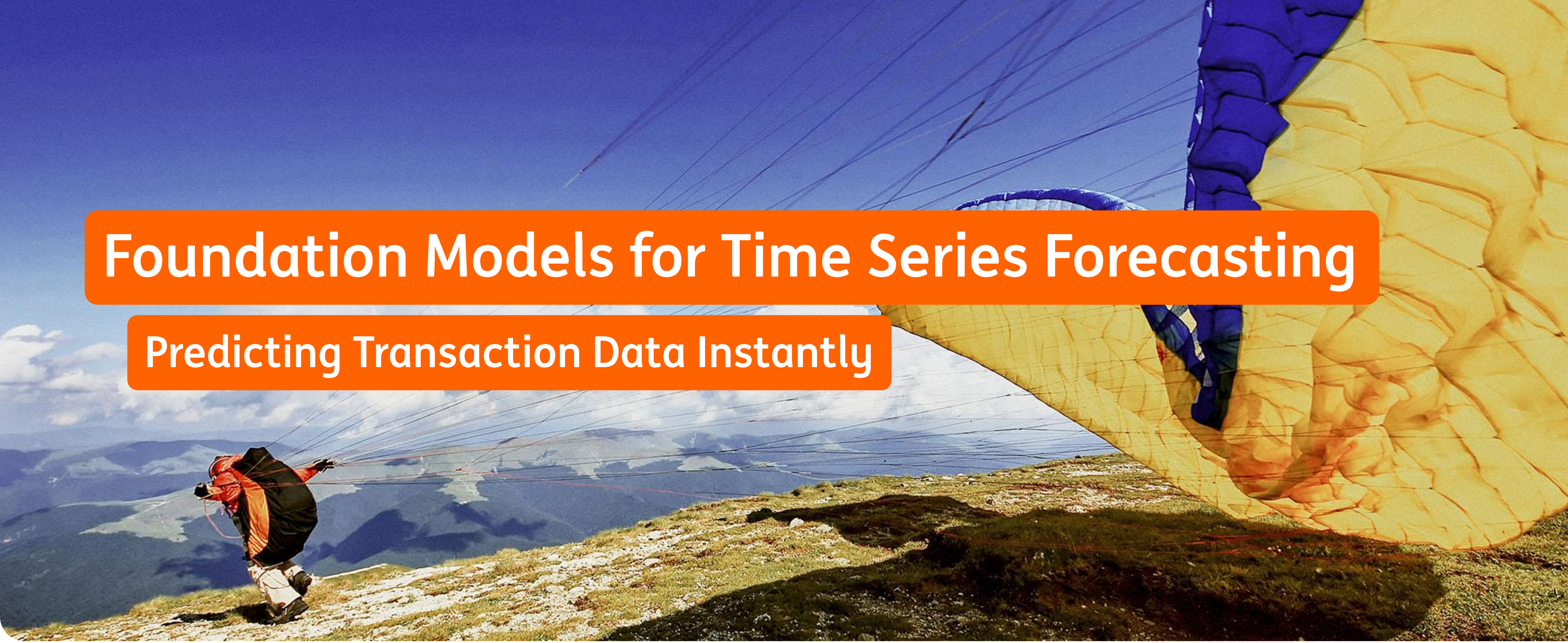


Foundation Models for Time Series Forecasting

Predicting Transaction Data Instantly

Didier Merk

ING DSCC – December 10th, 2024

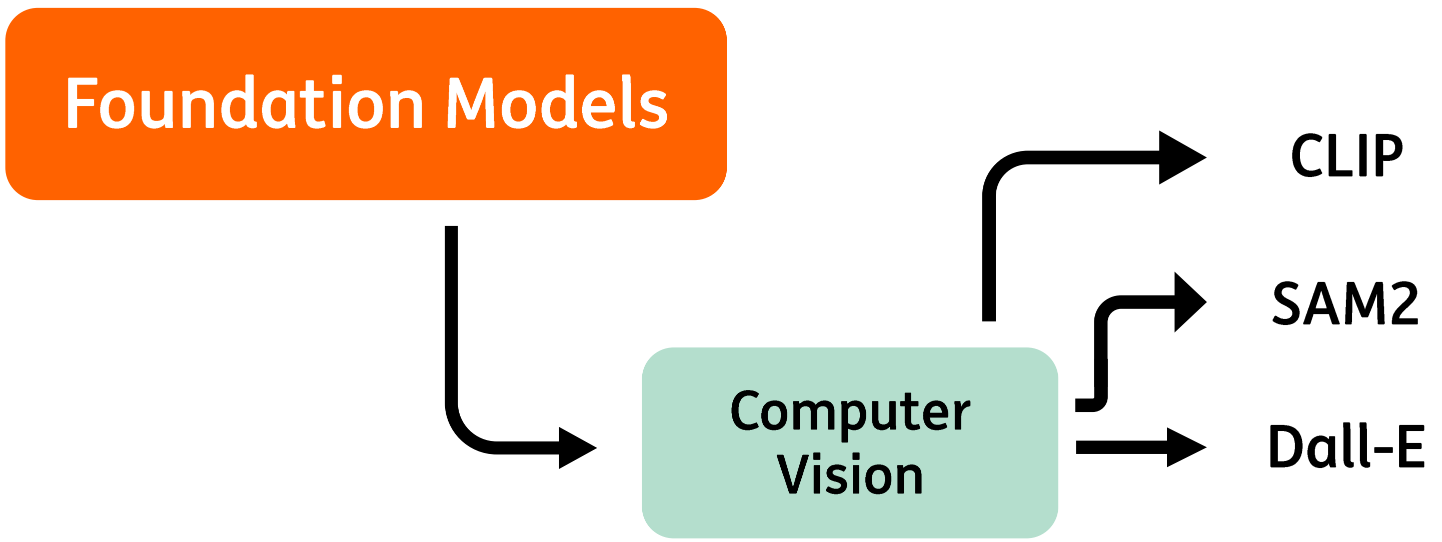
ING Bank & University of Amsterdam

do your thing

Foundation Models: Large-scale, general-purpose learners

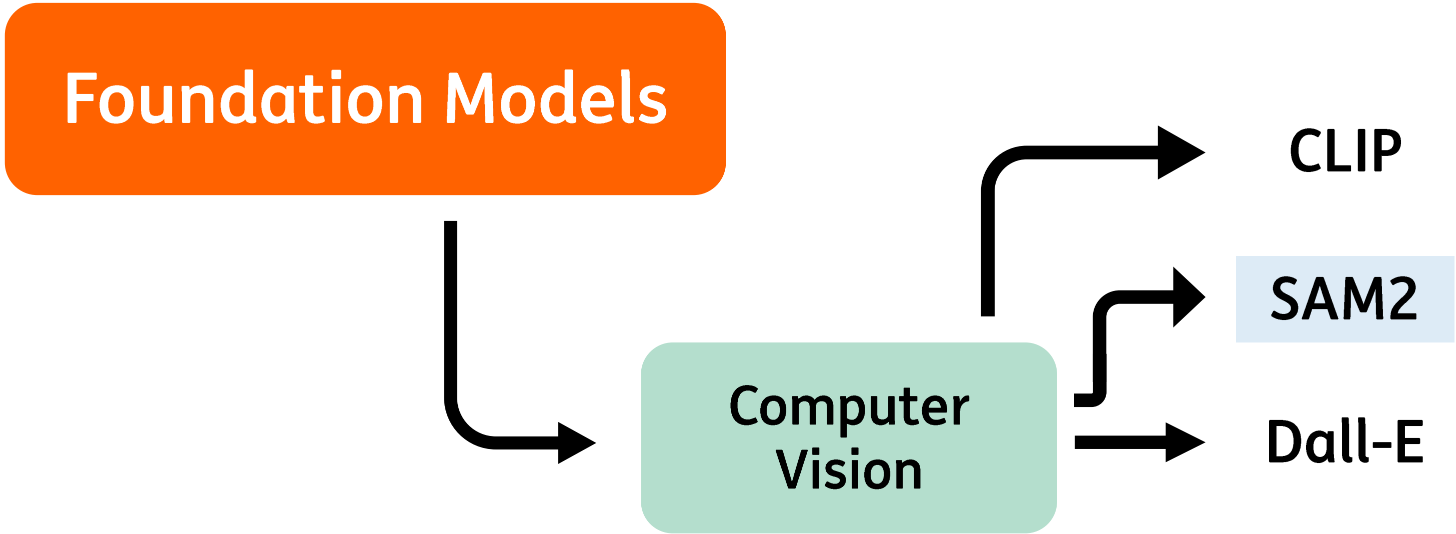
Foundation Models

Foundation Models: Large-scale, general-purpose learners

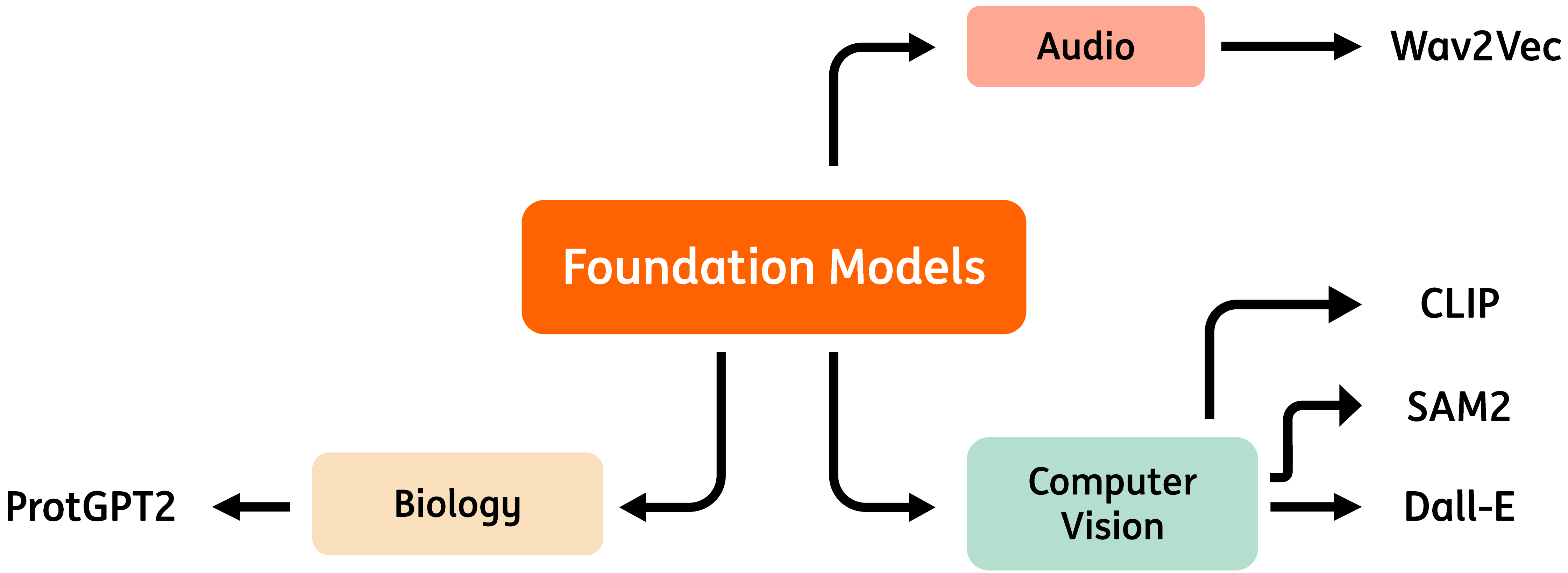


Foundation Models: Large-scale, general-purpose learners

Model: **SAM2**

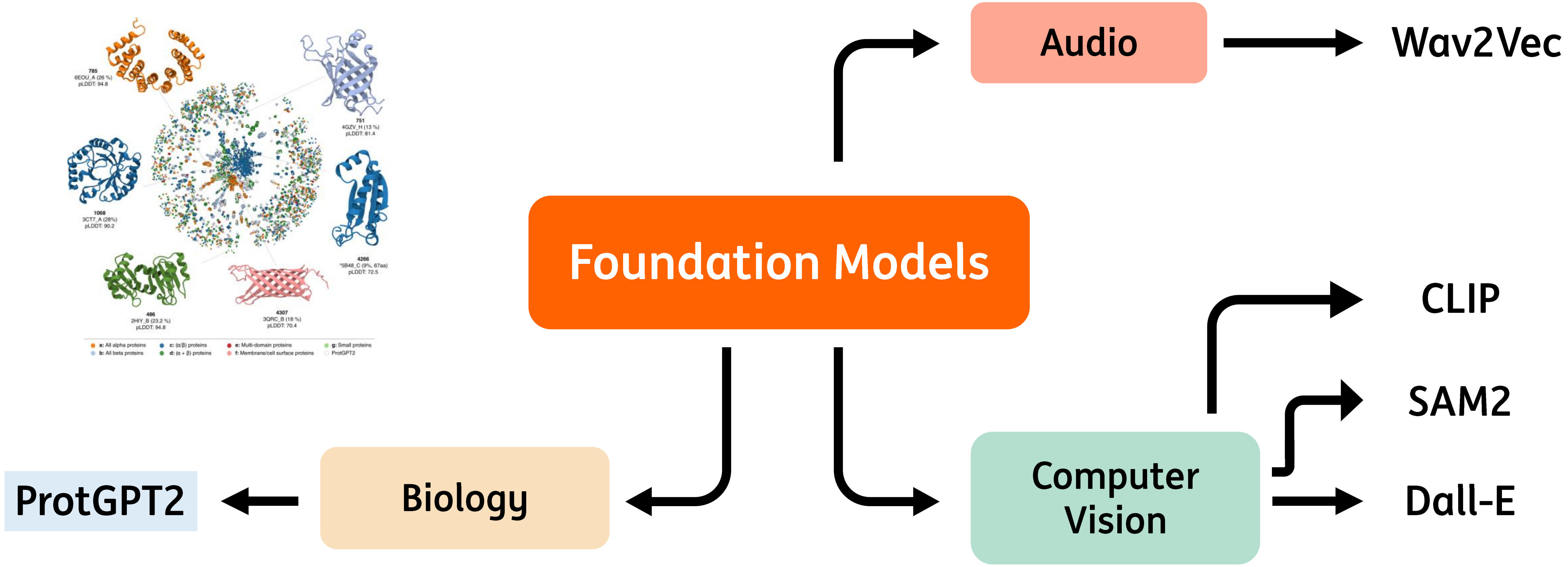


Foundation Models: Large-scale, general-purpose learners

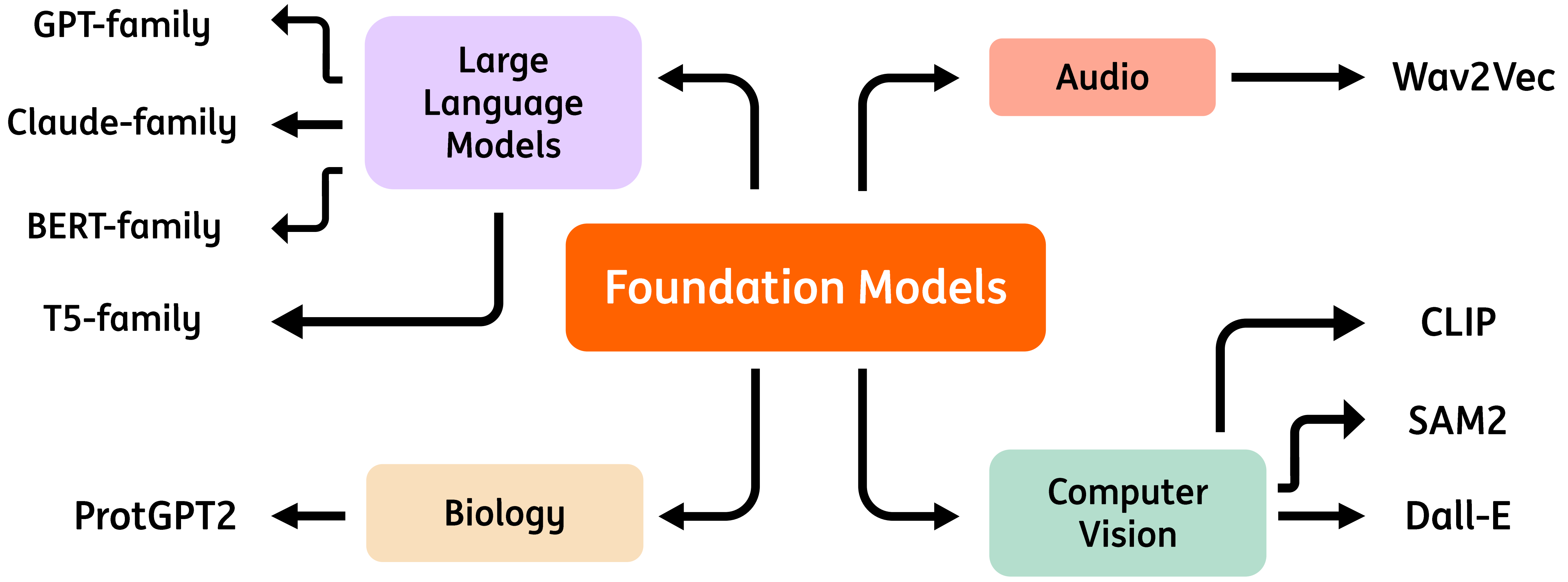


Foundation Models: Large-scale, general-purpose learners

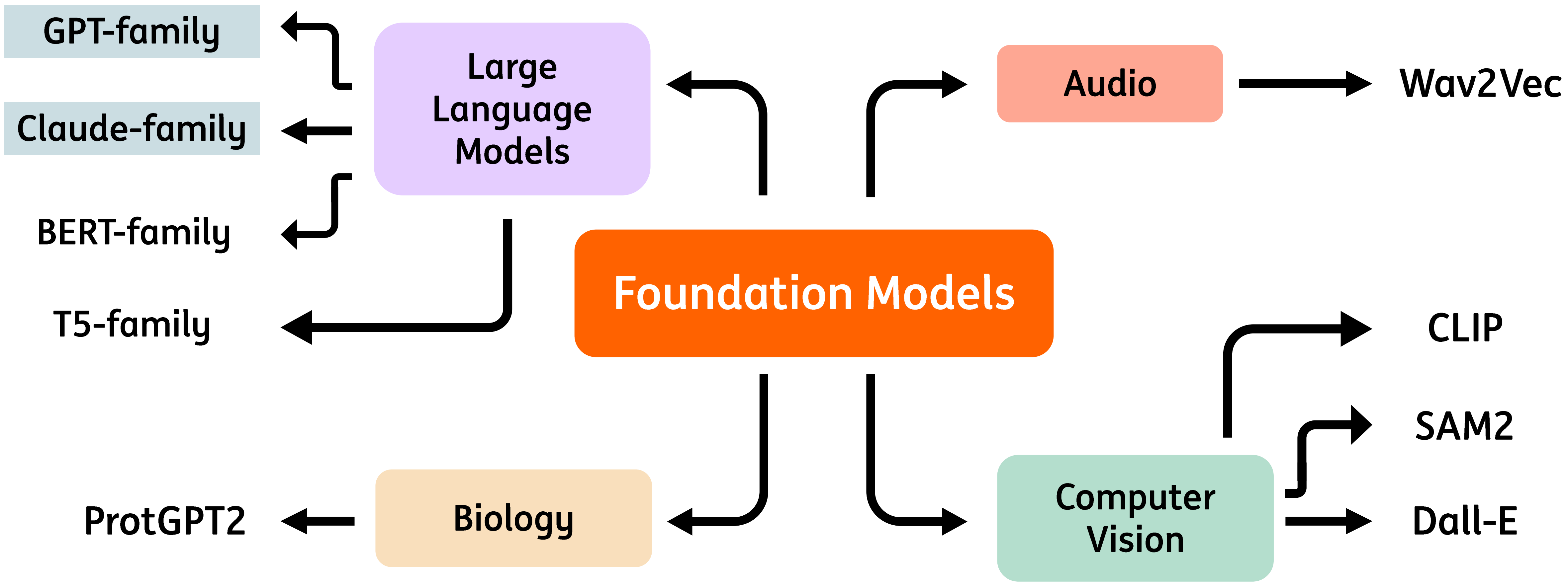
Model: **ProtGPT2**



Foundation Models: Large-scale, general-purpose learners

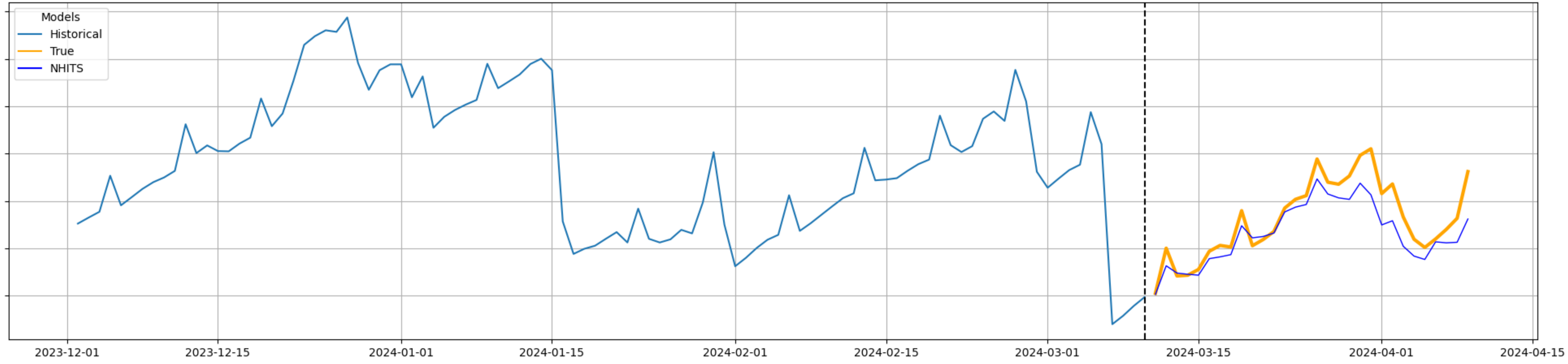


Foundation Models: Large-scale, general-purpose learners



