Learning Deep Time-index Models for Time Series Forecasting

Gerald Woo

Authors: Gerald Woo[1,2], Chenghao Liu[1], Doyen Sahoo[1], Akshat Kumar[2], Steven Hoi[1]

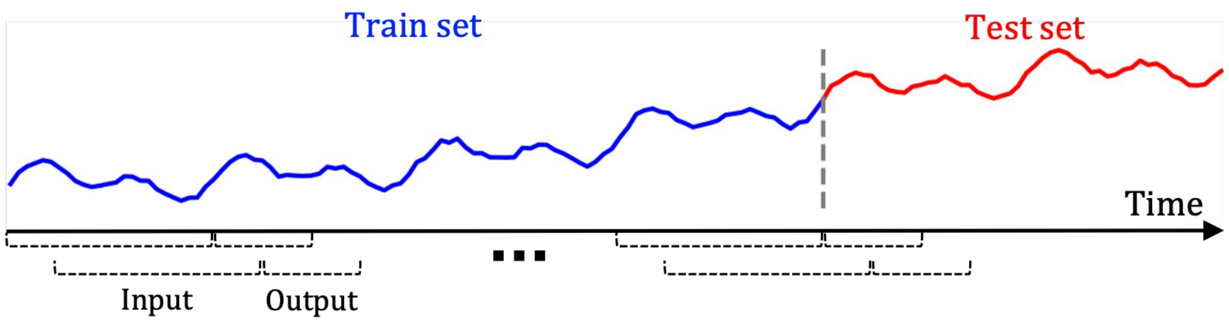
[1]Salesforce Research Asia

[2]School of Computing and Information Systems, Singapore Management University

Approaches to Time Series Forecasting

2 different methods

Historical-value Models



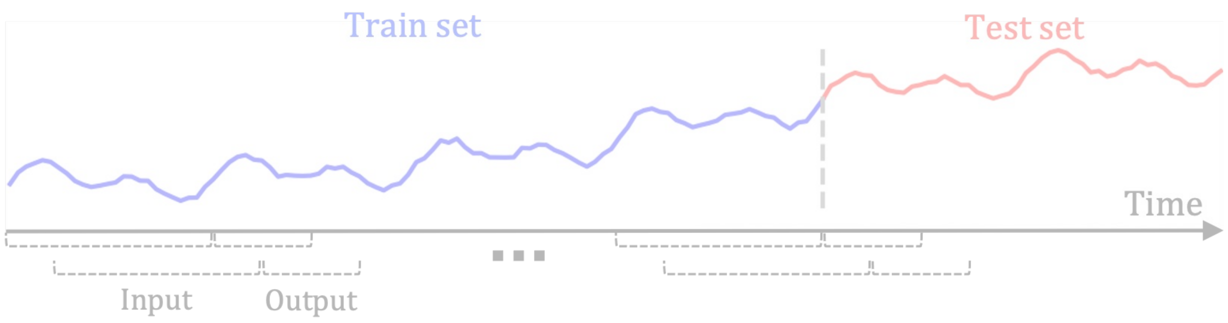
Function of historical data

$$y_t = f(y_{t-1}, y_{t-2}, \ldots) + \varepsilon_t$$

Approaches to Time Series Forecasting

2 different methods

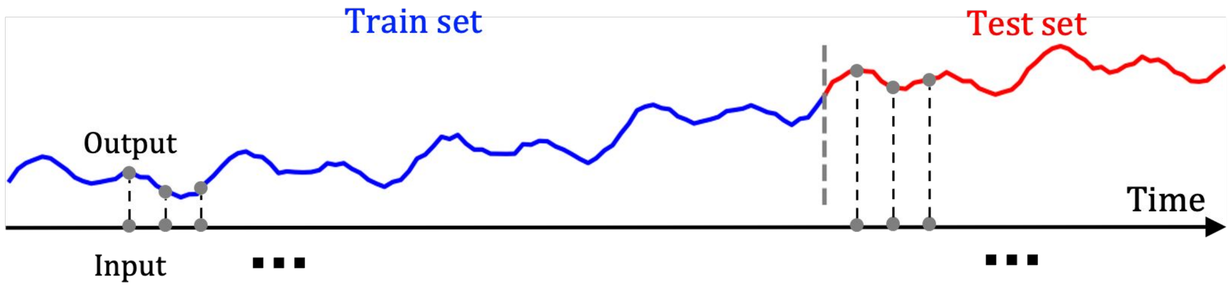
Historical-value Models



Function of historical data

$$y_t = f(y_{t-1}, y_{t-2}, \ldots) + \varepsilon_t$$

Time-index Models



Function of predictor variables

$$y_t = f(x_t) + \varepsilon_t$$

Taxonomy

Classical vs Deep, Historical-value vs Time-index

	Classical	**Deep Learning**
Historical-value	• ARIMA • ETS • …	
Time-index		

Taxonomy

Classical vs Deep, Historical-value vs Time-index

	Classical	Deep Learning
Historical-value	• ARIMA • ETS • …	• DeepAR • N-BEATS • Autoformer • …
Time-index		

