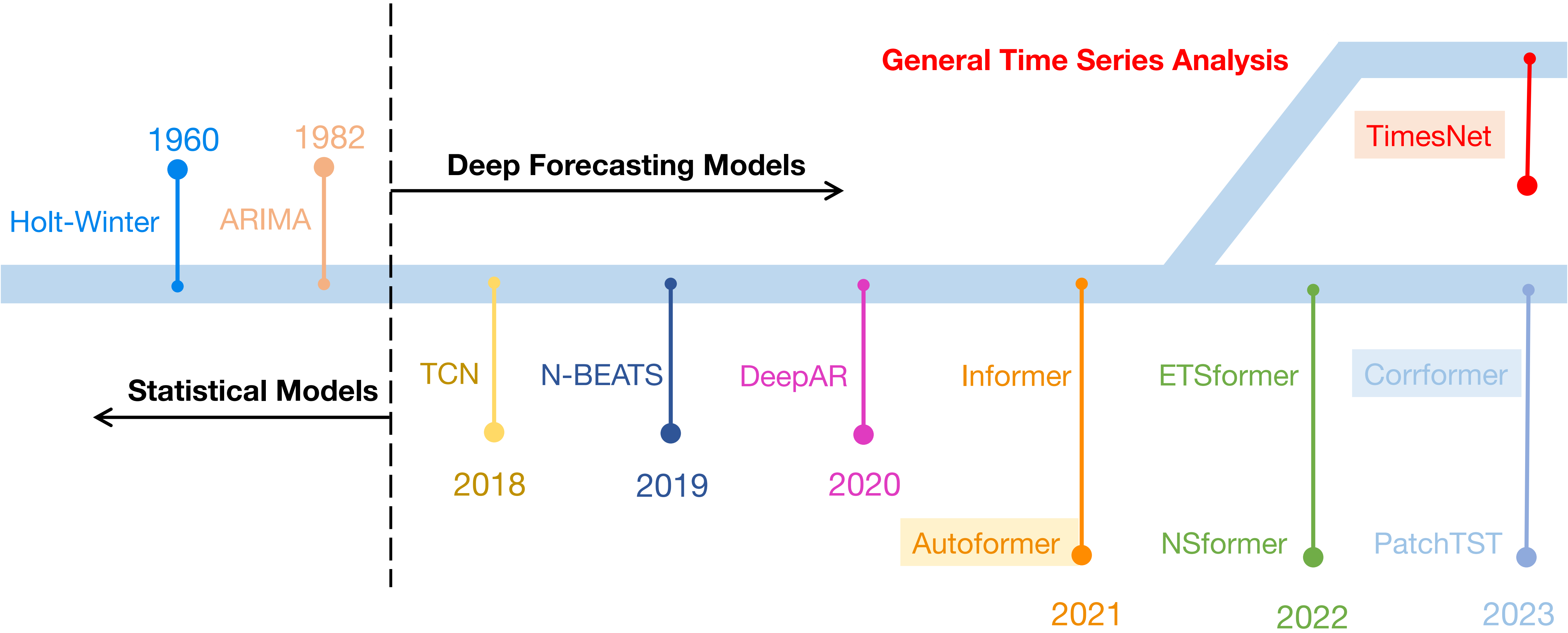
Foundation Models for Time Series Analysis

Mingsheng Long

School of Software, Tsinghua University

June 10, 2023

Timeline


General Time Series Analysis
1960
1982
Holt-Winter
ARIMA
Deep Forecasting Models
TimesNet
Statistical Models
TCN
N-BEATS
DeepAR
Informer
ETSformer
Corrformer
2018
2019
2020
Autoformer
NSformer
PatchTST
2021
2022
2023

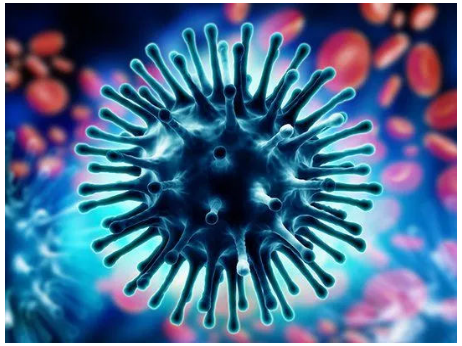
Time Series Forecasting

Energy Consumption

Traffic Flow

Economic Changes

Weather Variations



Disease Propagation

Past Observations

Future Time Series

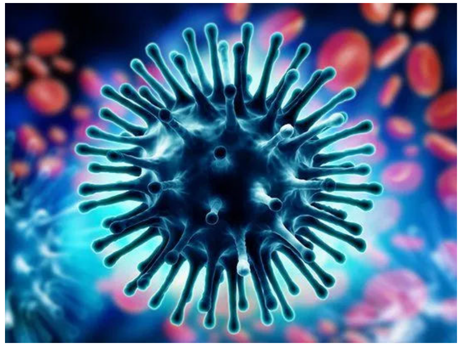
Time Series Forecasting

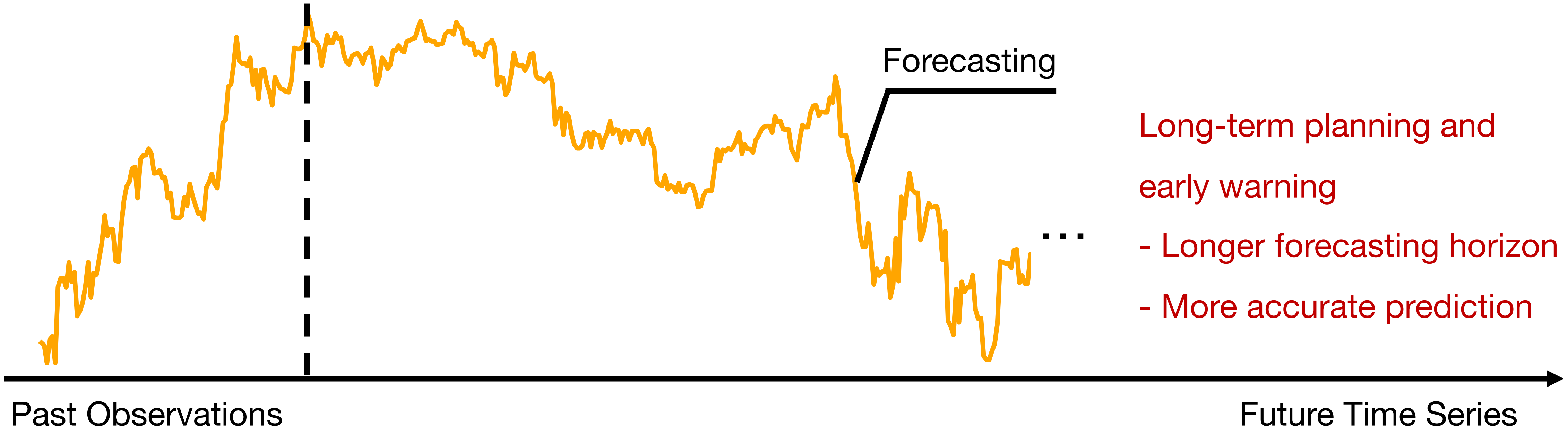

Energy Consumption


Traffic Flow


Economic Changes


Weather Variations


Disease Propagation



Long-term planning and early warning

- Longer forecasting horizon
- More accurate prediction

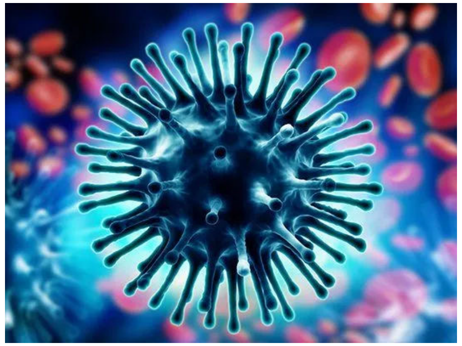
Long-Term Time Series Forecasting

Energy Consumption

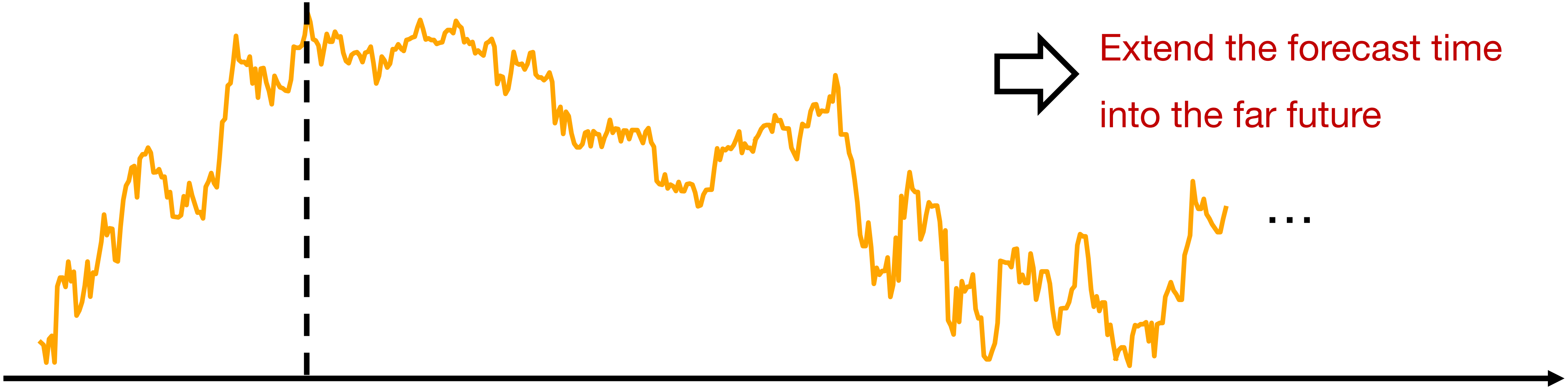
Traffic Flow

Economic Changes

Weather Variations



Disease Propagation



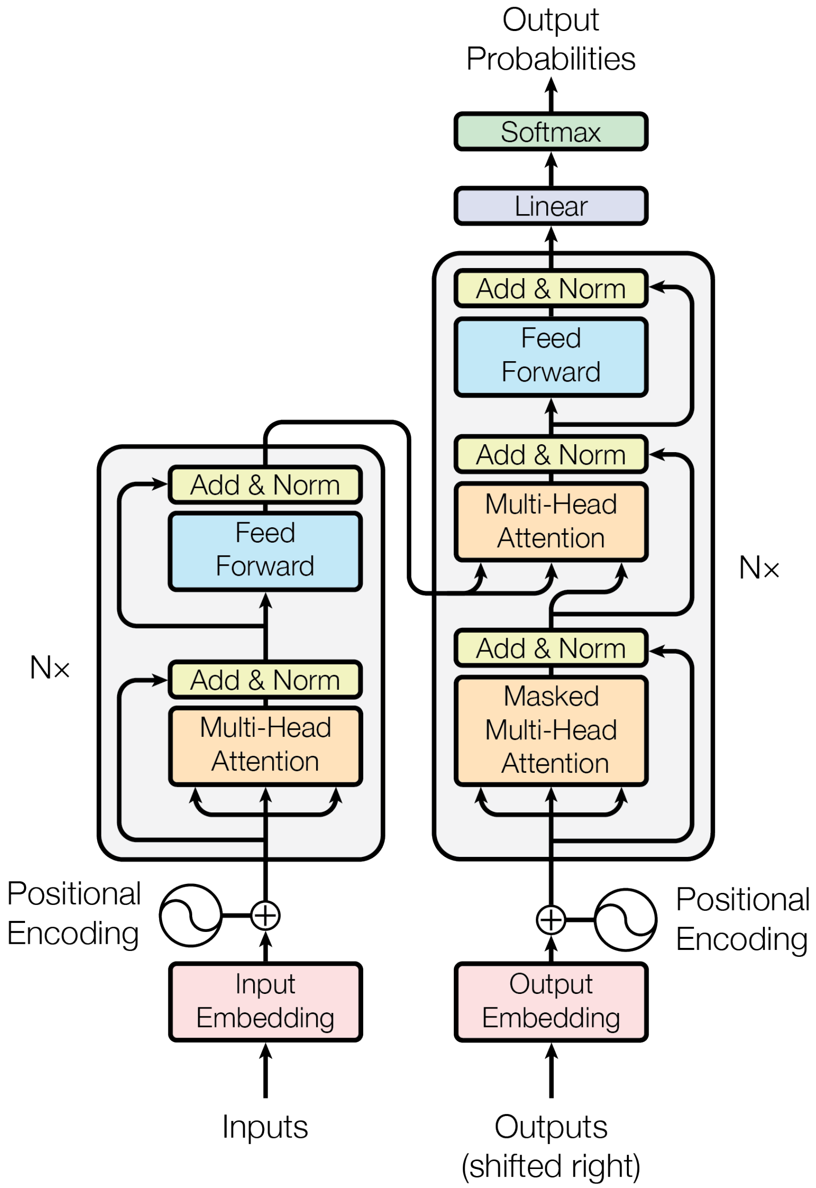
Extend the forecast time into the far future

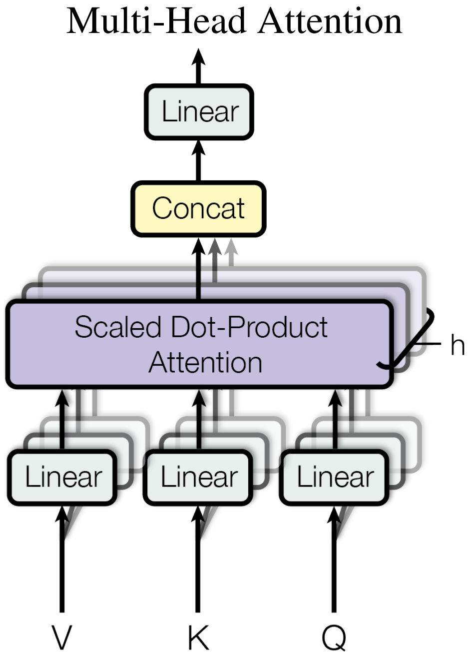
…

Past Observations

Future Time Series

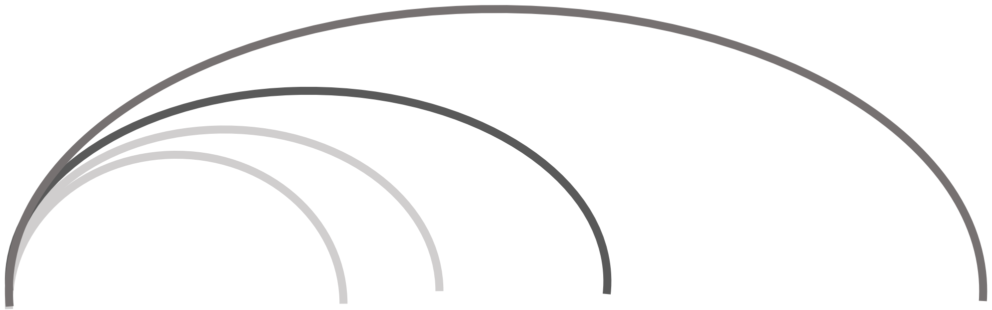
Transformers





Modeling the relation of words

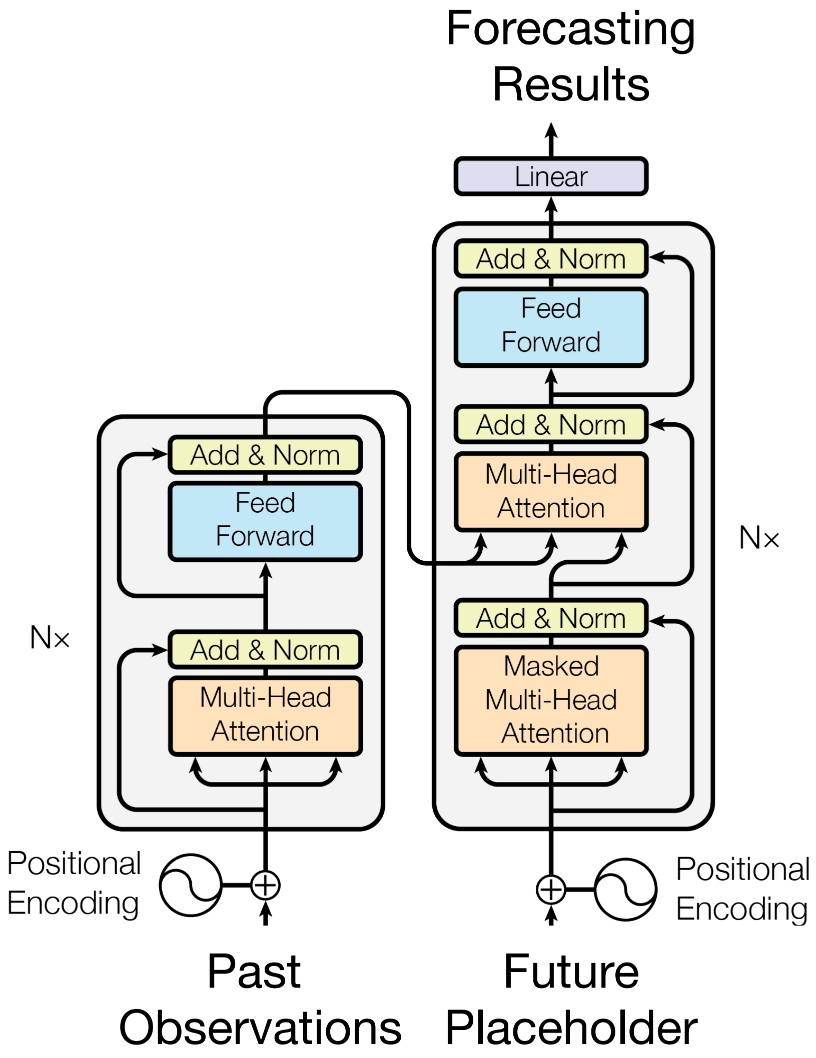
with point-wise Self-Attention



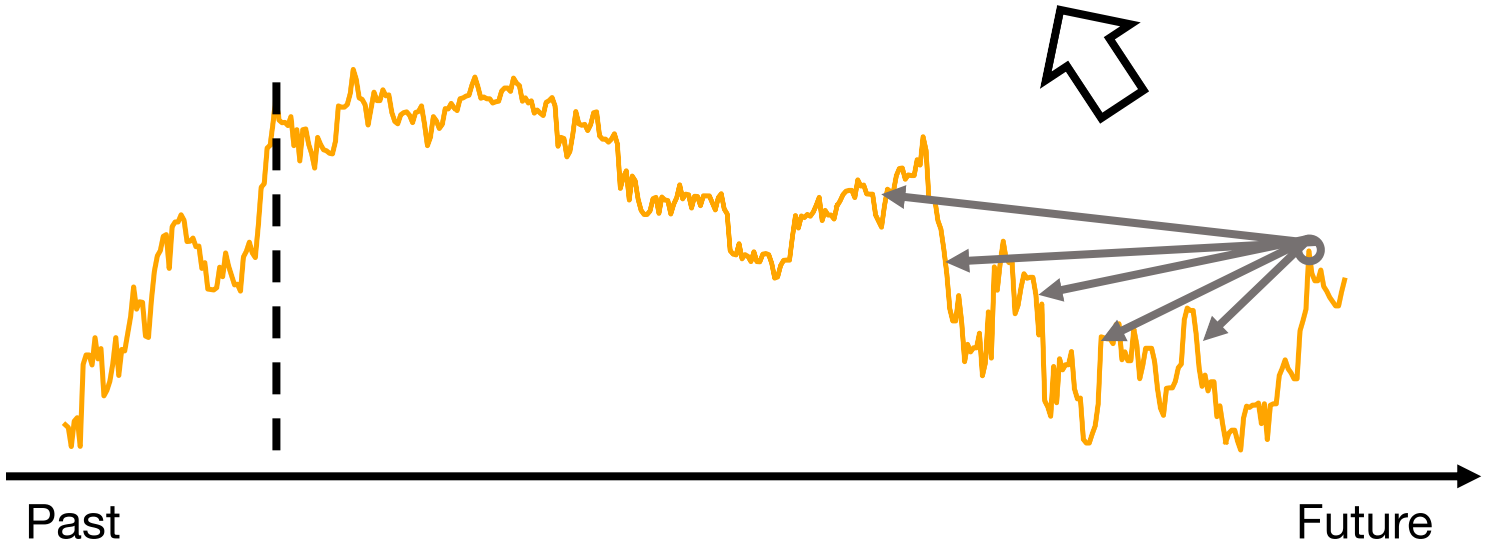
Quadratic complexity for long-term forecasting

Vaswani et al. Attention is All Your Need. *NeurIPS 2017.*

Transformers for Time Series Forecasting



- Modeling the temporal dependencies with point-wise Self-Attention
- Aggregate the representations for forecasting

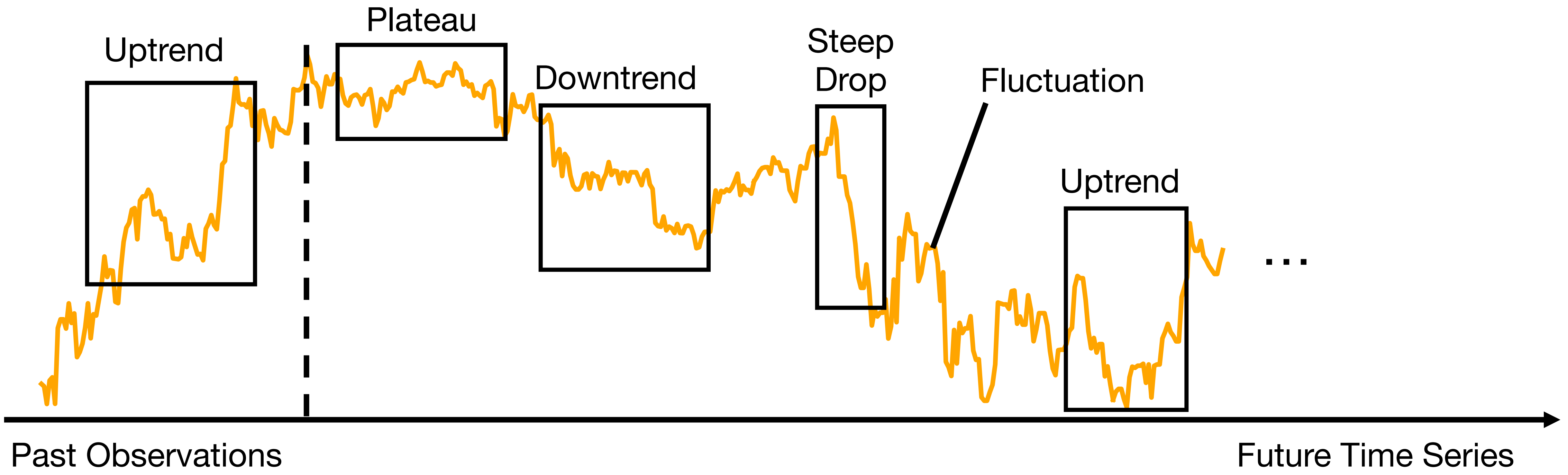


Informer [Zhou et al. AAAI 2021], Reformer [Kitaev et al. ICLR 2020], Log Trans [Li et al. NeurIPS 2019]

Long-Term Time Series Forecasting

Longer forecasting horizon ⇨ Intricate temporal patterns

Deal with long series (complexity)



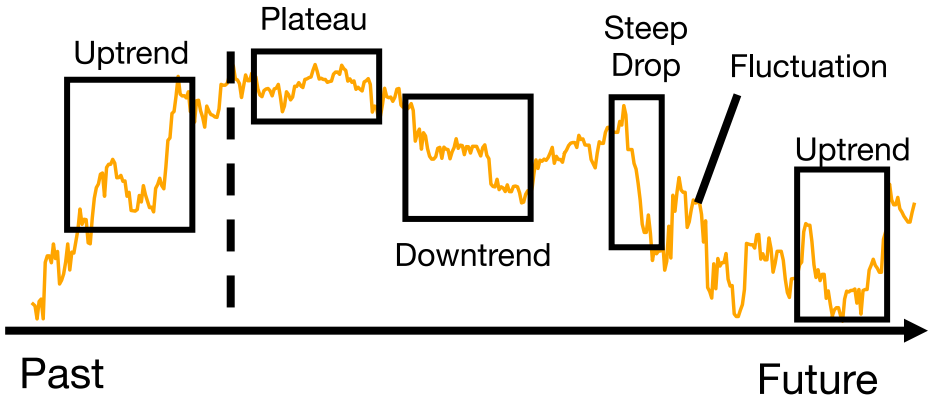
Autoformer

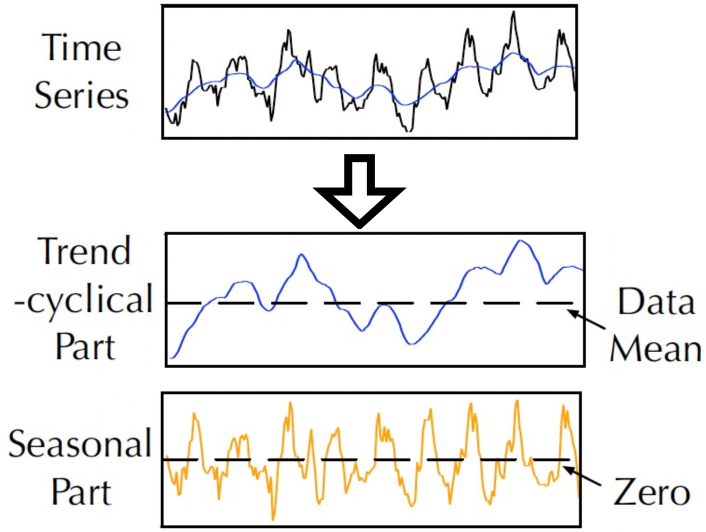
Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long (✉)
School of Software, BNRist, Tsinghua University, China
{whx20,xjh20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

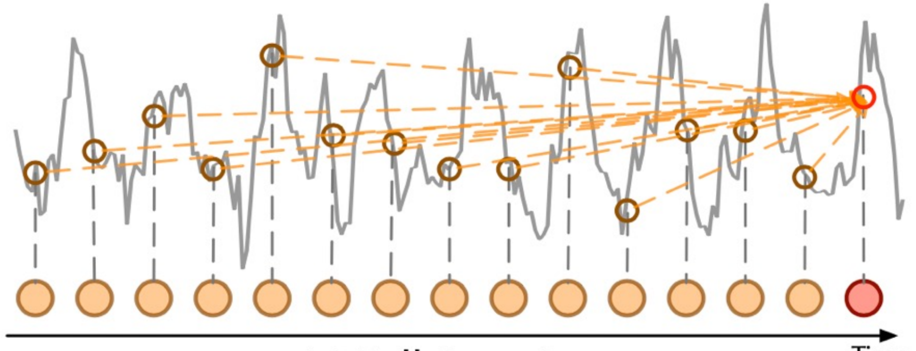
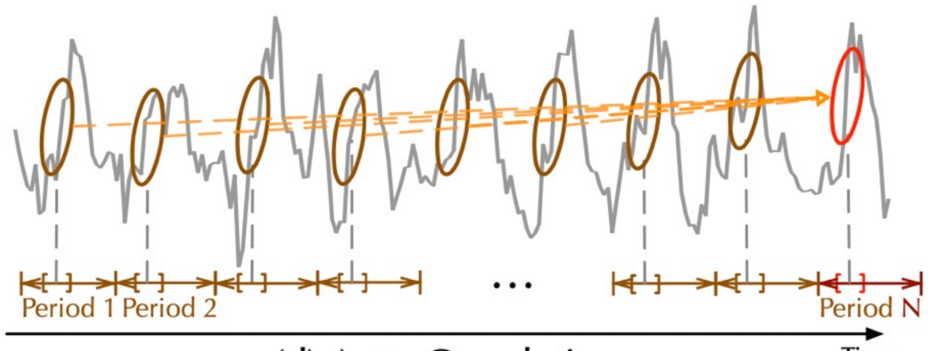
Autoformer vs. Transformer

	Transformer	Autoformer
Intricate Temporal Patterns	Hard to directly find reliable temporal dependencies from raw series ✗	Decomposition architecture to ravel out the entangled temporal patterns ✓

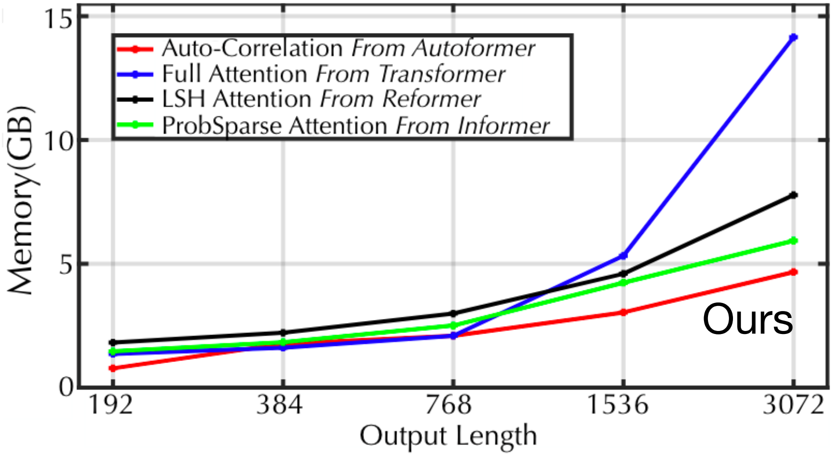
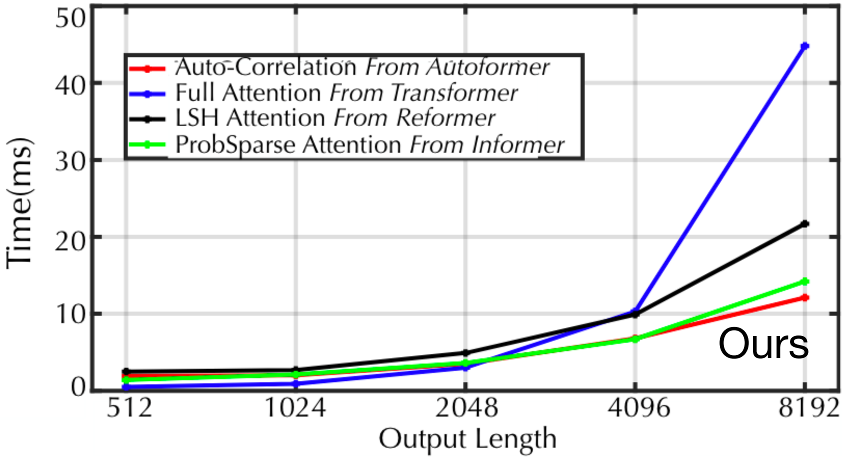




Autoformer vs. Transformer

	Transformer	Autoformer
Time Series Continuity	Self-Attention discover the temporal dependencies from scattered points ✗  (a) Full Attention	Auto-Correlation to discover the Series-wise temporal dependencies ✓  (d) Auto-Correlation

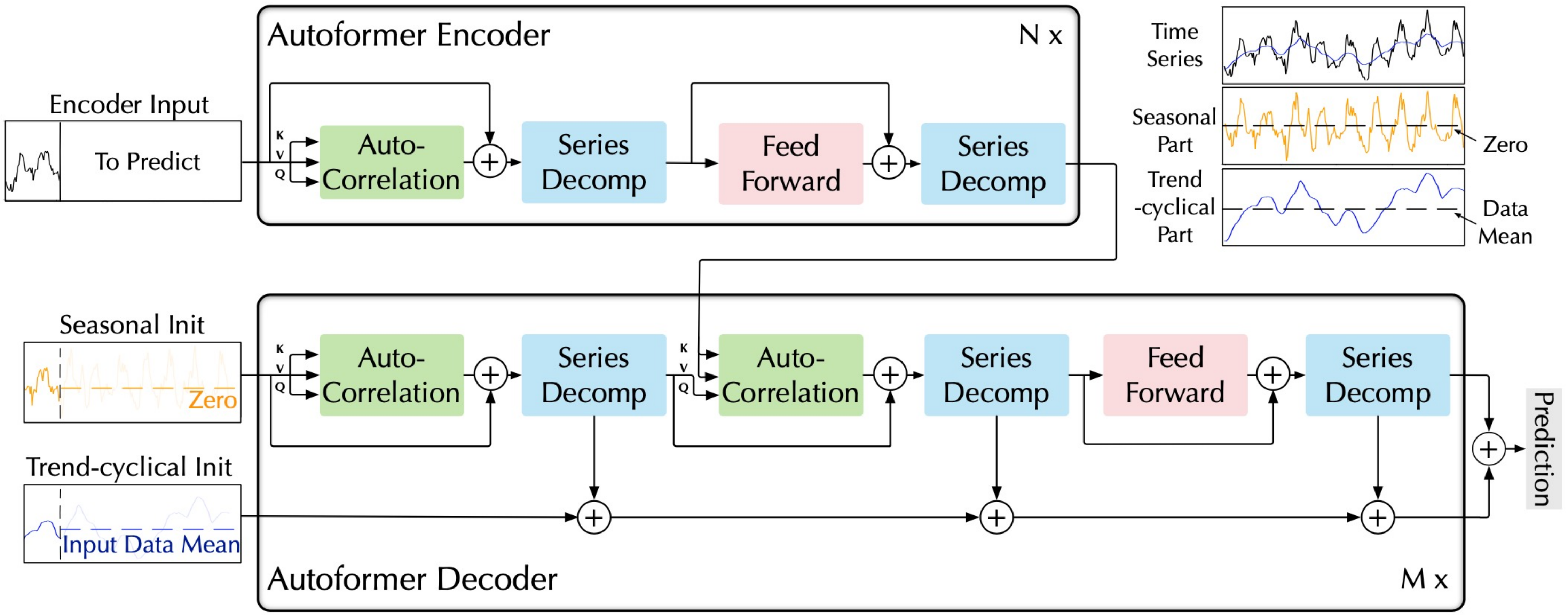
Autoformer vs. Transformer

	Transformer	Autoformer
Computation Efficiency	• Point-wise Self-Attention is $O(L^2)$ • Adopt sparse version for efficiency resulting in the trade-off dilemma ✗	Auto-Correlation mechanism based on stochastic process theory with inherent $O(L \log L)$ complexity ✓
		