Forecasting: A next-token prediction problem?

1 Use-case at ING: Univariate End-of-Day Balance Prediction



Thesis: Rethinking Models for Financial Time Series Forecasting

1 Research question:

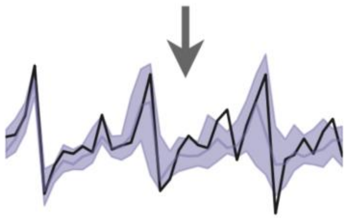
“To what extent can large language model architectures be applied to financial time series forecasting, in comparison to traditional statistical and deep learning models?”

2 Main sub-questions:

Accuracy of LLM-based forecasts	Effects of seasonality and predictability	Reliability of the probabilistic output

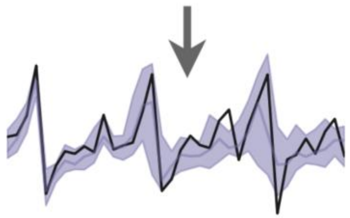
Aligning modalities: From language to numbers

Prompt		Samples	
"The dog jumped on the bed"	LLM	"and fell asleep" "and wagged its tail" "and bit my leg"	Classic text generating LLMs
"631, 656, … , 473, 487, 485"	LLM	"479, …, 371, 364" "492, …, 499, 501" "488, …, 421, 434"	Numerical sequences encoded as strings [1]



Aligning modalities: From language to numbers

Prompt		**Samples**
"The dog jumped on the bed"	**LLM**	"and fell asleep" "and wagged its tail" "and bit my leg"
"631, 656, … , 473, 487, 485"	**LLM**	"479, …, 371, 364" "492, …, 499, 501" "488, …, 421, 434"



Difficulties:

1. Tokenization

```
import tokenizer

number = "42235630"
tokens = tokenize(number)

print(tokens)
```

```
>>> [422, 35, 630]
```

2. Contrastive learning

Dedicated Time Series Foundation Models

TimeGPT: The First Foundation Model for Time Series Forecasting

Explore the first generative pre-trained forecasting model and apply it in a project with Python

Marco Peixeiro · Follow

Published in Towards Data Science · 12 min read · Oct 24, 2023

Dedicated Time Series Foundation Models

TimeGPT: The First Foundation Model for Time Series Forecasting

Explore the first genera
in a project with Python

Marco Peixeiro · F
Published in Towards Da

TimesFM: Google's Foundation Model For Time-Series Forecasting

A new age for time series

Nikos Kafritsas · Follow
Published in Towards Data Science · 9 min read · Feb 28, 2024