Taxonomy

Classical vs Deep, Historical-value vs Time-index

	Classical	**Deep Learning**
Historical-value	• ARIMA • ETS • …	• DeepAR • N-BEATS • Autoformer • …
Time-index	• Prophet • Gaussian Processes • …	

Taxonomy

Classical vs Deep, Historical-value vs Time-index

	Classical	**Deep Learning**
Historical-value	• ARIMA • ETS • …	• DeepAR • N-BEATS • Autoformer • …
Time-index	• Prophet • Gaussian Processes • …	**?**

Deep Time-index Models

A deep learning based approach

$$y_t = f(t) + \varepsilon_t$$

- where f is a neural network!
- Learn the appropriate function representation based on the time-index!

Deep Time-index Models

A deep learning based approach

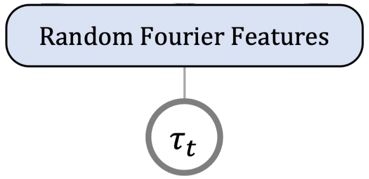
$$y_t = f(t) + \varepsilon_t$$

- where *f* is a neural network!
- Learn the appropriate function representation based on the time-index!

Our proposed instantiation of deep time-index models

random fourier features

$$z^{(0)} = \gamma(\boldsymbol{\tau}) = [\sin(2\pi \boldsymbol{B}\boldsymbol{\tau}), \cos(2\pi \boldsymbol{B}\boldsymbol{\tau})]^T$$



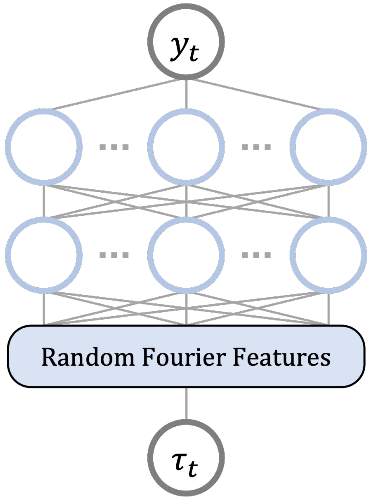
Deep Time-index Models

A deep learning based approach

$$y_t = f(t) + \varepsilon_t$$

- where *f* is a neural network!
- Learn the appropriate function representation based on the time-index!

Our proposed instantiation of deep time-index models



random fourier features

$$\boldsymbol{z}^{(0)} = \gamma(\boldsymbol{\tau}) = [\sin(2\pi \boldsymbol{B}\boldsymbol{\tau}), \cos(2\pi \boldsymbol{B}\boldsymbol{\tau})]^T$$

multi-layered perceptron

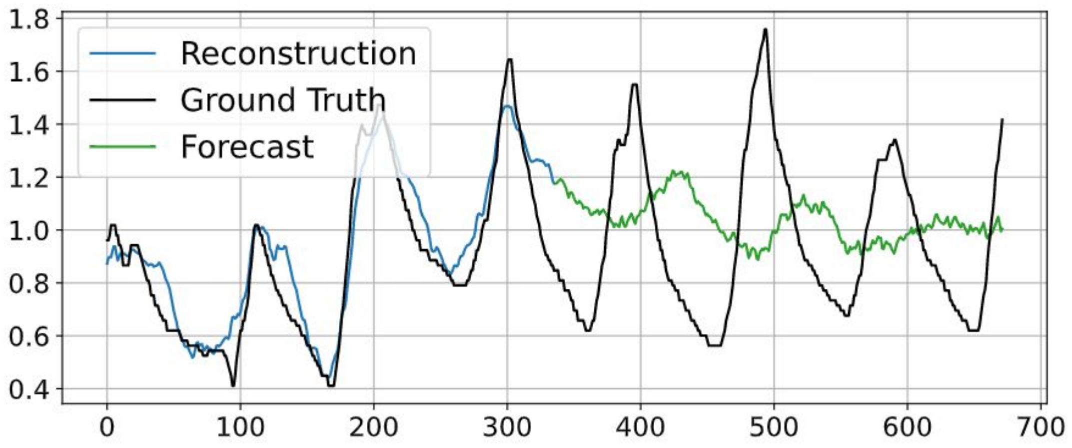
$$\begin{cases} \boldsymbol{z}^{(k+1)} = \max(0, \boldsymbol{W}^{(k)}\boldsymbol{z}^{(k)} + \boldsymbol{b}^{(k)}), \quad k = 0, \ldots, K-1 \\ f_\theta(\boldsymbol{\tau}) = \boldsymbol{W}^{(K)}\boldsymbol{z}^{(K)} + \boldsymbol{b}^{(K)} \end{cases}$$

Deep Time-index Models

Pitfalls of deep time-index models

Pitfalls

- No inductive biases (such as linear trend, periodicity) unlike classical time-index models
- How to extrapolate across forecast horizon (generalize to future time steps)?



