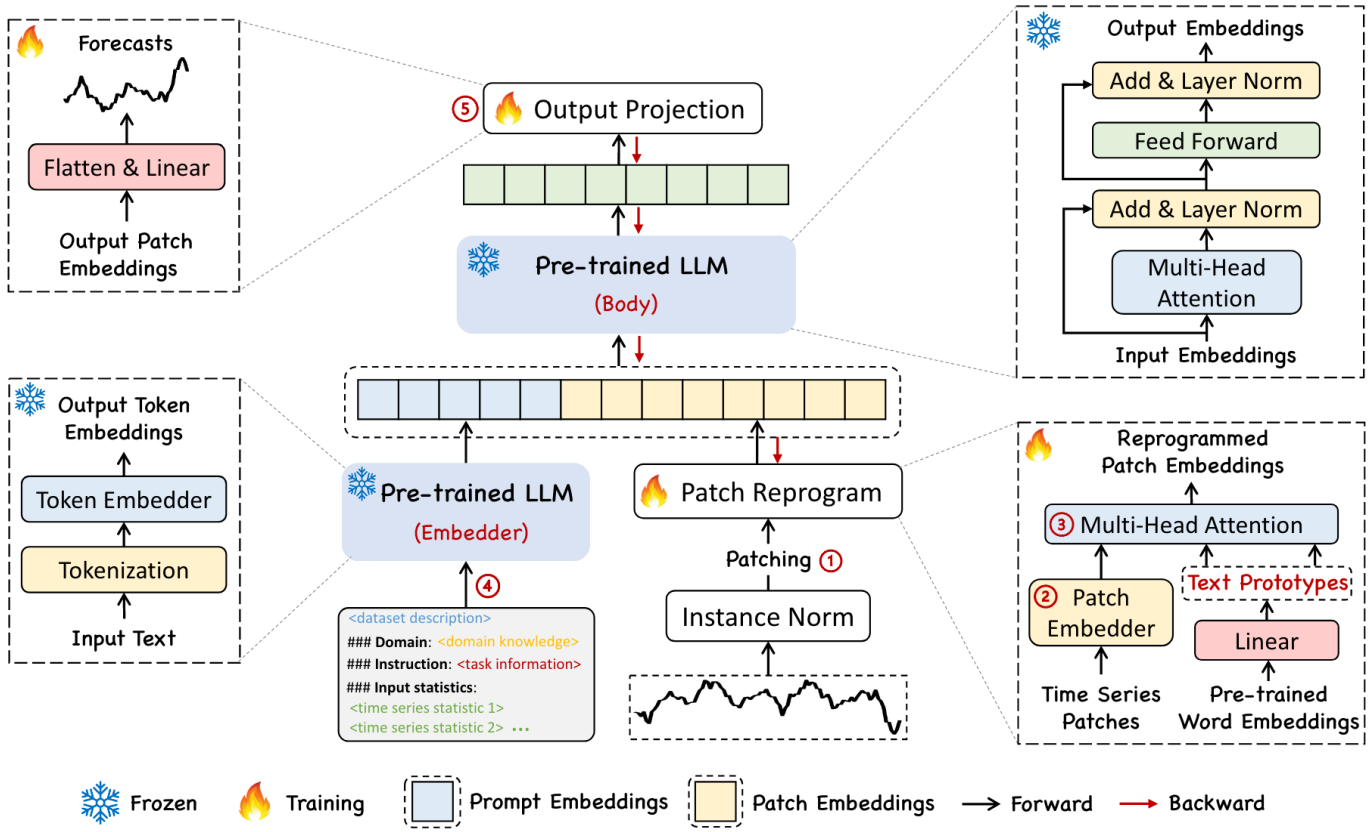


Figure 2: The model framework of TIME-LLM. Given an input time series, we first tokenize and embed it via ① patching along with a ② customized embedding layer. ③ These patch embeddings are then reprogrammed with condensed text prototypes to align two modalities. To augment the LLM's reasoning ability, ④ additional prompt prefixes are added to the input to direct the transformation of input patches. ⑤ The output patches from the LLM are projected to generate the forecasts.

- **Frozen LLM Parameters**
- Concat. patched series with designed **language prompts**
- Flattenen & Proj. (**BERT-style**)

✓ Introduce **textual modality**

✗ Obscure mechanism of utilizing LLMs (Results are still good without LLMs)

✗ Costly to adapt (8 x A100)

Jin et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. ICLR 2024.

Unsolved Questions

Are Language Models Actually Useful for Time Series Forecasting?

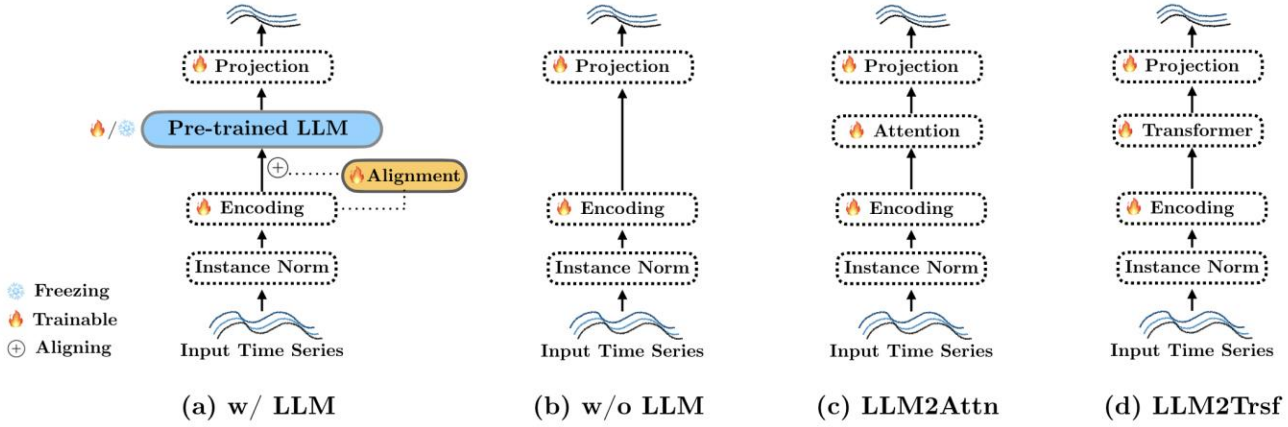
Mingtian Tan
University of Virginia
wtd3gz@virginia.edu

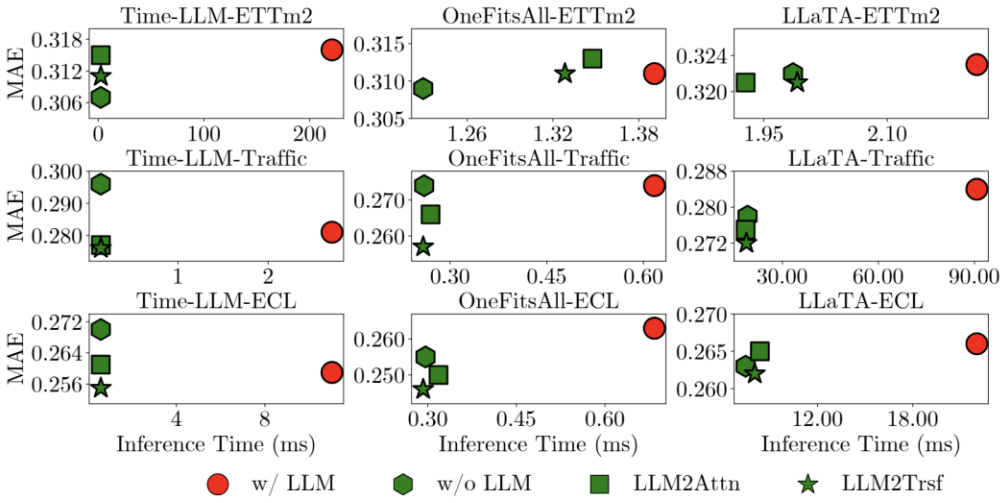
Mike A. Merrill
University of Washington
mikeam@cs.washington.edu

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





- ✗ High adaptation cost (**7B+ Params. In a LLM**)
- ✗ Results are still good **without LLMs**
- ✗ **Patch + Project** is already a simple & effective choice

Tan et al. Are Language Models Actually Useful for Time Series Forecasting? arXiv 2024.