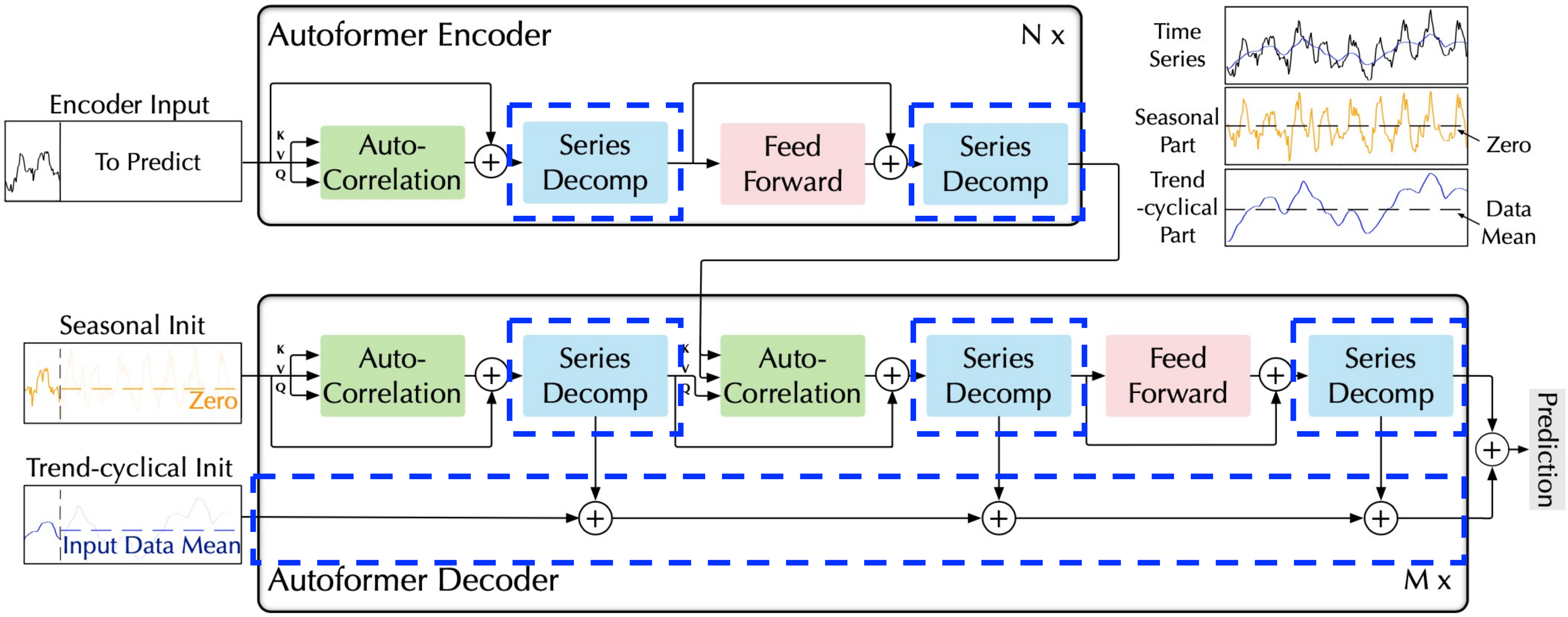
(a) Memory Efficiency Analysis

(b) Running Time Efficiency Analysis

Autoformer Architecture

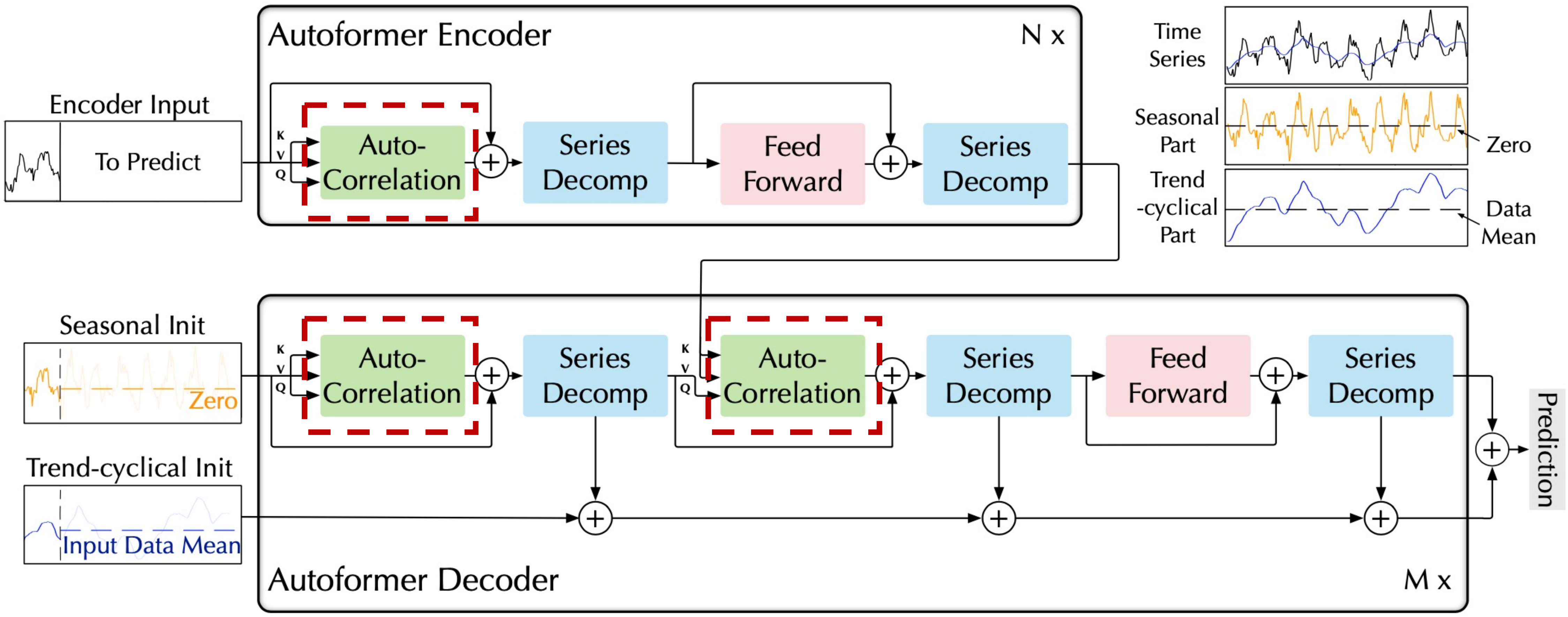


Autoformer Architecture



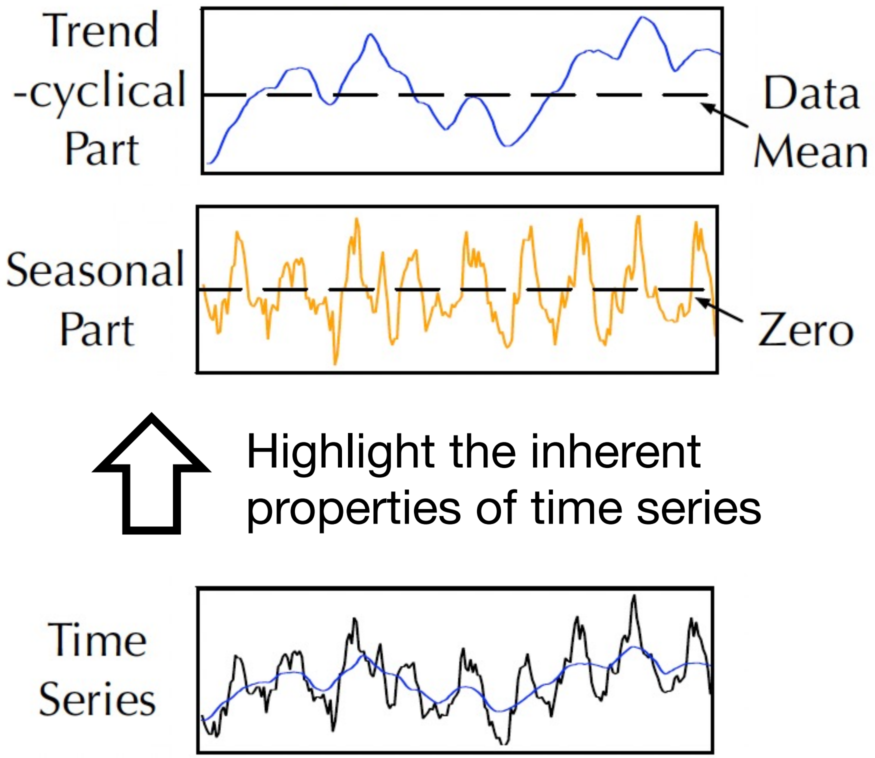
Decomposition architecture for intricate temporal patterns.

Autoformer Architecture



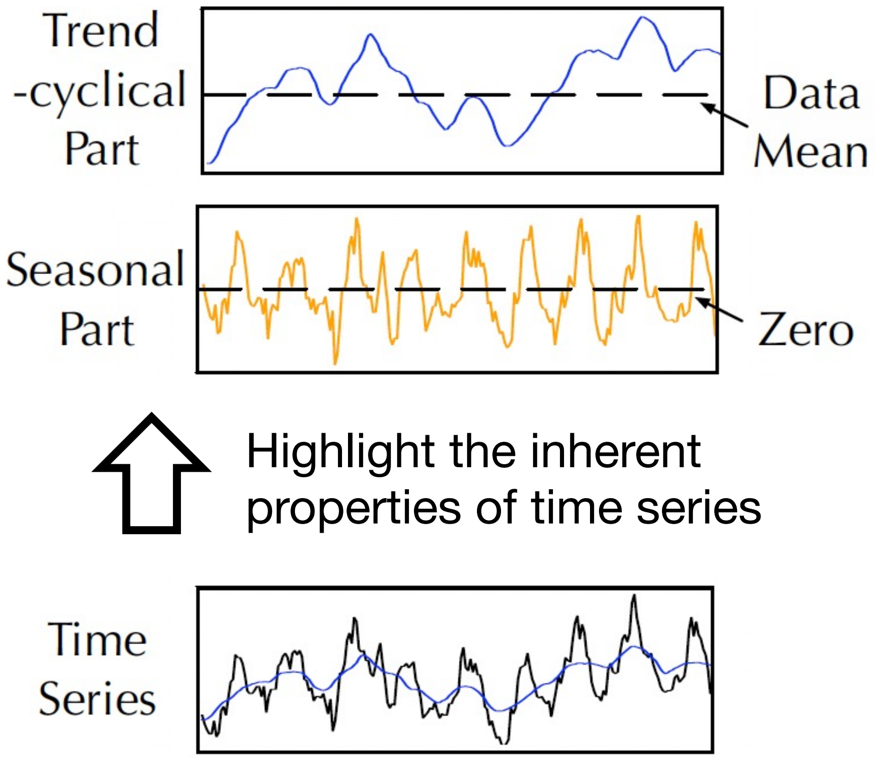
Series-wise Auto-Correlation for information utilization bottleneck.

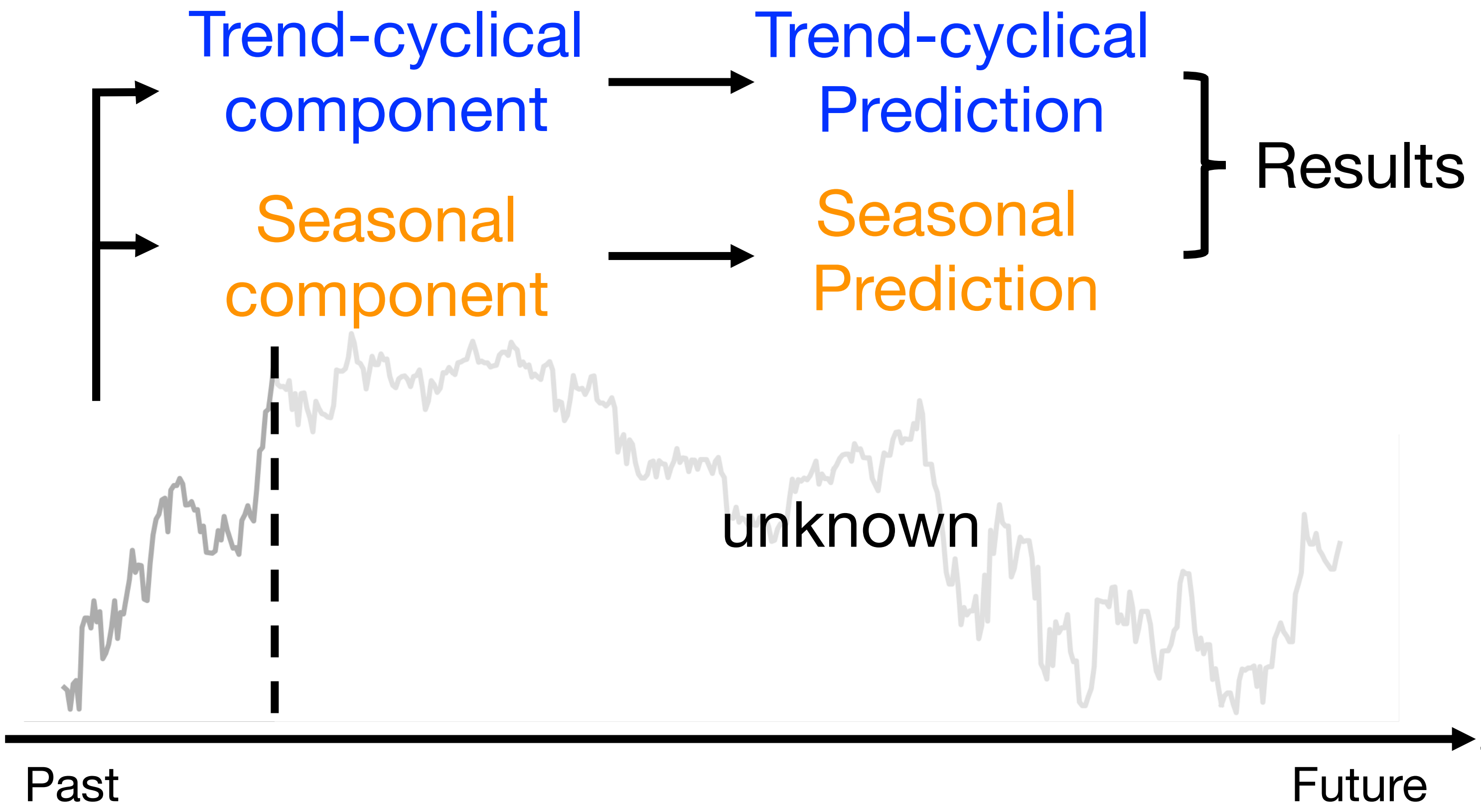
Decomposition



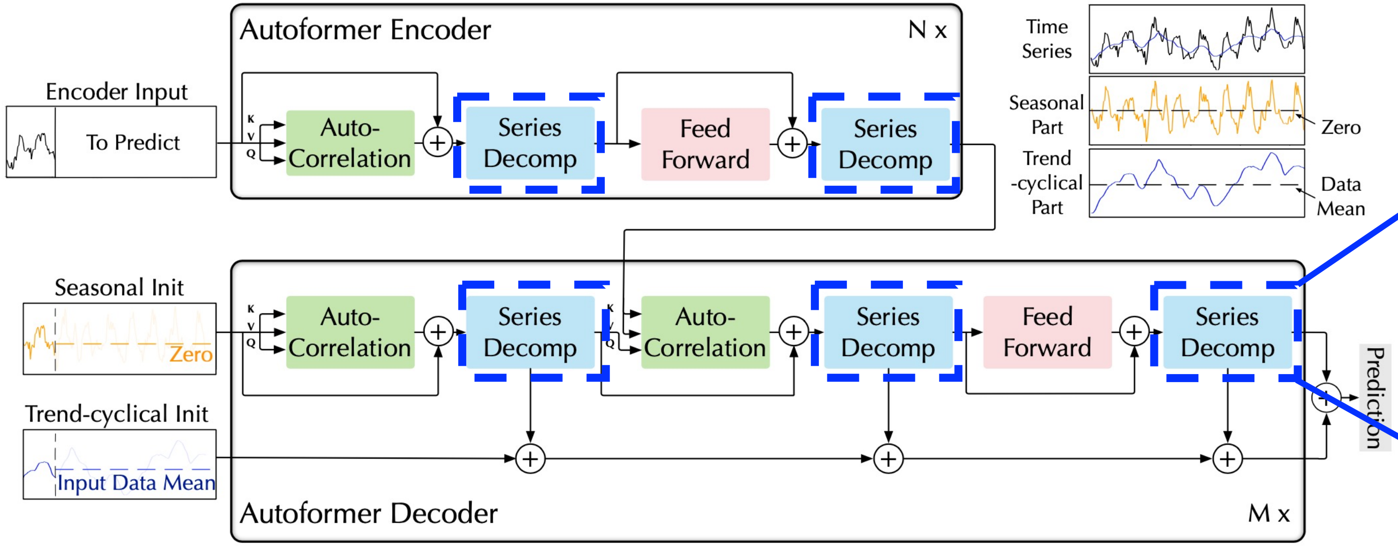
Decomposition as Pre-processing

- Limited by the capabilities of decomposition
- Overlooks the potential future interactions among components





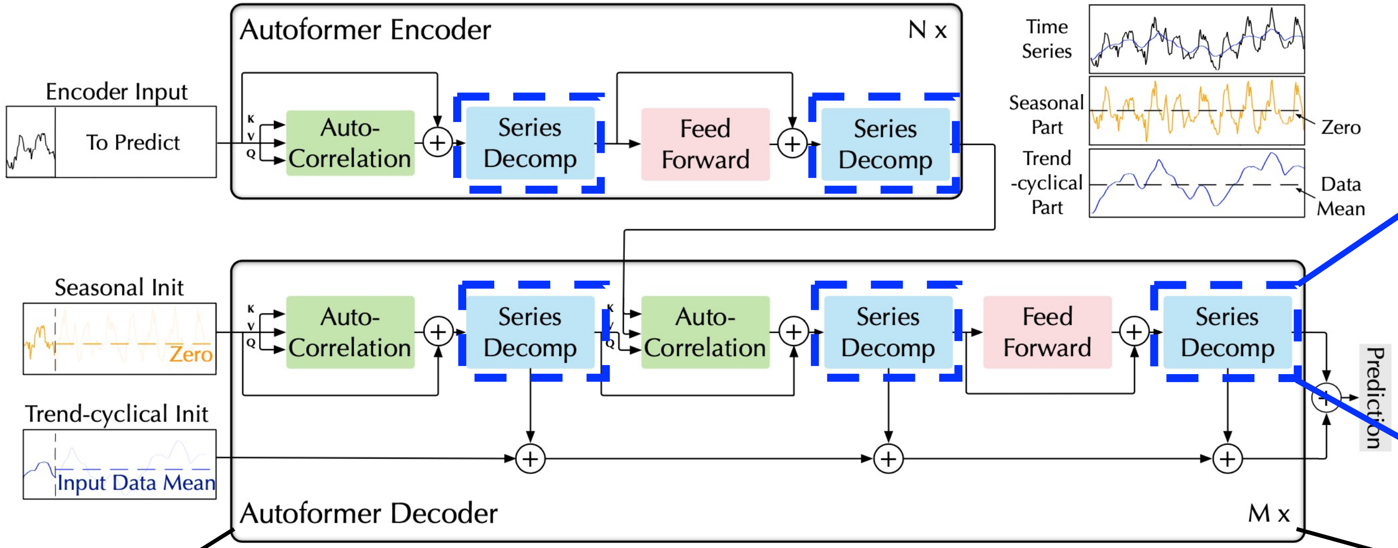
Decomposition in Architecture



Decomposition with Moving Average

$$\mathcal{X}_{\mathrm{t}} = \mathrm{AvgPool}(\mathrm{Padding}(\mathcal{X}))$$
$$\mathcal{X}_{\mathrm{s}} = \mathcal{X} - \mathcal{X}_{\mathrm{t}},$$

Decomposition in Architecture



Decomposition with Moving Average

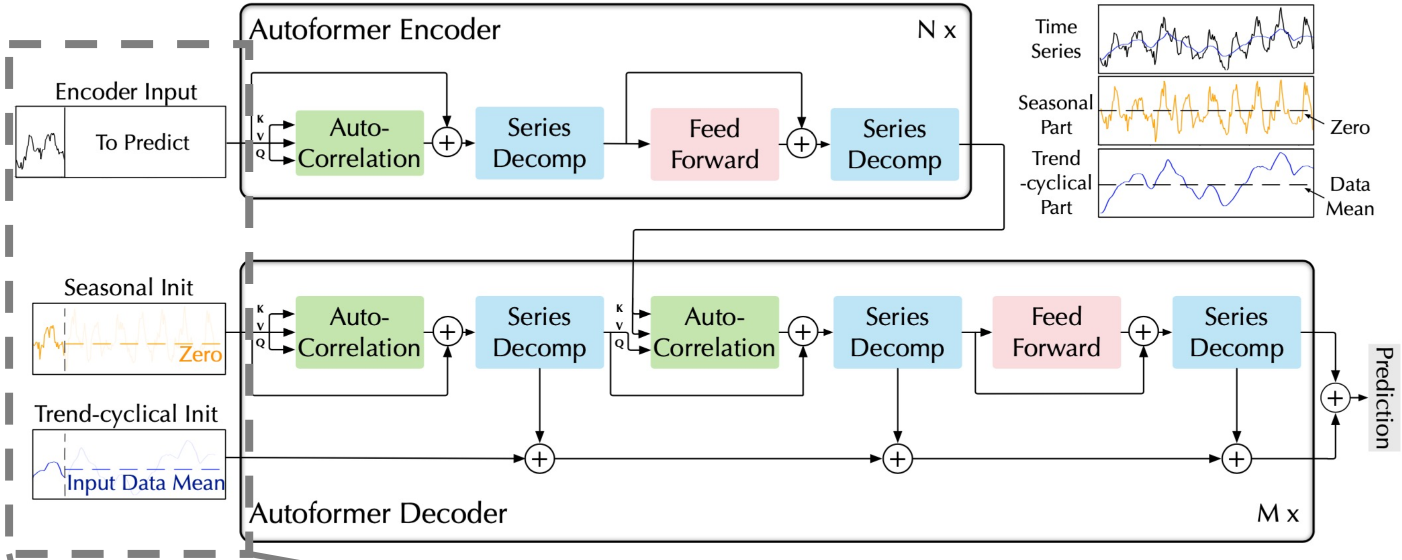
$$\mathcal{X}_{\mathrm{t}} = \mathrm{AvgPool}(\mathrm{Padding}(\mathcal{X}))$$
$$\mathcal{X}_{\mathrm{s}} = \mathcal{X} - \mathcal{X}_{\mathrm{t}},$$

→ Predict → Decompose → Refine Prediction → Decompose → Refine Prediction →

Progressive decomposition capacity

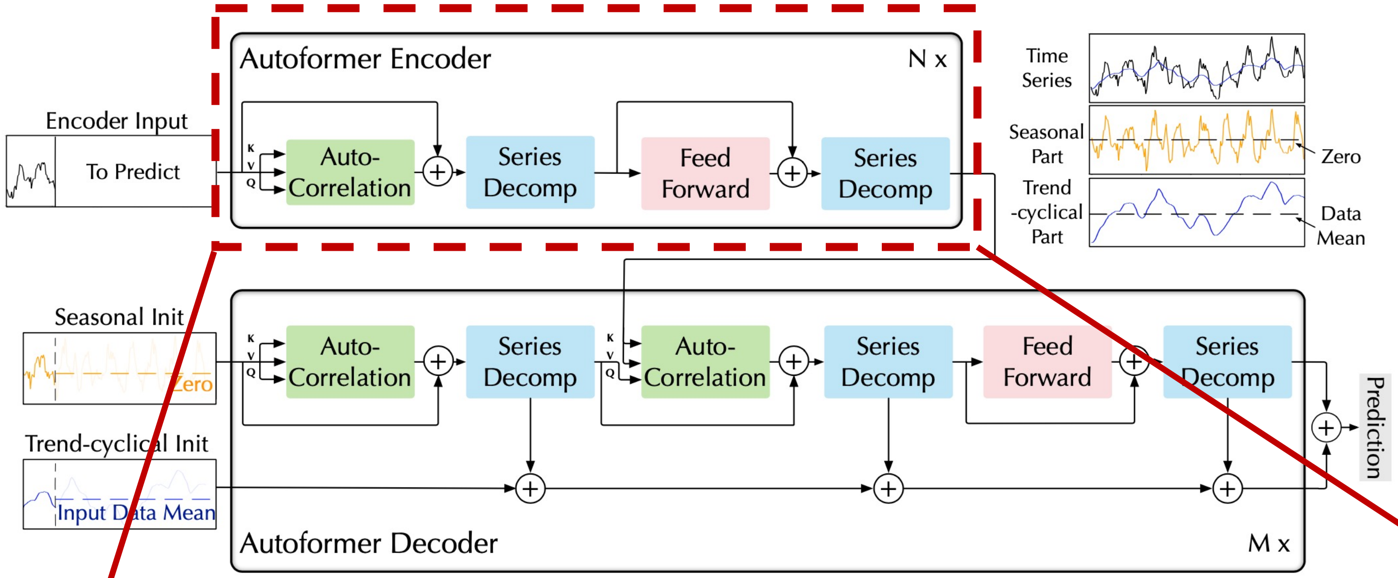
Decompose the trend from the intermediate "future" and refine it in the decoder.

Decomposition in Architecture: Input



$$\mathcal{X}_{\mathrm{ens}}, \mathcal{X}_{\mathrm{ent}} = \mathrm{SeriesDecomp}(\mathcal{X}_{\mathrm{en}\,\frac{I}{2}:I})$$
$$\mathcal{X}_{\mathrm{des}} = \mathrm{Concat}(\mathcal{X}_{\mathrm{ens}}, \mathcal{X}_0)$$
$$\mathcal{X}_{\mathrm{det}} = \mathrm{Concat}(\mathcal{X}_{\mathrm{ent}}, \mathcal{X}_{\mathrm{Mean}}),$$

Decomposition in Architecture: Encoder



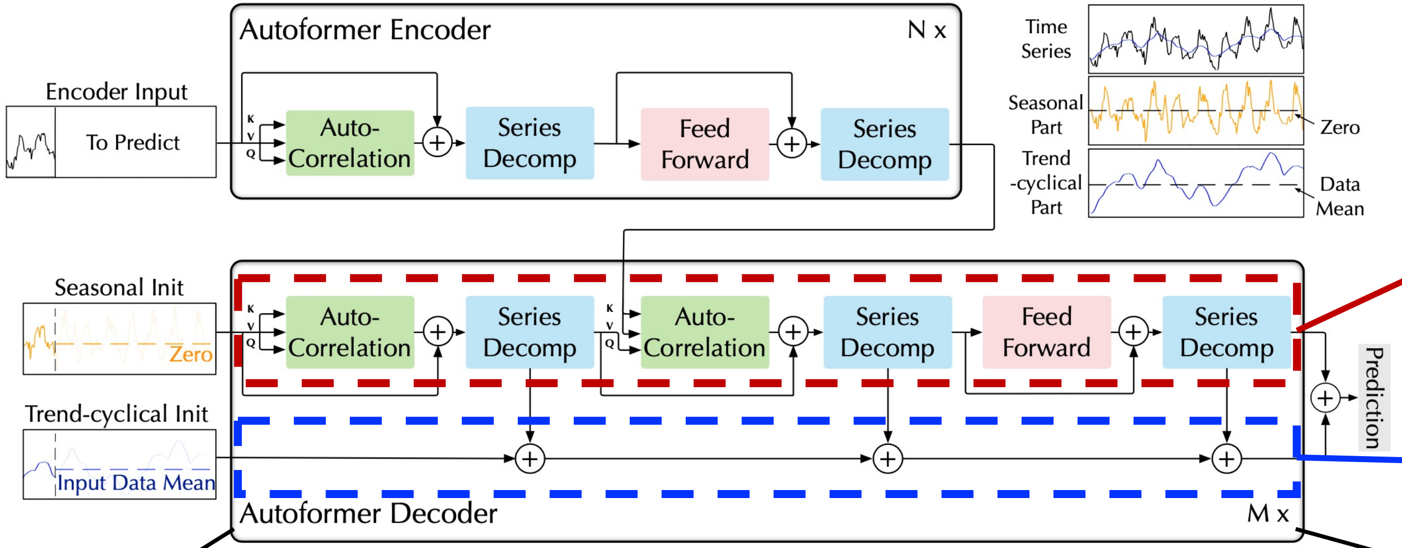
Focus on seasonal part modeling,

Provide cross information for decoder

$$\mathcal{S}_{\mathrm{en}}^{l,1}, \_ = \mathrm{SeriesDecomp}\Big(\mathrm{AutoCorrelation}(\mathcal{X}_{\mathrm{en}}^{l-1}) + \mathcal{X}_{\mathrm{en}}^{l-1}\Big)$$

$$\mathcal{S}_{\mathrm{en}}^{l,2}, \_ = \mathrm{SeriesDecomp}\Big(\mathrm{FeedForward}(\mathcal{S}_{\mathrm{en}}^{l,1}) + \mathcal{S}_{\mathrm{en}}^{l,1}\Big),$$

Decomposition in Architecture: Decoder



Adopt the Auto-Correlation for the seasonal prediction

Accumulation for the trend-cyclical prediction

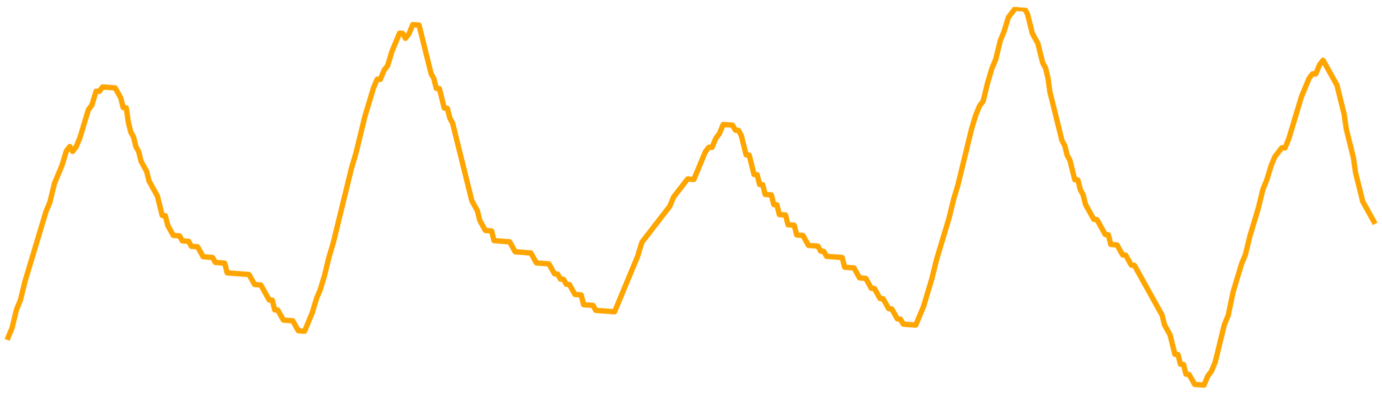
$$\mathcal{S}_{\mathrm{de}}^{l,1}, \mathcal{T}_{\mathrm{de}}^{l,1} = \mathrm{SeriesDecomp}\Big(\mathrm{AutoCorrelation}(\mathcal{X}_{\mathrm{de}}^{l-1}) + \mathcal{X}_{\mathrm{de}}^{l-1}\Big)$$

$$\mathcal{S}_{\mathrm{de}}^{l,2}, \mathcal{T}_{\mathrm{de}}^{l,2} = \mathrm{SeriesDecomp}\Big(\mathrm{AutoCorrelation}(\mathcal{S}_{\mathrm{de}}^{l,1}, \mathcal{X}_{\mathrm{en}}^{N}) + \mathcal{S}_{\mathrm{de}}^{l,1}\Big)$$

$$\mathcal{S}_{\mathrm{de}}^{l,3}, \mathcal{T}_{\mathrm{de}}^{l,3} = \mathrm{SeriesDecomp}\Big(\mathrm{FeedForward}(\mathcal{S}_{\mathrm{de}}^{l,2}) + \mathcal{S}_{\mathrm{de}}^{l,2}\Big)$$

$$\mathcal{T}_{\mathrm{de}}^{l} = \mathcal{T}_{\mathrm{de}}^{l-1} + \mathcal{W}_{l,1} * \mathcal{T}_{\mathrm{de}}^{l,1} + \mathcal{W}_{l,2} * \mathcal{T}_{\mathrm{de}}^{l,2} + \mathcal{W}_{l,3} * \mathcal{T}_{\mathrm{de}}^{l,3},$$

Auto-Correlation Mechanism



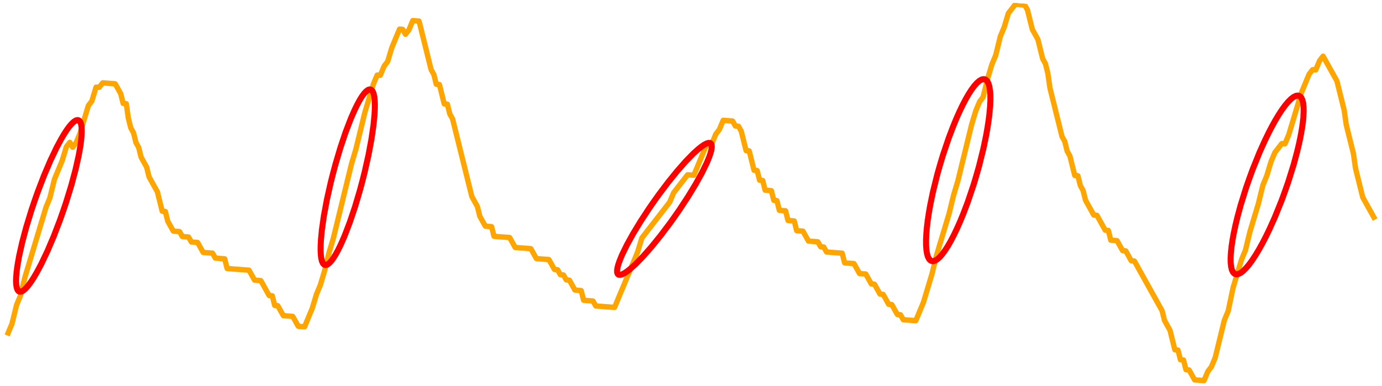
Benefited from the deep decomposition,
the seasonal part is highlighted with periodicity.

Conduct the dependencies discovery and representation aggregation at the series level.