Dedicated Time Series Foundation Models

TimeGPT: The First Foundation Model for Time Series Forecasting

Explore the first genera
in a project with Python

TimesFM: Google's Foundation
orecasting

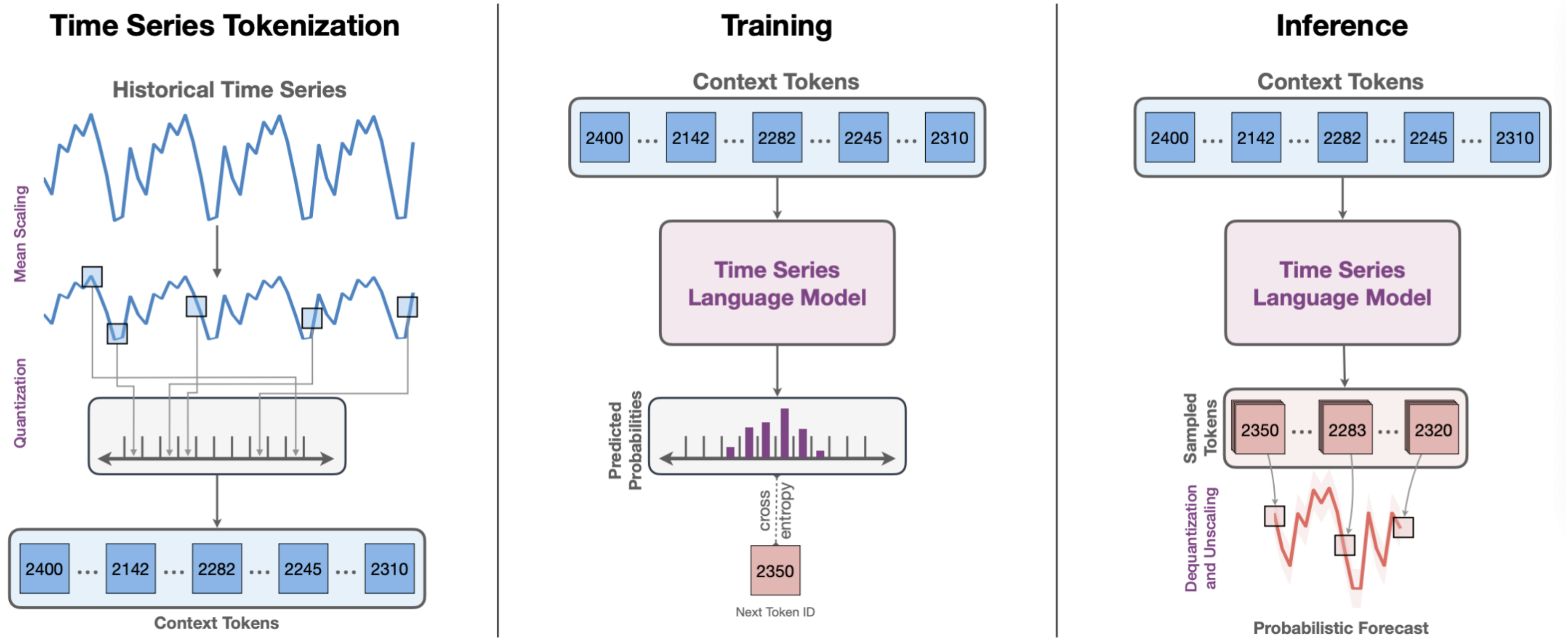
Chronos: The Latest Time Series Forecasting Foundation Model by Amazon

024

Take a deep dive into Chronos, its inner workings, and how to apply it in your forecasting projects using Python.

Marco Peixeiro · Follow
Published in Towards Data Science · 12 min read · Mar 27, 2024

Chronos: A dedicated time series Foundation Model



from [2]

Time Series Language Model = Google's T5 LLM-family

Zero-Shot Forecasting = No additional training!

Model comparisons:

Model	Architecture	Number of params.
Chronos-T5 (small)	Pre-trained Transformer	46M
Chronos-T5 (large)	Pre-trained Transformer	710M
Chronos-T5 (Finetuned)	Pre-trained Transformer	46M
PatchTST	Transformer	604K
NHITS	MLP	3.6M
TimesNet	CNN	4.9M
DeepAR	LSTM + MLP decoder	199K
Naive	Statistical	-
AutoARIMA	Statistical	-
AutoETS	Statistical	-

Pre-trained Models

Deep Models

Statistical models

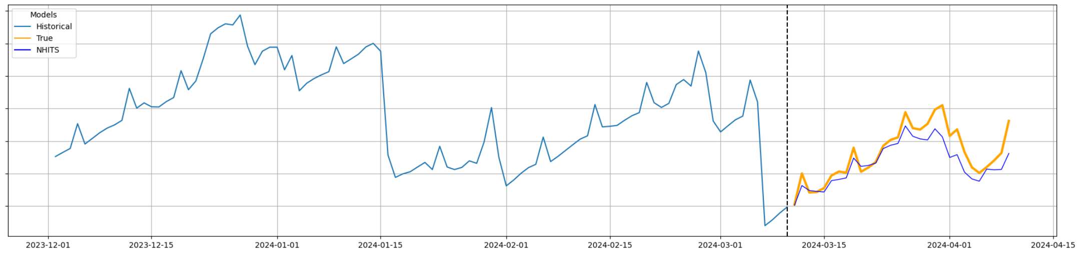
Forecasting balances

1 Use-case at ING: **Uni**variate End-of-Day Balance Prediction

Forecasting balances

1 **Use-case at ING: Uni**variate End-of-Day Balance Prediction

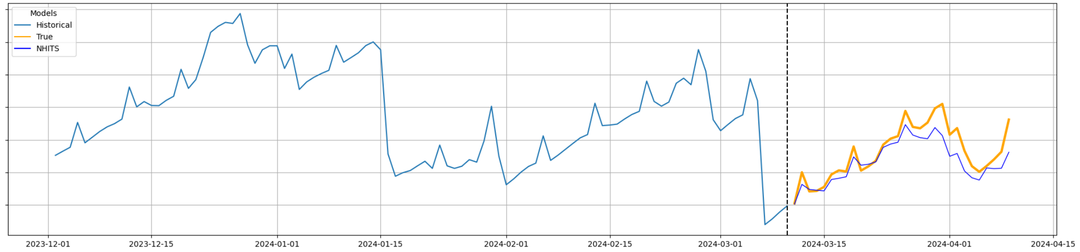
2 **Data:** "profile. Redacted for privacy reasons
& "prd Redacted for privacy reasons



Forecasting balances

1 Use-case at ING: **Uni**variate End-of-Day Balance Prediction

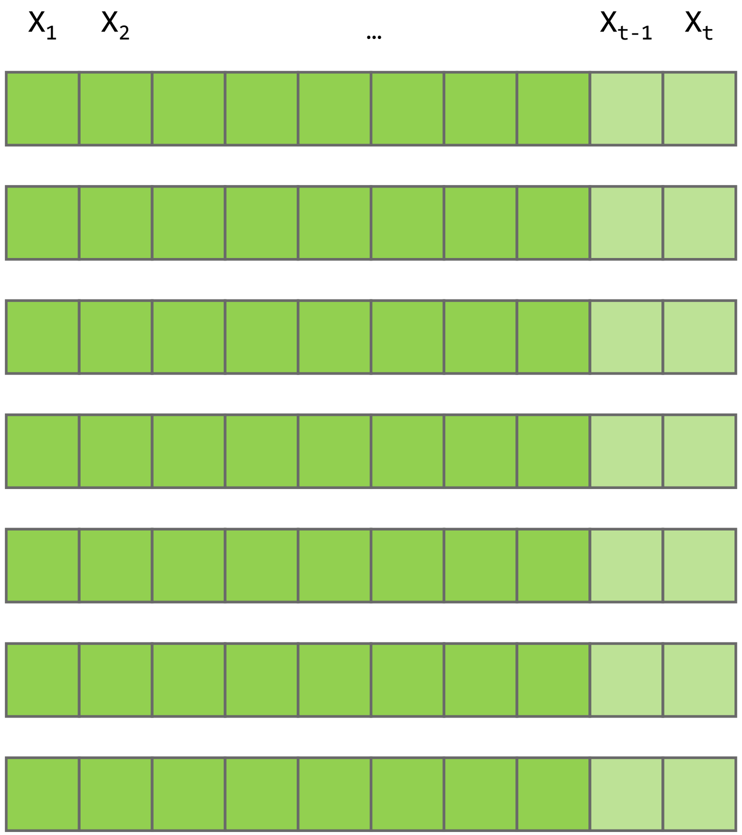
2 Data: "profile Redacted for privacy reasons
& "pr Redacted for privacy reasons