AutoTimes (Ours): Exploring LLM's Potentials for TSF

AutoTimes: Autoregressive Time Series Forecasters via Large Language Models

Yong Liu [*1] **Guo Qin** [*1] **Xiangdong Huang** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Yong Liu

Guo Qin

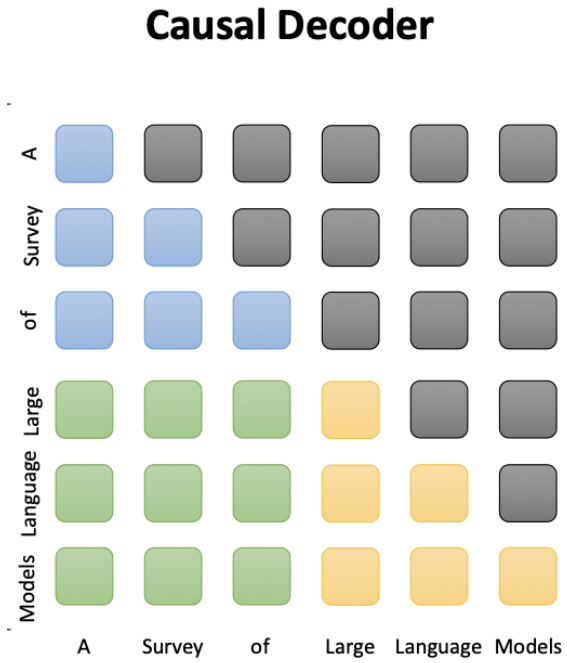
Xiangdong Huang

Jianmin Wang

Mingsheng Long

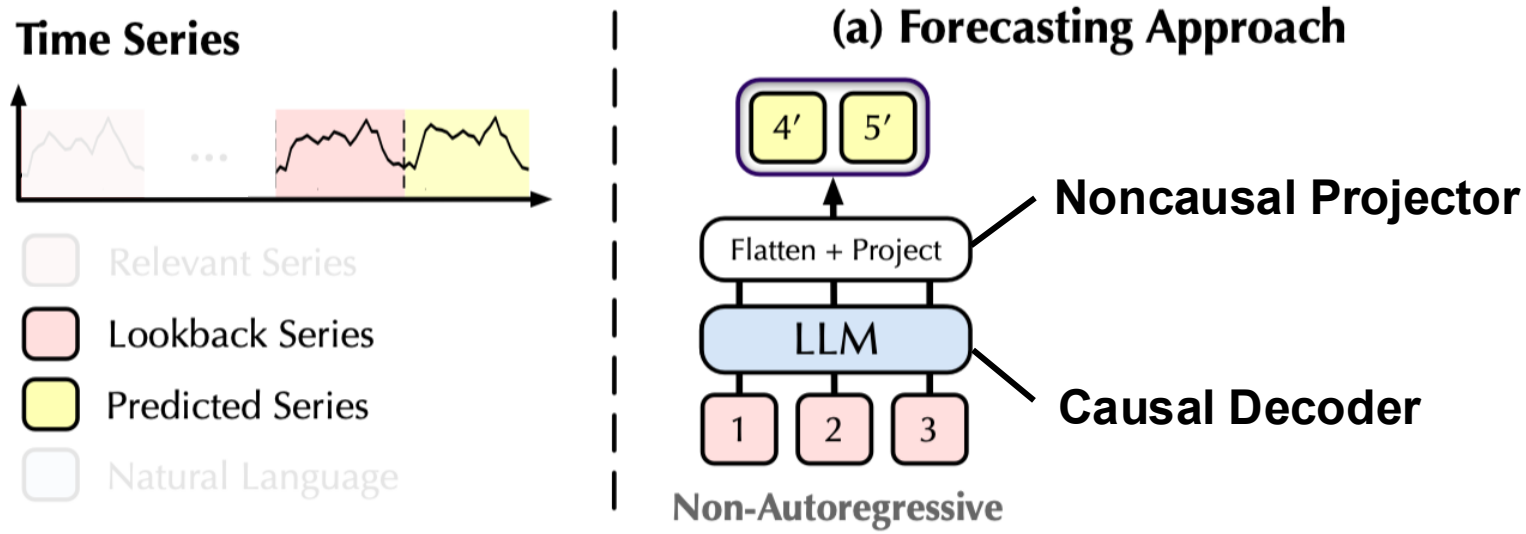
Rethinking Previous LLM4TS Methods

Insufficient utilization of LLMs is caused by several inconsistencies



Casual mask inside each LLM block

✗ **Architecture:** Previous works adapt LLMs, which are GPT-style causal decoders, as encoder-only models in a BERT-style



☹ The token causality are broken in the last projector

Rethinking Previous LLM4TS Methods

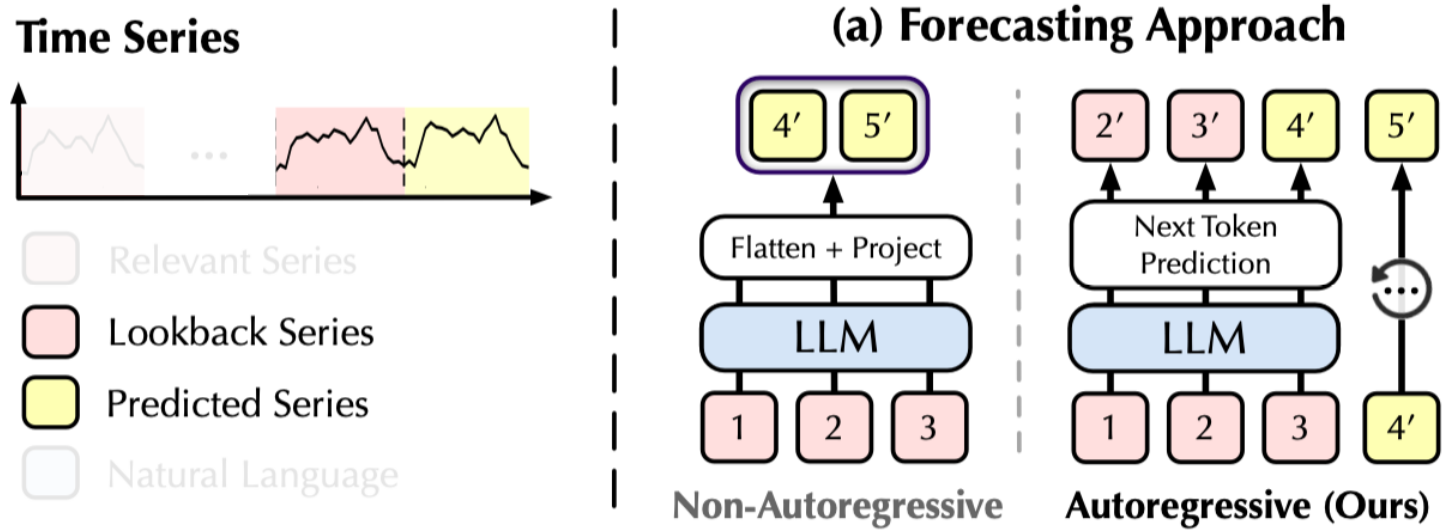
Insufficient utilization of LLMs is caused by several inconsistencies

$$P(\mathcal{U}) = \prod_{i=1}^{N} p(u_i | u_{<i})$$

Multiple supervision under different lengths

Inference with different lengths of input tokens

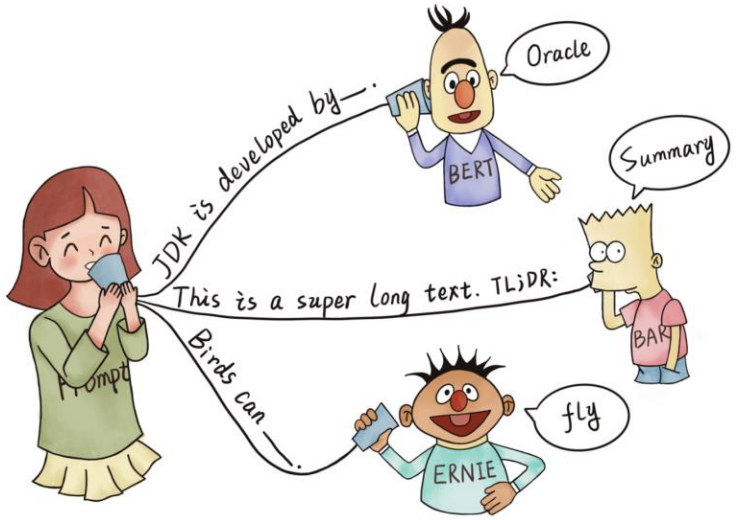
✗ **Autoregression:** LLM predicts the next tokens iteratively, while prevalent forecasters obtain all tokens in one step



☹ The outcome forecaster is only available for specific length

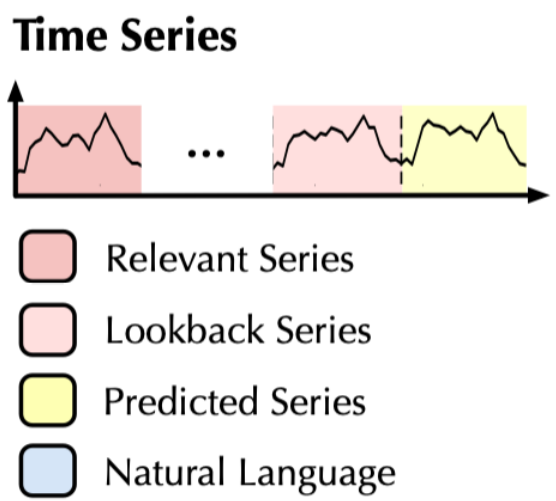
Revitalize LLMs for Time Series Modality