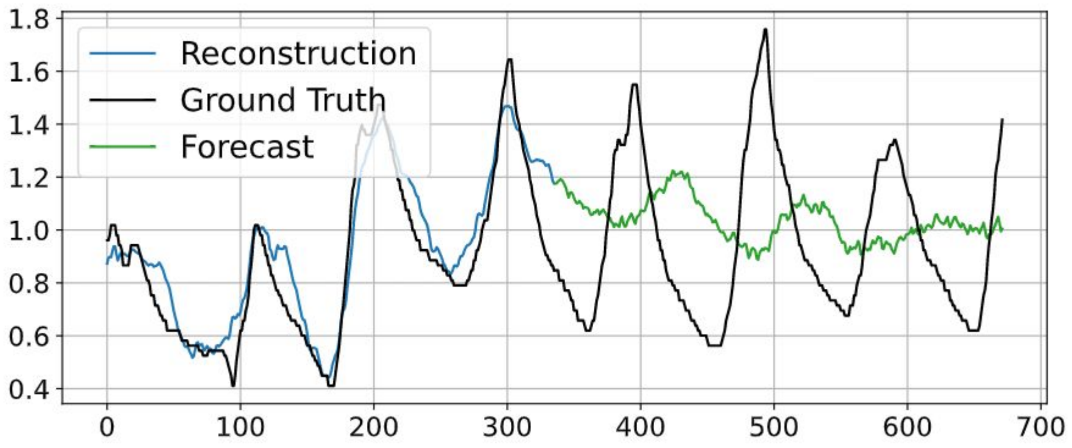
(a) Naive Deep Time-index Model

Deep Time-index Models

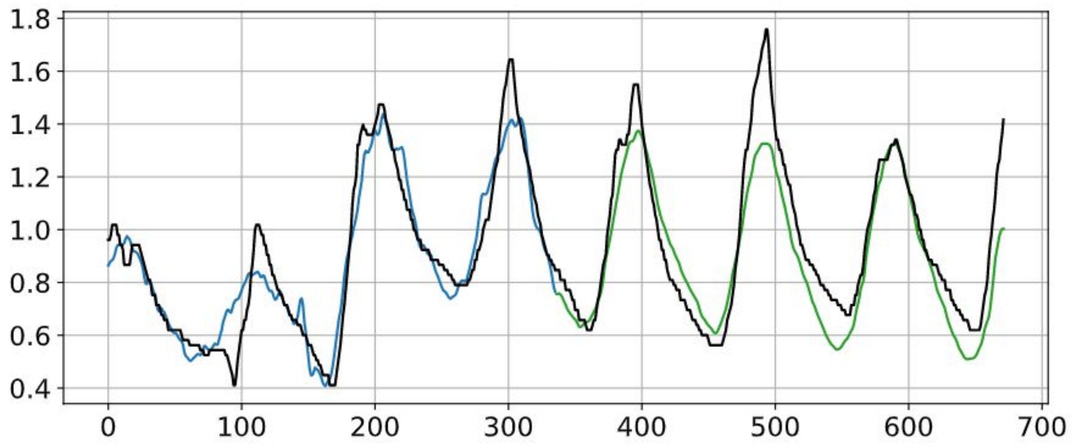
Pitfalls of deep time-index models

Pitfalls

- No inductive biases (such as linear trend, periodicity) unlike classical time-index models
- How to extrapolate across forecast horizon (generalize to future time steps)?