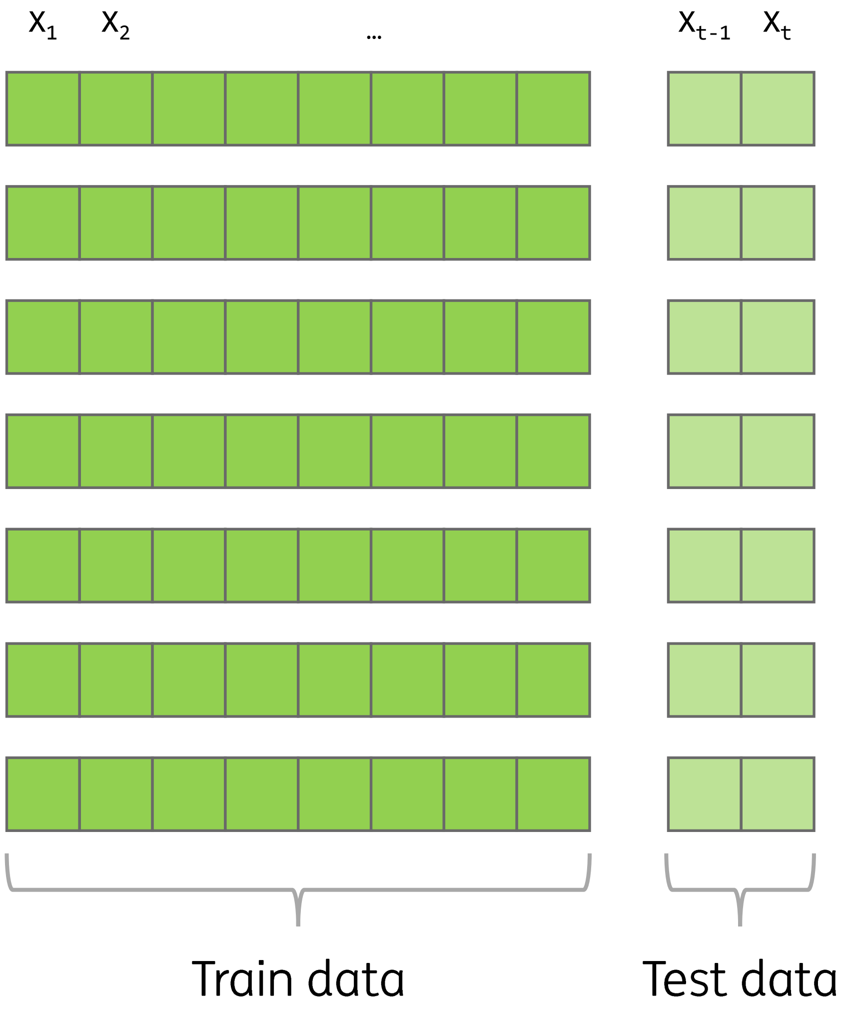
Data filtering and processing:
1. Dutch Transaction Services Wholesale Banking Clients
2. Active between 2022 and 2024
3. Grouped under *ultimate parents*
4. Forward-filled and min-max scaled

Result: **278** time series, each with **1014** timesteps

Evaluation

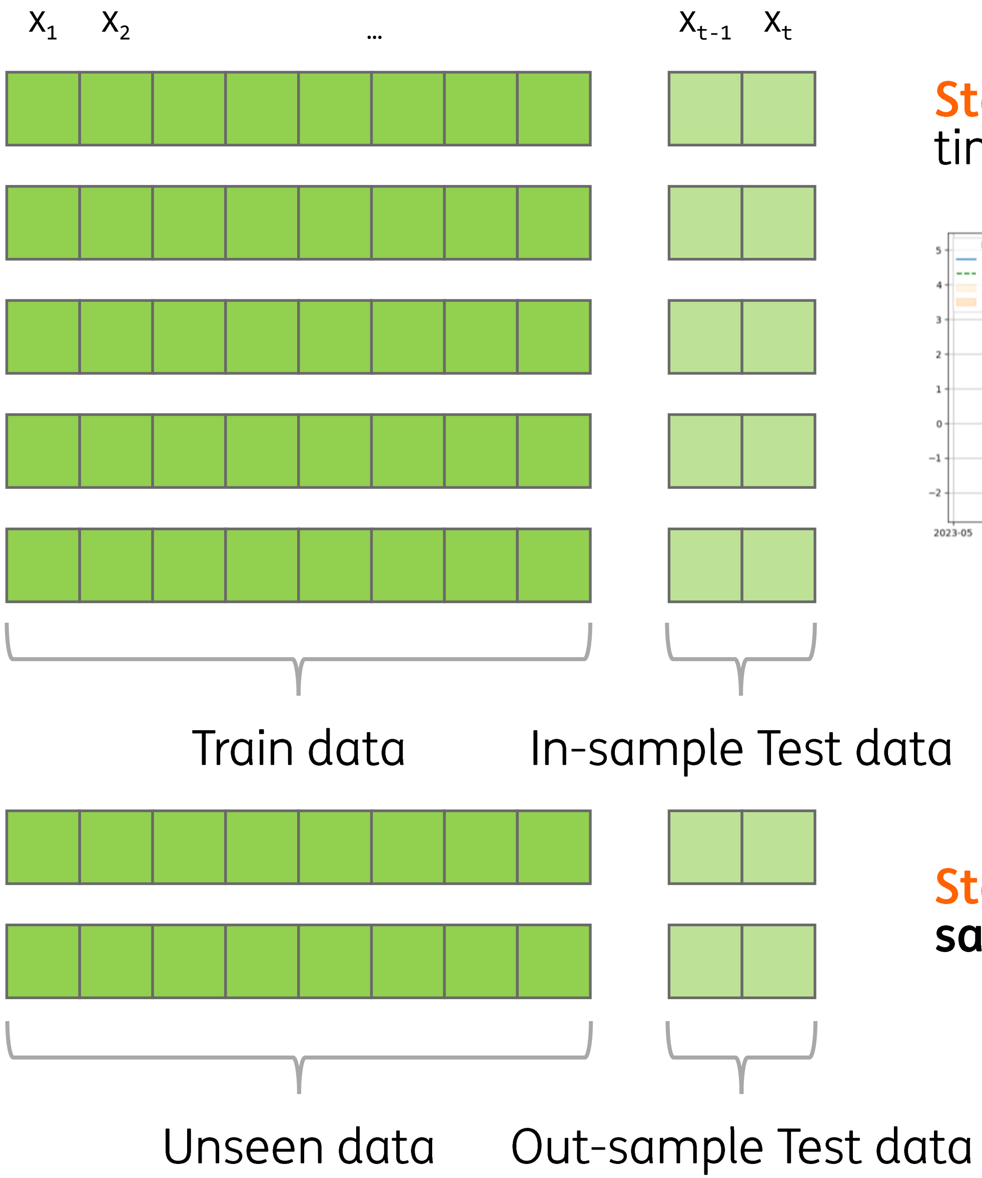


Evaluation



Step 1: The forecasting horizon is cut-off from the original timeseries and used as **test data**

Evaluation

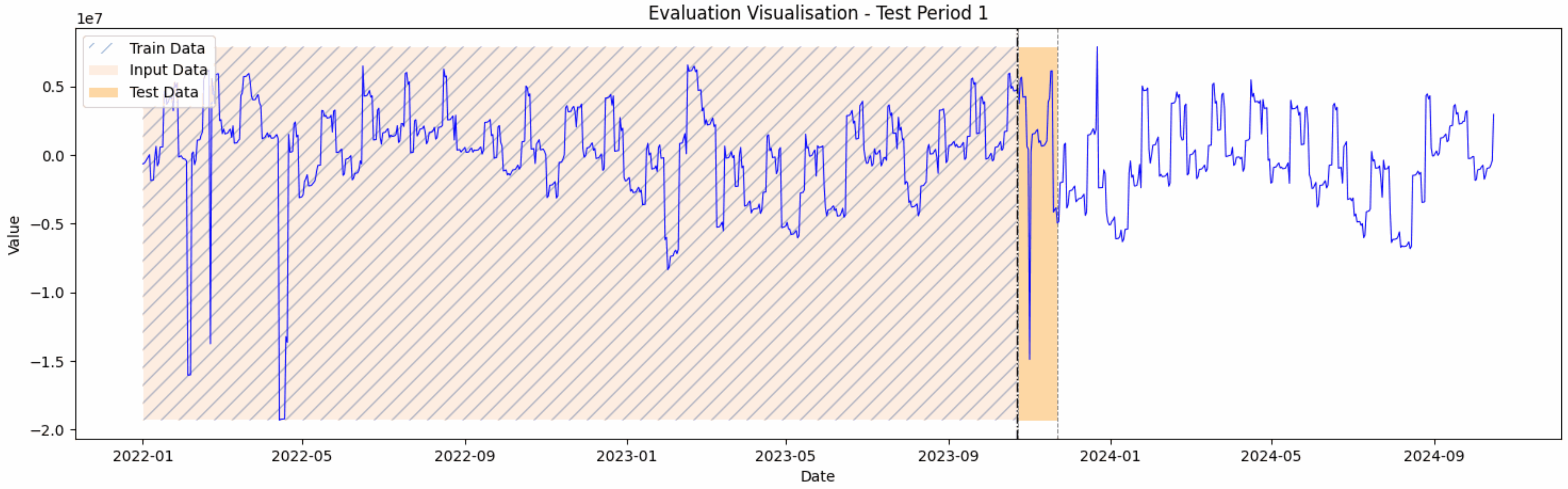


Step 1: The forecasting horizon is cut-off from the original timeseries and used as **test data**

Step 2: Data is divided into **in-sample** (80% of total) and **out-sample** (20% of total) timeseries