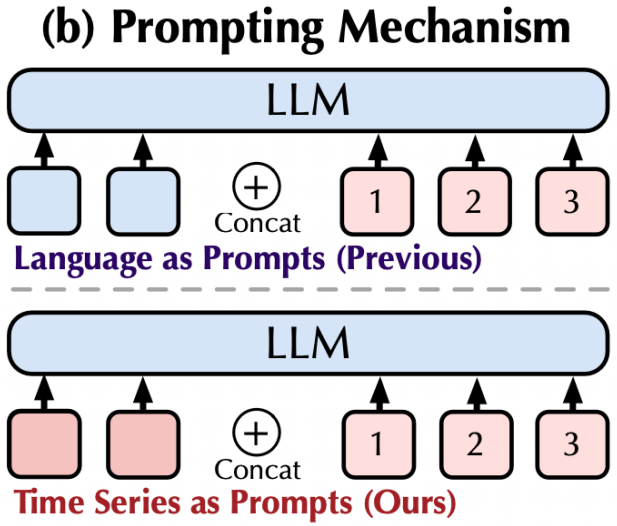
Exploration of advanced capabilities of language models



Prompts aim to elicit better responses from large models

- **Prompting**: we formulate time series as prompts, extending the context for prediction beyond the lookback window





☹ Language prompts for TSF lead to modality gap

Liu et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM 2023.

Revitalize LLMs for Time Series Modality

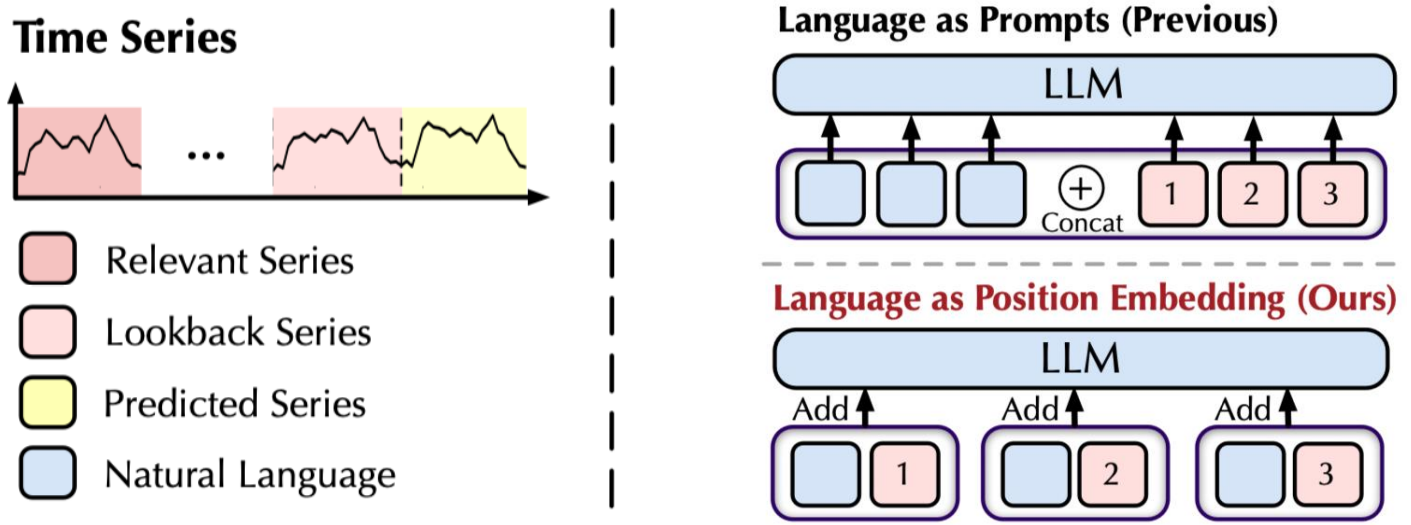
Exploration of advanced capabilities of language models

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>.
[END DATA]

Long language prompts designed for time series

- **Multimodal:** we use LLM-embedded textual timestamps to utilize chronological information and align multivariate series

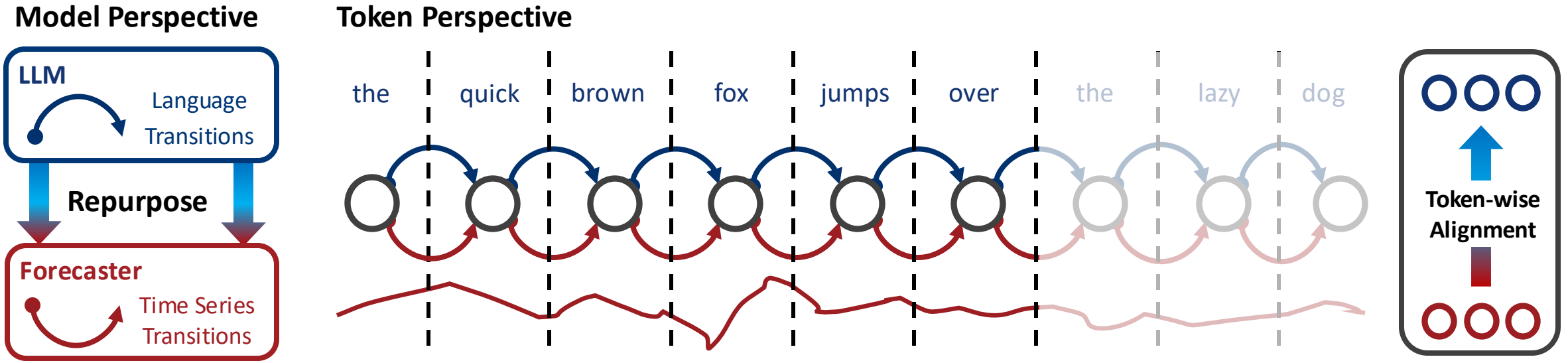


☹ **Language prompts for TSF lead to excessive contexts**

Jin et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. ICLR 2024.

Key Idea

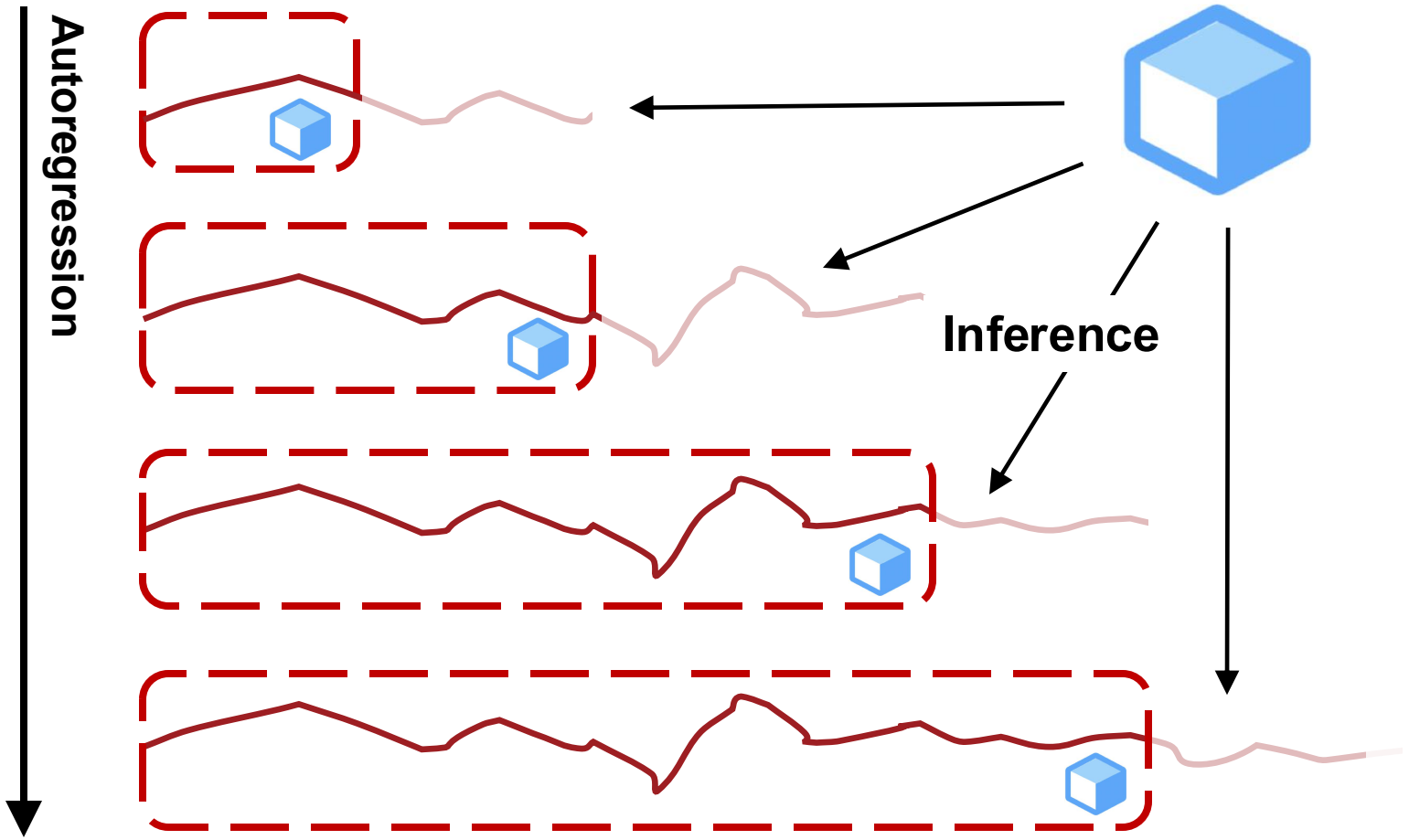
Language token transitions are general-purpose and transferable