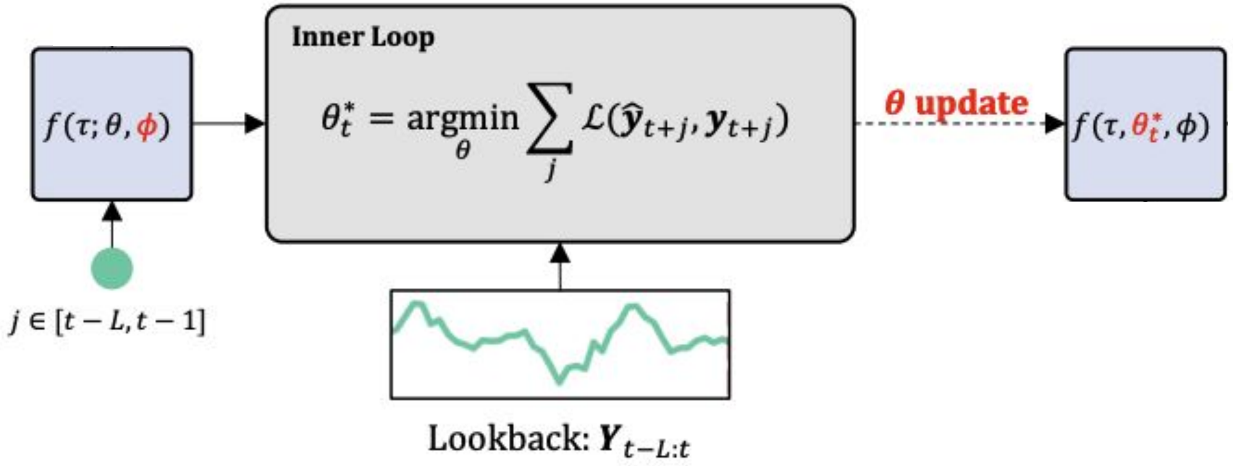
(a) Naive Deep Time-index Model



(b) With meta-optimization formulation

Learning Deep Time-index Models

Meta-optimization framework

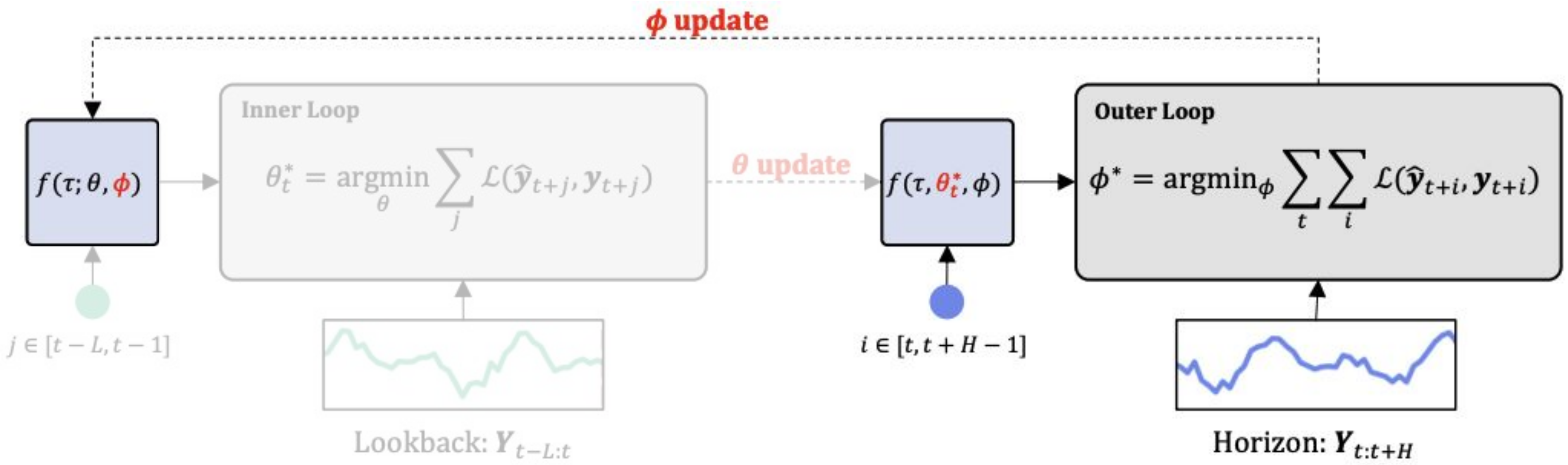


Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**

Learning Deep Time-index Models

Meta-optimization framework

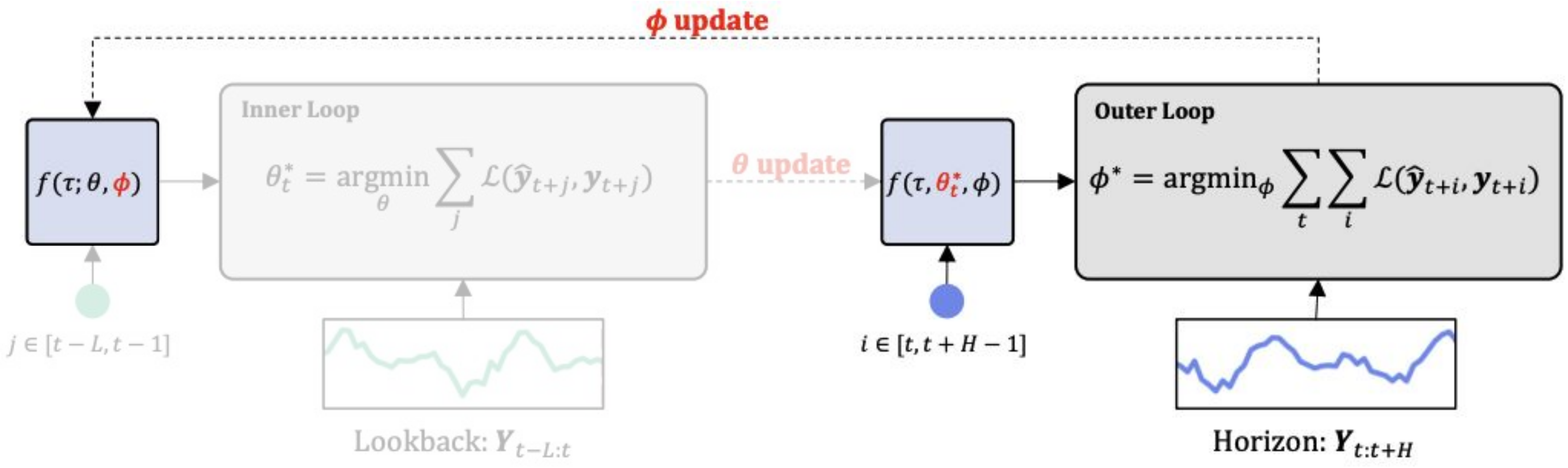


Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**
- Outer loop updates the **global meta parameters** over the **forecast horizon**

Learning Deep Time-index Models

Meta-optimization framework

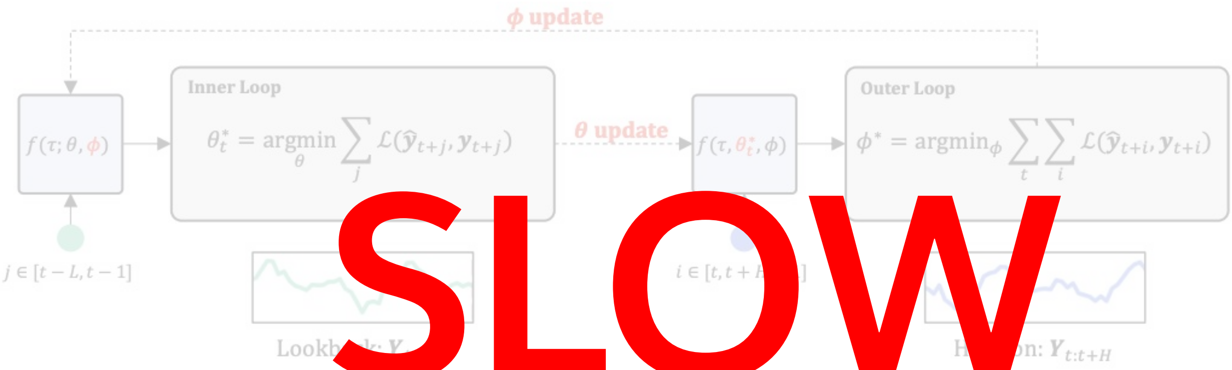


Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**
- Outer loop updates the **global meta parameters** over the **forecast horizon**
- Global meta parameters encode the inductive bias, learning to enable extrapolation across the forecast horizon

Learning Deep Time-index Models

Meta-optimization framework



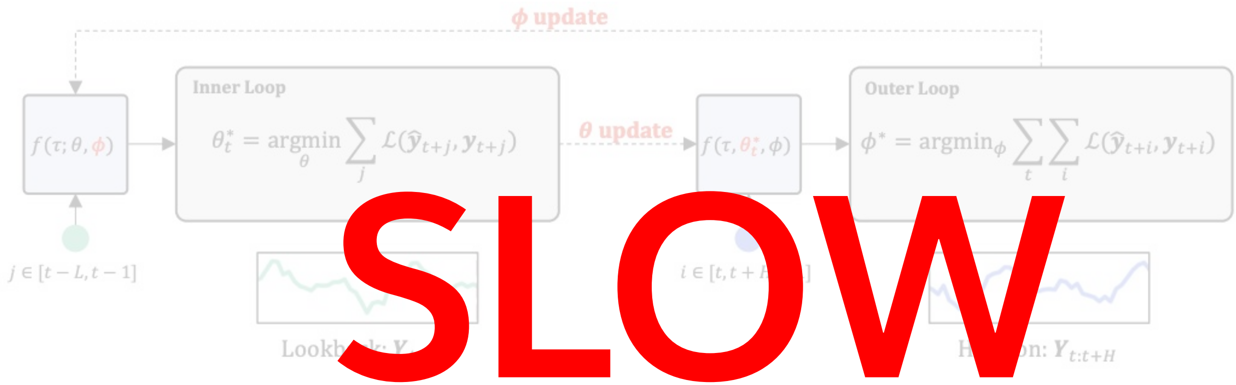
(b) Meta-optimization Framework

Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**
- Outer loop updates the **global meta parameters** over the **forecast horizon**
- Global meta parameters encode the inductive bias, learning to enable extrapolation across the forecast horizon