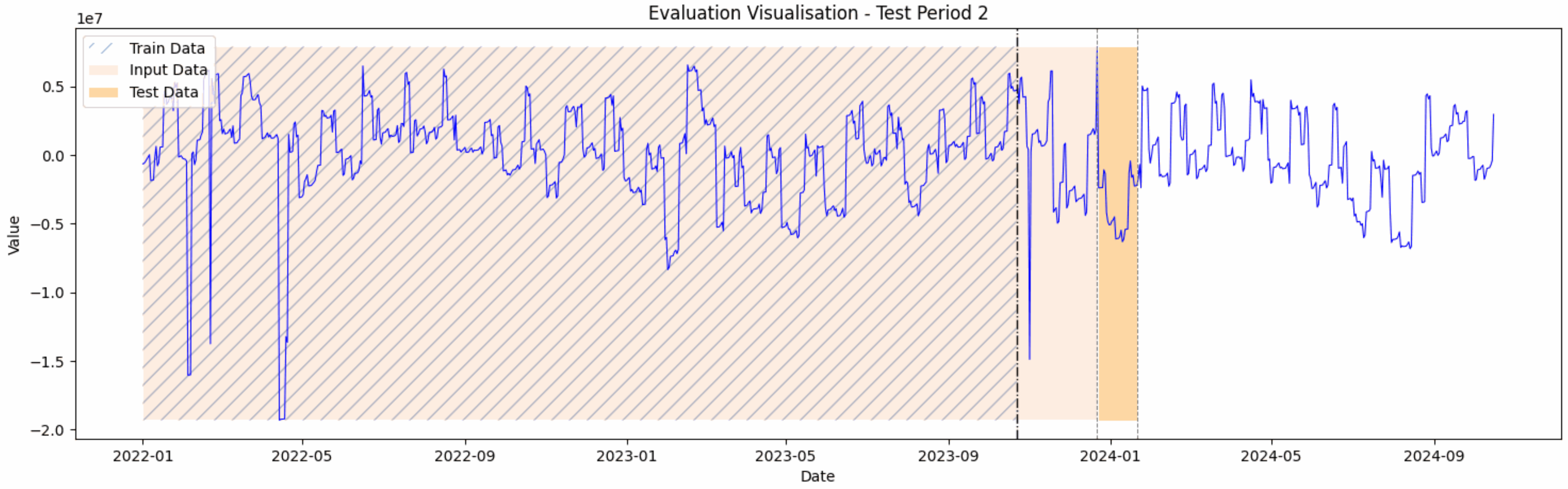
Evaluation: Multiple test months

We don’t have enough computational resources and time to train each model for each test period



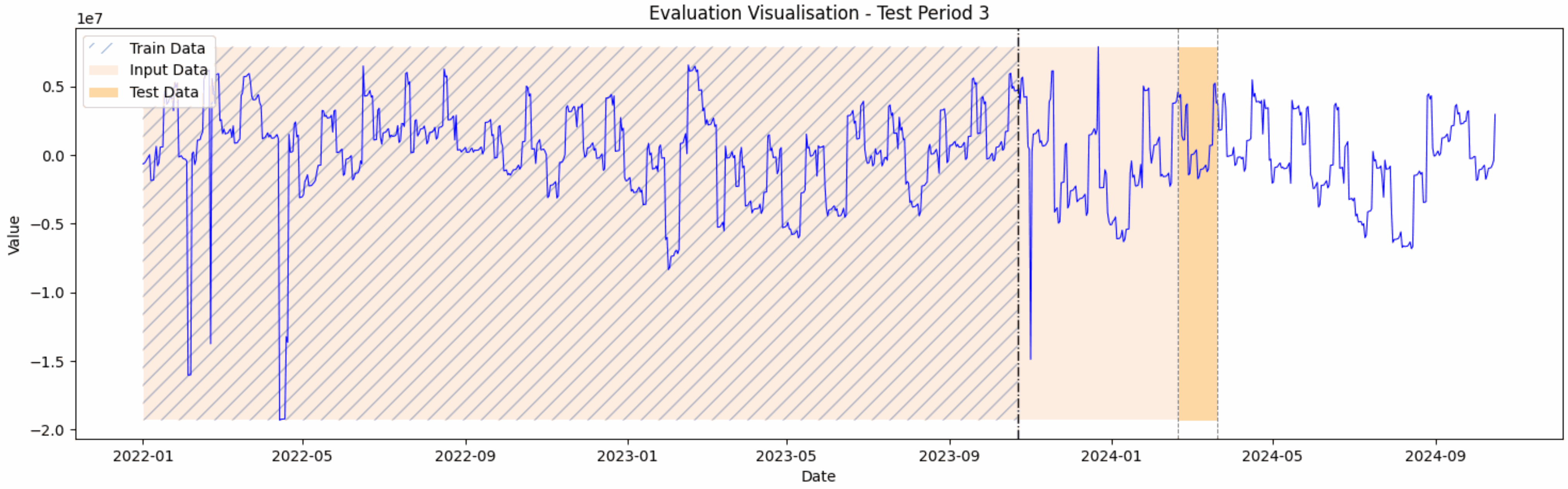
Evaluation: Multiple test months

We don’t have enough computational resources and time to train each model for each test period



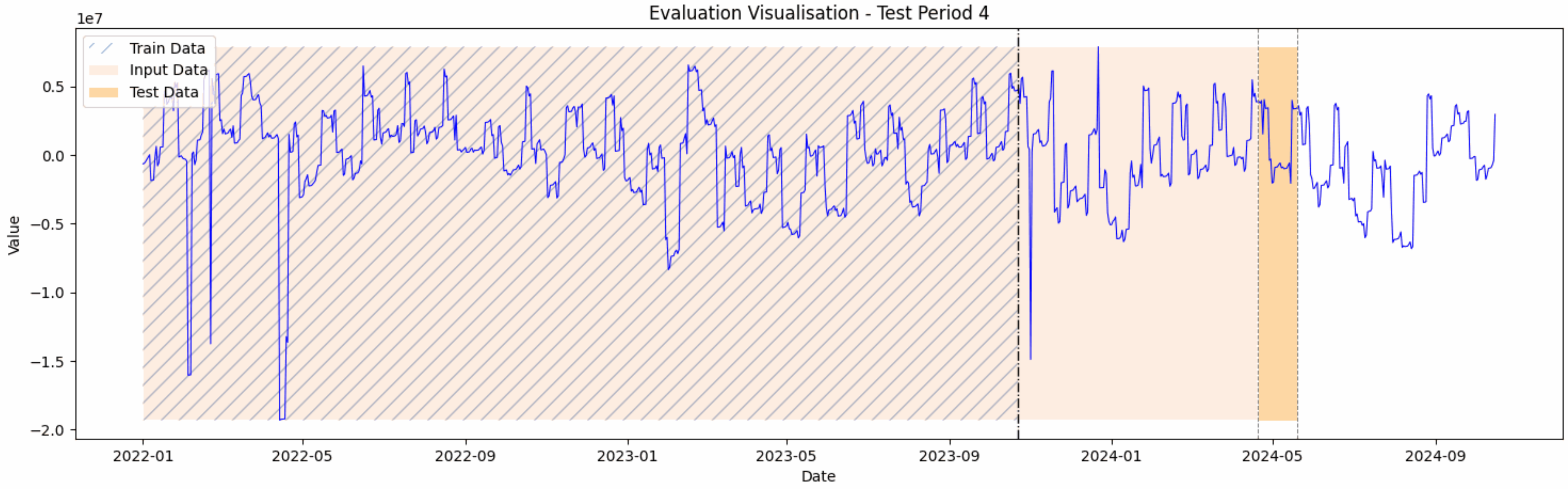
Evaluation: Multiple test months

We don't have enough computational resources and time to train each model for each test period