Auto-Correlation Mechanism

Period-based dependencies

The same phase position of different periods



Benefited from the deep decomposition,
the seasonal part is highlighted with periodicity.

Conduct the dependencies discovery and representation aggregation at the series level.

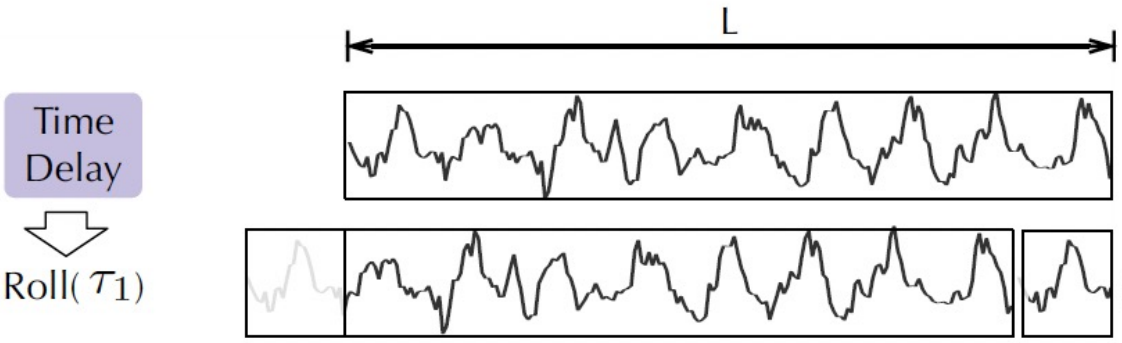
Auto-Correlation Mechanism

Discover period-based dependencies with autocorrelation in stochastic process:

$$\mathcal{R}_{\mathcal{X}\mathcal{X}}(\tau) = \lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} \mathcal{X}_t \mathcal{X}_{t-\tau}.$$

Autocorrelation reflects the time delay similarity,

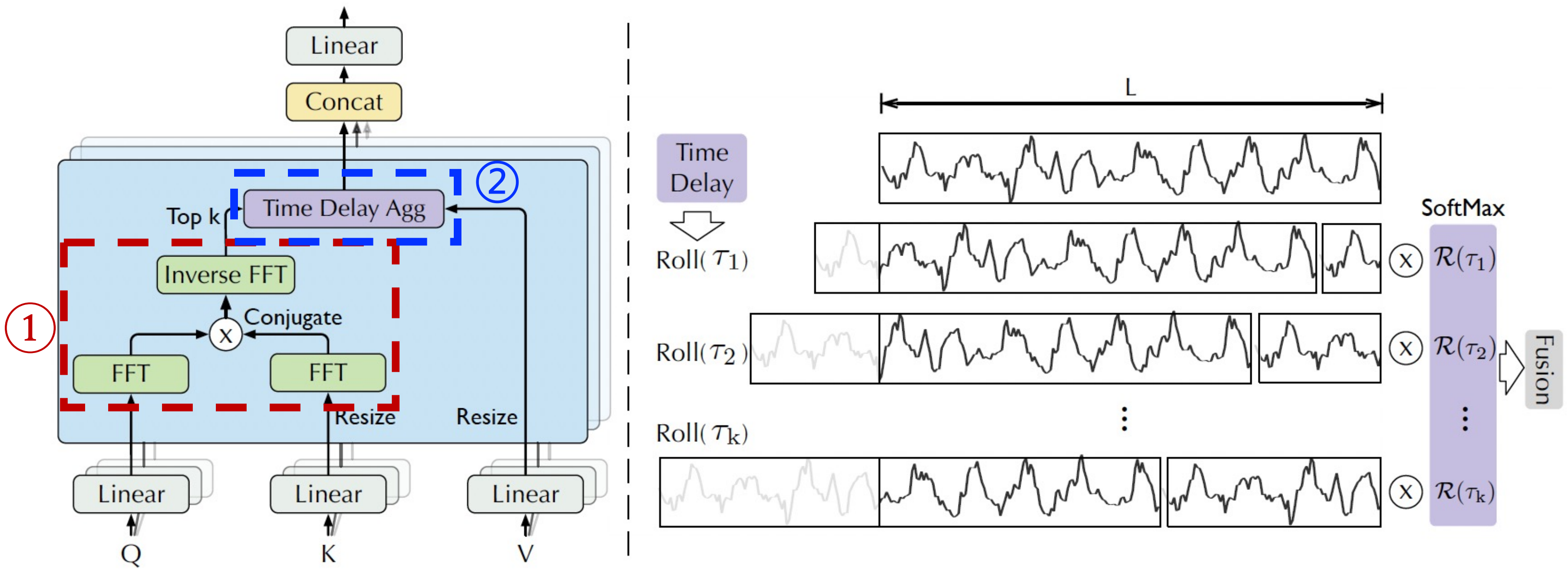
and corresponds to the confidence of period estimation.

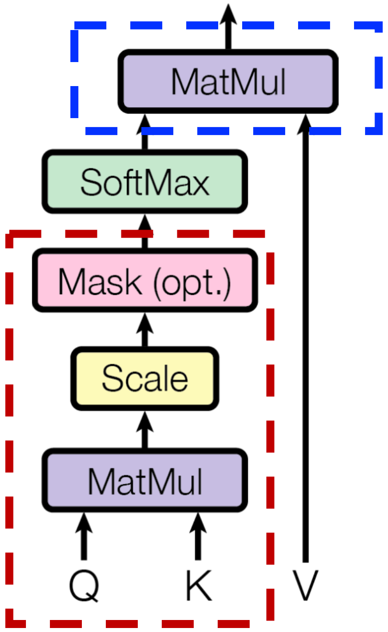


Larger autocorrelation $\mathcal{R}(\tau)$ means

- stronger time delay similarity w.r.t. τ
- more confidence of period length as τ

Auto-Correlation Mechanism

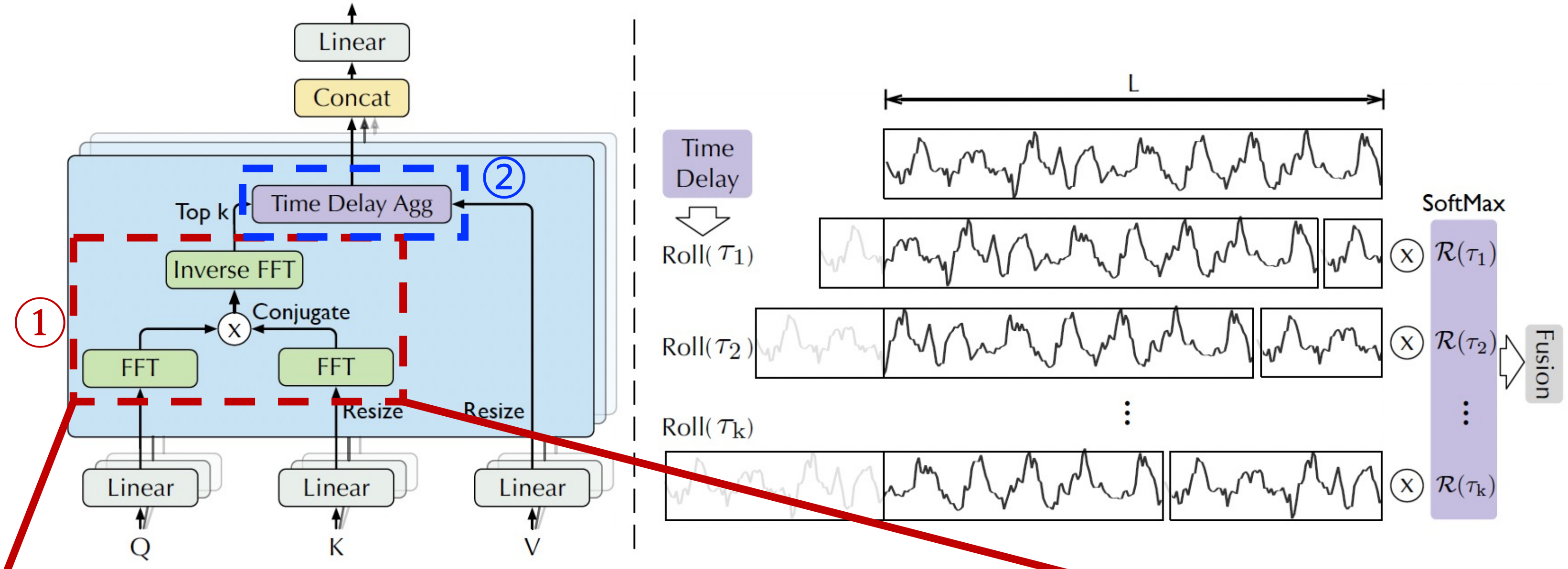




① Discover period-based dependencies

② Aggregate similar sub-processes from different periods

Auto-Correlation Mechanism



Efficient computation of autocorrelation

with Wiener–Khinchin theorem by FFT

$$\mathcal{S}_{\mathcal{X}\mathcal{X}}(f) = \mathcal{F}\left(\mathcal{X}_t\right)\mathcal{F}^*\left(\mathcal{X}_t\right) = \int_{-\infty}^{\infty} \mathcal{X}_t e^{-i2\pi tf} \mathrm{d}t \overline{\int_{-\infty}^{\infty} \mathcal{X}_t e^{-i2\pi tf} \mathrm{d}t}$$

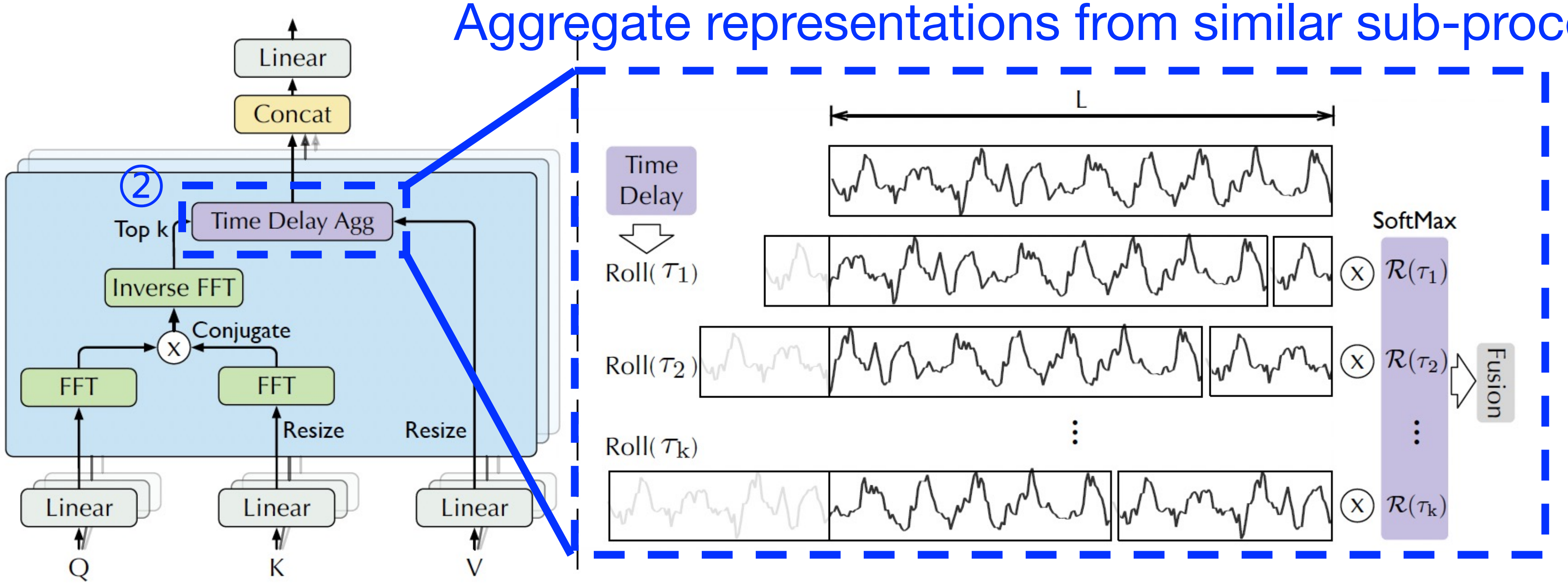
$$\mathcal{R}_{\mathcal{X}\mathcal{X}}(\tau) = \mathcal{F}^{-1}\left(\mathcal{S}_{\mathcal{X}\mathcal{X}}(f)\right) = \int_{-\infty}^{\infty} \mathcal{S}_{\mathcal{X}\mathcal{X}}(f) e^{i2\pi f\tau} \mathrm{d}f,$$

Discover period-based dependencies

with inherent $O(L \log L)$ complexity

Auto-Correlation Mechanism

Aggregate representations from similar sub-processes



$$\tau_1, \cdots, \tau_k = \underset{\tau \in \{1, \cdots, L\}}{\arg \operatorname{Topk}} \left(\mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau)\right)$$

Select the Top k period lengths

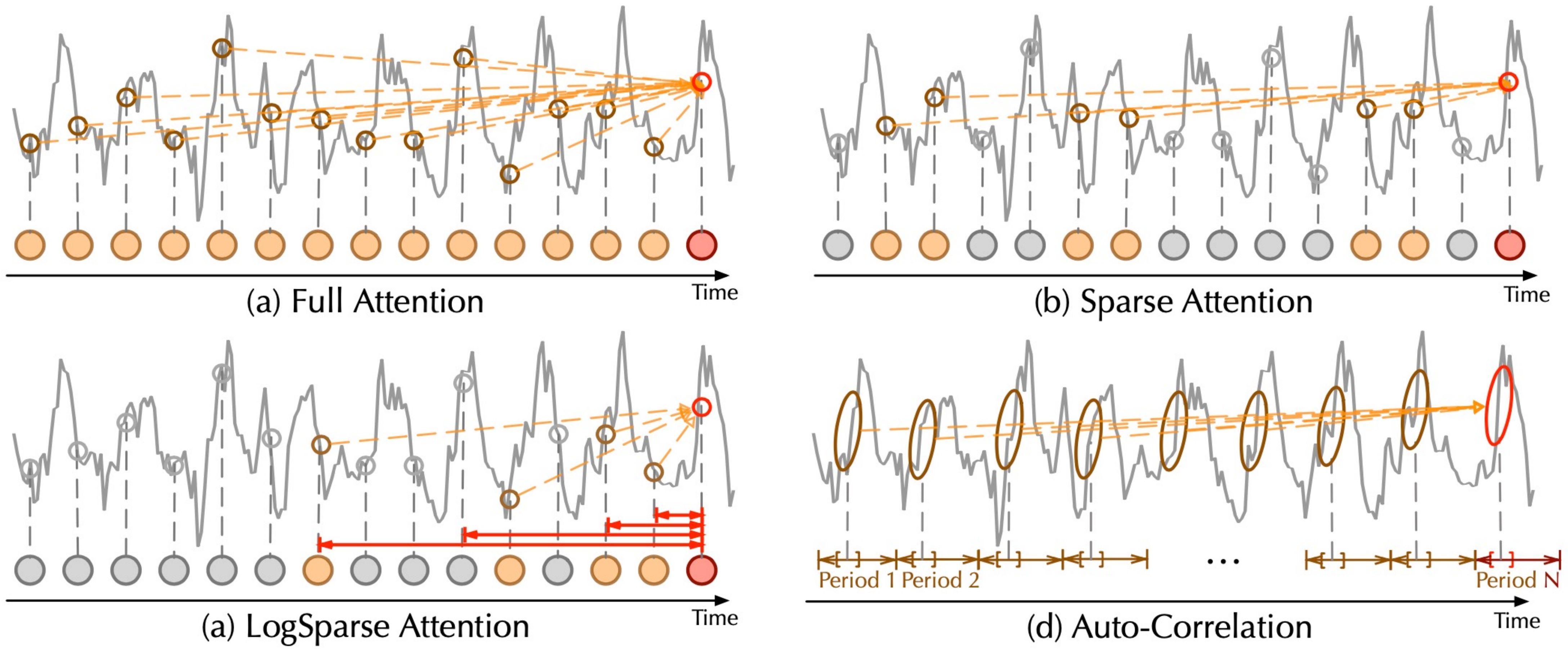
$$\widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_1), \cdots, \widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_k) = \operatorname{SoftMax}\left(\mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau_1), \cdots, \mathcal{R}_{\mathcal{Q},\mathcal{K}}(\tau_k)\right)$$

Normalization

$$\operatorname{AutoCorrelation}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \boxed{\sum_{i=1}^{k} \operatorname{Roll}(\mathcal{V}, \tau_k) \widehat{\mathcal{R}}_{\mathcal{Q},\mathcal{K}}(\tau_k),}$$

Align the delayed series,
Aggregate sub-series representations

Auto-Correlation vs. Self-Attention



Auto-Correlation can extend the point-wise aggregation to series-wise.

Informer [Zhou et al. AAAI 2021], Reformer [Kitaev et al. ICLR 2020], Log Trans [Li et al. NeurIPS 2019]

Experiments: Multivariate Setting

					Transformers						LSTMs				TCN	
	Models		**Autoformer**		Informer[41]		LogTrans[20]		Reformer[17]		LSTNet[19]		LSTM[13]		TCN[3]	
	Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Energy	ETT*	96	**0.255**	**0.339**	0.365	0.453	0.768	0.642	0.658	0.619	3.142	1.365	2.041	1.073	3.041	1.330
		192	**0.281**	**0.340**	0.533	0.563	0.989	0.757	1.078	0.827	3.154	1.369	2.249	1.112	3.072	1.339
		336	**0.339**	**0.372**	1.363	0.887	1.334	0.872	1.549	0.972	3.160	1.369	2.568	1.238	3.105	1.348
		720	**0.422**	**0.419**	3.379	1.388	3.048	1.328	2.631	1.242	3.171	1.368	2.720	1.287	3.135	1.354
	Electricity	96	**0.201**	**0.317**	0.274	0.368	0.258	0.357	0.312	0.402	0.680	0.645	0.375	0.437	0.985	0.813
		192	**0.222**	**0.334**	0.296	0.386	0.266	0.368	0.348	0.433	0.725	0.676	0.442	0.473	0.996	0.821
		336	**0.231**	**0.338**	0.300	0.394	0.280	0.380	0.350	0.433	0.828	0.727	0.439	0.473	1.000	0.824
		720	**0.254**	**0.361**	0.373	0.439	0.283	0.376	0.340	0.420	0.957	0.811	0.980	0.814	1.438	0.784
Economics	Exchange	96	**0.197**	**0.323**	0.847	0.752	0.968	0.812	1.065	0.829	1.551	1.058	1.453	1.049	3.004	1.432
		192	**0.300**	**0.369**	1.204	0.895	1.040	0.851	1.188	0.906	1.477	1.028	1.846	1.179	3.048	1.444
		336	**0.509**	**0.524**	1.672	1.036	1.659	1.081	1.357	0.976	1.507	1.031	2.136	1.231	3.113	1.459
		720	**1.447**	**0.941**	2.478	1.310	1.941	1.127	1.510	1.016	2.285	1.243	2.984	1.427	3.150	1.458
Traffic	Traffic	96	**0.613**	**0.388**	0.719	0.391	0.684	0.384	0.732	0.423	1.107	0.685	0.843	0.453	1.438	0.784
		192	**0.616**	**0.382**	0.696	0.379	0.685	0.390	0.733	0.420	1.157	0.706	0.847	0.453	1.463	0.794
		336	**0.622**	**0.337**	0.777	0.420	0.733	0.408	0.742	0.420	1.216	0.730	0.853	0.455	1.479	0.799
		720	**0.660**	**0.408**	0.864	0.472	0.717	0.396	0.755	0.423	1.481	0.805	1.500	0.805	1.499	0.804
Weather	Weather	96	**0.266**	**0.336**	0.300	0.384	0.458	0.490	0.689	0.596	0.594	0.587	0.369	0.406	0.615	0.589
		192	**0.307**	**0.367**	0.598	0.544	0.658	0.589	0.752	0.638	0.560	0.565	0.416	0.435	0.629	0.600
		336	**0.359**	**0.395**	0.578	0.523	0.797	0.652	0.639	0.596	0.597	0.587	0.455	0.454	0.639	0.608
		720	**0.419**	**0.428**	1.059	0.741	0.869	0.675	1.130	0.792	0.618	0.599	0.535	0.520	0.639	0.610
Disease	ILI	24	**3.483**	**1.287**	5.764	1.677	4.480	1.444	4.400	1.382	6.026	1.770	5.914	1.734	6.624	1.830
		36	**3.103**	**1.148**	4.755	1.467	4.799	1.467	4.783	1.448	5.340	1.668	6.631	1.845	6.858	1.879
		48	**2.669**	**1.085**	4.763	1.469	4.800	1.468	4.832	1.465	6.080	1.787	6.736	1.857	6.968	1.892
		60	**2.770**	**1.125**	5.264	1.564	5.278	1.560	4.882	1.483	5.548	1.720	6.870	1.879	7.127	1.918

* *ETT* means the ETTm2. See supplementary materials for the **full benchmark** of ETTh1, ETTh2, ETTm1.

Prediction Accuracy Relative Promotion (In MSE)

↑ 74%
Input-96-predict-336

↑ 18%
Input-96-predict-336

↑ 61%
Input-96-predict-336

↑ 15%
Input-96-predict-336

↑ 21%
Input-96-predict-336

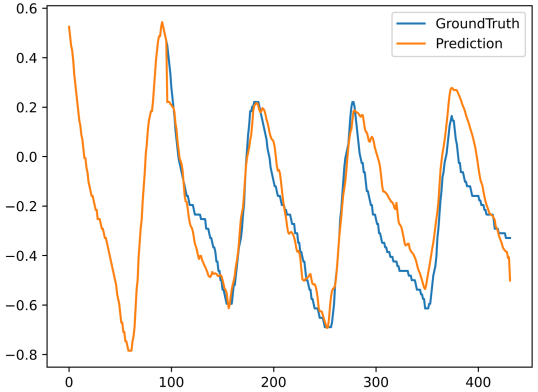
↑ 43%
Input-24-predict-48

Experiments: Univariate Setting

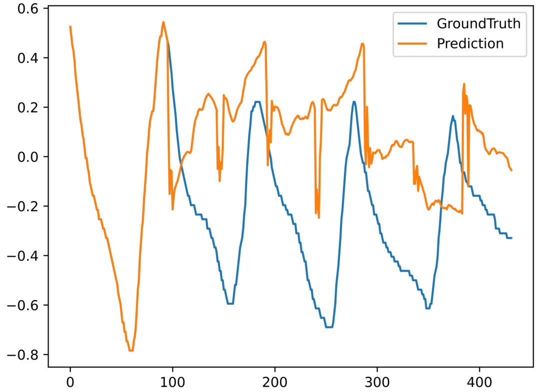
Models		Autoformer		N-BEATS[23]		Informer[41]		LogTrans[20]		Reformer[17]		DeepAR[28]		Prophet[33]		ARIMA[1]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	96	**0.065**	**0.189**	0.082	0.219	0.088	0.225	0.082	0.217	0.131	0.288	0.099	0.237	0.287	0.456	0.211	0.362
	192	**0.118**	**0.256**	0.120	0.268	0.132	0.283	0.133	0.284	0.186	0.354	0.154	0.310	0.312	0.483	0.261	0.406
	336	**0.154**	**0.305**	0.226	0.370	0.180	0.336	0.201	0.361	0.220	0.381	0.277	0.428	0.331	0.474	0.317	0.448
	720	**0.182**	**0.335**	0.188	0.338	0.300	0.435	0.268	0.407	0.267	0.430	0.332	0.468	0.534	0.593	0.366	0.487
Exchange	96	0.241	0.387	0.156	0.299	0.591	0.615	0.279	0.441	1.327	0.944	0.417	0.515	0.828	0.762	**0.112**	**0.245**
	192	**0.273**	**0.403**	0.669	0.665	1.183	0.912	1.950	1.048	1.258	0.924	0.813	0.735	0.909	0.974	0.304	0.404
	336	**0.508**	**0.539**	0.611	0.605	1.367	0.984	2.438	1.262	2.179	1.296	1.331	0.962	1.304	0.988	0.736	0.598
	720	**0.991**	**0.768**	1.111	0.860	1.872	1.072	2.010	1.247	1.280	0.953	1.894	1.181	3.238	1.566	1.871	0.935

Competitive baseline
N-BEATS

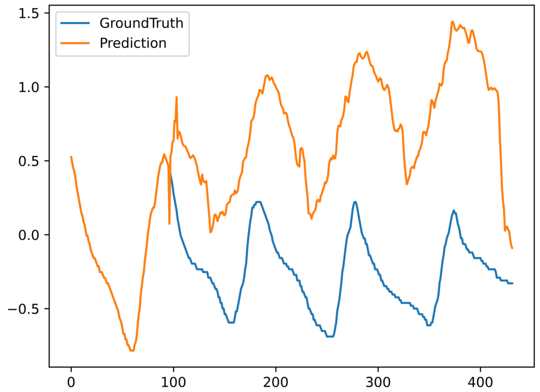
Showcases



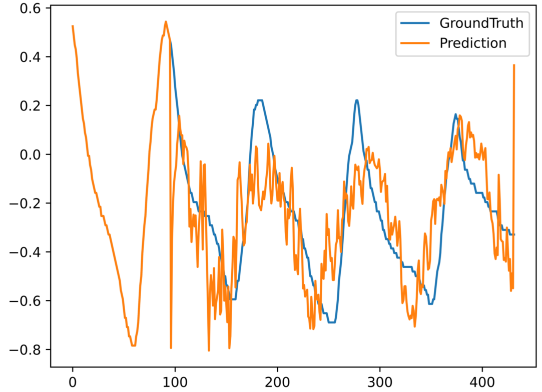
Autoformer



Informer

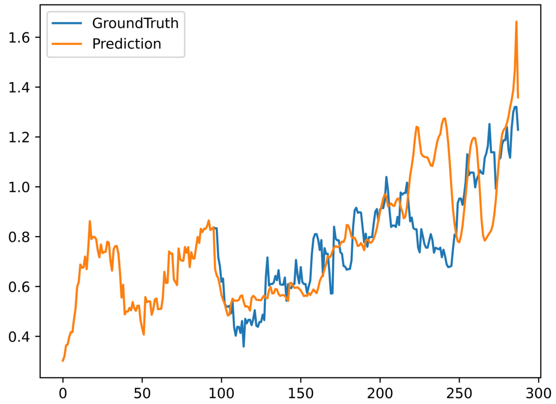


LogTrans

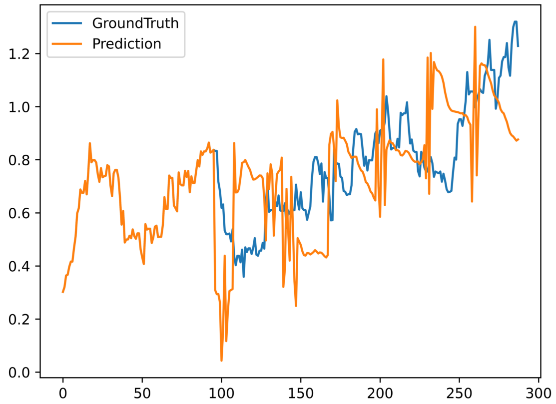


Reformer

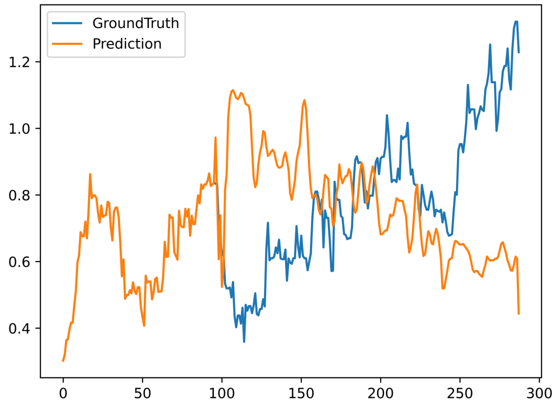
(1) ETT dataset with input-96-predict-336 (Energy, with obvious periodicity)



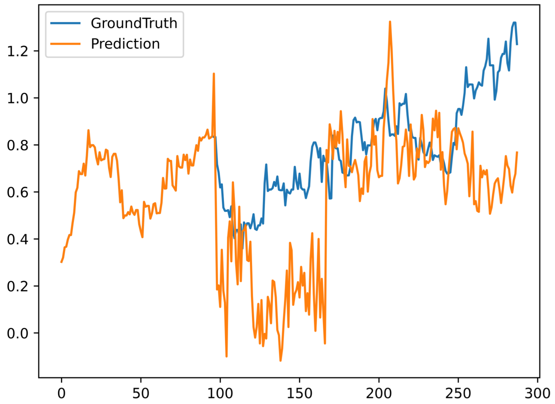
Autoformer



Informer



LogTrans



Reformer

(2) Exchange dataset with input-96-predict-192 (Economics, without obvious periodicity)

Learned Dependencies

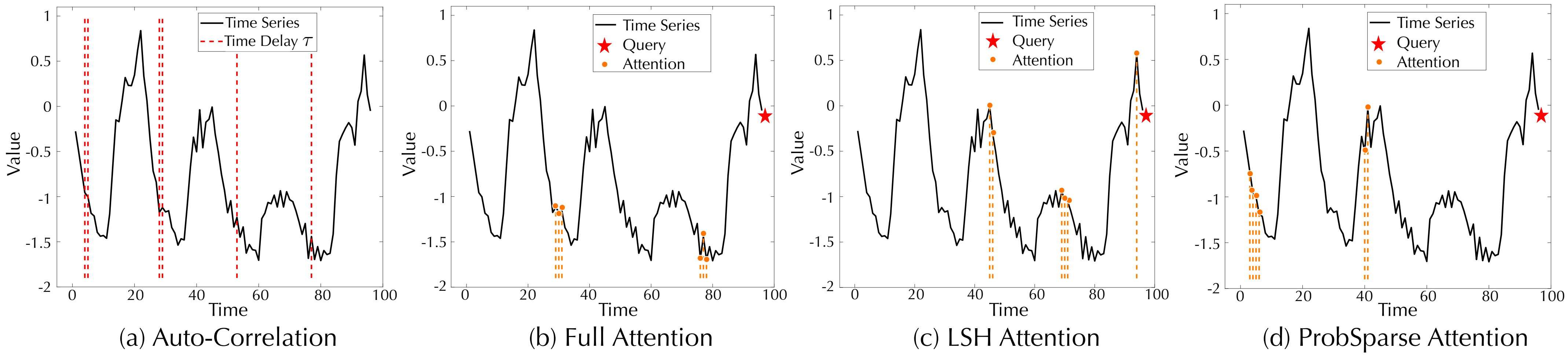


Figure 5: Visualization of learned dependencies. For clearness, we select the top-6 time delay sizes $\tau_1, \cdots, \tau_6$ of Auto-Correlation and mark them in raw series (red lines). For self-attentions, top-6 similar points with respect to the last time step (red stars) are also marked by orange points.

Auto-Correlation can discover the relevant information more sufficiently and precisely.

Transformers [Vaswani et al. NeurIPS 2017], Informer [Zhou et al. AAAI 2021], Reformer [Kitaev et al. ICLR 2020]

Learned Lags

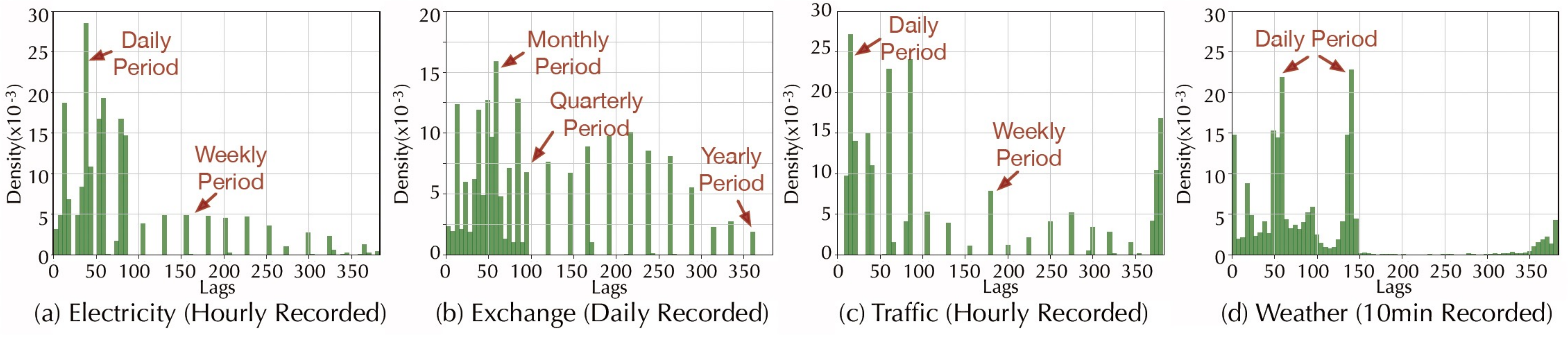
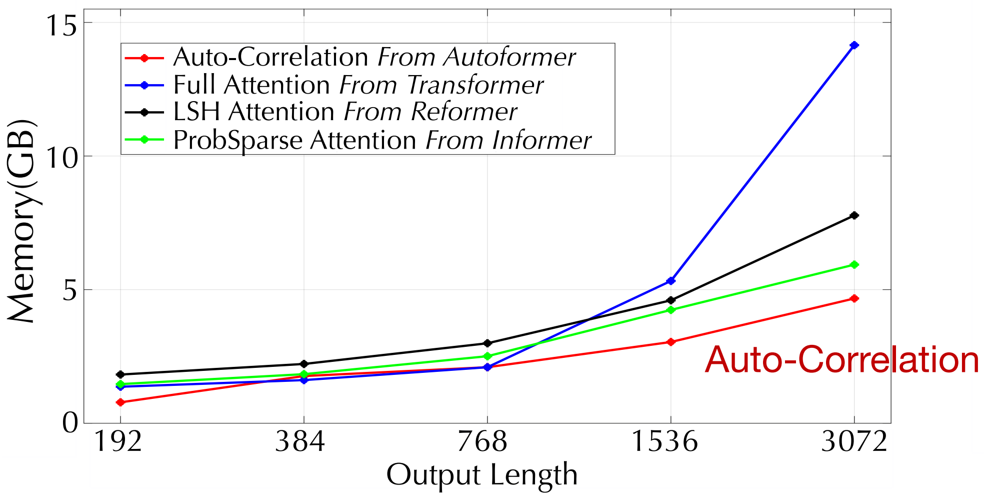


Figure 6: Statistics of learned lags. For each time series in the test set, we count the top 10 lags learned by decoder for the input-96-predict-336 task. Figure (a)-(d) are the density histograms.