Learning Deep Time-index Models

Meta-optimization framework



(b) Meta-optimization Framework

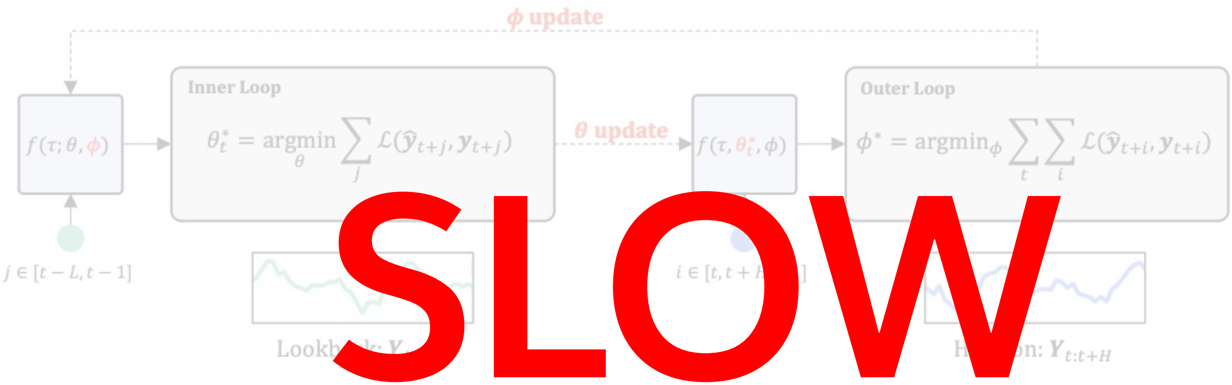
Each forecast is treated as an optimization problem!

Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**
- Outer loop updates the **global meta parameters** over the **forecast horizon**
- Global meta parameters encode the inductive bias, learning to enable extrapolation across the forecast horizon

Learning Deep Time-index Models

Meta-optimization framework



(b) Meta-optimization Framework

SLOW

Each forecast is treated as an optimization problem! → Fast adaptation is needed!

Solution: Meta-optimization formulation:

- Inner loop updates the **local base parameters** to the **lookback window**
- Outer loop updates the **global meta parameters** over the **forecast horizon**
- Global meta parameters encode the inductive bias, learning to enable extrapolation across the forecast horizon

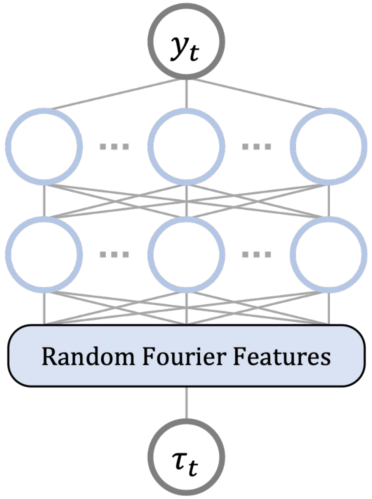
Learning Deep Time-index Models

Fast and efficient meta-optimization

$$z^{(0)} = \gamma(\tau) = [\sin(2\pi B\tau), \cos(2\pi B\tau)]^T$$

$$z^{(k+1)} = \max(0, W^{(k)} z^{(k)} + b^{(k)}), \quad k = 0, \ldots, K-1$$

$$f_\theta(\tau) = W^{(K)} z^{(K)} + b^{(K)}$$



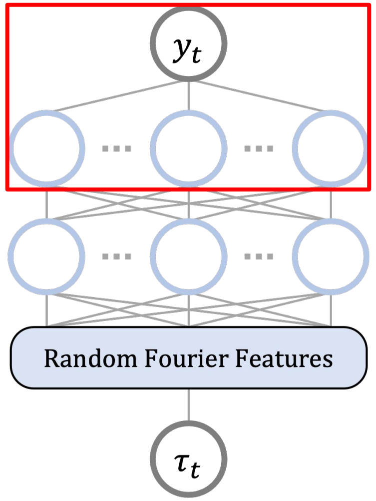
Learning Deep Time-index Models

Fast and efficient meta-optimization

$$\boldsymbol{z}^{(0)} = \gamma(\boldsymbol{\tau}) = [\sin(2\pi \boldsymbol{B}\boldsymbol{\tau}), \cos(2\pi \boldsymbol{B}\boldsymbol{\tau})]^T$$

$$\boldsymbol{z}^{(k+1)} = \max(0, \boldsymbol{W}^{(k)}\boldsymbol{z}^{(k)} + \boldsymbol{b}^{(k)}), \quad k = 0, \ldots, K-1$$

$$f_\theta(\boldsymbol{\tau}) = \boldsymbol{W}^{(K)}\boldsymbol{z}^{(K)} + \boldsymbol{b}^{(K)}$$



Apply inner loop updates to last linear layer only!