- ✓ **Approach:** Reuse the general-purpose token transition
- ✓ **Alignment**: Embed time series into latent language representations
- ✓ **Potentials:** Autoregressive generation with inherited LLM capabilities

Key Idea

Autoregressive LLMs are arbitrary-length time series forecasters



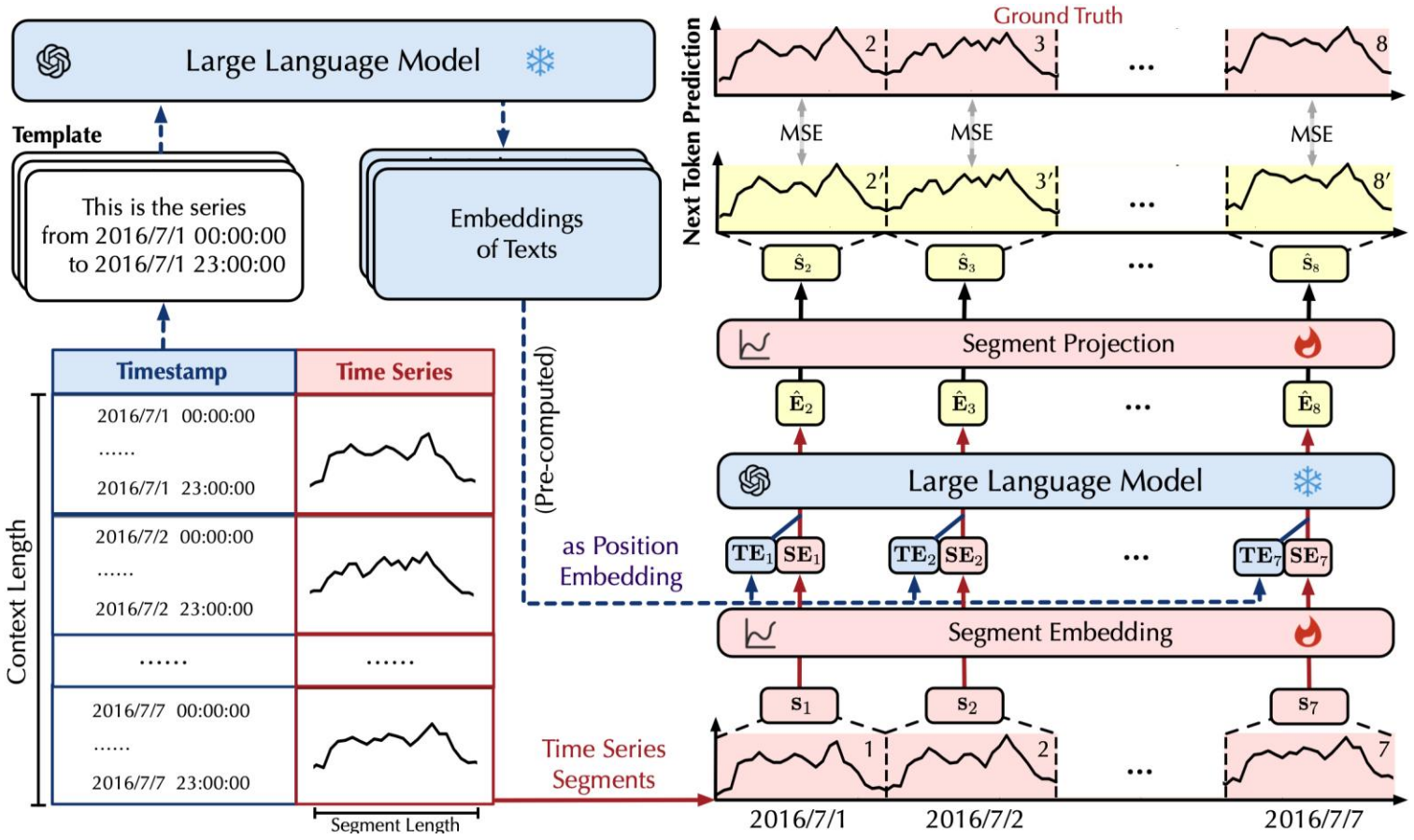
$$P(\mathcal{U}) = \prod_{i=1}^{N} p(u_i | u_{<i})$$

Token-wise supervision

$$f : (\mathbf{x}_{1:L}, \mathbf{a}_{1:L+F}) \mapsto \hat{\mathbf{x}}_{L+1:L+F}.$$

- ✓ Arbitrary lookback length L
- ✓ Arbitrary prediction length F
- ✓ Covariates: $\mathbf{a}_{1:L+F}$

Method Pipeline



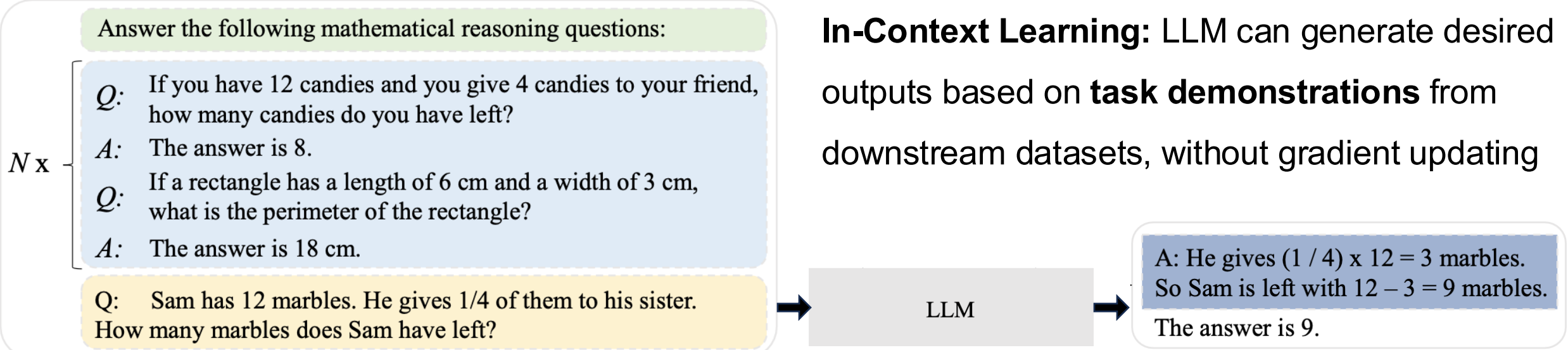
Tokenization: regard time series segments as basic language tokens

Modality-Mixing: Incorporate textual covariates (timestamp) to align variates

Freeze the LLM: Train minimal parameters by next token prediction

Inference: Generate arbitrary-length time series autoregressively like LLMs

In-Context Learning



In-Context Learning: LLM can generate desired outputs based on **task demonstrations** from downstream datasets, without gradient updating