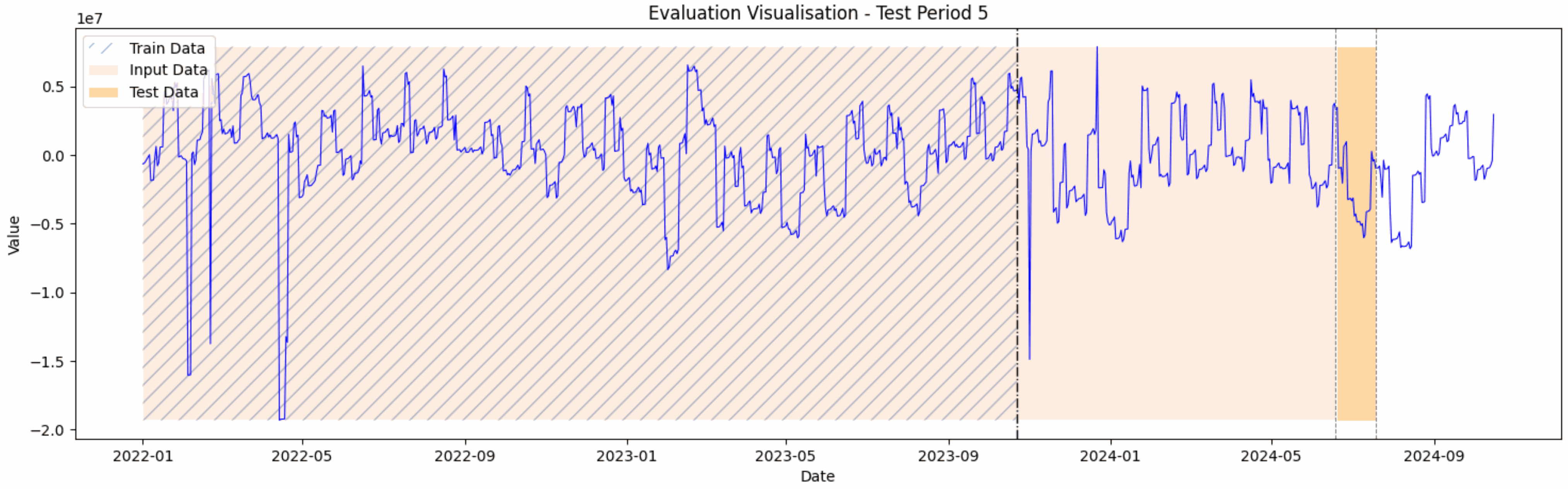
Evaluation: Multiple test months

We don't have enough computational resources and time to train each model for each test period



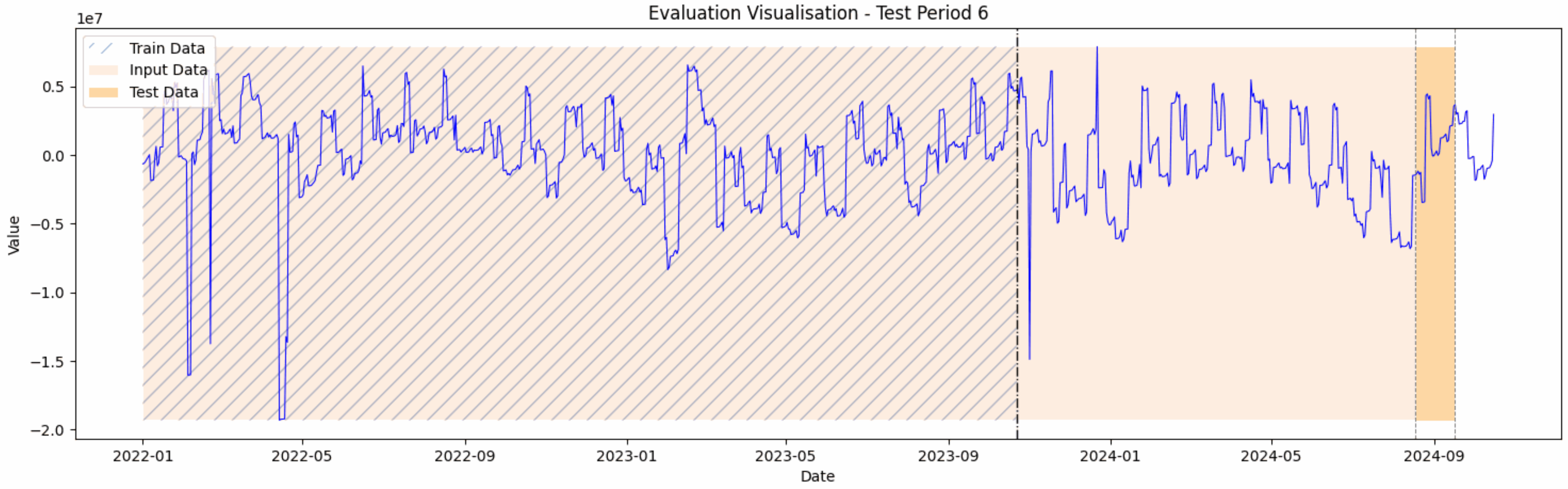
Evaluation: Multiple test months

We don't have enough computational resources and time to train each model for each test period



Evaluation: Multiple test months

We don't have enough computational resources and time to train each model for each test period



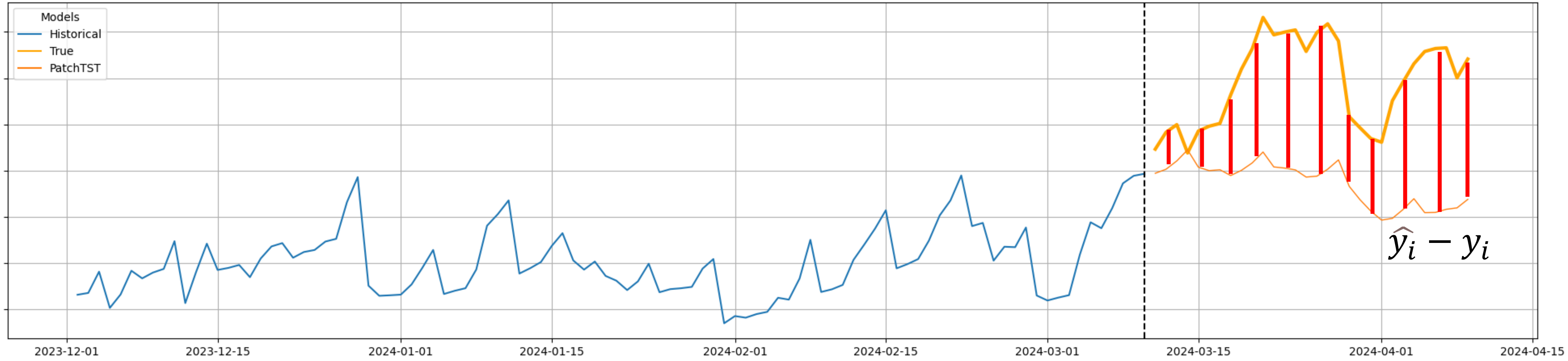
Evaluation: Metric Calculation

For each model we calculate **three** different **metrics** for **four forecasting horizons** for **each timeseries**

1 **MAE** (Mean Absolute Error)

$$\frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i|$$

Forecasting Horizons:
1 day, 7 days, 14 days and 30 days



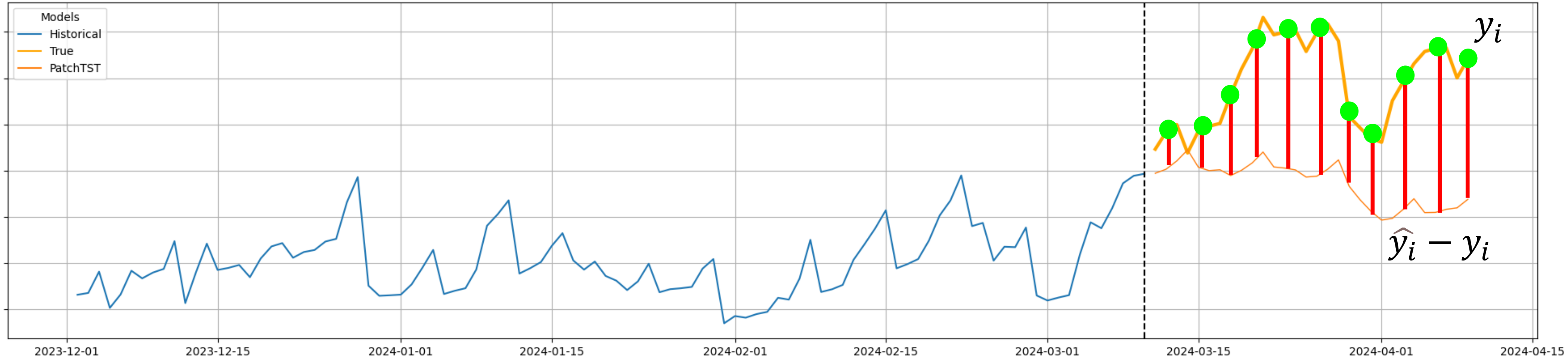
Evaluation: Metric Calculation

For each model we calculate **three** different **metrics** for **four forecasting horizons** for **each timeseries**

2 **MAPE** (Mean Absolute Percentage Error)

$$100\,\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y_i}-y_i}{y_i}\right|$$