Base params $\theta = \{\boldsymbol{W}^{(K)}\}$

Meta params $\phi = \{\boldsymbol{W}^{(0)}, \boldsymbol{b}^{(0)}, \ldots, \boldsymbol{W}^{(K-1)}, \boldsymbol{b}^{(K-1)}, \lambda\}$

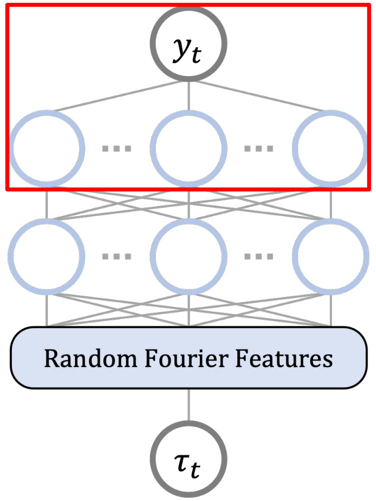
Learning Deep Time-index Models

Fast and efficient meta-optimization

$$z^{(0)} = \gamma(\boldsymbol{\tau}) = [\sin(2\pi \boldsymbol{B}\boldsymbol{\tau}), \cos(2\pi \boldsymbol{B}\boldsymbol{\tau})]^T$$

$$z^{(k+1)} = \max(0, \boldsymbol{W}^{(k)} z^{(k)} + \boldsymbol{b}^{(k)}), \quad k = 0, \dots, K-1$$

$$f_\theta(\boldsymbol{\tau}) = \boldsymbol{W}^{(K)} z^{(K)} + \boldsymbol{b}^{(K)}$$



Apply inner loop updates to last linear layer only!

Base params $\theta = \{\boldsymbol{W}^{(K)}\}$

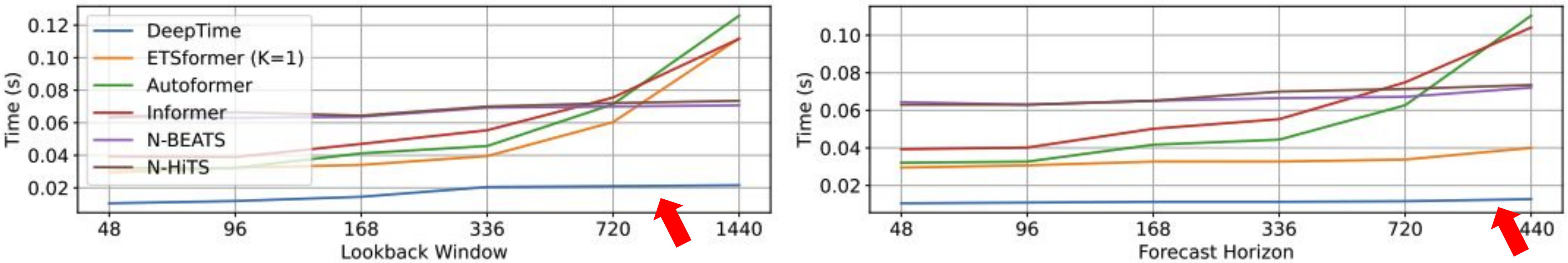
Meta params $\phi = \{\boldsymbol{W}^{(0)}, \boldsymbol{b}^{(0)}, \dots, \boldsymbol{W}^{(K-1)}, \boldsymbol{b}^{(K-1)}, \lambda\}$ $\longrightarrow$ **Closed-form solution to Ridge Regression!**

DeepTime obtains state-of-the-art results!