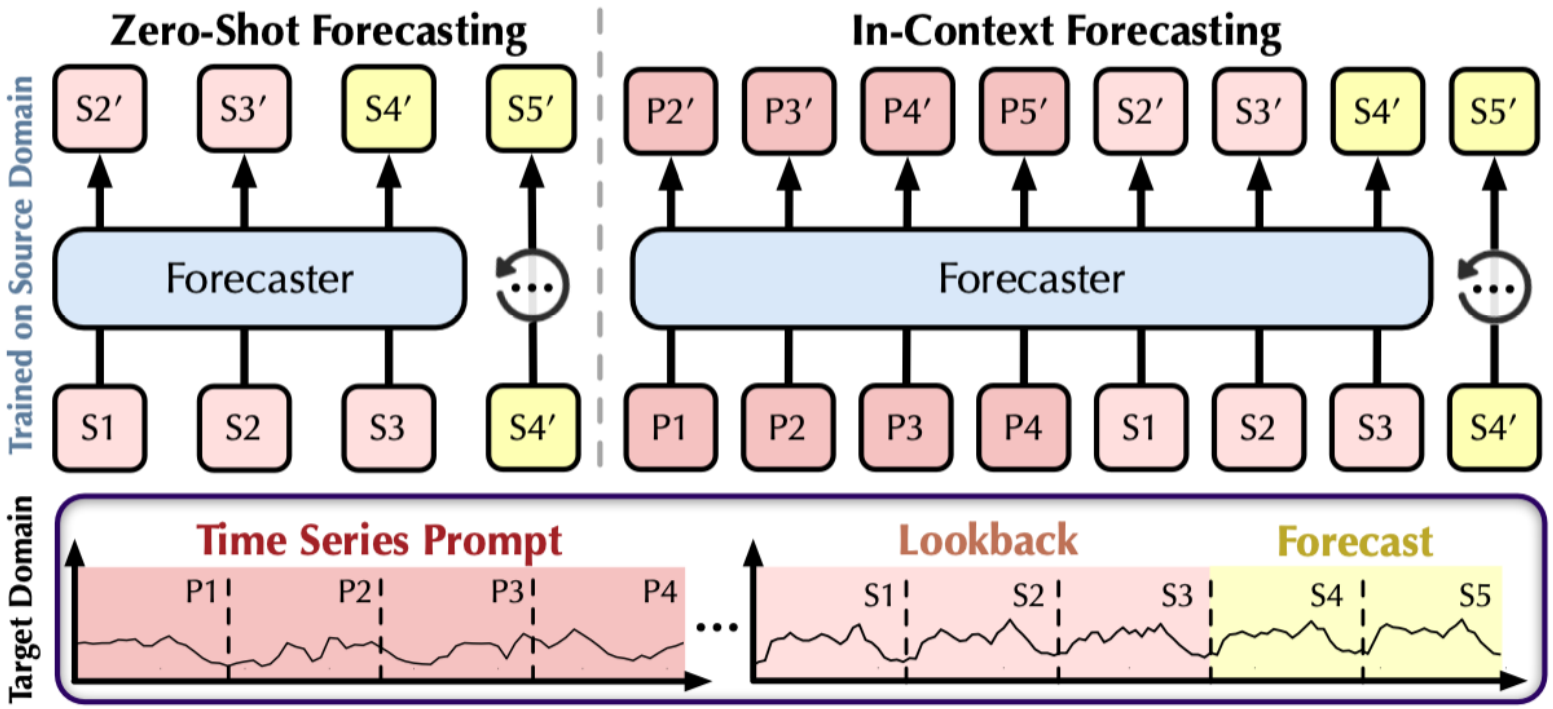
Task Demonstrations: Question-answer pairs in natural language, from an unseen task

Inference: Combine the current question with task demonstrations (prompt) as the input

Based on the token-wise alignment and full reutilization of token transition, AutoTimes can seamlessly transfer ICL to the time series modality

In-Context Forecasting

We propose in-context forecasting for time series



Time Series Forecasting:

$$f : (\mathbf{x}_{1:L}, \mathbf{a}_{1:L+F}) \mapsto \hat{\mathbf{x}}_{L+1:L+F}.$$

Time Series Prompt:

$$\mathcal{C} = \{tsp^{(j)} = \mathbf{x}_{\leq t_j}\}, t_j \leq L.$$

Earlier historical time series
(perhaps non-consecutive)

In-Context Forecasting:

$$f : (\underline{\mathcal{C}}, \mathbf{x}_{1:L}, \mathbf{a}_{1:L+F}) \mapsto \hat{\mathbf{x}}_{L+1:L+F}.$$

Prediction Demonstrations: Retrieve time series as prompts from the target domain

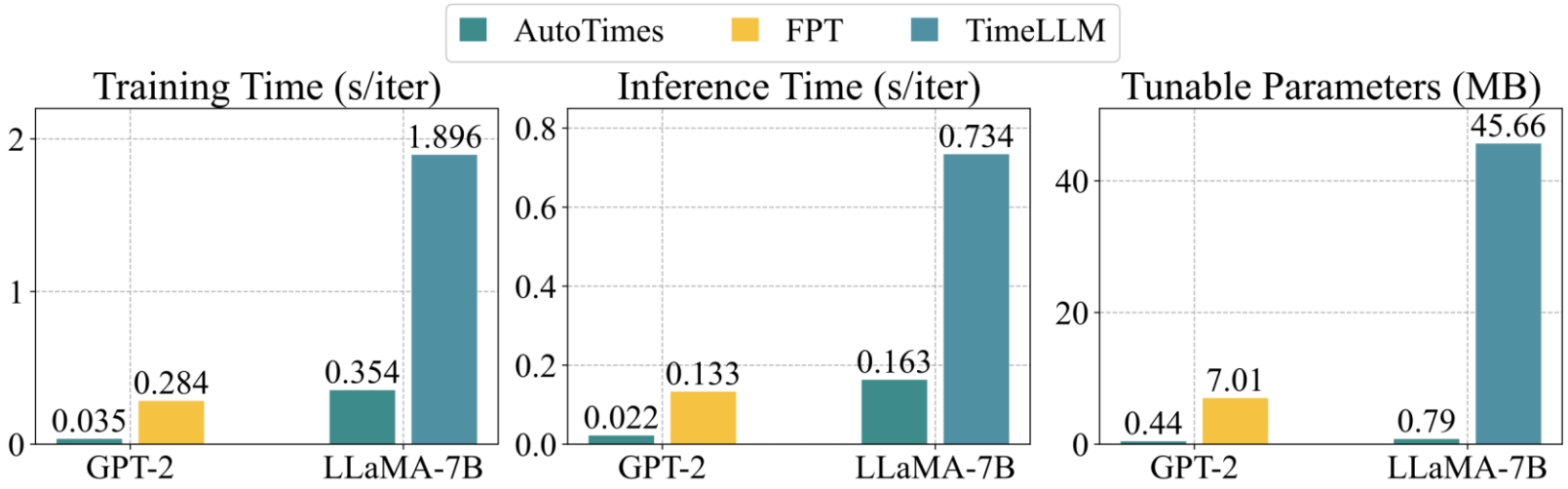
Inference: Input "prompt-lookback" sentence into our model without updating parameters

Comparison of LLM4TS

Quality assessments (none of prior LLM4TS methods achieved all three)

Method	**AutoTimes**	TimeLLM [15]	UniTime [21]	FPT [49]	LLMTime [13]	TEST [34]	TEMPO [7]	PromptCast [44]
Autoregressive	✓	✗	✗	✗	✓	✗	✗	✗
Freeze LLM	✓	✓	✗	✗	✓	✓	✗	✓
Multimodal	✓	✓	✓	✗	✗	✓	✓	✓

Minimal tunable parameters -> Better performance/model efficiency



15min to repurpose **LLaMA-7B** on **a RTX 3090-24G**

(8 x A100 for Time-LLM)

Ablation Study

True utilization of large language model (different from non-autoregressive LLM4TS methods)

Table 6: We follow the protocol of LLM4TS ablation studies [35] to verify whether the LLM is truly useful in our AutoTimes: (1) *w/o LLM* replaces the language model entirely and passing input tokens directly to the last layer; (2) *LLM2Attn* replaces the language model with a single multi-head attention layer; (3) *LLM2Trsf* replaces the language model with a single transformer block.

Dataset	ETTh1								ECL							
Type	**AutoTimes**		w/o LLM		LLM2Attn		LLM2Trsf		**AutoTimes**		w/o LLM		LLM2Attn		LLM2Trsf	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Pred-96	**0.360**	**0.400**	0.365	0.399	0.383	0.404	0.377	0.401	**0.129**	**0.225**	0.171	0.263	0.156	0.255	0.162	0.263
Pred-192	**0.388**	**0.419**	0.405	0.425	0.414	0.422	0.406	0.420	**0.147**	**0.241**	0.192	0.282	0.178	0.276	0.189	0.287
Pred-336	**0.401**	**0.429**	0.429	0.441	0.431	0.432	0.421	0.431	**0.162**	**0.258**	0.216	0.304	0.198	0.295	0.216	0.309
Pred-720	**0.406**	**0.440**	0.450	0.468	0.456	0.454	0.449	0.452	**0.199**	**0.288**	0.264	0.342	0.230	0.320	0.258	0.340

Tan et al. Are Language Models Actually Useful for Time Series Forecasting? NeurIPS 2024.