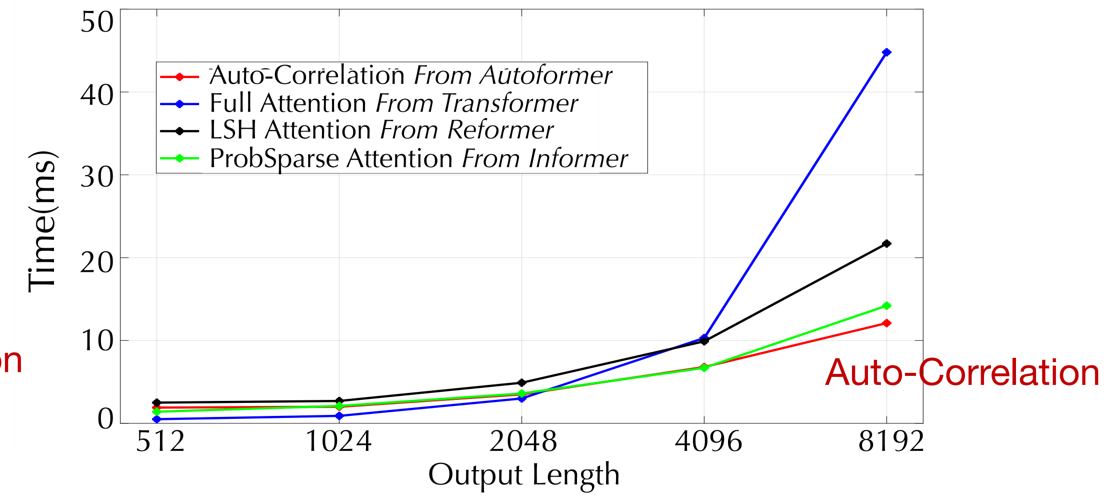
Learned lags can reflect the

human-interpretable prediction.

Efficiency Analysis



(a) Memory Efficiency Analysis



(b) Running Time Efficiency Analysis

Figure 6: Efficiency Analysis. For memory, we replace Auto-Correlation with self-attention family in Autoformer and record the memory with input 96. For running time, we run the Auto-Correlation or self-attentions 10^3 times to get the execution time per step. The output length increases exponentially.

Auto-Correlation presents remarkable $O(L \log L)$ complexity

in both memory and computation.

Summary

	Motivation		Autoformer
Intricate Temporal Patterns	Classic method of time series analysis	⇨	Decomposition architecture to ravel out the entangled temporal patterns
Deal with Long Series	Stochastic process theory	⇨	Series-wise Auto-Correlation with $\mathcal{O}(L \log L)$ complexity

Autoformer achieves the remarkable state-of-the-art on extensive benchmarks.

Open Source

wuhaixu2016 Merge pull request #134 from elisim/patch-2 … e6371e2 on Mar 6 **75** commits

data_provider	Update data_factory.py	6 months ago
exp	Update exp_main.py	4 months ago
layers	adding the length of the output signal to irfft	3 months ago
models	Update Reformer.py	last year
pic	init code	2 years ago
scripts	updated scripts for all experiments using the ETTm1 and ETTm2 data...	4 months ago
utils	Update metrics.py	6 months ago
.gitignore	Initial commit	2 years ago
Dockerfile	add docker, make and conda env	2 years ago
LICENSE	Initial commit	2 years ago
Makefile	add docker, make and conda env	2 years ago

About Code release for "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting" (NeurIPS 2021), https://arxiv.org/abs/2106.13008

deep-learning time-series

Readme

MIT license

1.1k stars

14 watching

285 forks

Report repository

Releases

No releases published

https://github.com/thuml/Autoformer

Well-organized code and pre-processed dataset

Corrformer

nature machine intelligence

Article
https://doi.org/10.1038/s42256-023-00667-9

Interpretable weather forecasting for worldwide stations with a unified deep model

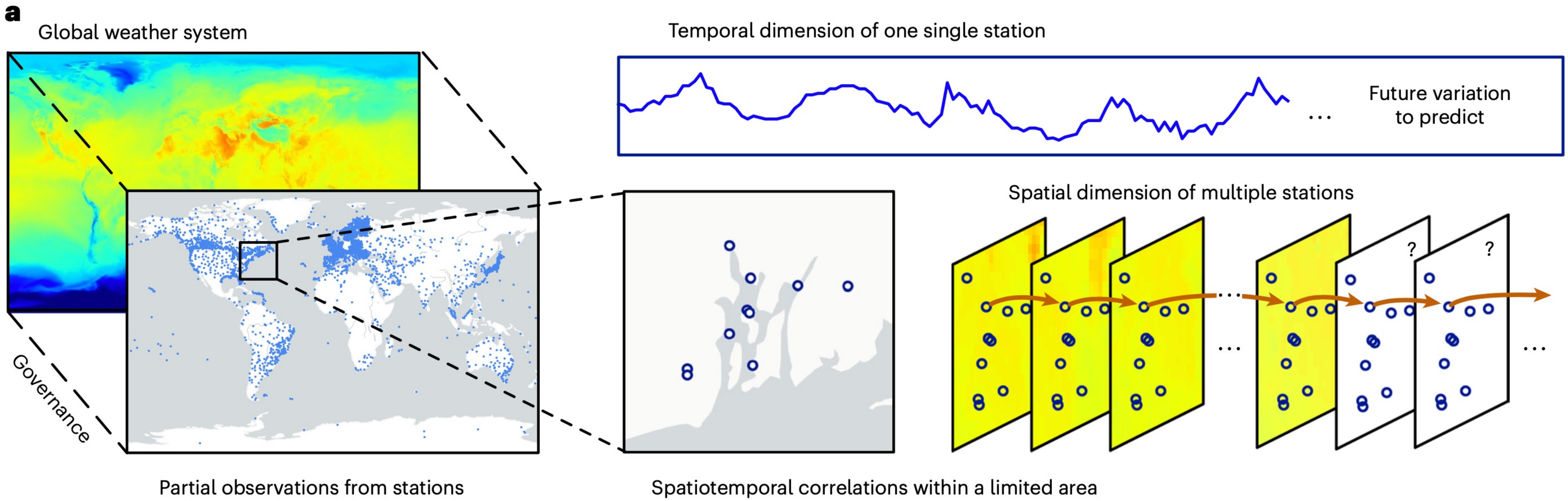
Received: 3 July 2022
Accepted: 26 April 2023

Haixu Wu, Hang Zhou, Mingsheng Long & Jianmin Wang

https://www.nature.com/articles/s42256-023-00667-9

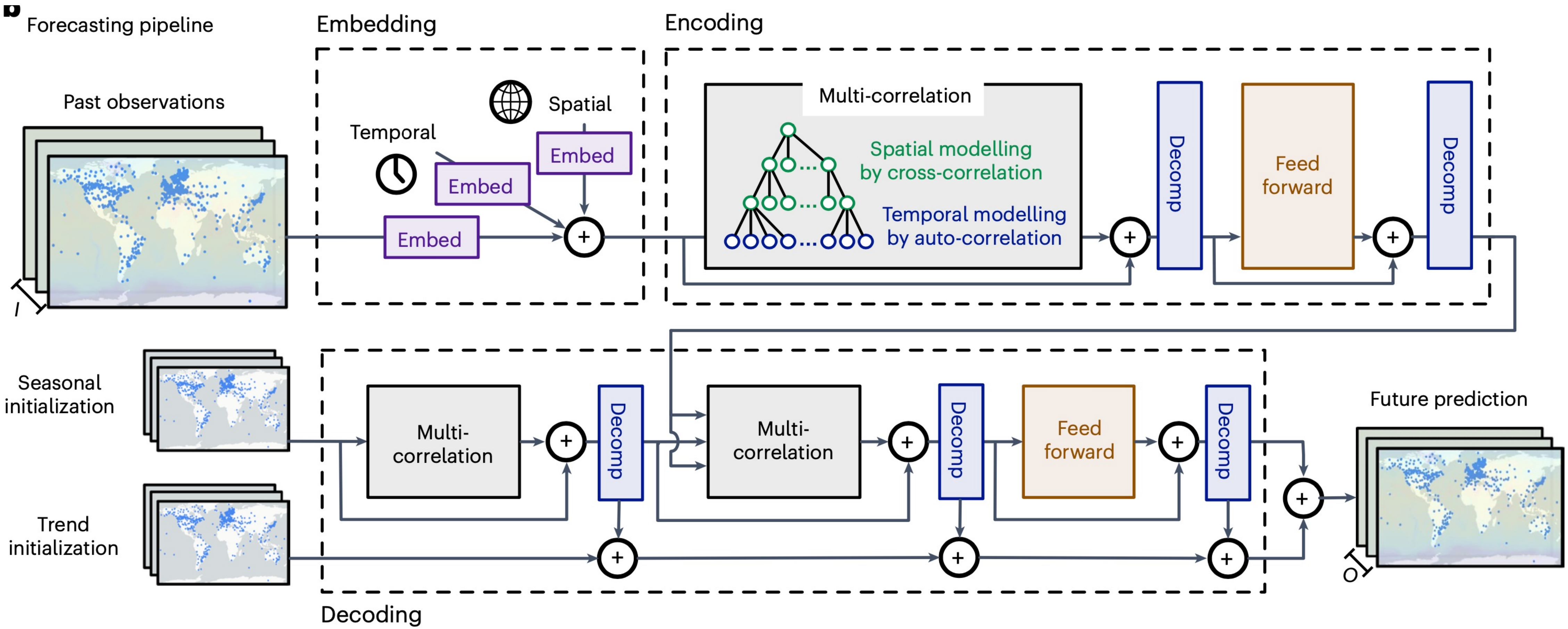
Weather Forecasting for Worldwide Stations

Real-time collaborative forecasts of **tens of thousands** automatic weather stations (Future prediction in **0-24 hours** near the ground)

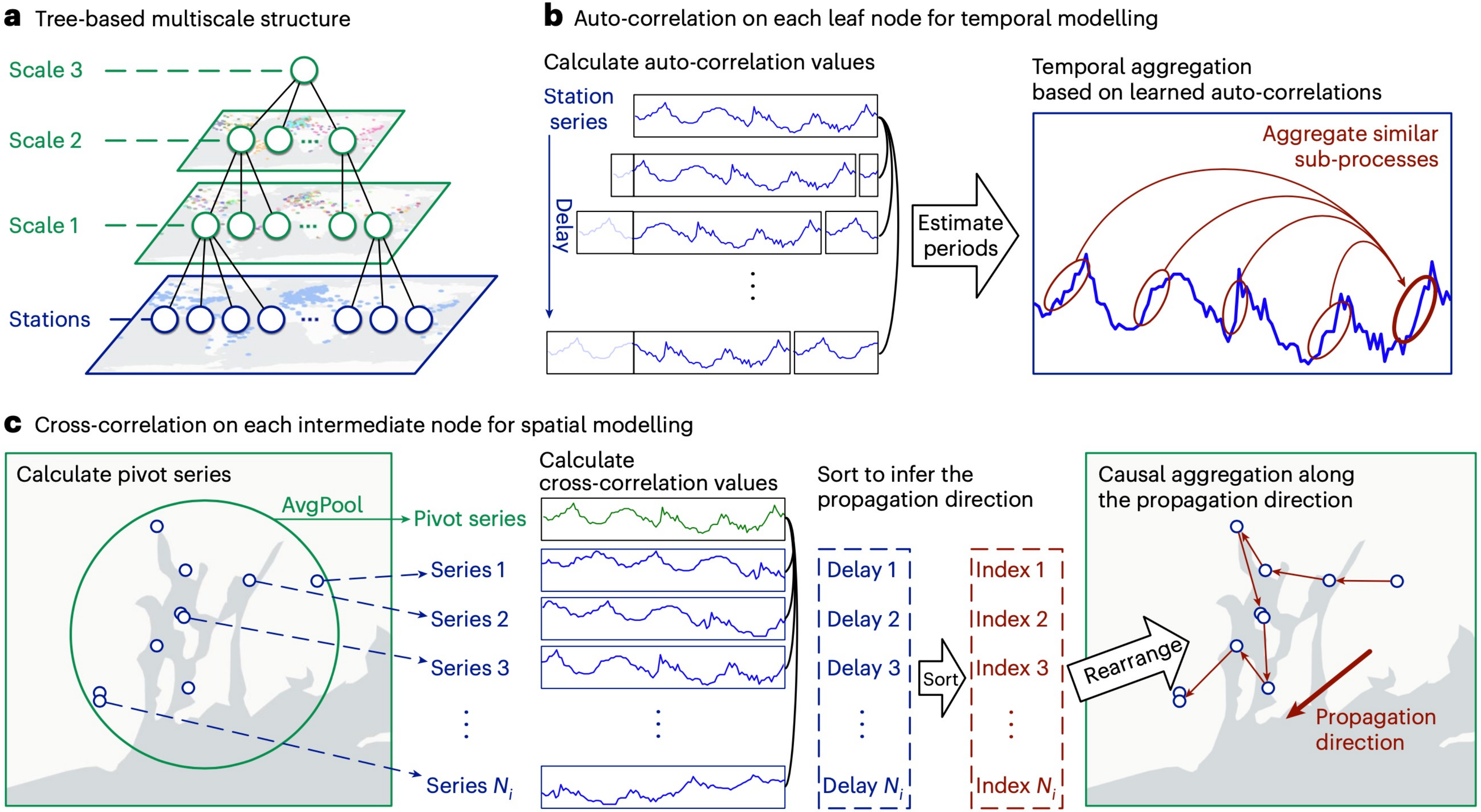


Corrformer: Model Architecture

- Inheriting Decomposition from Autoformer
- Utilizing Multi-Correlation to capture spacial-temporal correlations

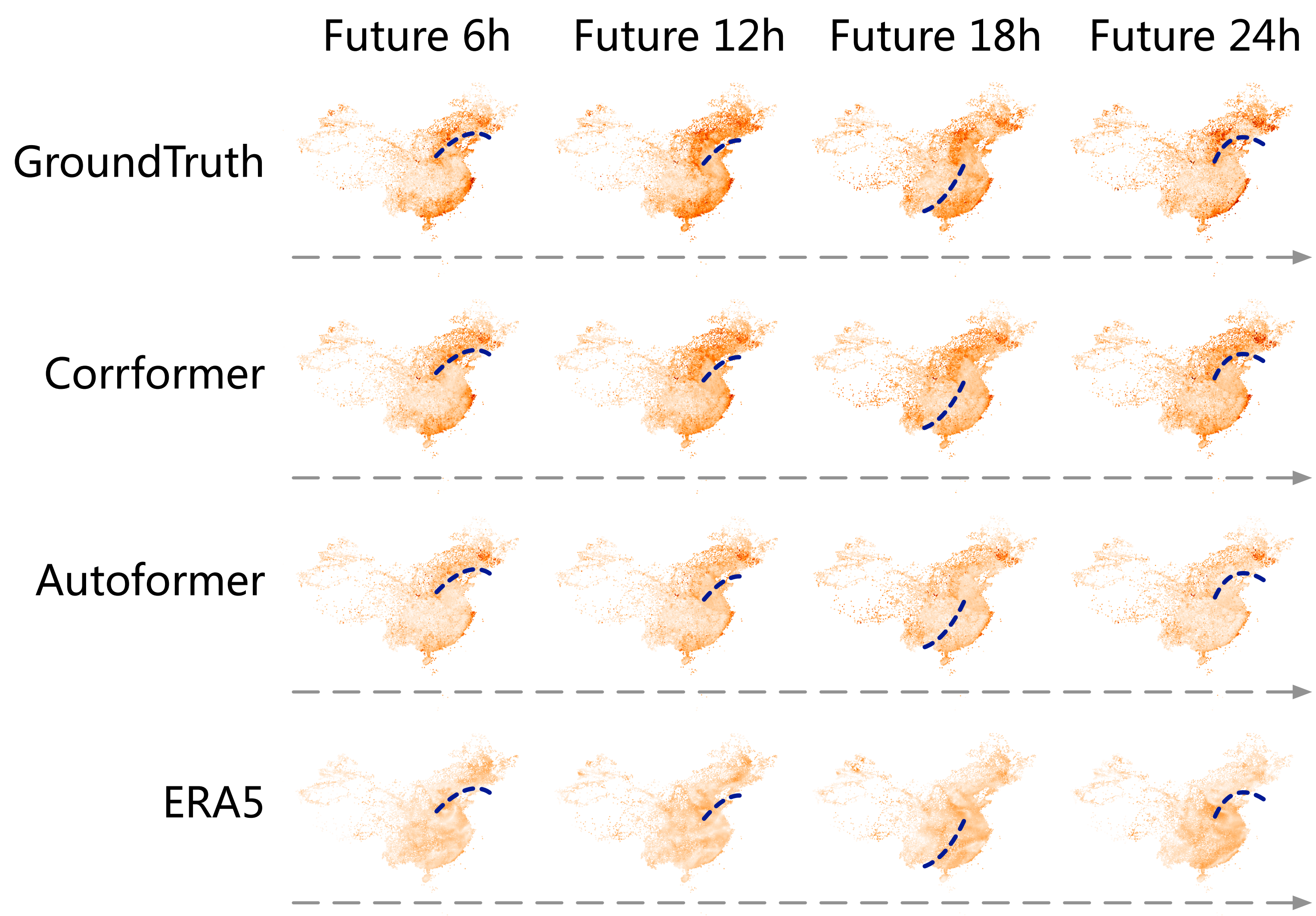


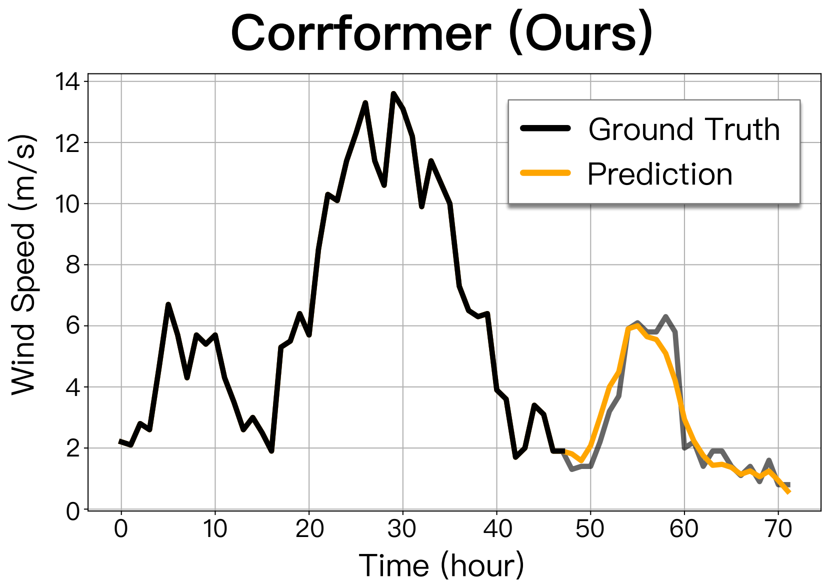
Corrformer: Multi-Correlation Mechanism



- Multi-scale structure
- Series-wise auto-correlation for temporal modeling
- Pivot-based cross-correlation for spatial modeling
- Log-linear Complexity
- $O(N^2L^2) \rightarrow O(NL\log L)$

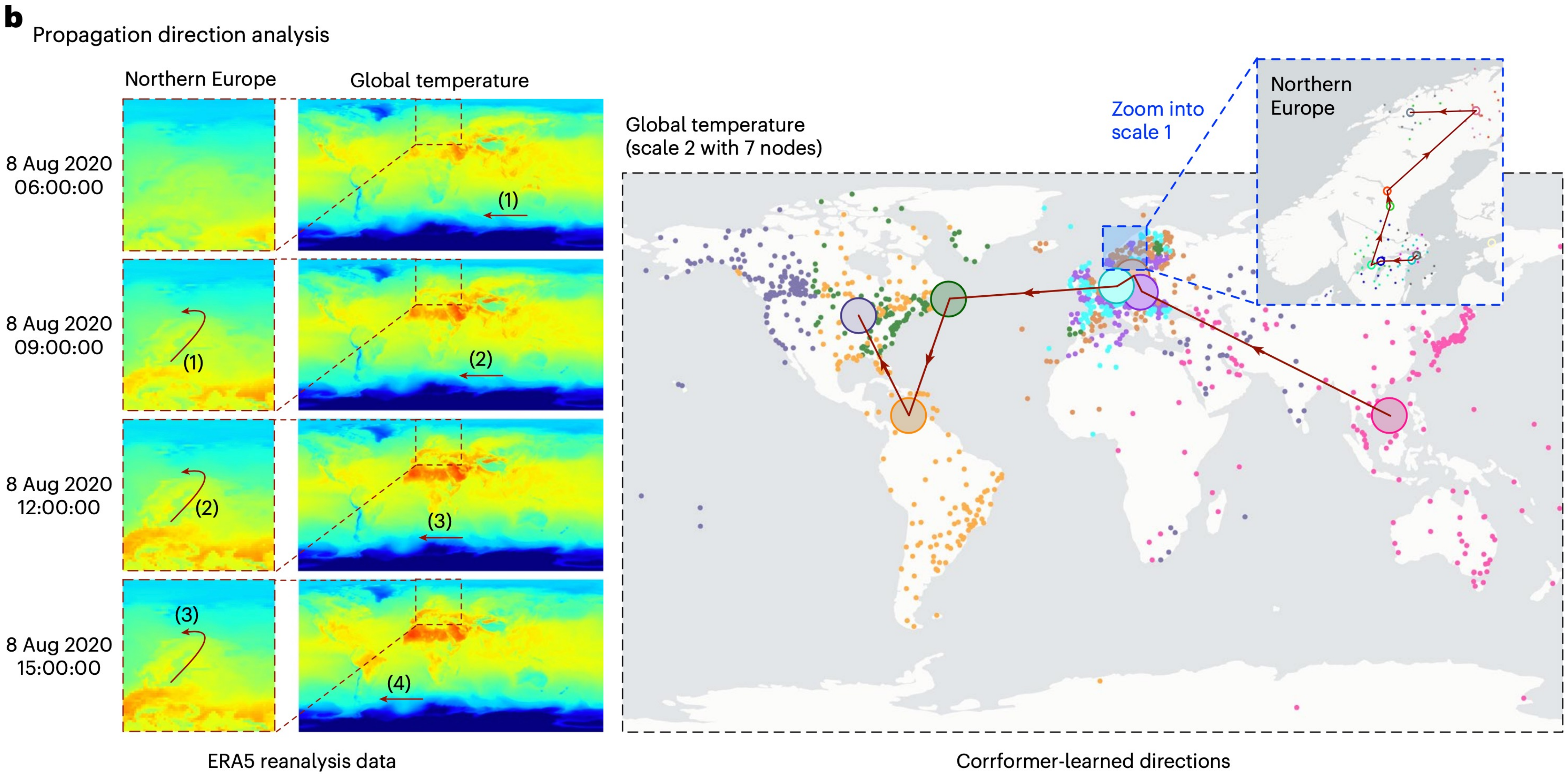
Regional Wind Forecasting



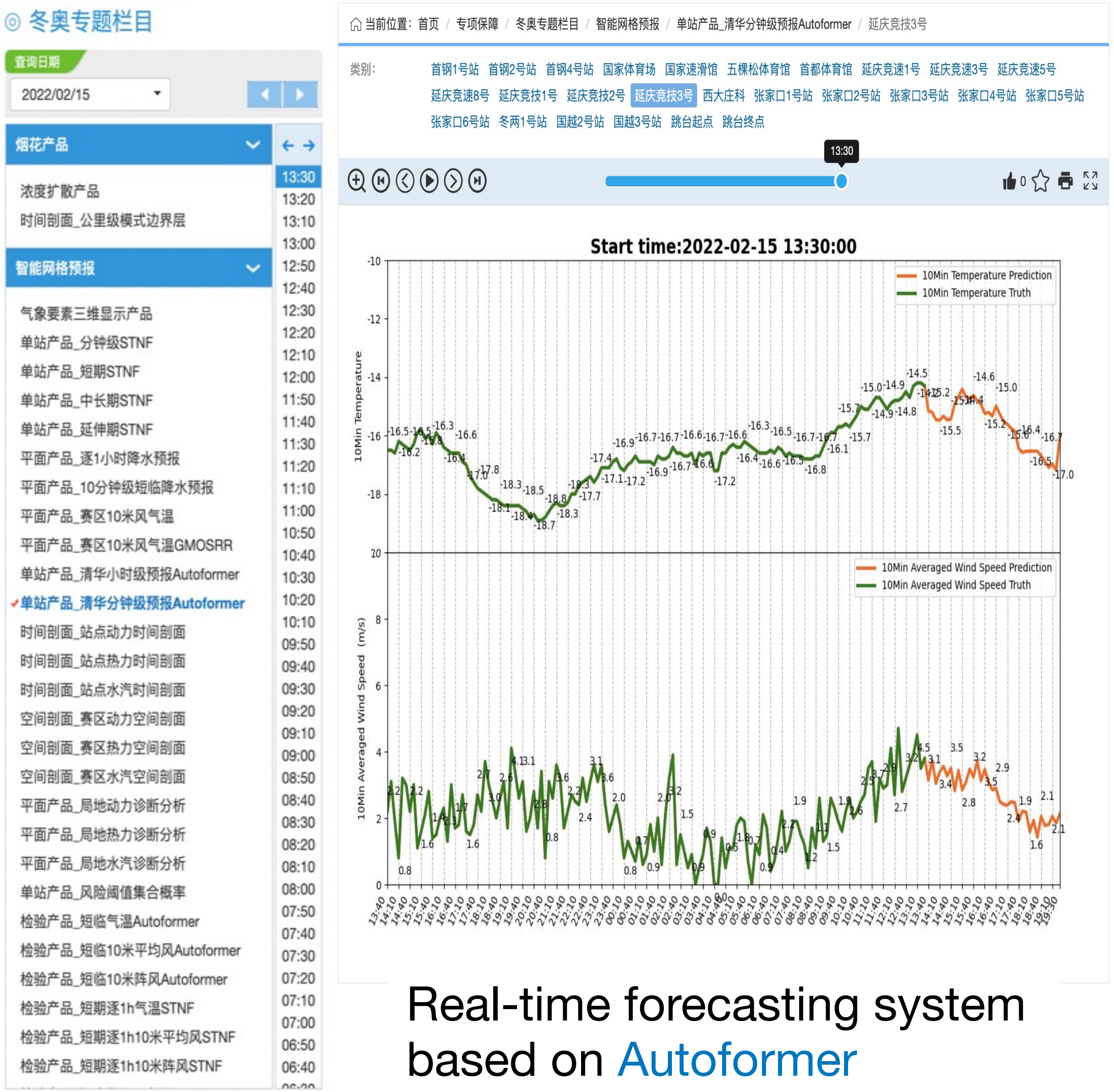


- ✓ Overall 34040 Stations
- ✓ Training one day on single GPU, forecast issued in one second
- ✓ Achieve the SOTA

Interpretable Worldwide Forecasting



Service in 2022 Winter Olympics



Real-time forecasting system based on Autoformer

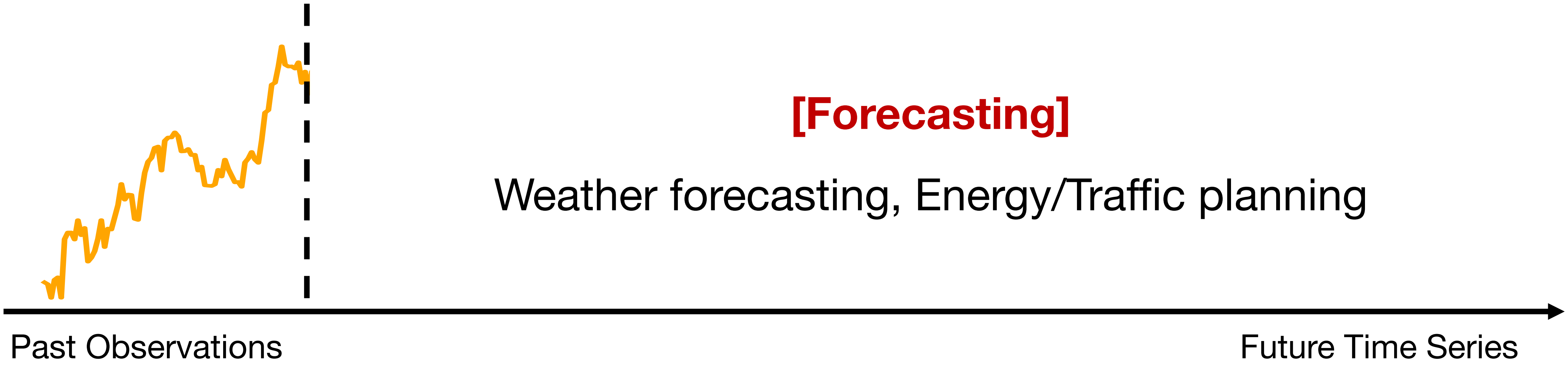
Indoors: Temperature

Outdoors: Wind speed

Provides online forecast service of temperature and wind speed for the 2022 Beijing Winter Olympics, assists athletes preparation and schedule planning, works as a solid support for the competition.

Achieves **10-minute** real-time temperature and wind speed forecast based on meteorological observation, and achieves **23% lower forecast error** than the mainstream numerical prediction methods.