MSE for each dataset averaged over 4 different horizons

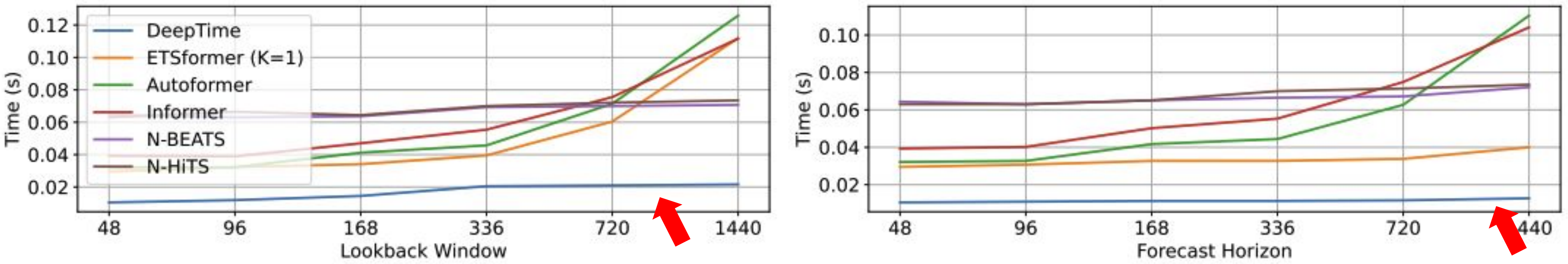
	DeepTime	NSTrans	N-HiTS	ETSformer	FEDformer	Autoformer	Informer	LogTrans	GP
ETTm2	**0.262**	0.306	0.279	0.293	0.305	0.324	1.410	1.535	0.684
ECL	**0.164**	0.193	0.186	0.208	0.205	0.227	0.311	0.272	0.568
Exchange	**0.351**	0.461	0.390	0.410	0.478	0.613	1.550	1.402	0.468
Traffic	**0.414**	0.624	0.452	0.621	0.573	0.628	0.764	0.705	1.200
Weather	**0.231**	0.288	0.249	0.271	0.309	0.338	0.634	0.696	0.463
ILI	2.257	**2.077**	2.210	2.497	2.307	3.006	5.137	4.839	2.642
Avg Rank	**1.42**	3.67	2.38	3.83	4.17	6.17	8.17	8.04	7.17

DeepTime is highly efficient!

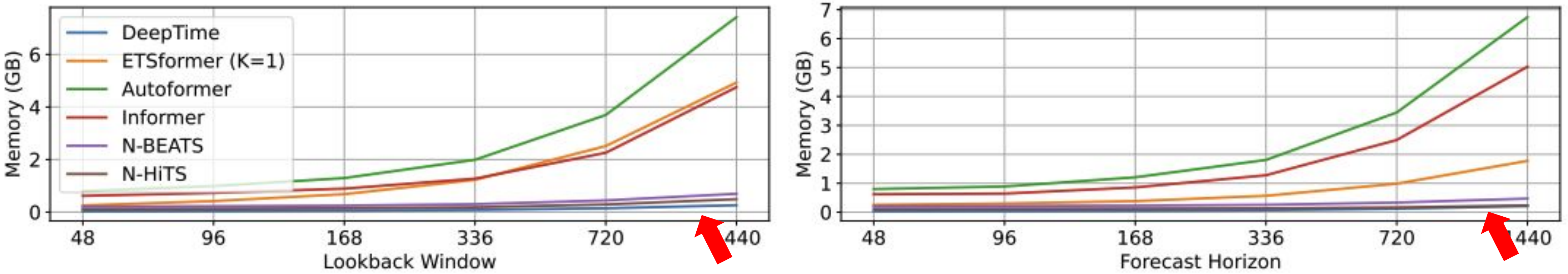


(a) Runtime Analysis

DeepTime is highly efficient!



(a) Runtime Analysis



(b) Memory Analysis

Conclusion

salesforce

- Demonstrate that naive deep time-index models are unable to perform forecasting
- DeepTime: deep time-index models + meta-optimization
- DeepTime achieves
  - state-of-the-art results,
  - superior time efficiency over existing approaches,
  - superior memory efficiency over existing approaches

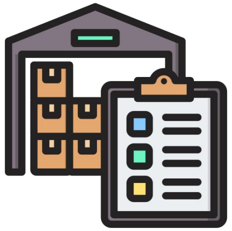
Paper can be found at: https://arxiv.org/abs/2207.06046
Code is available at: https://github.com/salesforce/DeepTime

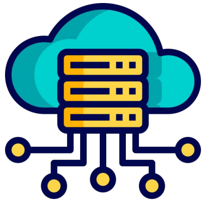
Conclusion

- Demonstrate that naive deep time-index models are unable to perform forecasting
- DeepTime: deep time-index models + meta-optimization
- DeepTime achieves
  - state-of-the-art results,
  - superior time efficiency over existing approaches,
  - superior memory efficiency over existing approaches

Limitations & Future Work

- Exogenous covariates
- Probabilistic forecasting





Paper can be found at: https://arxiv.org/abs/2207.06046
Code is available at: https://github.com/salesforce/DeepTime