Forecasting Performance

Long-term forecasting (one-for-all rolling forecasting)

Models	**AutoTimes**		TimeLLM [15]		UniTime [21]		FPT [48]		iTrans. [22]		DLinear [44]		PatchTST [26]		TimesNet [41]	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.389**	**0.422**	0.412	0.437	0.683	0.596	0.429	0.439	0.421	0.445	0.426	0.444	0.409	0.430	0.495	0.491
ECL	**0.159**	**0.253**	0.181	0.288	0.325	0.399	0.184	0.284	0.164	0.258	0.165	0.265	0.169	0.268	0.201	0.303
Weather	0.235	0.273	**0.225**	**0.266**	0.461	0.459	0.228	**0.266**	0.266	0.291	0.239	0.291	0.226	0.268	0.264	0.293
Traffic	**0.374**	**0.264**	0.410	0.303	0.584	0.367	0.461	0.326	0.384	0.274	0.423	0.298	0.391	0.275	0.602	0.322
Solar.	**0.197**	**0.242**	0.263	0.335	0.392	0.462	0.236	0.303	0.213	0.291	0.222	0.283	0.202	0.269	0.213	0.295

One LLM-forecasters can outperform each deep models trained on specific lengths

Short-term forecasting (in-distribution)

Models		**AutoTimes**	TimeLLM	FPT	Koopa	N-HiTS	DLinear	PatchTST	TimesNet	FiLM	N-BEATS
Average	sMAPE	**11.831**	11.983	11.991	11.863	11.960	12.418	13.022	11.930	12.489	11.910
	MASE	**1.585**	1.595	1.600	1.595	1.606	1.656	1.814	1.597	1.690	1.613
	OWA	**0.850**	0.859	0.861	0.858	0.861	0.891	0.954	0.867	0.902	0.862

State-of-the-art performance

Zero-shot forecasting (out-of-distribution)

Models	**AutoTimes**	FPT	DLinear	PatchTST	TimesNet	NSFormer	FEDFormer	Informer	Reformer
M4 → M3	**12.75**	13.06	14.03	13.06	14.17	15.29	13.53	15.82	13.37
M3 → M4	**13.036**	13.125	15.337	13.228	14.553	14.327	15.047	19.047	14.092

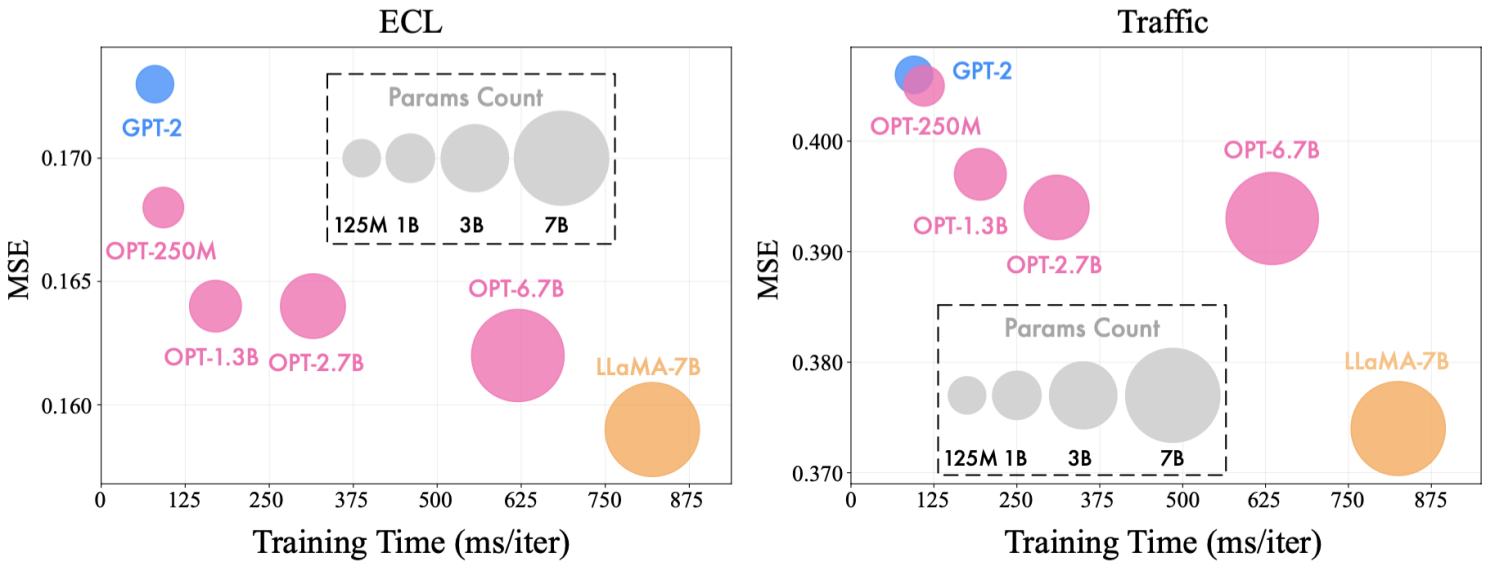
Compatibility of Language Models

AutoTimes configuration

Base LLM	GPT-2 (124M)	OPT-350M	OPT-1.3B	OPT-2.7B	OPT-6.7B	LLaMA-7B
Hidden Dim.	768	1024	2048	2560	4096	4096
Embedding	2-layer MLP	2-layer MLP	2-layer MLP	2-layer MLP	2-layer MLP	Linear
Trainable Param. (M)	0.44	0.58	1.10	1.36	2.15	0.79

Large model tuned with small amount of params

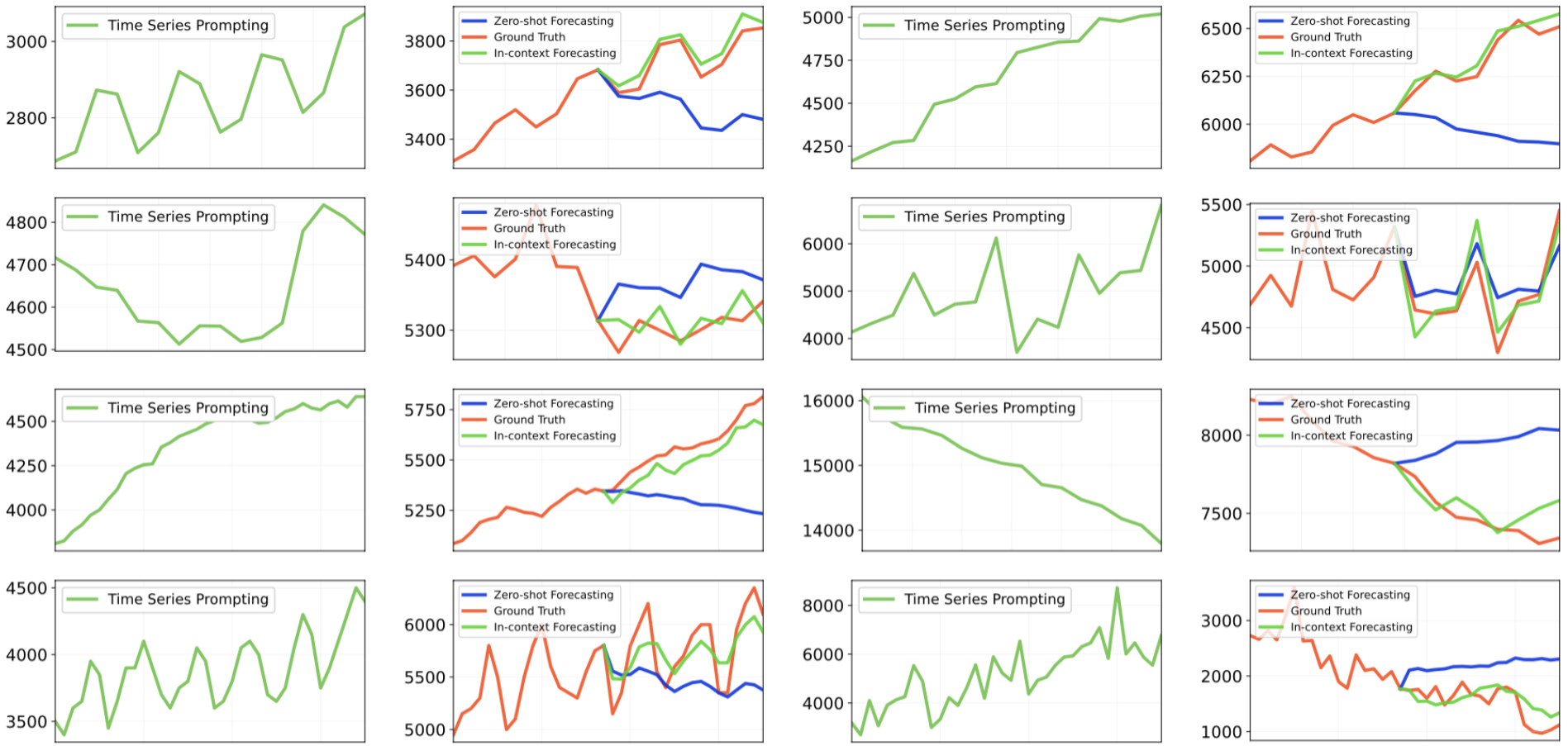
Scaling law of LLM-forecasters



Larger language models, more accurate predictions

In-Context Forecasting Showcases

Facilitate an interactive experience of forecasting via prediction samples



Open Source - AutoTimes

- ✓ **Efficient:** Only 15min to repurpose LLaMA-7B on **one single RTX 3090-24G** (8 x A100 for Time-LLM)
- ✓ **Compatible:** **Support any decoder-only LLMs**: GPT, LLaMA of different sizes, the OPT family…
- ✓ **Well-organized:** Pretty code implementations for multi-step **autoregressive forecasting** and **in-context forecasting**

GitHub: https://github.com/thuml/AutoTimes

README MIT license

AutoTimes (Large Language Models for Time Series Forecasting)

The repo is the official implementation: AutoTimes: Autoregressive Time Series Forecasters via Large Language Models.