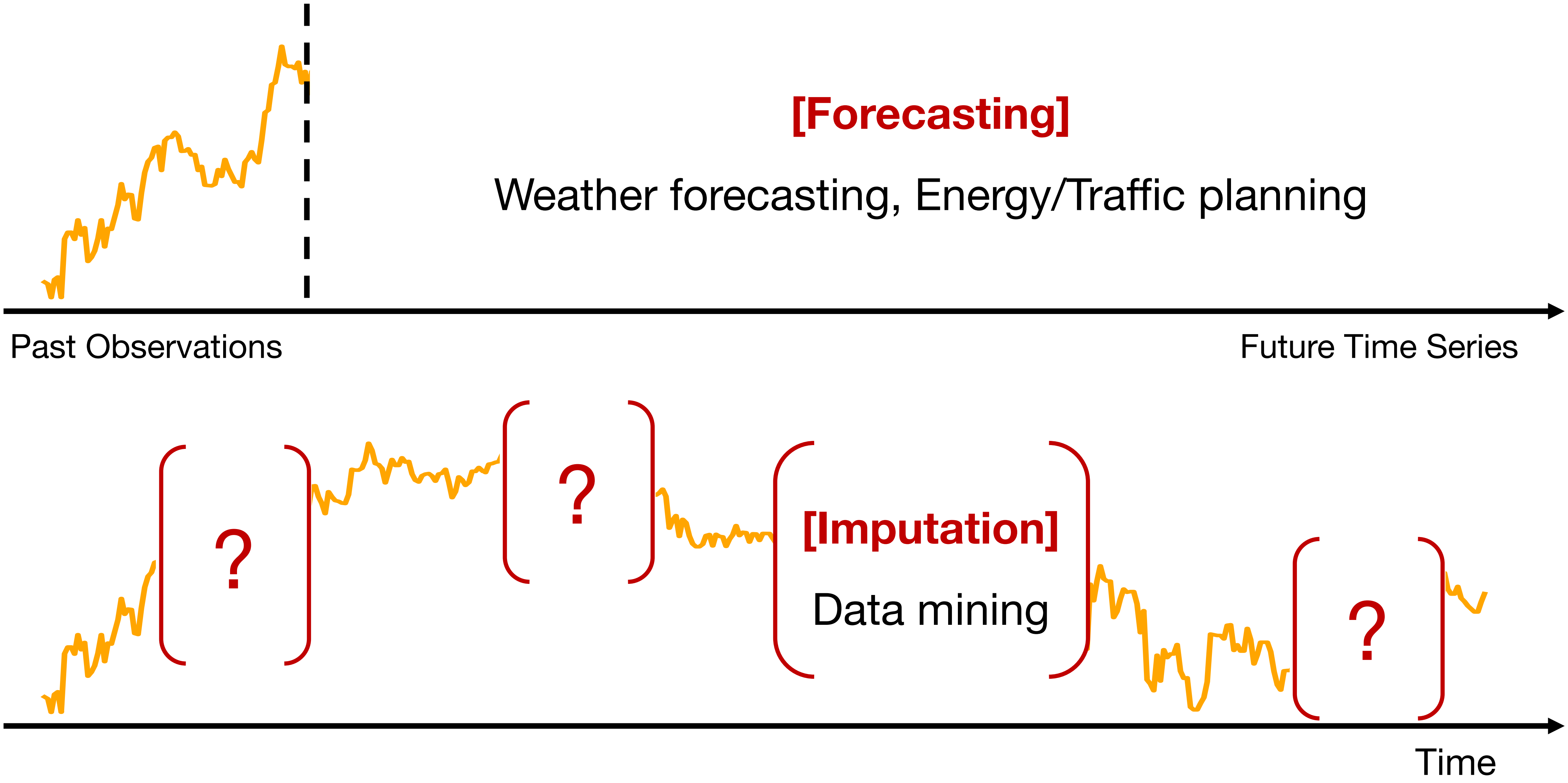
General Time Series Analysis



[Forecasting]

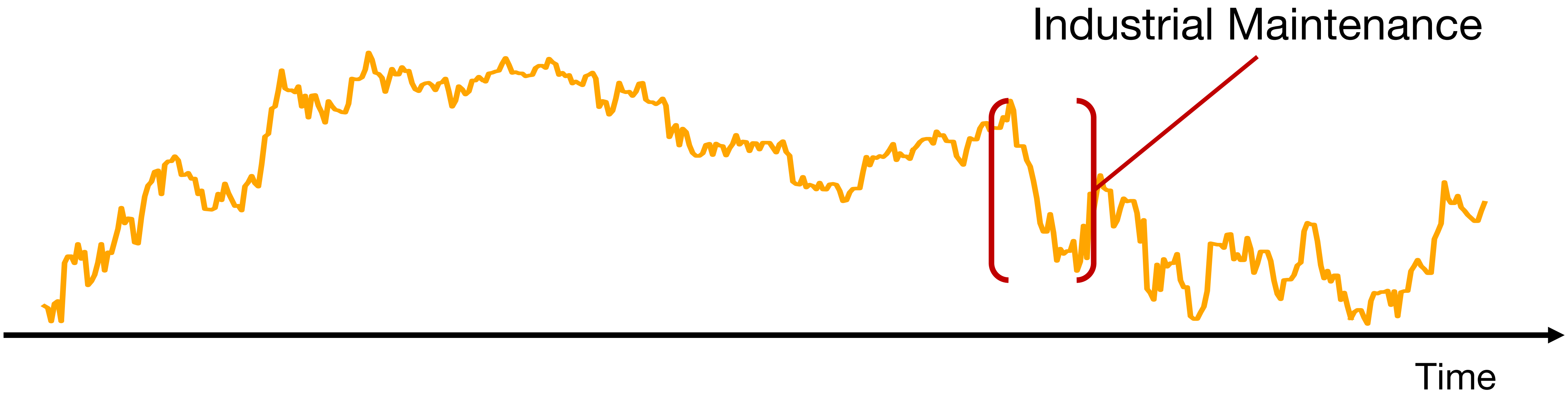
Weather forecasting, Energy/Traffic planning

General Time Series Analysis



General Time Series Analysis

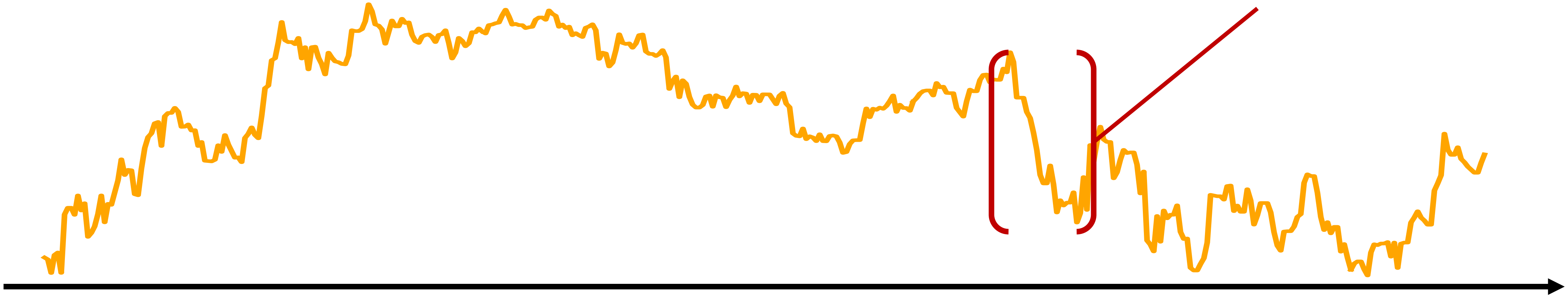
[Anomaly Detection]



General Time Series Analysis

[Anomaly Detection]

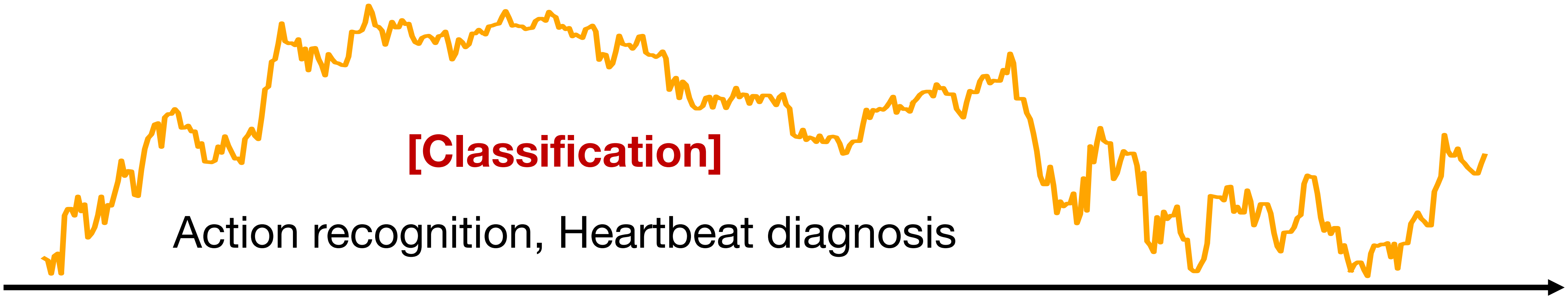
Industrial Maintenance



Time

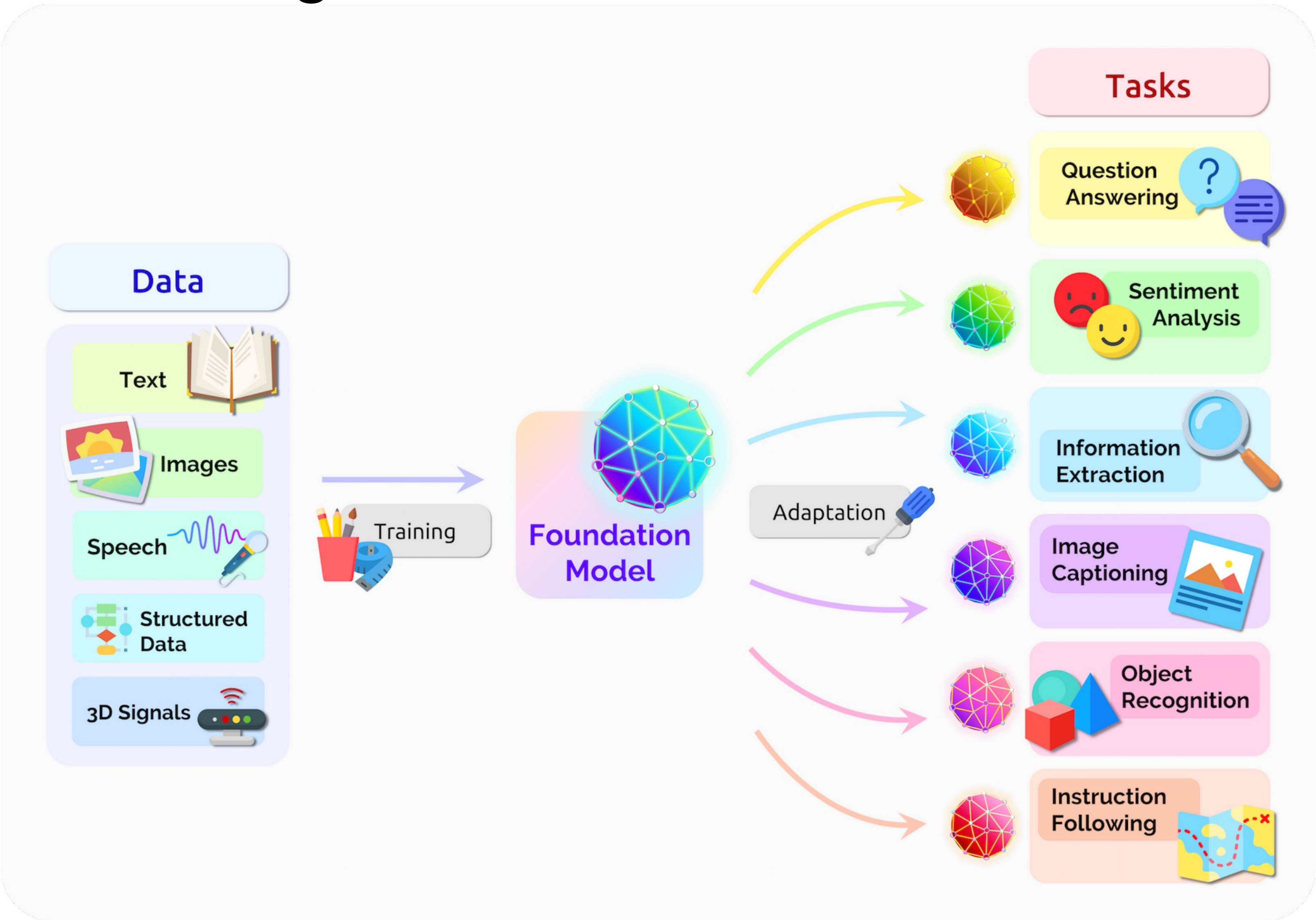
[Classification]

Action recognition, Heartbeat diagnosis



Time

In Pursing Foundation Models



[Data Universal]

Learn from various modalities

[Task Universal]

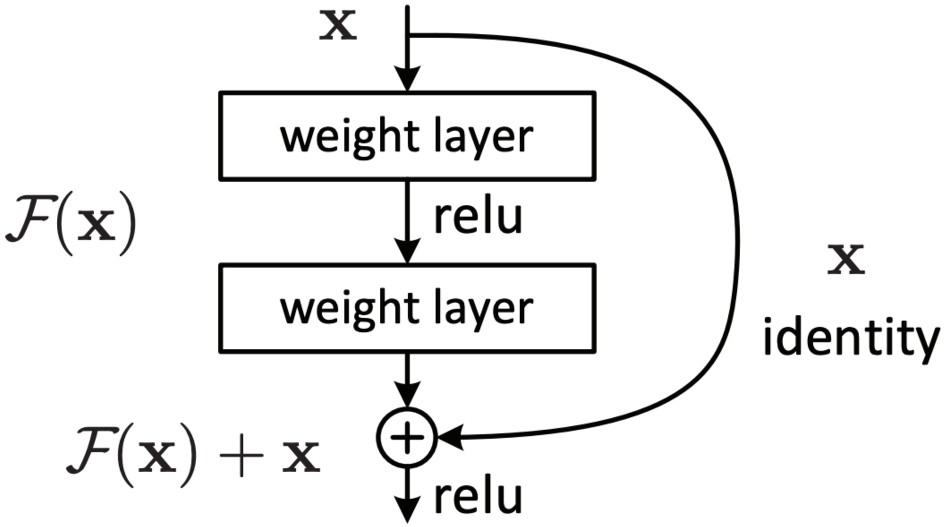
Adapt to a wide range of downstream tasks

Bommasani et al. On the Opportunities and Risks of Foundation Models. *Arxiv 2021.*

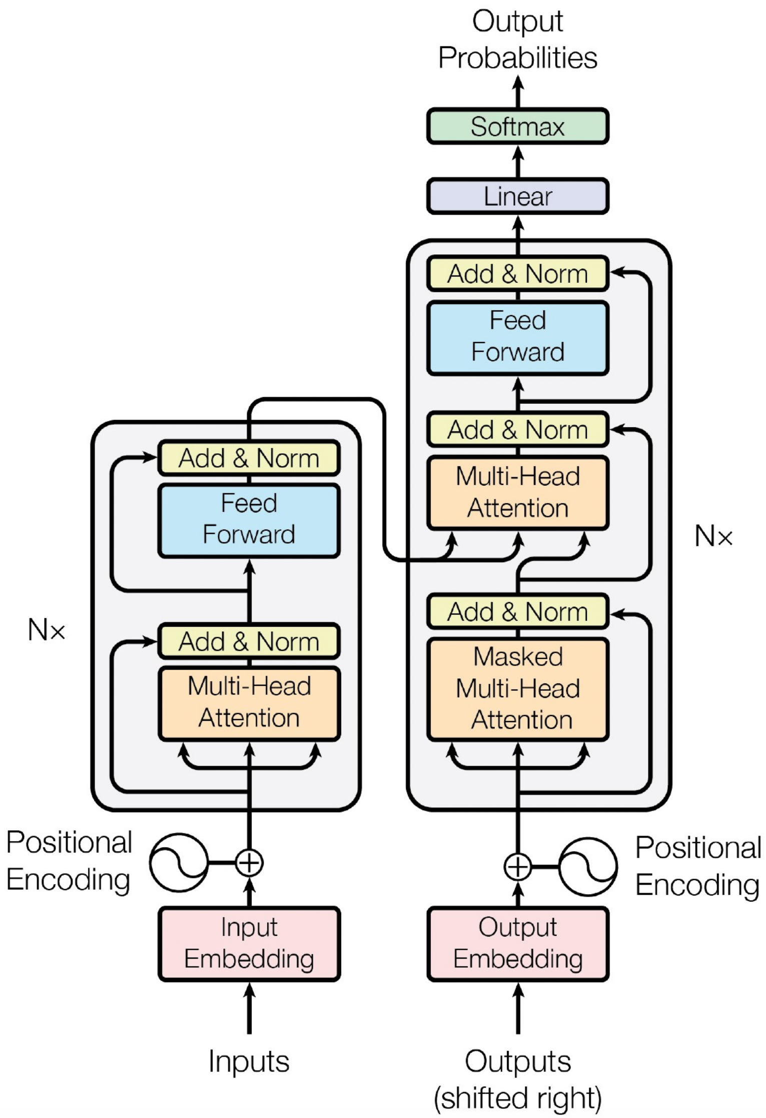
Foundation Models in CV and NLP

Universal backbone with

task-specific heads for different tasks.



Classification, Object detection, Segmentation



Classification, Generation

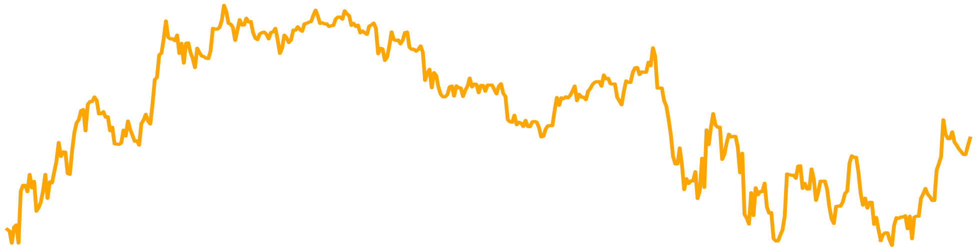
TimesNet

TIMESNET: TEMPORAL 2D-VARIATION MODELING FOR GENERAL TIME SERIES ANALYSIS

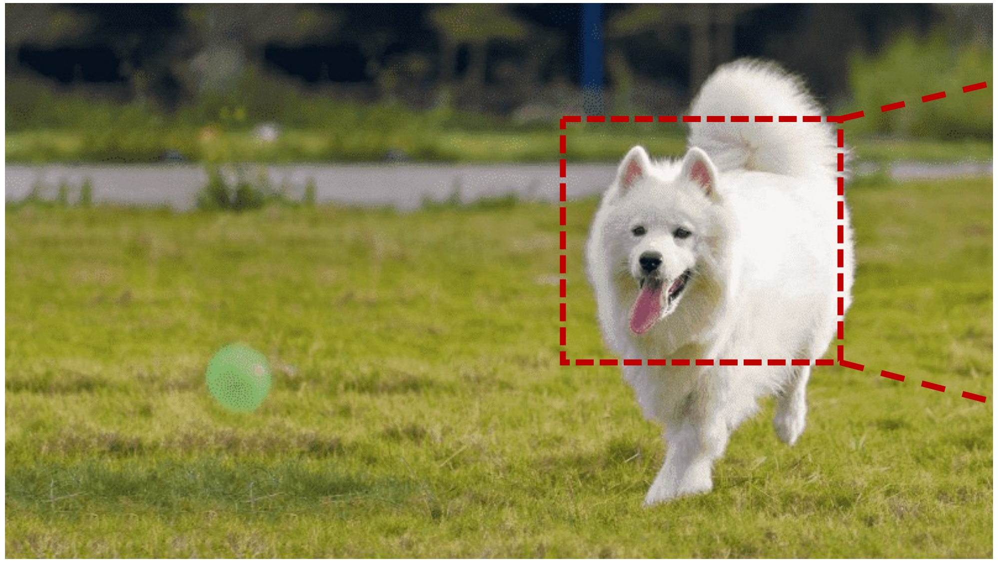
Haixu Wu,* Tengge Hu,* Yong Liu,* Hang Zhou, Jianmin Wang, Mingsheng Long✉
School of Software, BNRist, Tsinghua University, Beijing 100084, China
{whx20,liuyong21,htg21,h-zhou18}@mails.tsinghua.edu.cn
{jimwang,mingsheng}@tsinghua.edu.cn

Differences among Image, Language, Time Series

TimesNet is for time series analysis



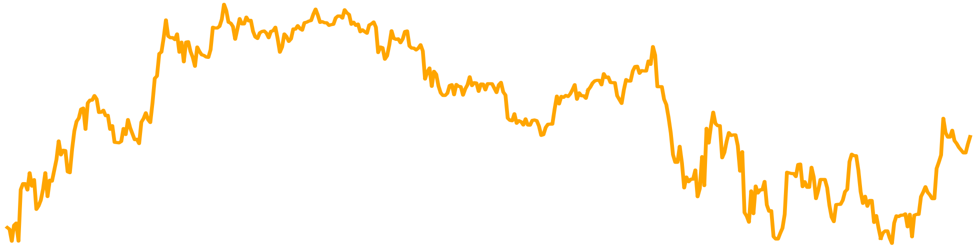
Differences among Image, Language, Time Series



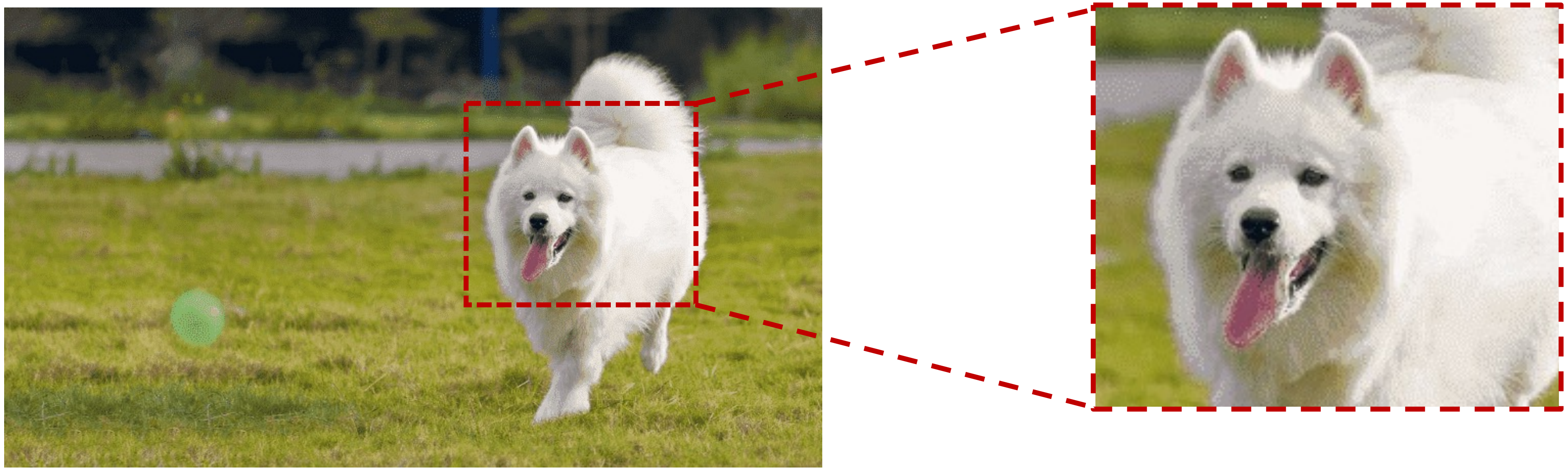
TimesNet is for time series analysis

Analysis is the process of breaking a complex topic into smaller parts for a better understanding.

WIKIPEDIA
The Free Encyclopedia



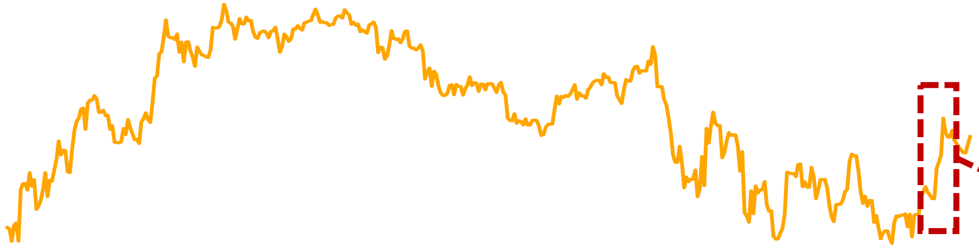
Differences among Image, Language, Time Series



TimesNet is for time series analysis

Analysis is the process of breaking a complex topic into smaller parts for a better understanding.

Wikipedia
The Free Encyclopedia

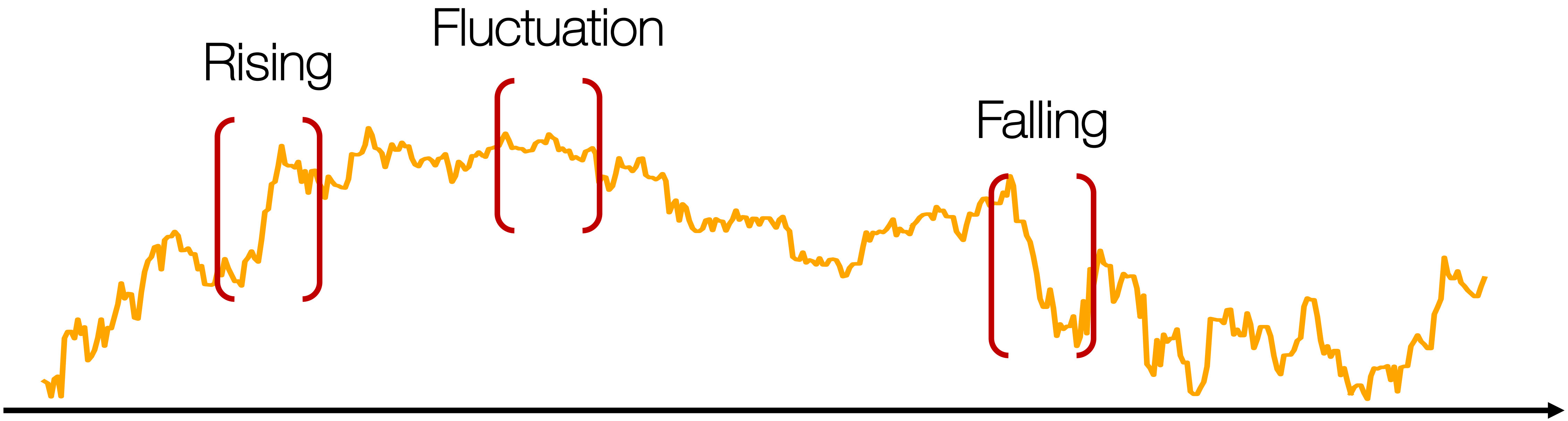


Each time point only saves some scalars.

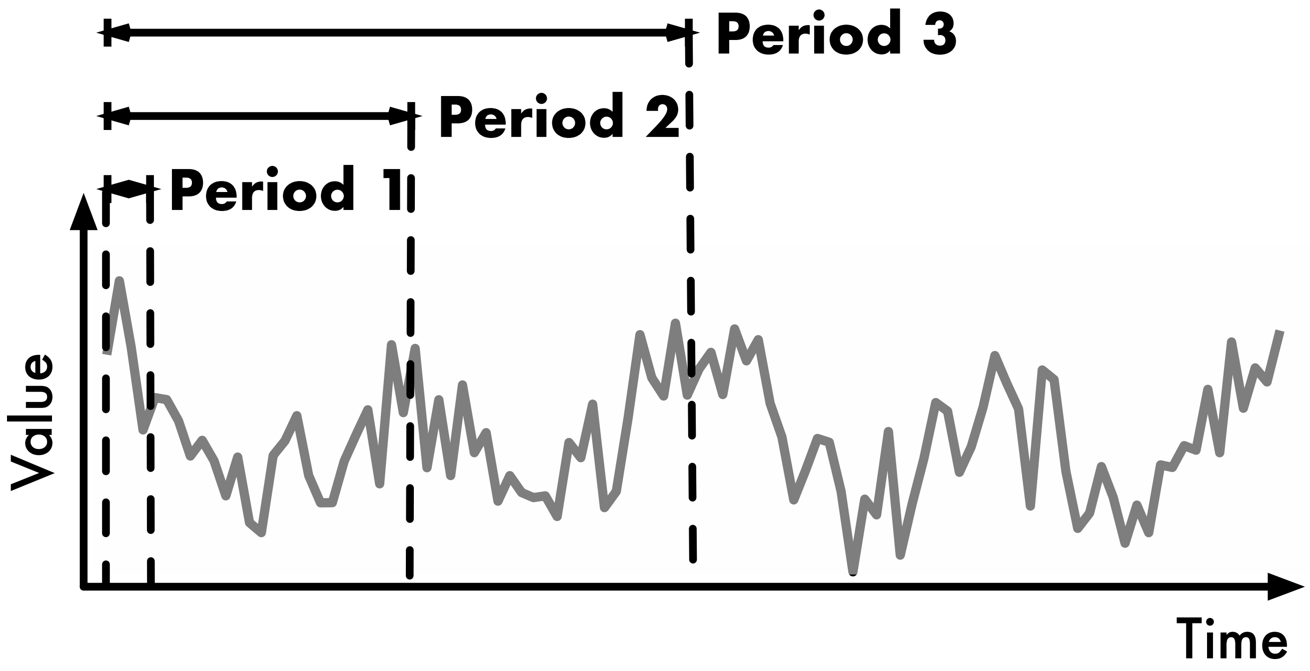
Temporal Variations of Time Series

More information of time series is in **temporal variations**,

such as continuity, periodicity, trend and etc.



Multi-periodicity View of Time Series

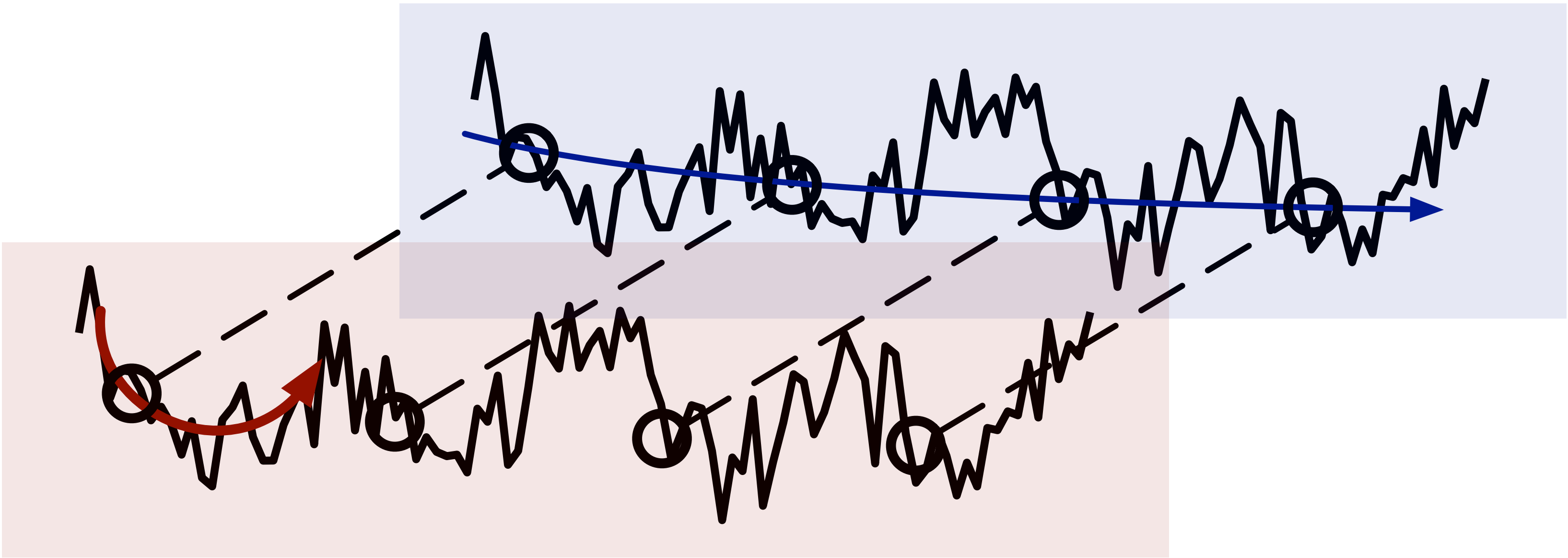


- ✓ Traffic: daily and weekly
- ✓ Weather: daily and yearly

Real-world time series usually present multi-periodicity.

Multiple periods overlap and interact with each other.

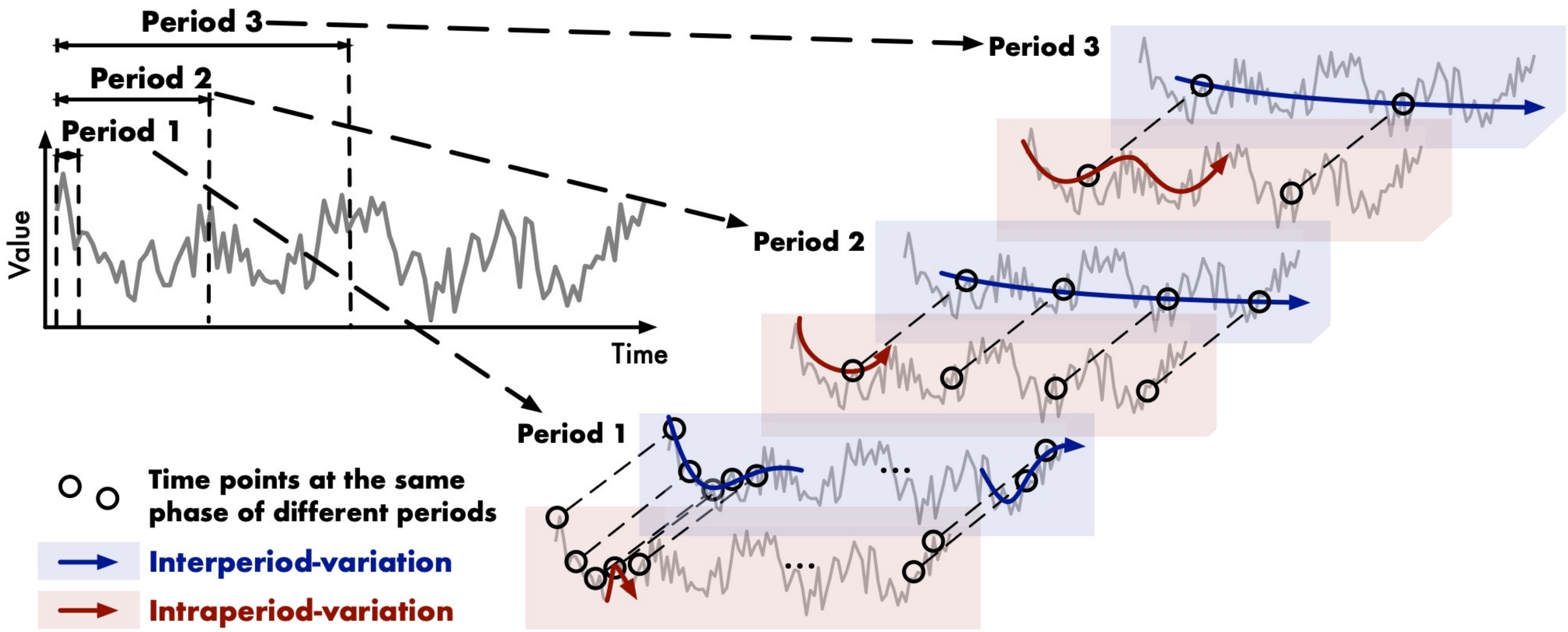
Intraperiod- and Interperiod-Variations



- ✓ **Intraperiod**: adjacent area, **short-term variations**
- ✓ **Interperiod**: same phase in adjacent periods, **long-term variations**

Non-periodic cases, the variations will be dominated by intraperiod-variations.

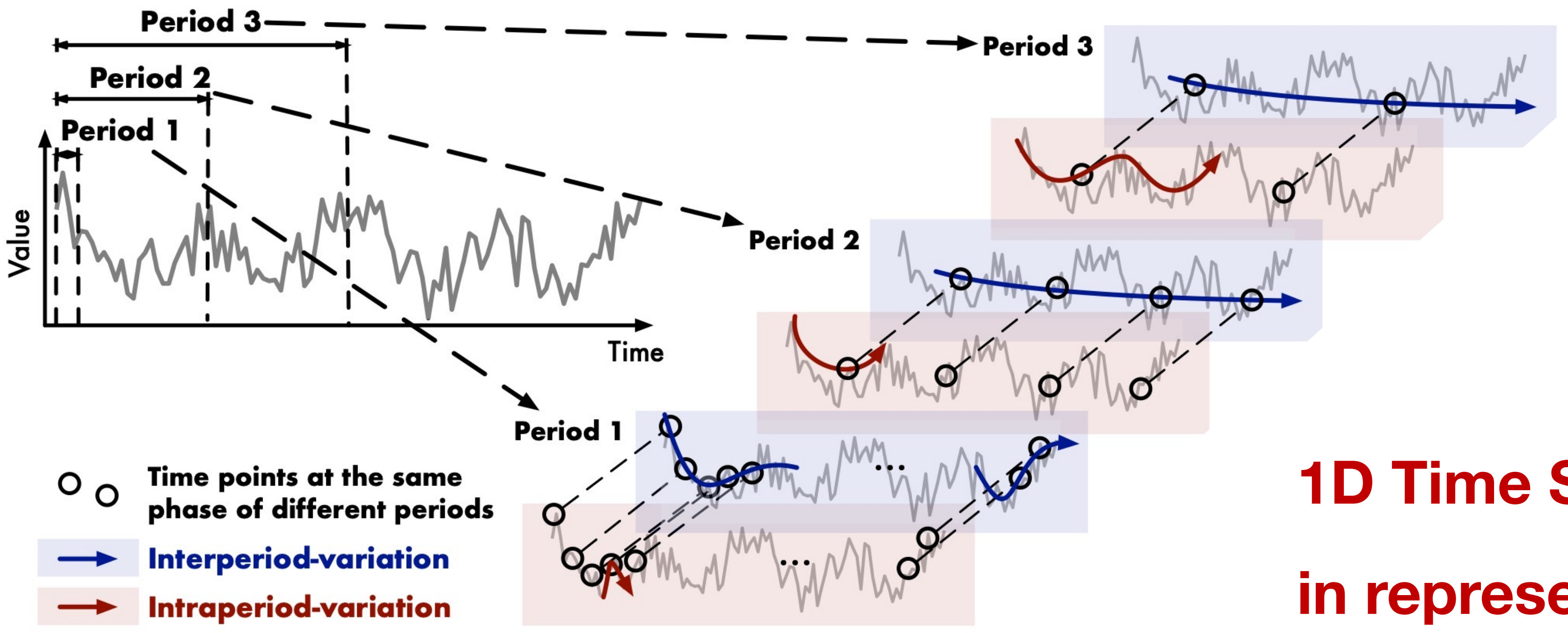
TimesNet: Overall Design



① Multi-periodicity

A modular architecture to disentangle intricate temporal patterns

TimesNet: Overall Design

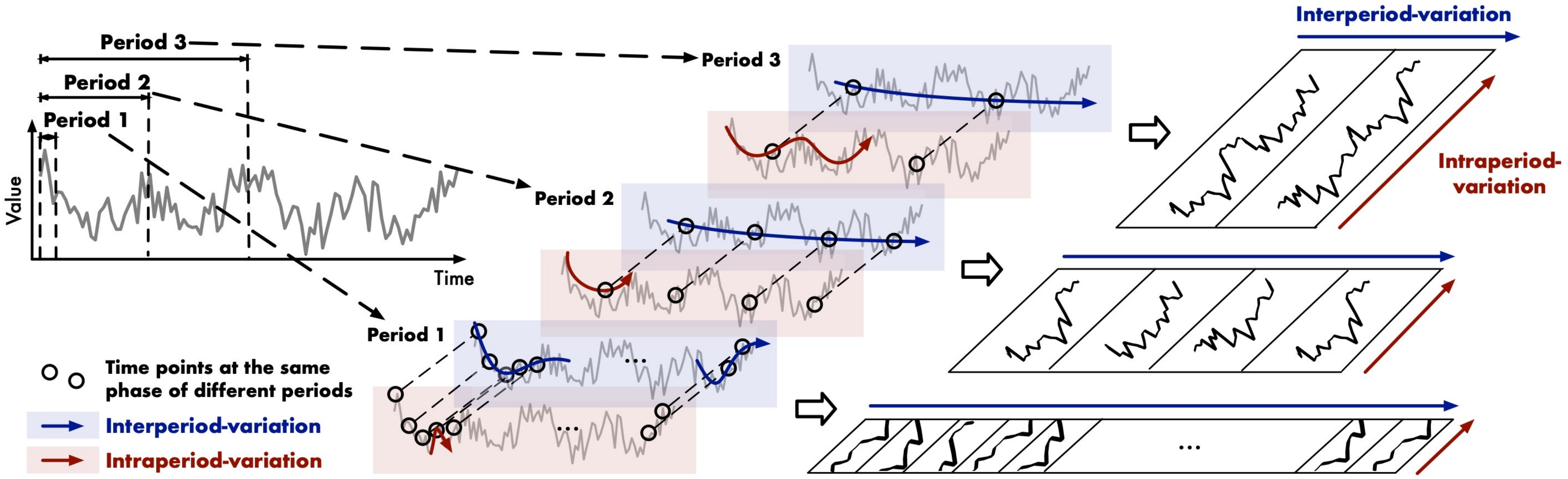


1D Time Series has limitations in representation capability.

① Multi-periodicity

A modular architecture to disentangle intricate temporal patterns

TimesNet: Overall Design

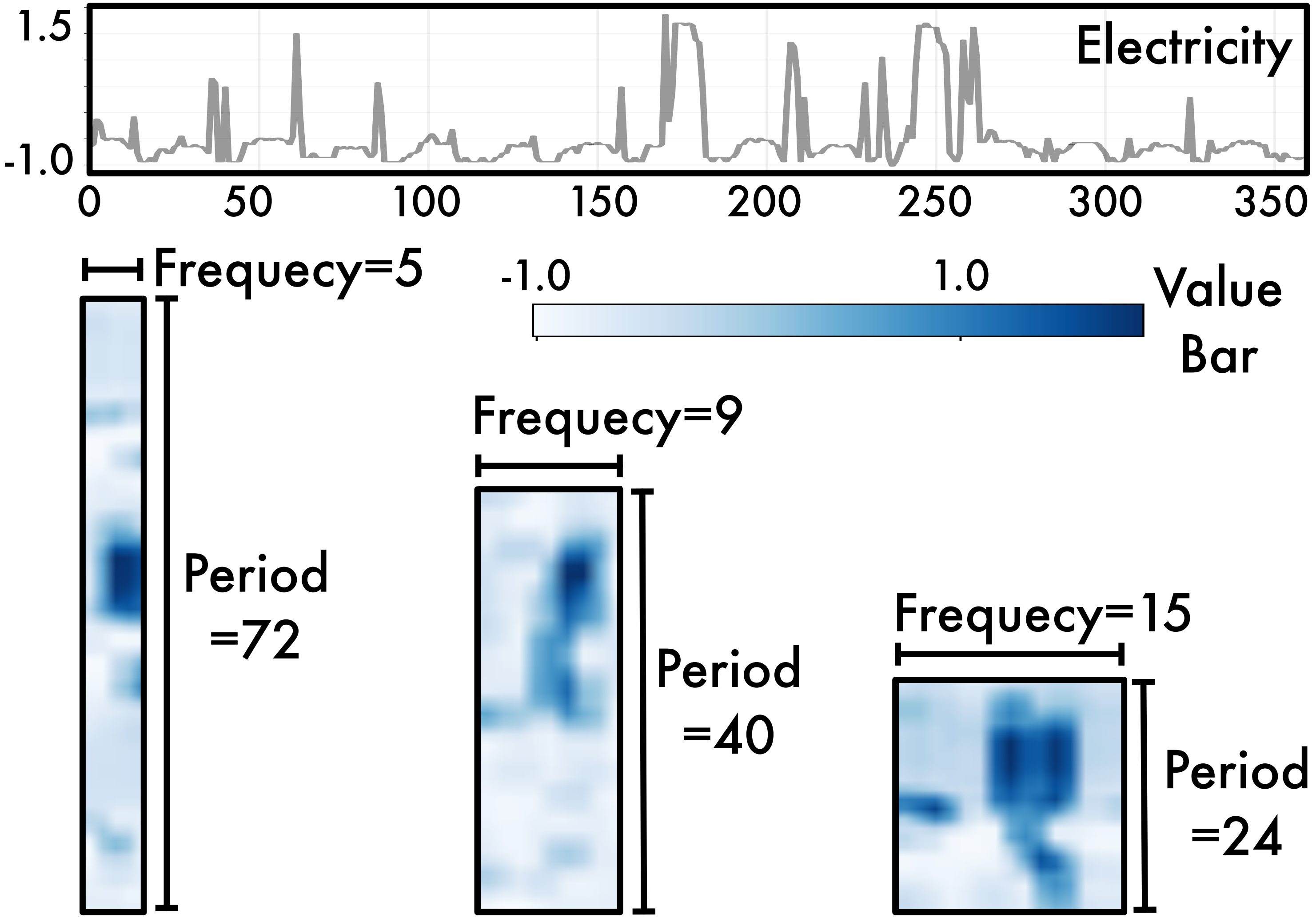


① Multi-periodicity ② Temporal 2D-variation

Unify intraperiod- and interperiod-variations in 2D space by reshape

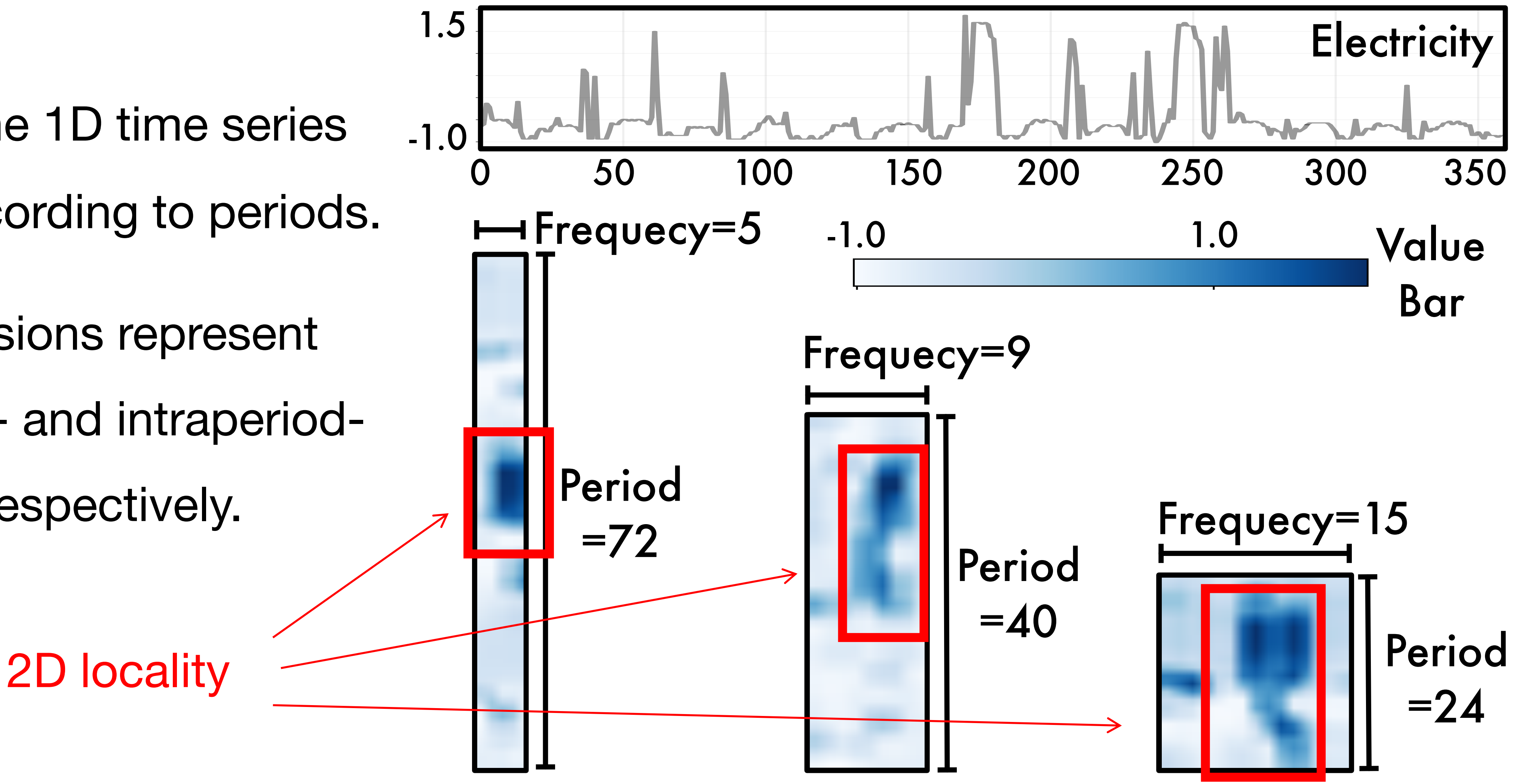
Temporal 2D-Variation: A Case Study

- ✓ Reshape the 1D time series into 2D according to periods.
- ✓ Two dimensions represent interperiod- and intraperiod-variations respectively.



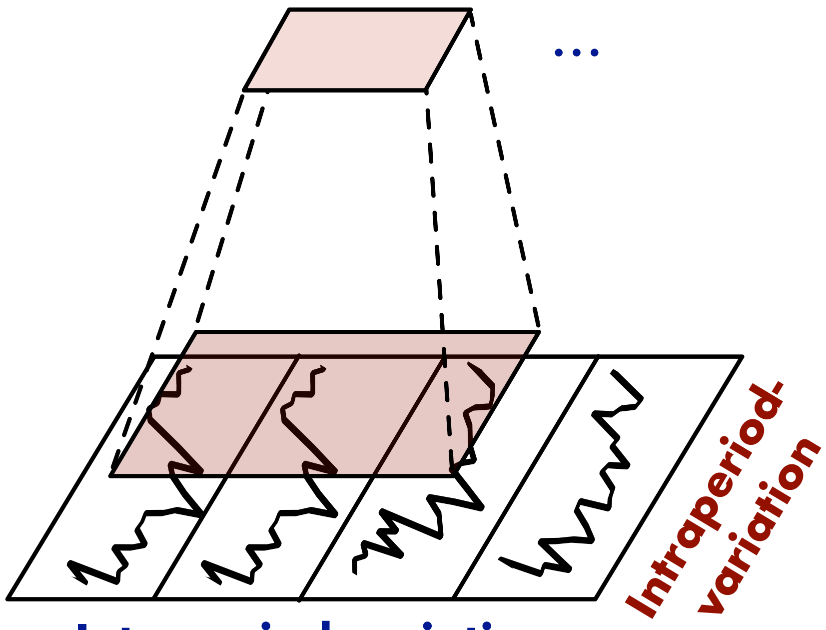
Temporal 2D-Variation: A Case Study

- ✓ Reshape the 1D time series into 2D according to periods.
- ✓ Two dimensions represent interperiod- and intraperiod-variations respectively.



Temporal 2D-Variation: A Case Study

Capture Temporal 2D-variations
by 2D Kernels

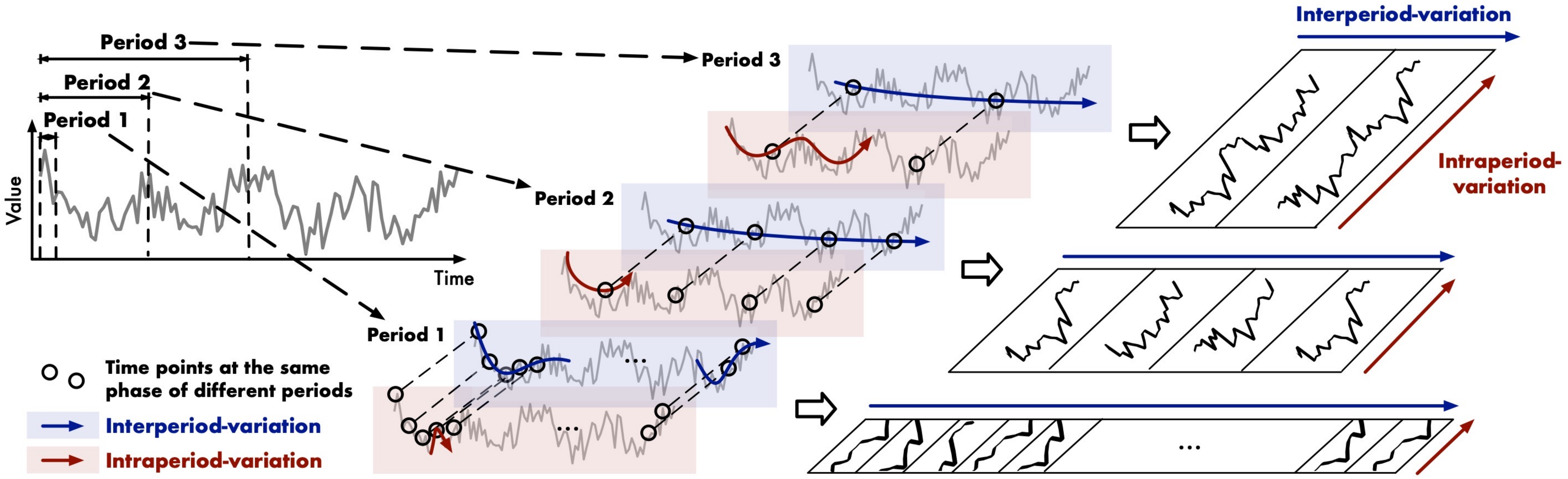


With temporal 2D-variations, we can

✓ Unify intraperiod- interperiod-variations

✓ Learn representations by 2D kernels

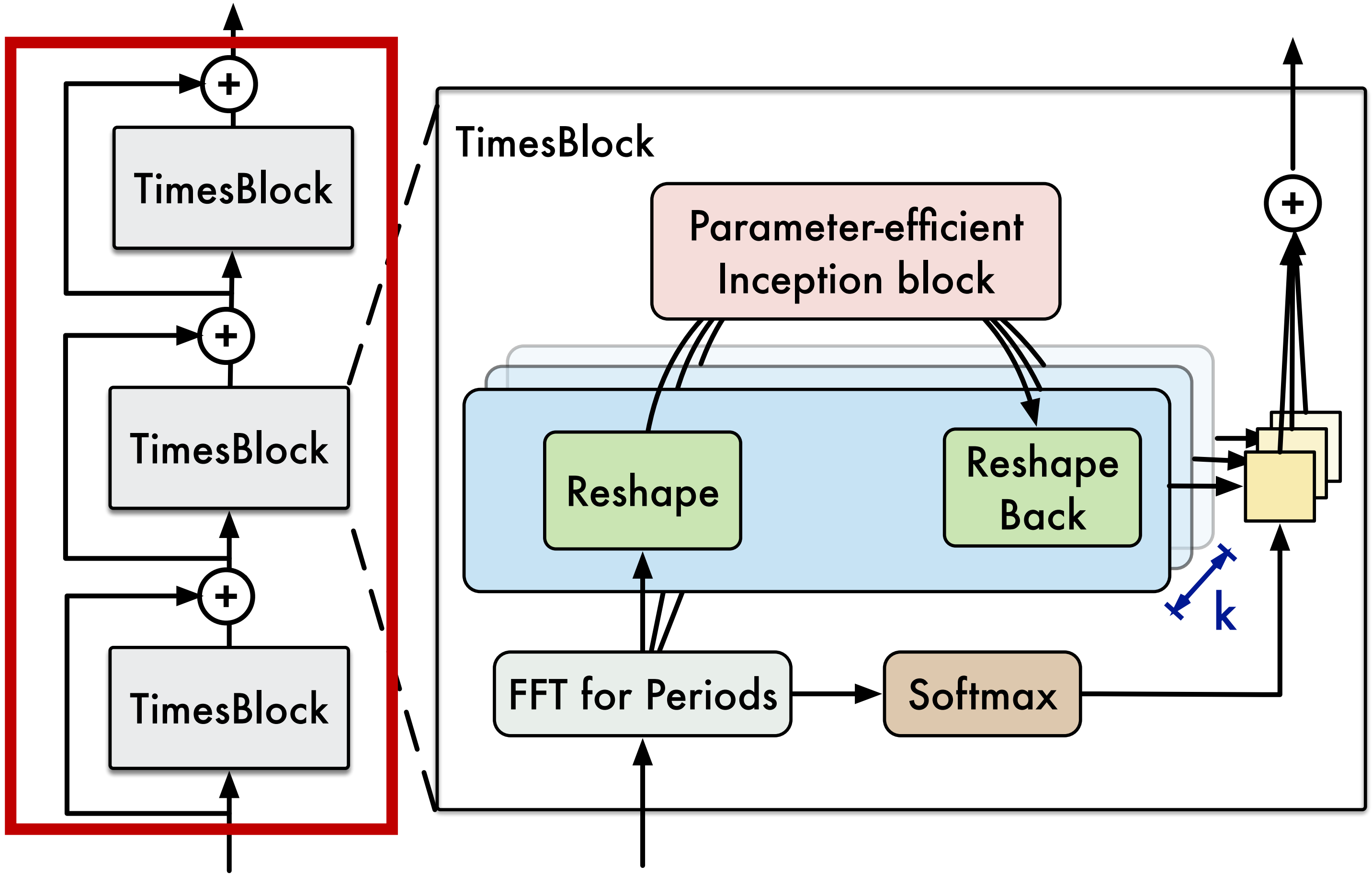
TimesNet: Overall Design



① Multi-periodicity ② Temporal 2D-variation

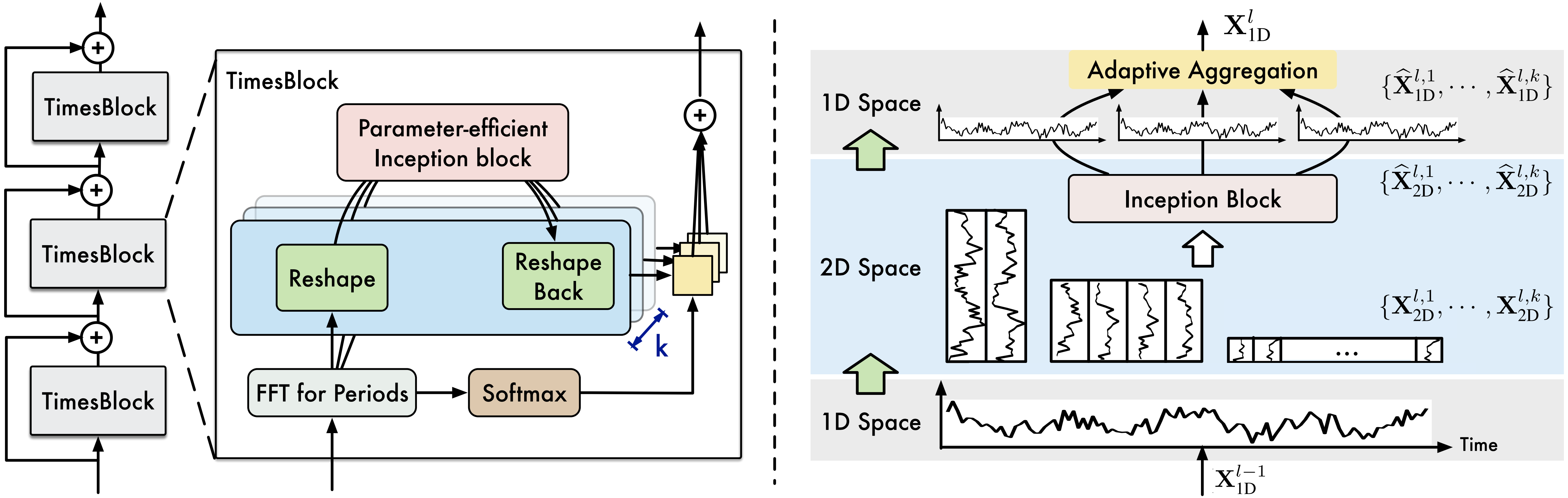
Unify intraperiod- and interperiod-variations in 2D

TimesNet



TimesNet consists of residual-connected **TimesBlocks**.

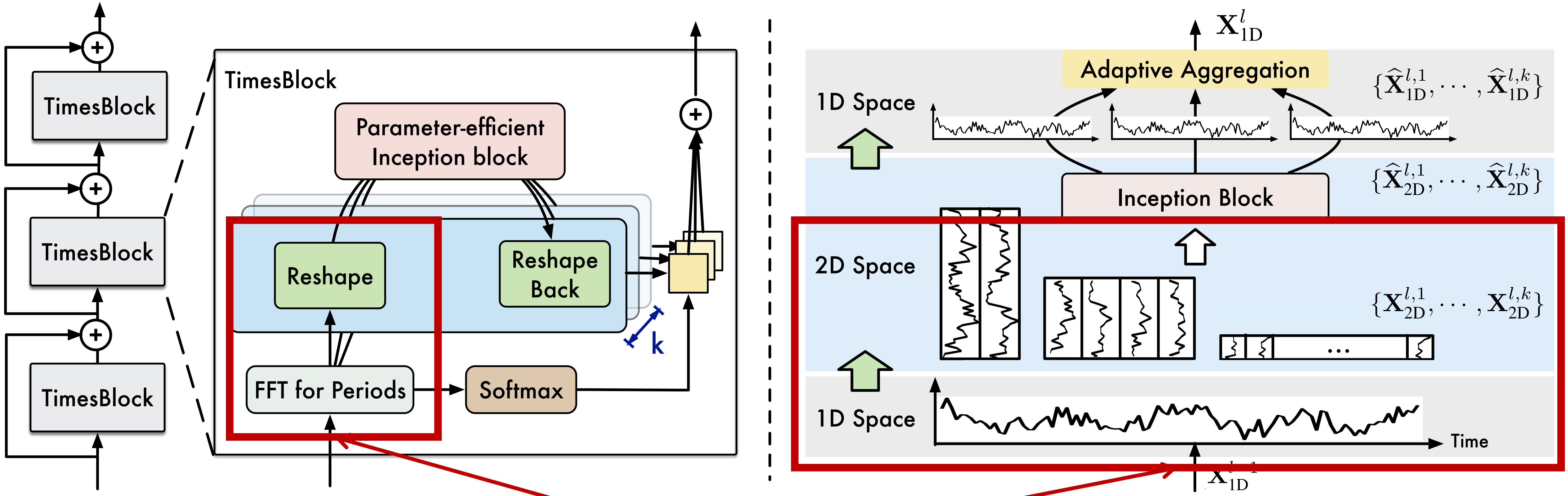
TimesBlock



TimesBlock learns representations in 2D space.

① 1D → 2D ② 2D representation learning ③ 2D → 1D

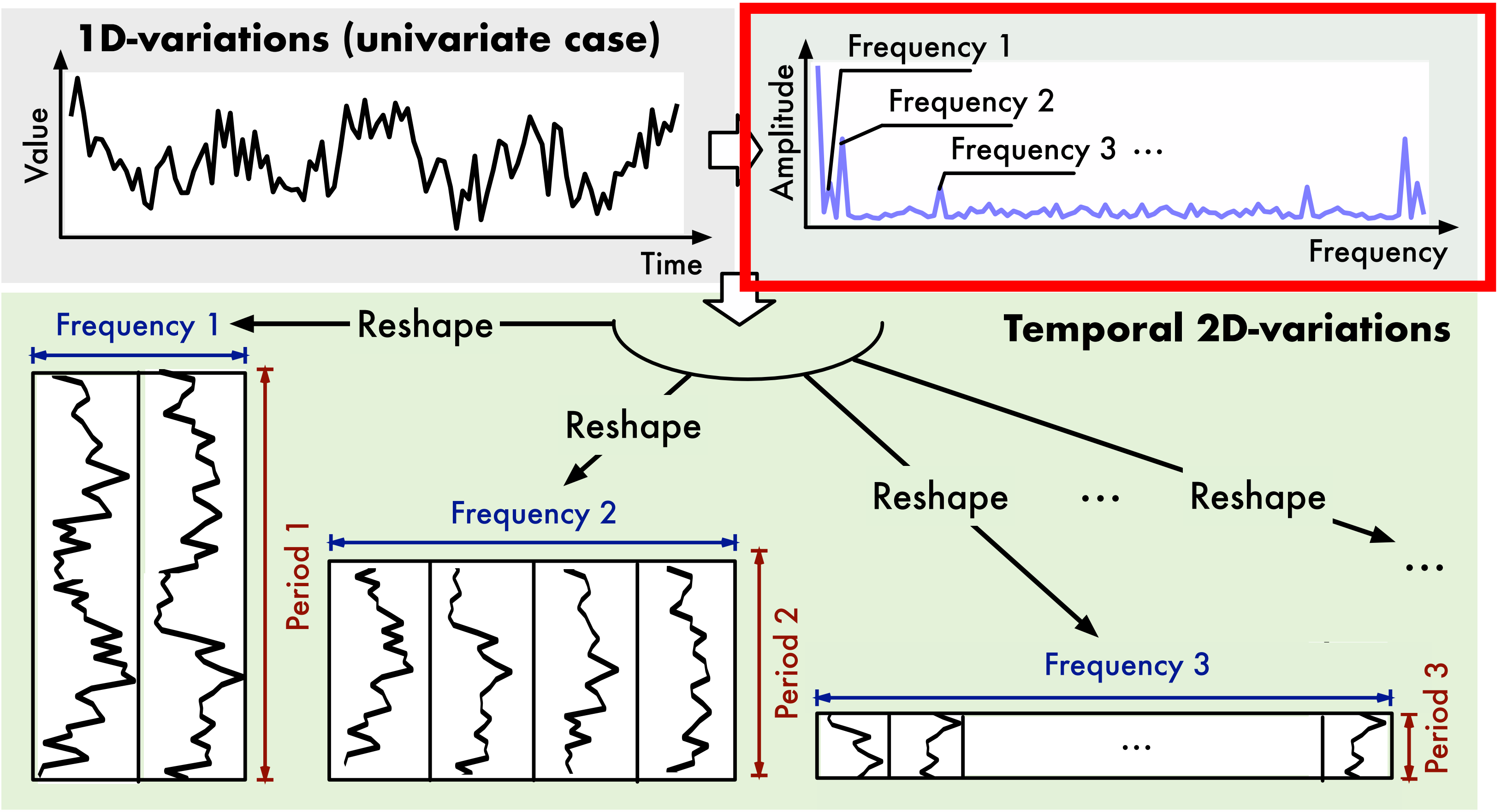
TimesBlock



① 1D→2D

Obtain temporal 2D-variations

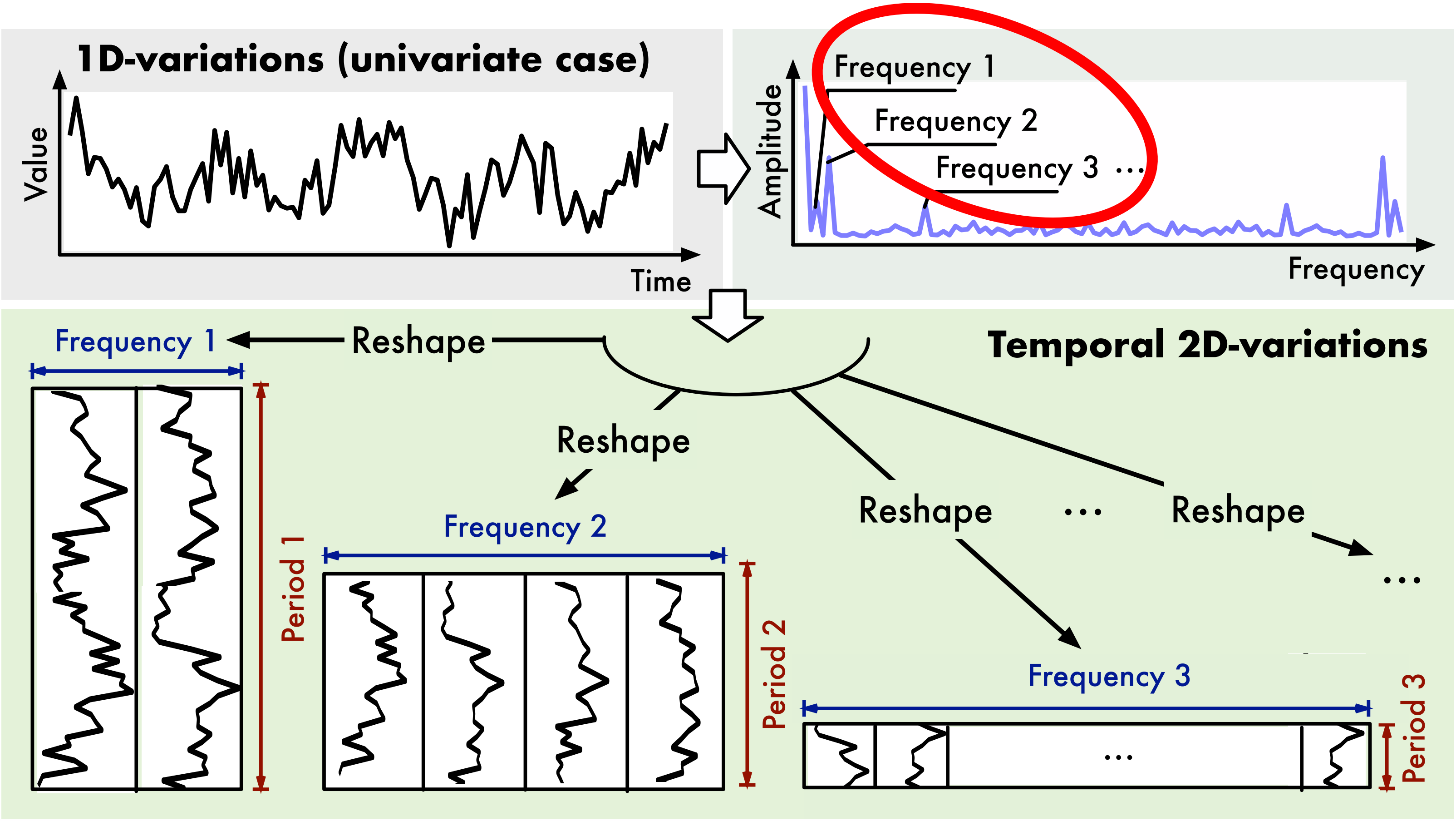
1D → 2D



1. Calculate the spectrum by Fast Fourier Transform

$$\mathbf{A} = \mathrm{Avg}\left(\mathrm{Amp}\left(\mathrm{FFT}(\mathbf{X}_{1D})\right)\right)$$

1D → 2D

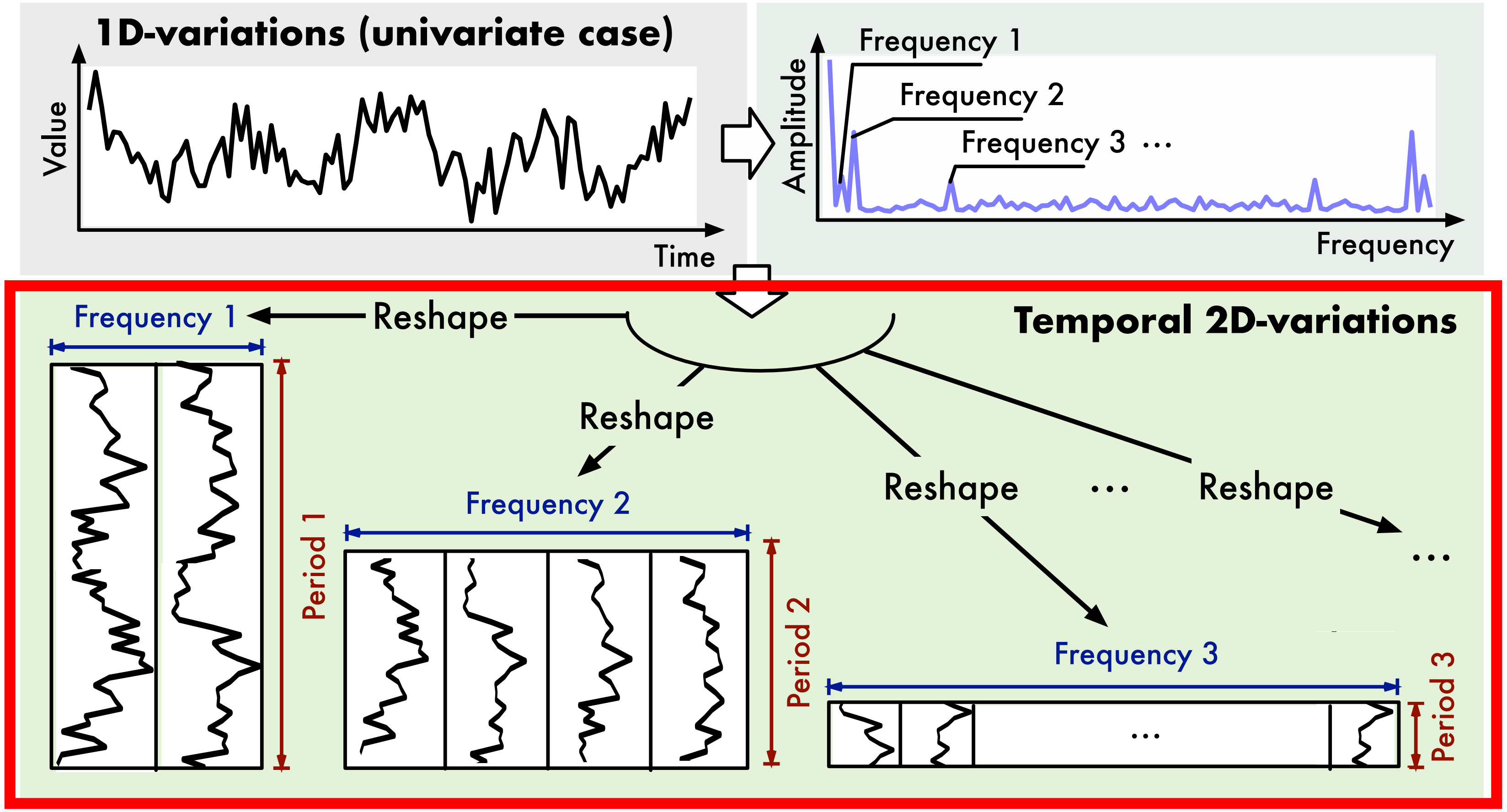


1. Calculate the spectrum by Fast Fourier Transform

2. Choose Topk Frequency

$$\mathbf{A} = \mathrm{Avg}\left(\mathrm{Amp}\left(\mathrm{FFT}(\mathbf{X}_{1\mathrm{D}})\right)\right), \{f_1, \cdots, f_k\} = \underset{f_* \in \{1, \cdots, [\frac{T}{2}]\}}{\arg \mathrm{Topk}} (\mathbf{A}), p_i = \left\lceil \frac{T}{f_i} \right\rceil, i \in \{1, \cdots, k\}.$$