> **Time Series Forecasting**: AutoTimes repurpose LLMs as autoregressive multivariate time series forecasters. Different from previous models, our repurposed forecaster can be applied on various lookback/forecast lengths.

> **Zero-Shot Forecasting**: AutoTimes takes advantage of LLM's general-purposed token transition as the future extrapolation of time series, demonstrating good performance without downstream samples.

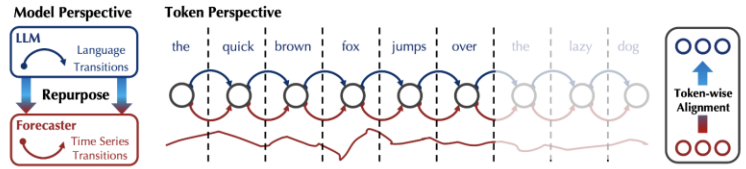
> **In-Context Forecasting**: We propose in-context forecasting for the first time, where time series prompts can further incorporated into the context to enhance forecasting.

> **Easy-to-Use**: AutoTimes is compatiable with any decoder-only large language models, demonstrating generality and proper scaling behavior.

Updates

► News (2024.10): AutoTimes has been accepted by **NeurIPS 2024**. A revised version (**25 Pages**) is now available, including prompt engineering of in-context forecasting, adaptation cost evaluations, textual embeddings of metadata, and low-rank adaptation techique.

► News (2024.08): Recent work (code) has also raised questions about previous non-autoregressive LLM4TS methods. We conduct ablations here, highlighting AutoTimes can truly utilize LLMs. Instead of adopting LLMs in a BERT-style, **the general-purpose token transition is transferable among time series and natural language**.

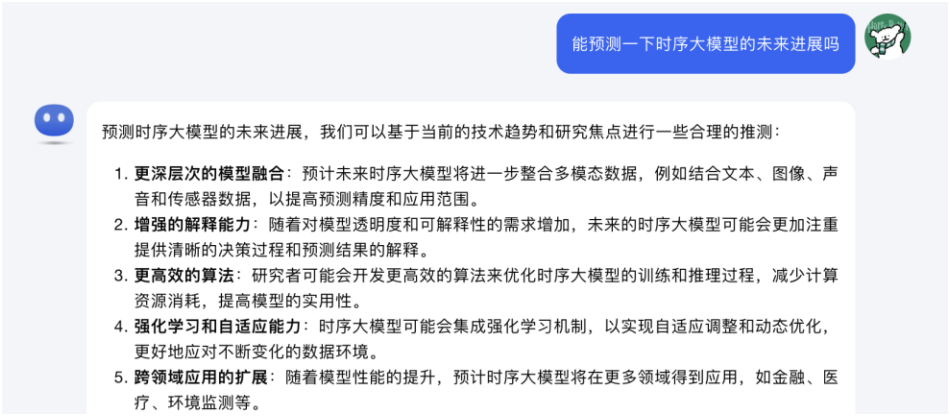


► **News** (2024.2) Scripts for the above tasks in our paper are all available.

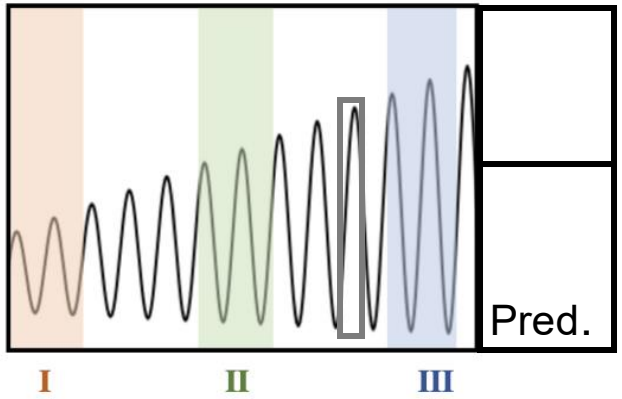
Limitations of LTMs

Deep Models Always Give Global-Optimal Predictions (Conservative)

- They are trained to make **minimal** prediction errors
- The **Scope** of LTMs is different from LLMs and LVMs



Generation V.S. Prediction



the same uptrend

sometimes up

sometimes down

?

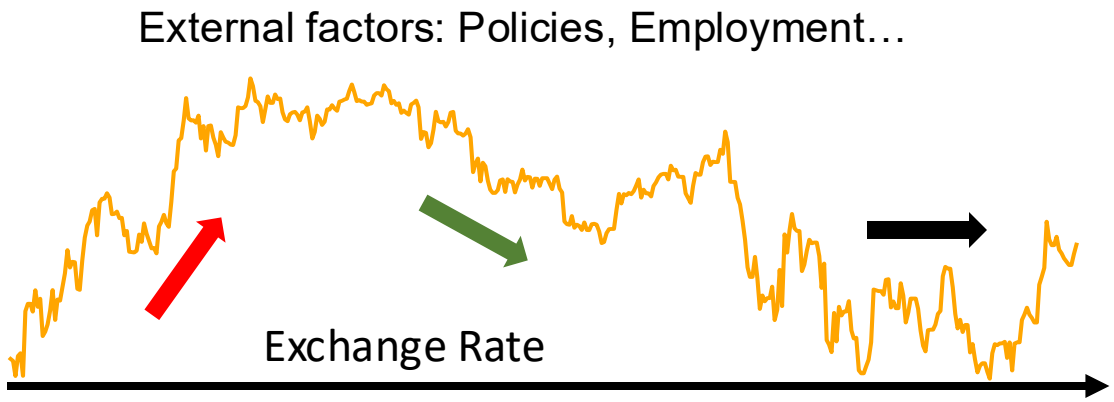
the middle will not be blamed

(statistical-optimal)

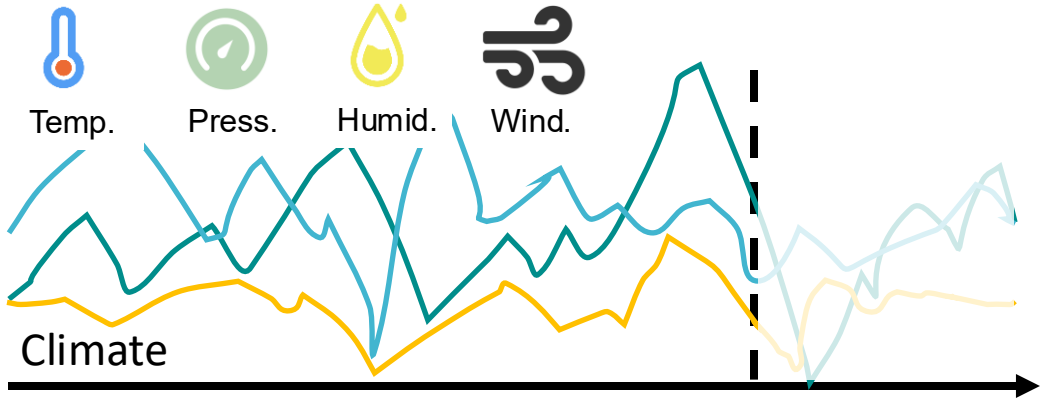
Limitations of LTMs

Learning/Inference on Single Time Series (Non-Multivariate)

- The single-variate formulation makes the training **simple** and **versatile**
- Fail to utilize **expertise knowledge / multivariate correlations**



Most time series are **unpredictable** and **external factors** that make the change are not considered