1D → 2D

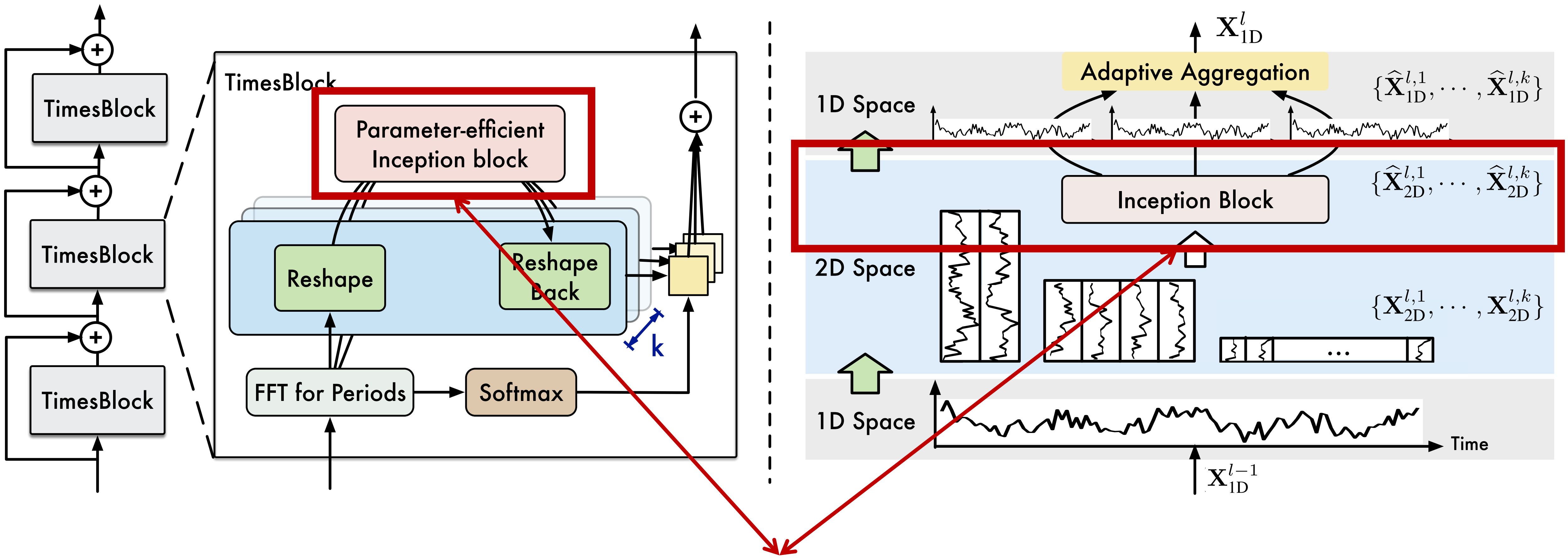


1. Calculate the spectrum by Fast Fourier Transform

2. Choose Topk Frequency

3. For each frequency, reshape 1D time series into 2D tensor

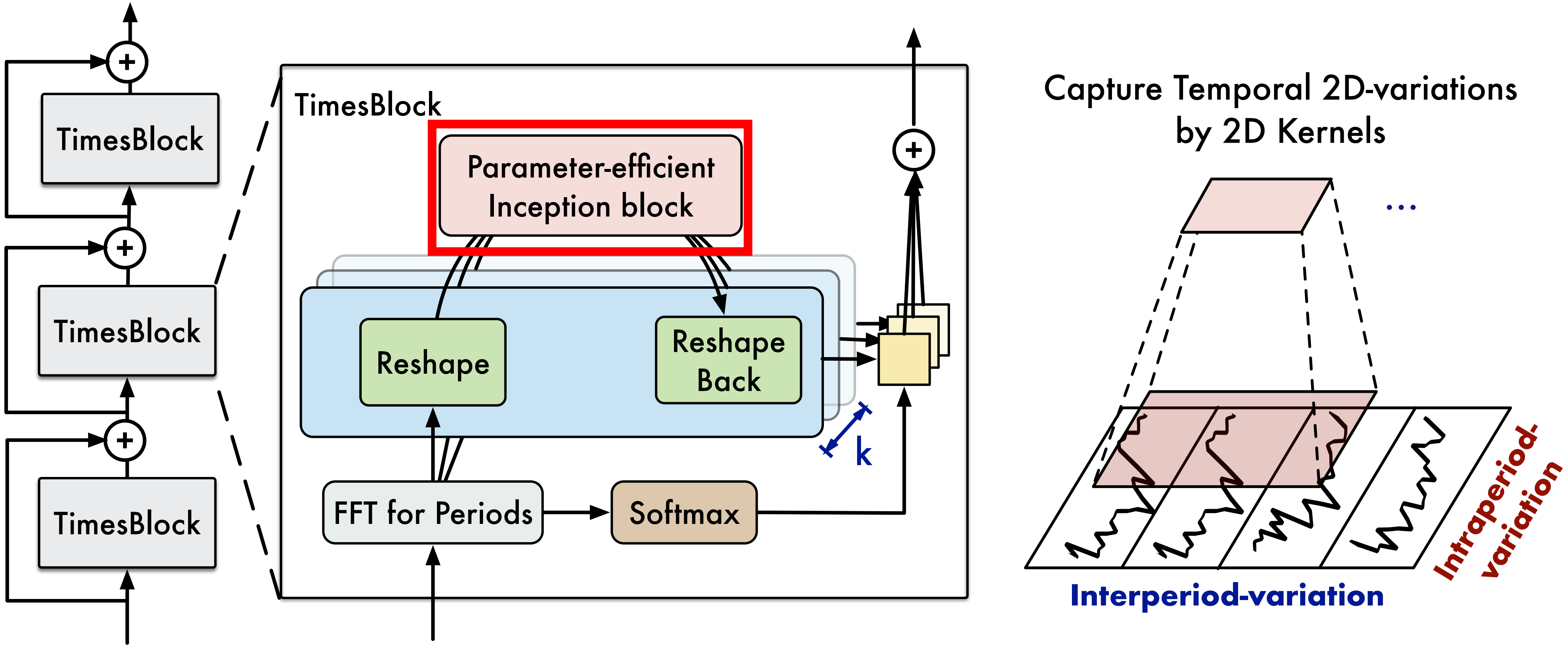
TimesBlock



② 2D representation learning

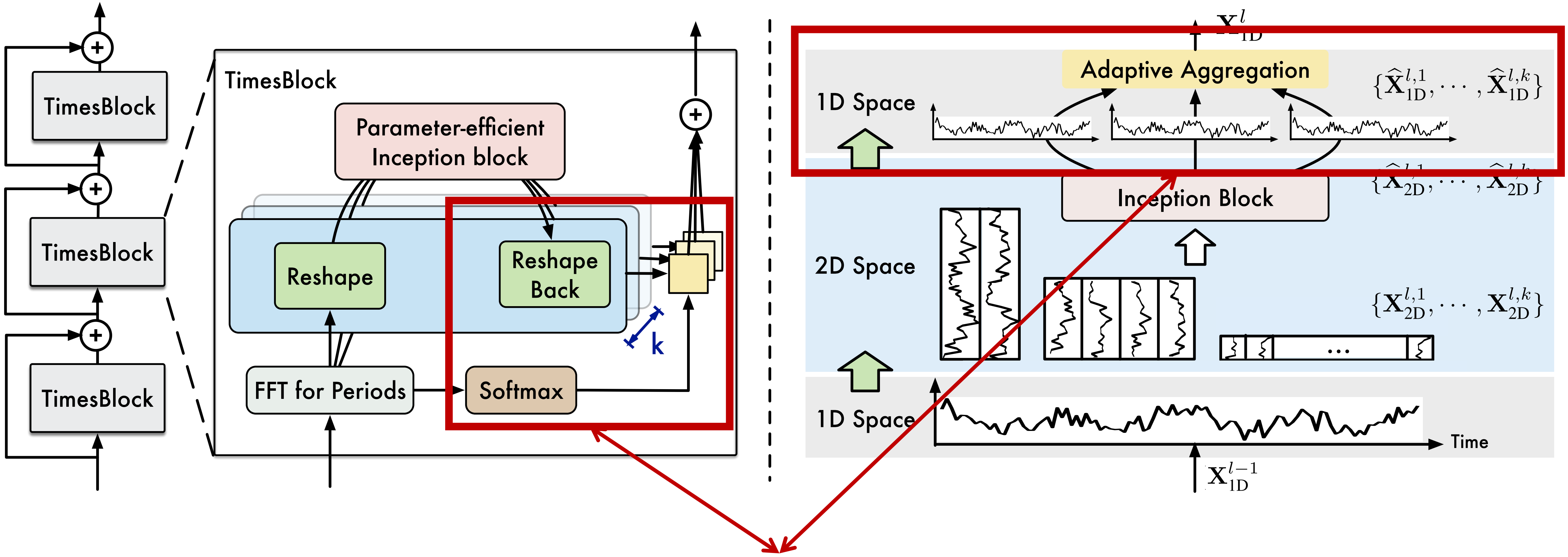
Extract temporal 2D-variations by 2D kernels

2D Representation Learning



- ✓ Inception block is shared in all selected periods for parameter efficiency.
- ✓ It can be replaced by any vision backbones, bridging time series and CV.

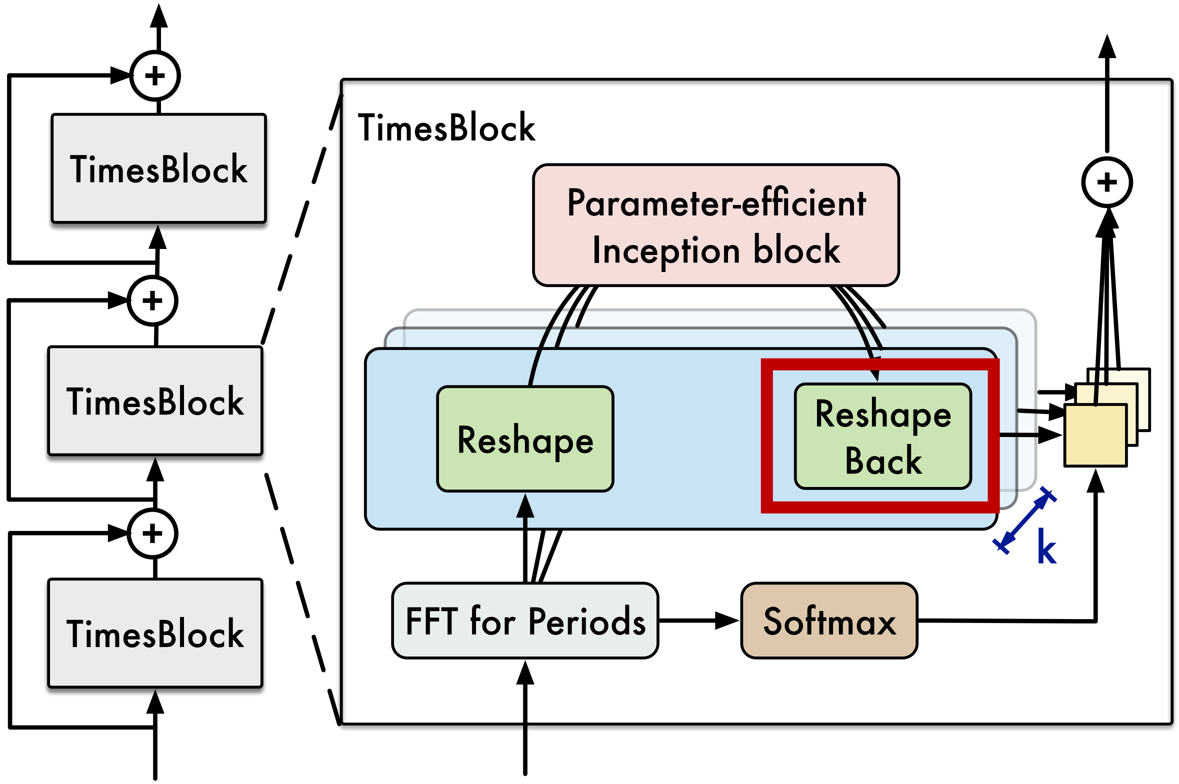
TimesBlock



③ 2D → 1D

Reshape to 1D space to aggregation

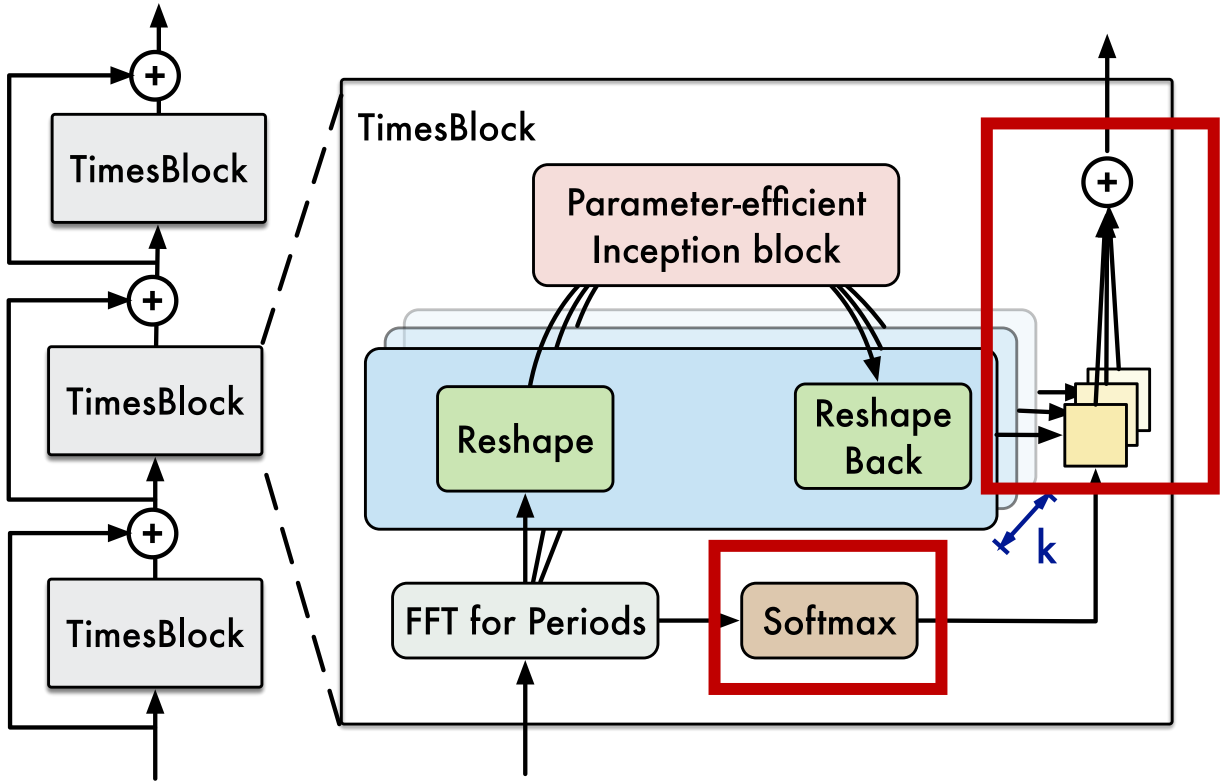
1D → 2D



✓ Reshape back to 1D space for k different representations:

$$\{\widehat{\mathbf{X}}^{l,1}_{\mathrm{1D}}, \cdots, \widehat{\mathbf{X}}^{l,k}_{\mathrm{1D}}\}$$

1D → 2D



✓ Reshape back to 1D space for k different representations.

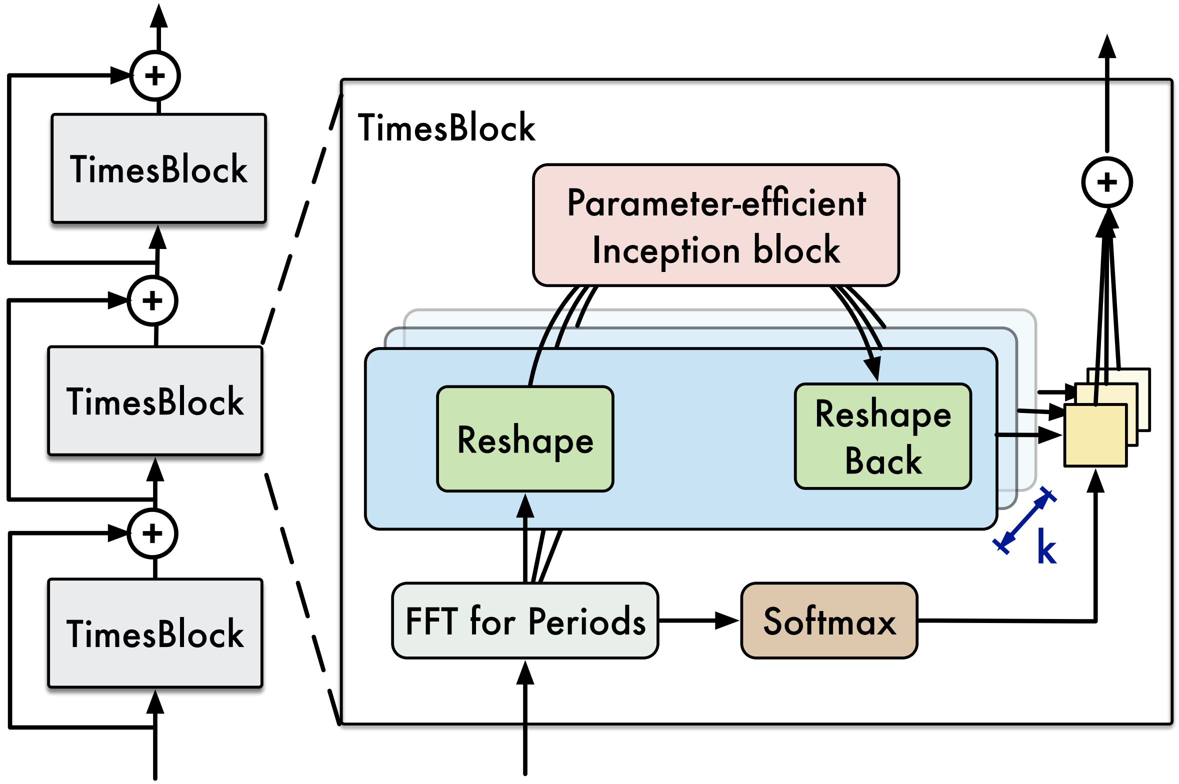
$$\{\widehat{\mathbf{X}}^{l,1}_{\text{1D}},\cdots,\widehat{\mathbf{X}}^{l,k}_{\text{1D}}\}$$

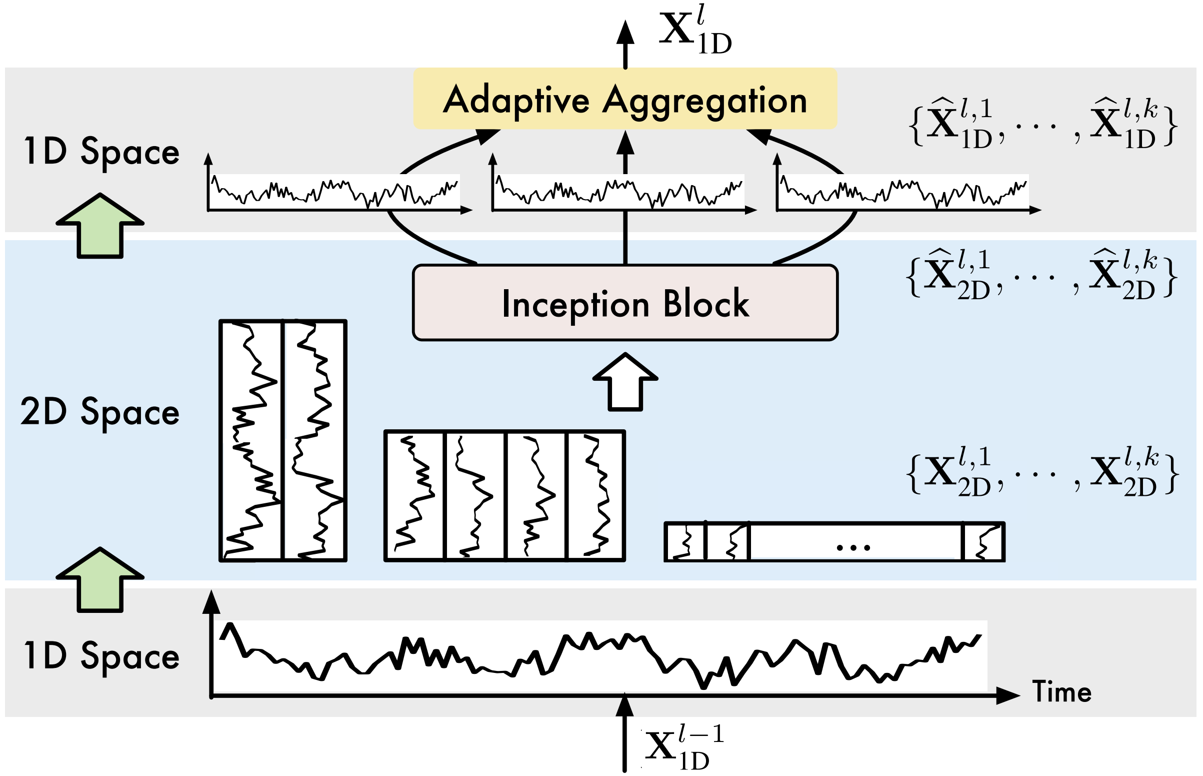
✓ Aggregate k different representations based on the periods' amplitudes $\mathbf{A}$:

$$\widehat{\mathbf{A}}^{l-1}_{f_1},\cdots,\widehat{\mathbf{A}}^{l-1}_{f_k}=\mathrm{Softmax}\left(\mathbf{A}^{l-1}_{f_1},\cdots,\mathbf{A}^{l-1}_{f_k}\right)$$

$$\mathbf{X}^{l}_{\text{1D}}=\sum_{i=1}^{k}\widehat{\mathbf{A}}^{l-1}_{f_i}\times\widehat{\mathbf{X}}^{l,i}_{\text{1D}}.$$

TimesBlock





TimesBlock learns representations in 2D space.

① 1D → 2D ② 2D representation learning ③ 2D → 1D

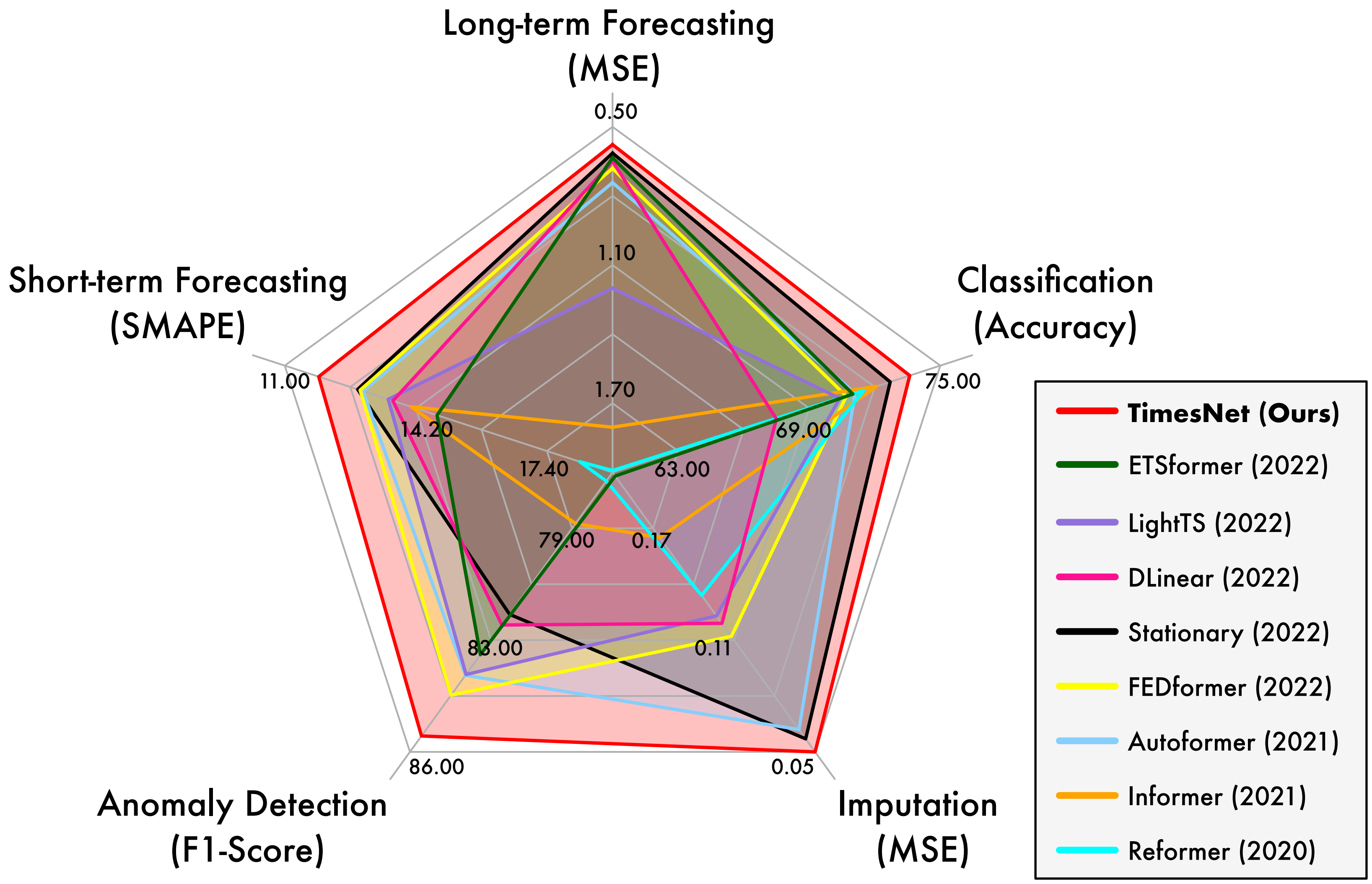
Experiment: Overview

Tasks	Benchmarks
Forecasting	**Long-term**: ETT (4 subsets), Electricity, Traffic, Weather, Exchange, ILI
	Short-term: M4 (6 subsets)
Imputation	ETT (4 subsets), Electricity, Weather
Classification	UEA (10 subsets)
Anomaly Detection	SMD, MSL, SMAP, SWaT, PSM

✓ Five mainstream time series analysis tasks.

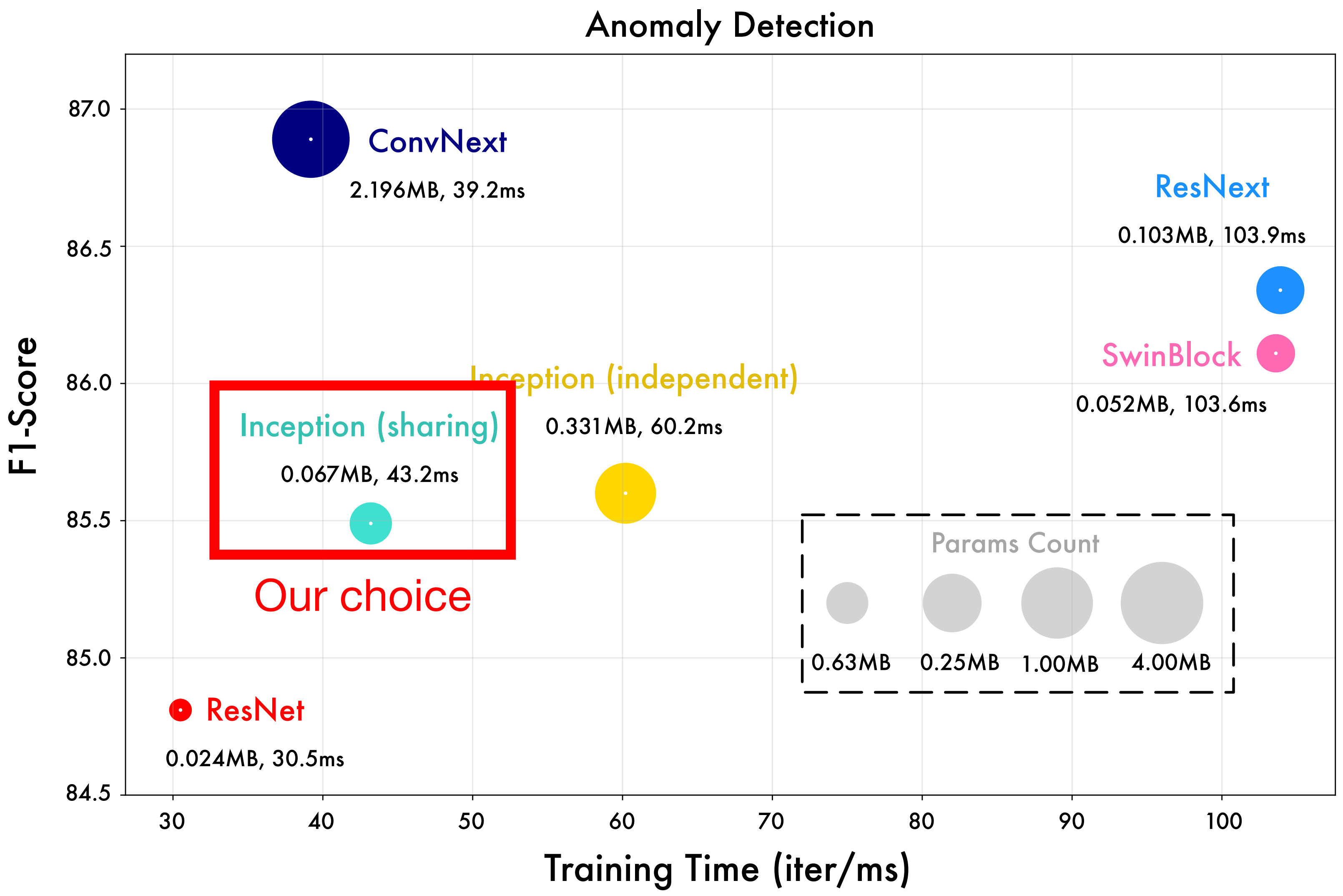
✓ 36 datasets, 81 settings, 20+ baselines

Experiment: Overview



TimesNet achieves state-of-the-art in all five tasks!