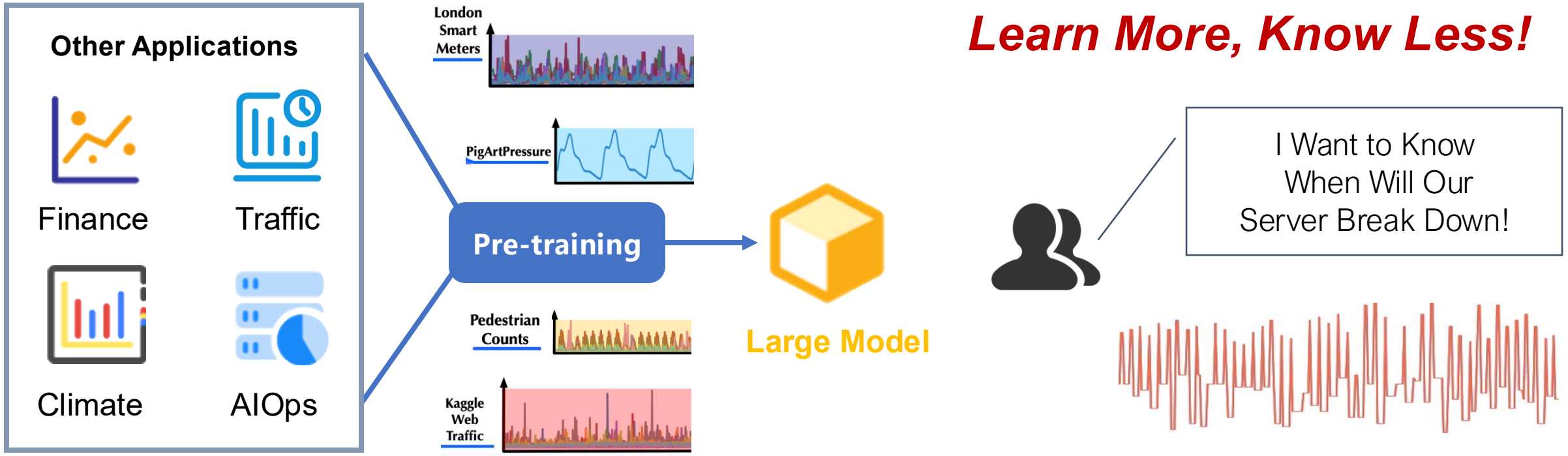


Applications are involved with **domain knowledge** while deep time series models are purely **data-driven**

Limitations of LTMs

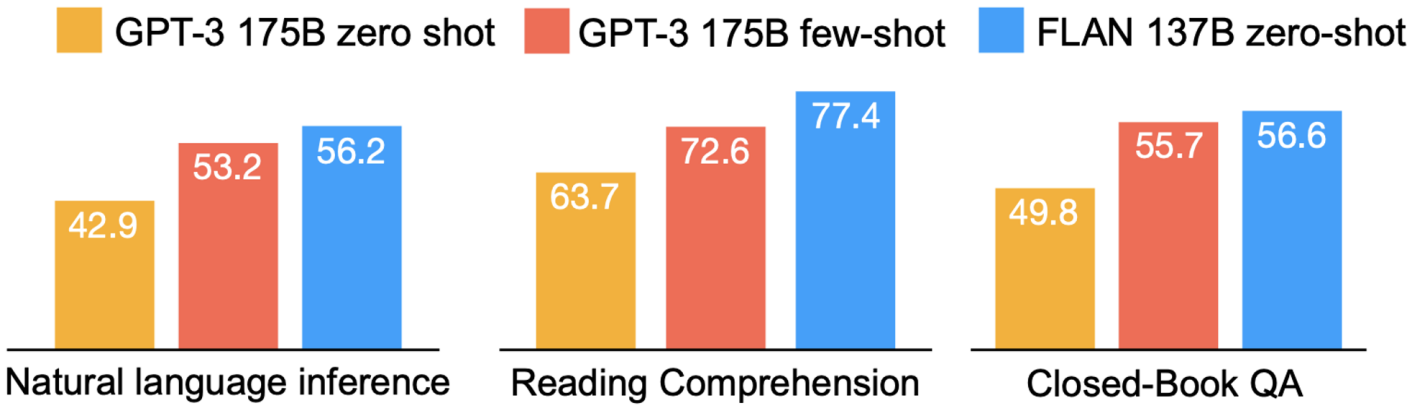
Series Variation is Distinct in Different Applications (Non-transferable)

- Unlike grammar in languages, the common sense of TS remains unclear
- Scaling does not bring continuous/explicit benefits to performance

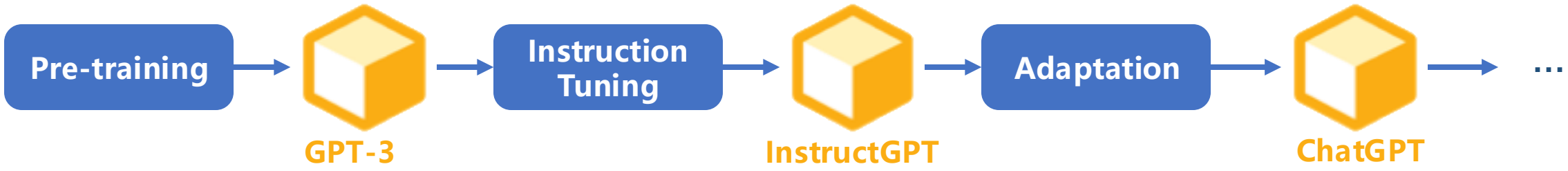


Future Directions

Large Model Does Not Grow in One Step

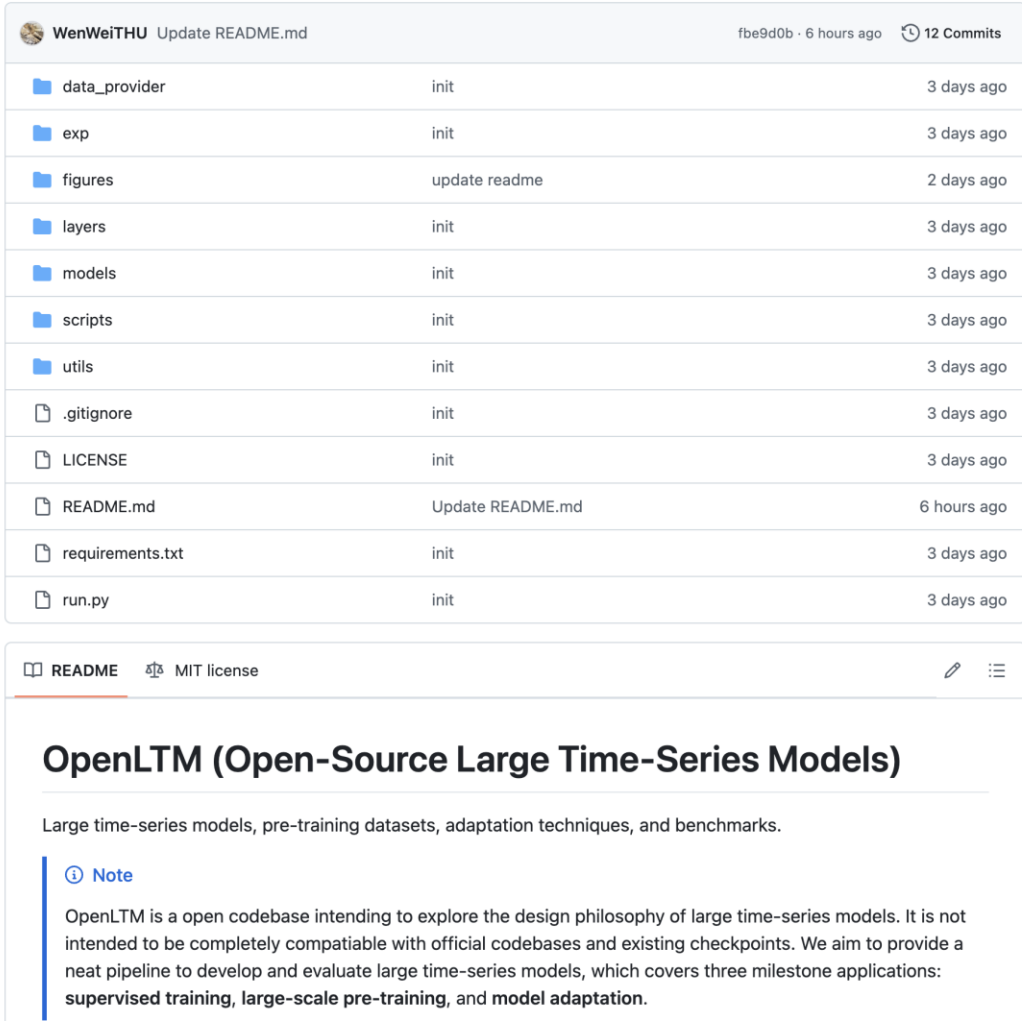


Large Model for Natural Language Was Also in Early Stages (GPT-3, 2020)



OpenLTM: Open-Source Large Time-Series Models

- ✓ **Inclusive:** Integrate mainstream large time-series models and datasets
- ✓ **Ease of Use:** Easy to pre-train and evaluate your large model design
- ✓ **Active:** We are engaging in discussion and welcome to any instructive PRs



GitHub: https://github.com/thuml/OpenLTM

Thank you!

Mingsheng Long
(龙明盛)
Tsinghua University
mingsheng@tsinghua.edu.cn

Jianmin Wang
(王建民)
Tsinghua University
jimwang@tsinghua.edu.cn

吴海旭

刘雍

董家祥

王雨轩

覃果

张渼然

清華大學 Tsinghua University

大数据系统软件国家工程研究中心

清华大学软件学院机器学习课题组