Experiment: Model Generality



Better vision backbones,

Better performance 🏆

Bridge Time Series and

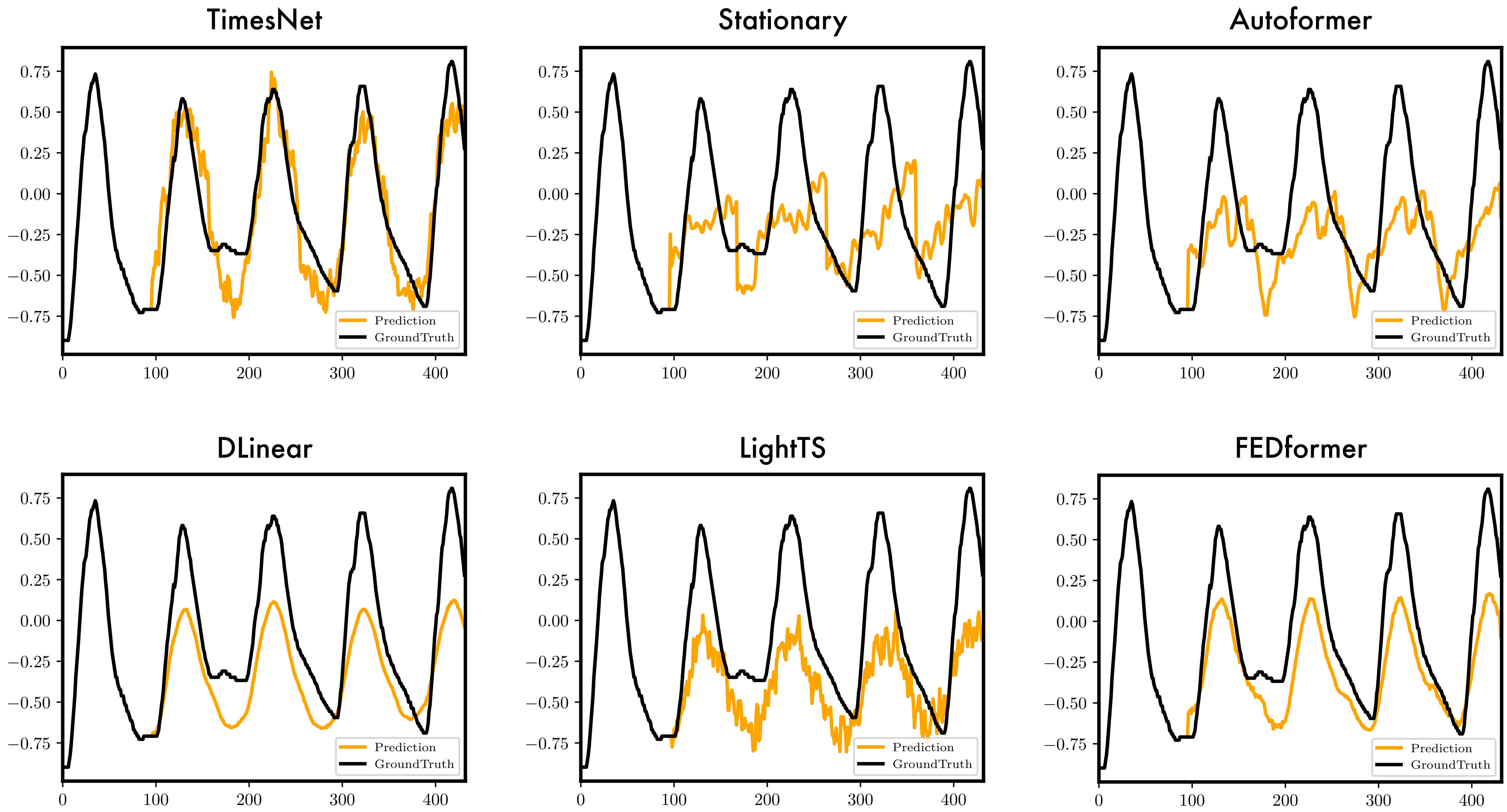
vision backbones 🏆

Experiment: Long-term Forecasting

Models	TimesNet (Ours)		ETSformer (2022)		LightTS (2022)		DLinear (2023)		FEDformer (2022)		Stationary (2022a)		Autoformer (2021)		Pyraformer (2021a)		Informer (2021)		LogTrans (2019)		Reformer (2020)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.400**	**0.406**	0.429	0.425	0.435	0.437	0.403	0.407	0.448	0.452	0.481	0.456	0.588	0.517	0.691	0.607	0.961	0.734	0.929	0.725	0.799	0.671
ETTm2	**0.291**	**0.333**	0.293	0.342	0.409	0.436	0.350	0.401	0.305	0.349	0.306	0.347	0.327	0.371	1.498	0.869	1.410	0.810	1.535	0.900	1.479	0.915
ETTh1	0.458	**0.450**	0.542	0.510	0.491	0.479	0.456	0.452	**0.440**	0.460	0.570	0.537	0.496	0.487	0.827	0.703	1.040	0.795	1.072	0.837	1.029	0.805
ETTh2	**0.414**	**0.427**	0.439	0.452	0.602	0.543	0.559	0.515	0.437	0.449	0.526	0.516	0.450	0.459	0.826	0.703	4.431	1.729	2.686	1.494	6.736	2.191
Electricity	**0.192**	**0.295**	0.208	0.323	0.229	0.329	0.212	0.300	0.214	0.327	0.193	0.296	0.227	0.338	0.379	0.445	0.311	0.397	0.272	0.370	0.338	0.422
Traffic	0.620	**0.336**	0.621	0.396	0.622	0.392	0.625	0.383	**0.610**	0.376	0.624	0.340	0.628	0.379	0.878	0.469	0.764	0.416	0.705	0.395	0.741	0.422
Weather	**0.259**	**0.287**	0.271	0.334	0.261	0.312	0.265	0.317	0.309	0.360	0.288	0.314	0.338	0.382	0.946	0.717	0.634	0.548	0.696	0.602	0.803	0.656
Exchange	0.416	0.443	0.410	0.427	0.385	0.447	**0.354**	**0.414**	0.519	0.500	0.461	0.454	0.613	0.539	1.913	1.159	1.550	0.998	1.402	0.968	1.280	0.932
ILI	2.139	0.931	2.497	1.004	7.382	2.003	2.616	1.090	2.847	1.144	**2.077**	**0.914**	3.006	1.161	7.635	2.050	5.137	1.544	4.839	1.485	4.724	1.445

TimesNet surpasses advanced Transformer-based and MLP-based models.

Experiment: Long-term Forecasting



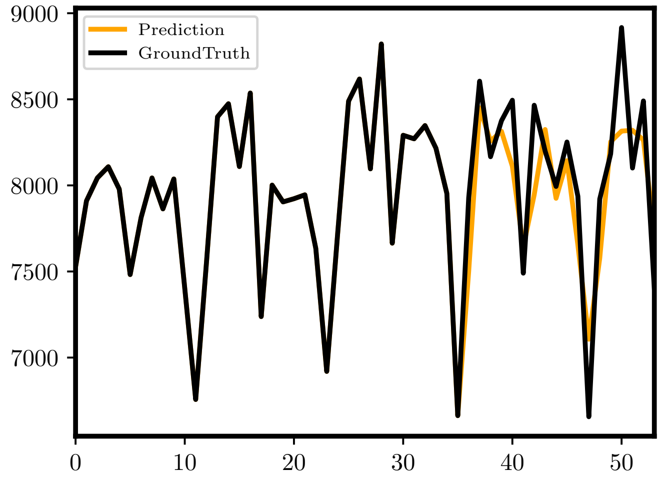
Experiment: Short-term Forecasting

- ✓ More complex temporal patterns: M4 dataset is composed of yearly, monthly, weekly, daily, hourly and quarterly collected univariate marketing data.
- ✓ TimesNet surpasses N-HiTs and N-BEATS.
- ✓ Simple Linear methods degenerate a lot.

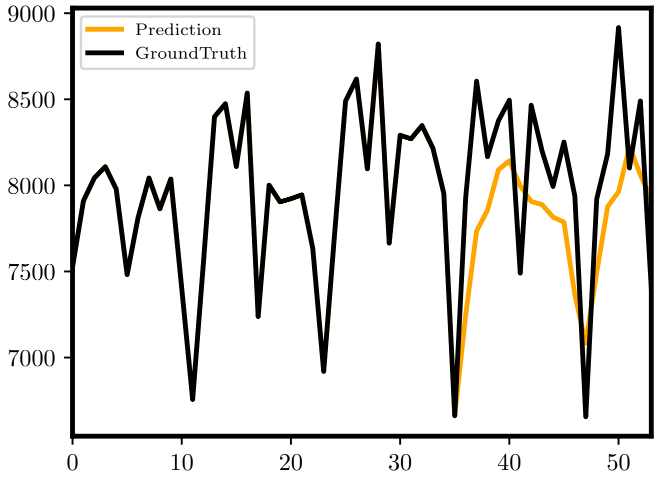
Models	**TimesNet (Ours)**	N-HiTS (2022)	N-BEATS (2019)	ETSformer (2022)	LightTS (2022)	DLinear (2023)	FEDformer (2022)	Stationary (2022a)	Autoformer (2021)	Pyraformer (2021a)	Informer (2021)	LogTrans (2019)	Reformer (2020)
SMAPE	**11.829**	11.927	11.851	14.718	13.525	13.639	12.840	12.780	12.909	16.987	14.086	16.018	18.200
MASE	**1.585**	1.613	1.599	2.408	2.111	2.095	1.701	1.756	1.771	3.265	2.718	3.010	4.223
OWA	**0.851**	0.861	0.855	1.172	1.051	1.051	0.918	0.930	0.939	1.480	1.230	1.378	1.775

Experiment: Short-term Forecasting

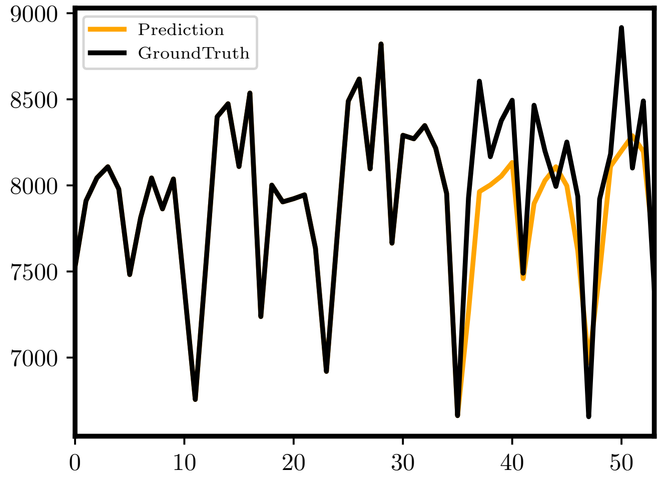
TimesNet



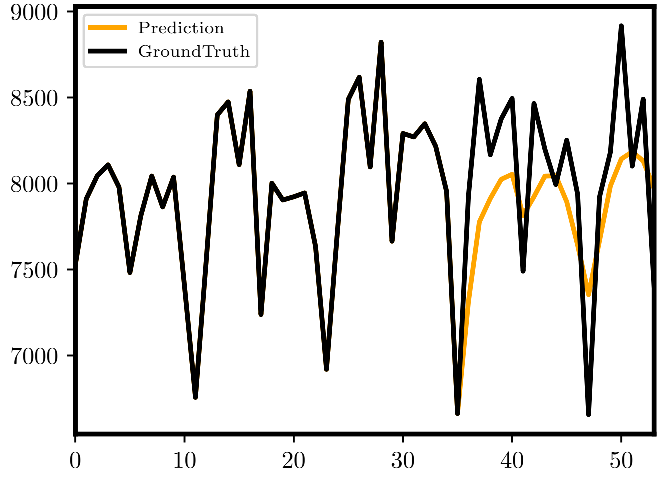
N-HiTS



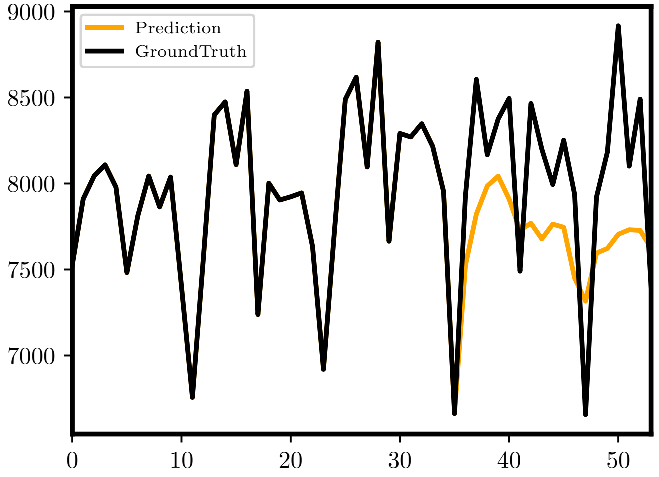
N-BEATS



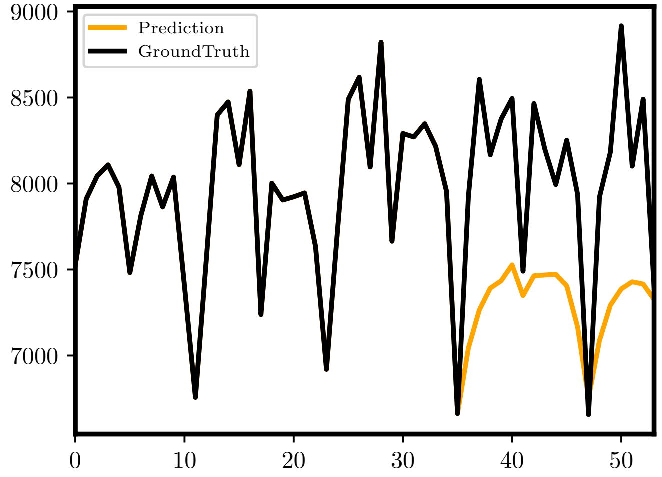
DLinear



Autoformer



FEDformer

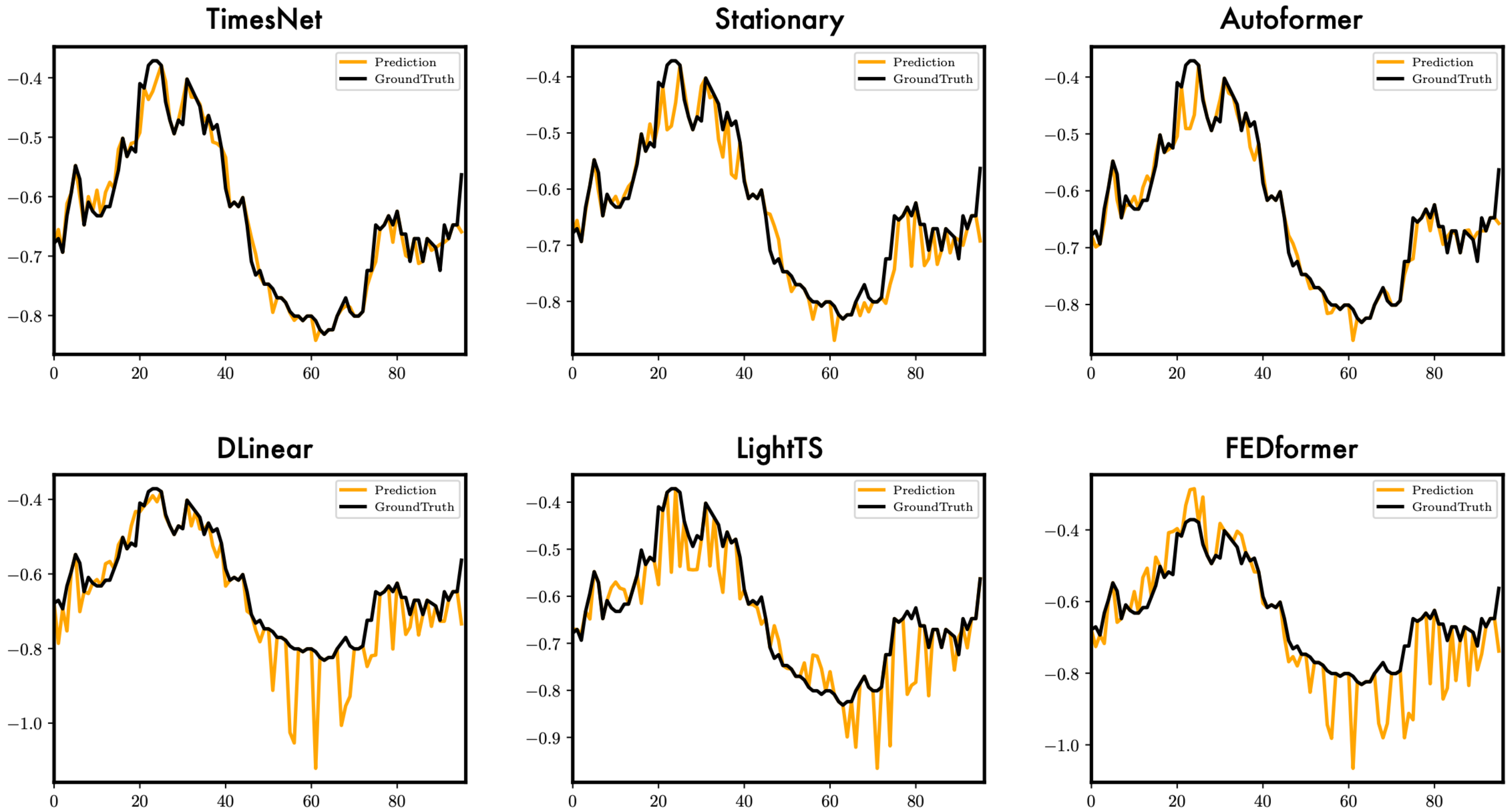


Experiment: Imputation

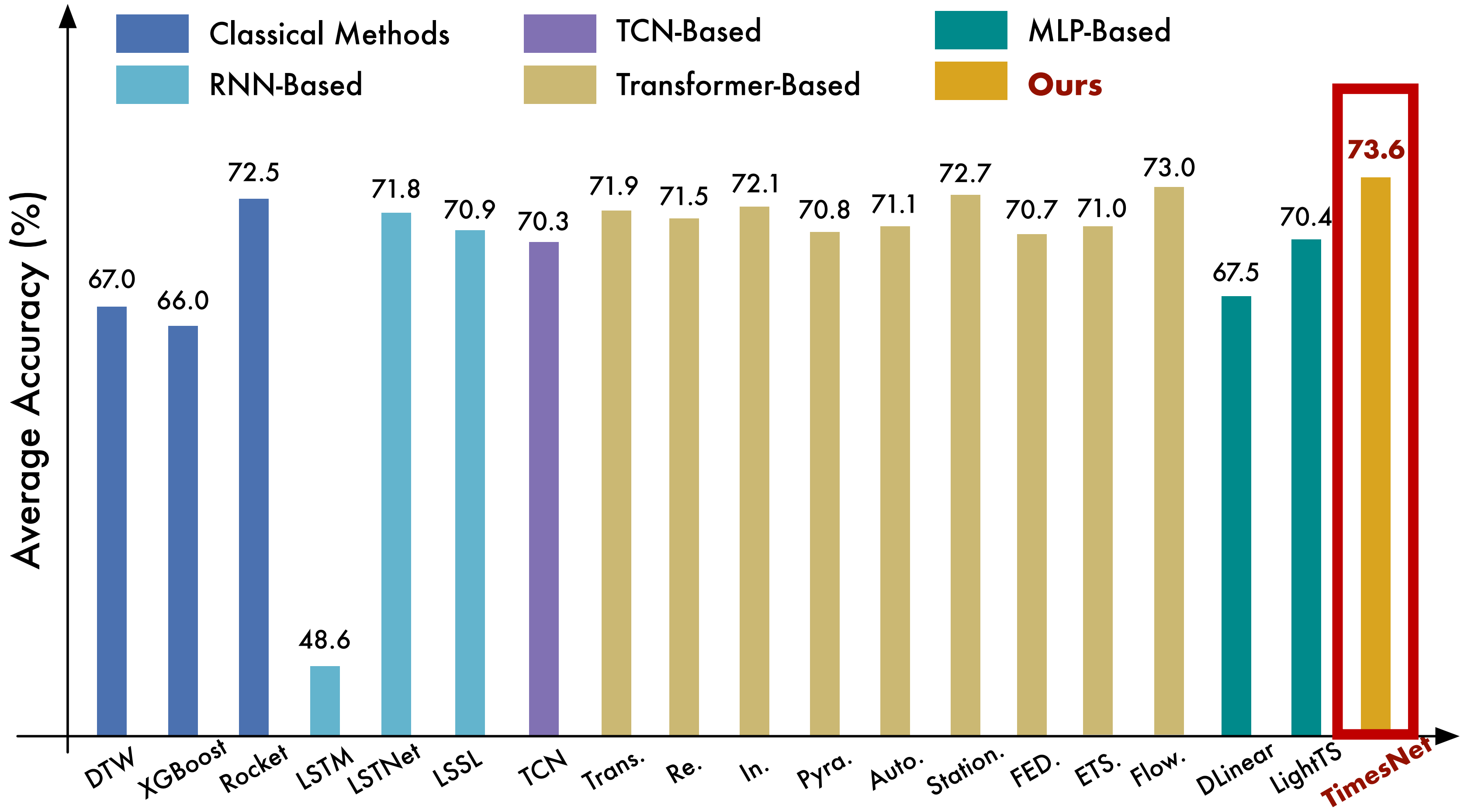
- ✓ Averaged from 4 different mask ratios: 12.5%, 25%, 37.5%, 50%
- ✓ Requires the model to handle irregular inputs.
- ✓ Non-stationary Transformer performs well but MLP-based models fail in this task.

Models	**TimesNet (Ours)**		ETSformer (2022)		LightTS (2022)		DLinear (2023)		FEDformer (2022)		Stationary (2022a)		Autoformer (2021)		Pyraformer (2021a)		Informer (2021)		LogTrans (2019)		Reformer (2020)	
Mask Ratio	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.027**	**0.107**	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150	0.717	0.570	0.071	0.188	0.050	0.154	0.055	0.166
ETTm2	**0.022**	**0.088**	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105	0.465	0.508	0.156	0.292	0.119	0.246	0.157	0.280
ETTh1	**0.078**	**0.187**	0.202	0.329	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214	0.842	0.682	0.161	0.279	0.219	0.332	0.122	0.245
ETTh2	**0.049**	**0.146**	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156	1.079	0.792	0.337	0.452	0.186	0.318	0.234	0.352
Electricity	**0.092**	**0.210**	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225	0.297	0.382	0.222	0.328	0.175	0.303	0.200	0.313
Weather	**0.030**	**0.054**	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057	0.152	0.235	0.045	0.104	0.039	0.076	0.038	0.087

Experiment: Imputation



Experiment: Classification



✓ TimesNet still achieves the best performance.

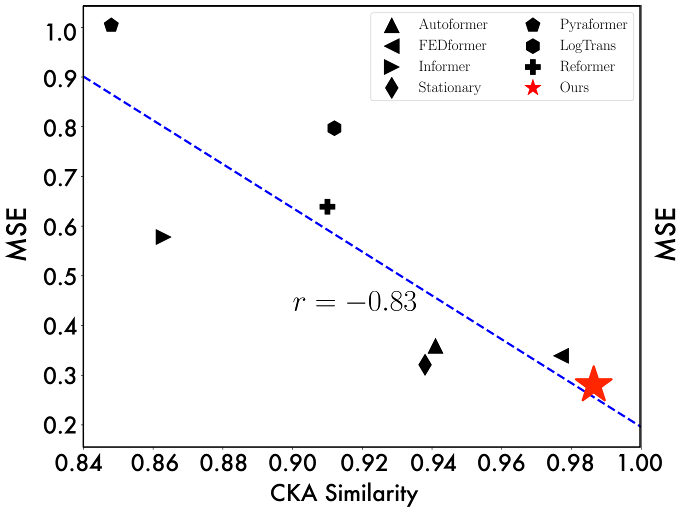
✓ Transformer-based models generally outperform MLP-based models

Experiment: Anomaly Detection

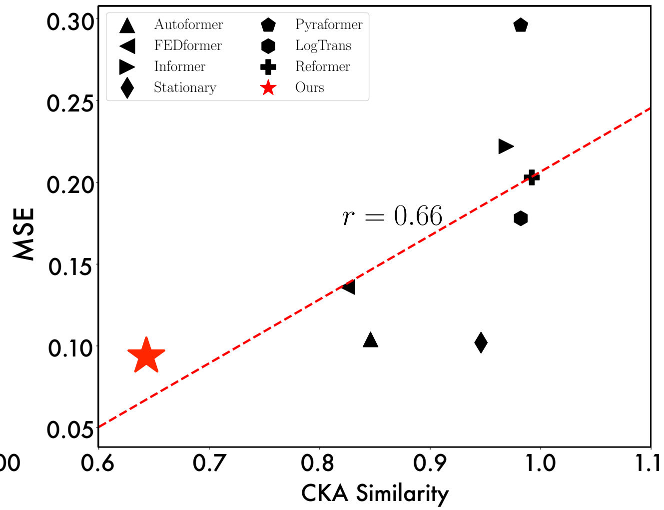
- ✓ Adopt the reconstruction error as the anomaly criterion.
- ✓ Better 2D backbones bring better performances.
- ✓ Transformer-based models performs well.

Models	**TimesNet (ResNeXt)**	**TimesNet (Inception)**	ETS. (2022)	FED. (2022)	LightTS (2022)	DLinear (2023)	Stationary (2022a)	Auto. (2021)	Pyra. (2021a)	Anomaly* (2021)	In. (2021)	Re. (2020)	LogTrans (2019)	Trans. (2017)
SMD	**85.81**	85.12	83.13	85.08	82.53	77.10	84.72	85.11	83.04	85.49	81.65	75.32	76.21	79.56
MSL	**85.15**	84.18	85.03	78.57	78.95	84.88	77.50	79.05	84.86	83.31	84.06	84.40	79.57	78.68
SMAP	**71.52**	70.85	69.50	70.76	69.21	69.26	71.09	71.12	71.09	71.18	69.92	70.40	69.97	69.70
SWaT	91.74	92.10	84.91	93.19	**93.33**	87.52	79.88	92.74	91.78	83.10	81.43	82.80	80.52	80.37
PSM	**97.47**	95.21	91.76	97.23	97.15	93.55	97.29	93.29	82.08	79.40	77.10	73.61	76.74	76.07
Avg F1	**86.34**	85.49	82.87	84.97	84.23	82.46	82.08	84.26	82.57	80.50	78.83	77.31	76.60	76.88

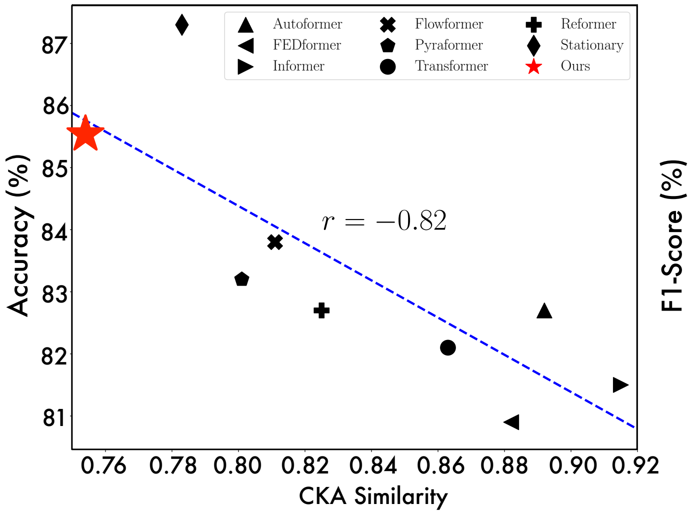
Representation Analysis



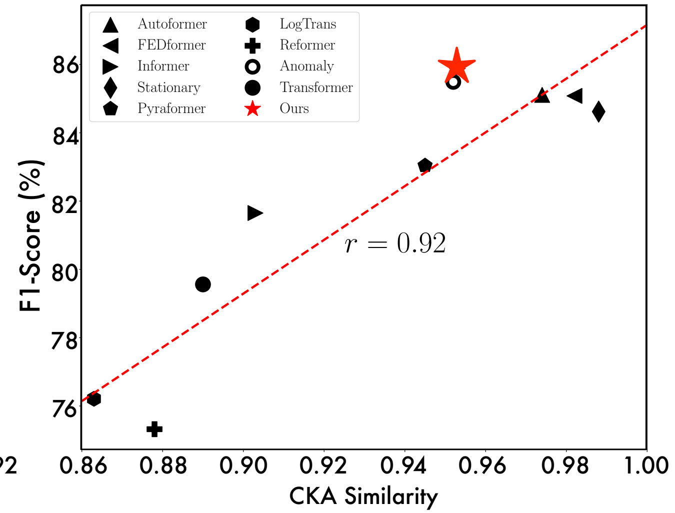
(a) Forecasting (Weather input-96-predict-336)



(b) Imputation (Electricity Mask 37.5%)



(c) Classification (PEMS-SF)



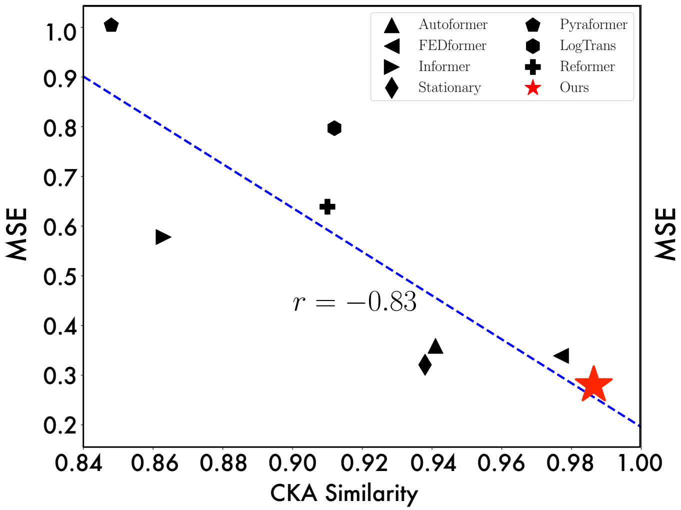
(d) Anomaly Detection (SMD)

Relation between top-bottom layer CKA similarity and performance

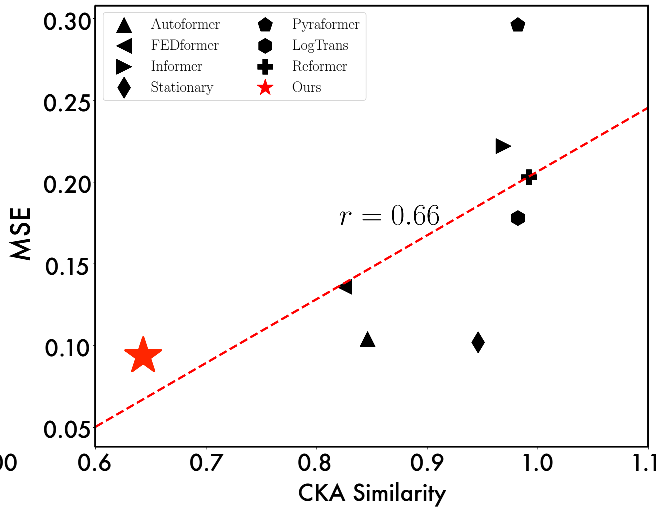
✓ **Why TimesNet achieves SOTA?**

Benefiting from temporal 2D-variations, it can learn proper representations for different tasks.

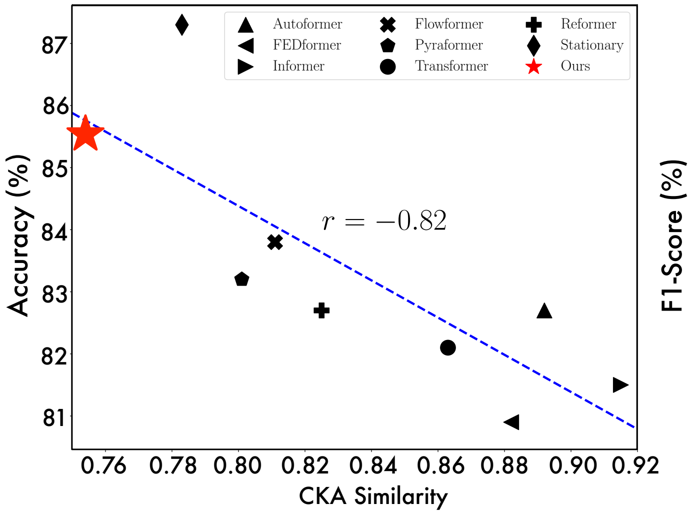
Representation Analysis



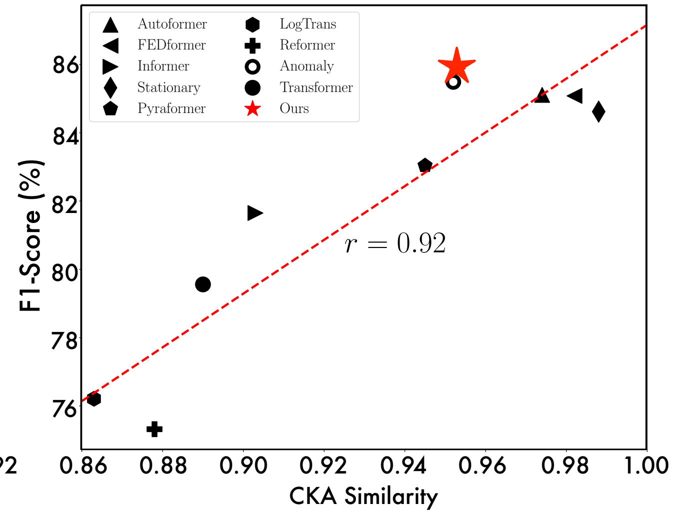
(a) Forecasting (Weather input-96-predict-336)



(b) Imputation (Electricity Mask 37.5%)



(c) Classification (PEMS-SF)



(d) Anomaly Detection (SMD)

Relation between top-bottom layer CKA similarity and performance

✓ **What is the design principle?**

- Classification & imputation need hierarchical representations.

- Anomaly detection & Forecasting expect low-level representations.

Performance Ranking

✓ After comparing with more than 20+ baselines, we get:

Model Ranking	Long-term Forecasting	Short-term Forecasting	Imputation	Anomaly Detection	Classification
🥇 1st	TimesNet	TimesNet	TimesNet	TimesNet	TimesNet
🥈 2nd	DLinear	Non-stationary Transformer	Non-stationary Transformer	Non-stationary Transformer	FEDformer
🥉 3rd	Non-stationary Transformer	FEDformer	Autoformer	Informer	Autoformer

Until 2023.06 (Keep updating)

Efficiency Comparison

Models	Series Length	Parameter (MB)	GPU Memory (MiB)	Running Time (s / iter)	Ranking: Five tasks	Ranking: Avg Ranking
TimesNet (ours)	384	0.067	1245	0.024	(1, 1, 1, 1, 1)	1.0
	768	0.067	1585	0.040		
	1536	0.067	2491	0.045		
	3072	0.067	2353	0.073		
Non-stationary Transformer	384	1.884	2321	0.046	(3, 2, 2, 2, 8)	3.4
	768	1.910	4927	0.118		
	1536	1.961	/	/		
	3072	/	/	/		
Autoformer	384	1.848	2101	0.070	(7, 4, 3, 5, 3)	4.4
	768	1.848	3209	0.071		
	1536	1.848	5395	0.129		
	3072	1.848	10043	0.255		
FEDformer	384	2.901	5977	0.807	(4, 3, 6, 9, 2)	4.8
	768	2.901	7111	1.055		
	1536	2.901	9173	1.482		
	3072	2.901	/	/		

Open Source

wuhaixu2016 Update MICN_ETTh2.sh — 5e2e887 2 weeks ago — **82** commits

data_provider	clean	3 months ago
exp	Improve annotations	2 months ago
layers	fix the model of PatchTST	3 months ago
models	Improve annotations	2 months ago
pic	update dataset discription	4 months ago
scripts	Update MICN_ETTh2.sh	2 weeks ago
utils	Improve annotations	2 months ago
.gitignore	Improve annotations	2 months ago
LICENSE	init	4 months ago
README.md	Update README.md	last month
requirements.txt	clean	3 months ago
run.py	Fix some errors	3 months ago

A Library for Advanced Deep Time Series Models.

deep-learning time-series

Readme

MIT license

918 stars

14 watching

205 forks

Report repository

Releases

No releases published

Packages

No packages published

Code is available at https://github.com/thuml/Time-Series-Library


王建民


龙明盛


吴海旭


刘雍


胡腾戈


周航


徐介晖

大数据系统软件国家工程研究中心

THUML