Forecasting Horizons:
1 day, 7 days, 14 days and 30 days



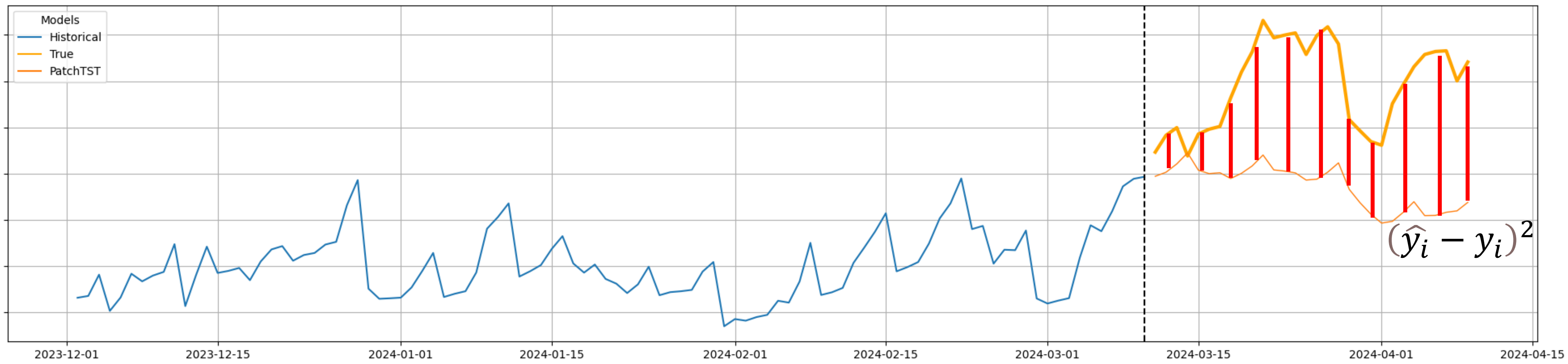
Evaluation: Metric Calculation

For each model we calculate **three** different **metrics** for **four forecasting horizons** for **each timeseries**

3 **RMSE** (Root Mean Squared Error)

$$\sqrt{\sum_{i=1}^{n} \frac{(\hat{y_i} - y_i)^2}{n}}$$

Forecasting Horizons:
1 day, 7 days, 14 days and 30 days



Results: Accuracy of in-sample forecasting

We can take the **median** of each metric for each forecasting horizon over all the timeseries:

Results: Accuracy of in-sample forecasting

We can take the **median** of each metric for each forecasting horizon over all the timeseries:

		Statistical			Deep Learning				Foundation Models		
Metric	**Horizon**	**Naive**	**ARIMA**	**ETS**	**NHITS**	**PatchTST**	**TimesNet**	**DeepAR**	**Chronos-S**	**Chronos-L**	**Chronos-FT**
MAE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	<u>0.0083</u>	**0.0077**	0.0102
	7 days	0.0323	0.0459	0.0456	0.0403	0.0320	0.0385	0.0346	<u>0.0293</u>	**0.0280**	0.0338
	14 days	0.0449	0.0571	0.0580	0.0485	0.0403	0.0473	0.0450	<u>0.0397</u>	**0.0389**	0.0429
	30 days	0.0517	0.0616	0.0636	0.0550	<u>0.0446</u>	0.0514	0.0524	0.0460	**0.0440**	0.0480
MAPE	1 day	3.5840	9.2107	8.9328	6.7932	5.0918	10.8874	5.2234	<u>3.3121</u>	**3.2282**	3.9072
	7 days	13.4556	18.2455	18.7574	16.2607	13.1425	16.6501	14.0777	<u>12.2991</u>	**12.1936**	14.4643
	14 days	19.8646	23.9450	24.3448	21.6301	17.7806	20.8744	19.2472	<u>17.5705</u>	**17.3186**	18.1345
	30 days	21.6435	24.8841	25.5652	22.6731	<u>18.9188</u>	21.5849	21.6900	19.1657	**18.3518**	19.6172
RMSE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	<u>0.0083</u>	**0.0077**	0.0102
	7 days	0.0415	0.0571	0.0584	0.0491	0.0398	0.0467	0.0433	<u>0.0368</u>	**0.0365**	0.0449
	14 days	0.0587	0.0739	0.0763	0.0616	**0.0520**	0.0588	0.0585	0.0548	<u>0.0529</u>	0.0581
	30 days	0.0704	0.0806	0.0830	0.0714	**0.0612**	0.0670	0.0685	0.0644	<u>0.0625</u>	0.0661

Intuitively: A MAPE of **18.3518** indicates 50% of our forecasts had a mean error lower than 18.35%

Results: Accuracy of in-sample forecasting

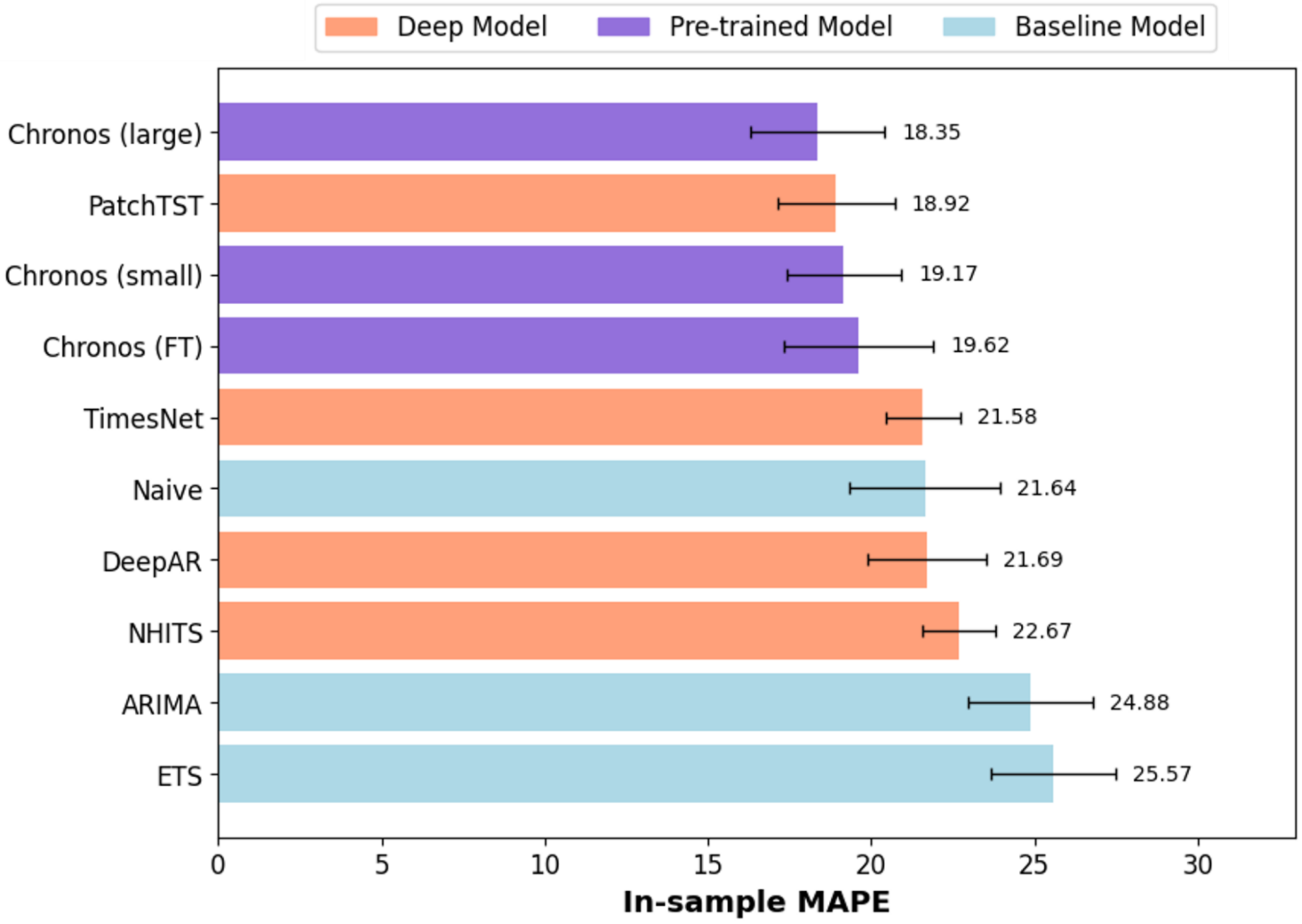
We can take the **median** of each metric for each forecasting horizon over all the timeseries:

		Statistical			Deep Learning				Foundation Models		
Metric	**Horizon**	**Naive**	**ARIMA**	**ETS**	**NHITS**	**PatchTST**	**TimesNet**	**DeepAR**	**Chronos-S**	**Chronos-L**	**Chronos-FT**
MAE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	0.0083	**0.0077**	0.0102
	7 days	0.0323	0.0459	0.0456	0.0403	0.0320	0.0385	0.0346	0.0293	**0.0280**	0.0338
	14 days	0.0449	0.0571	0.0580	0.0485	0.0403	0.0473	0.0450	0.0397	**0.0389**	0.0429
	30 days	0.0517	0.0616	0.0636	0.0550	0.0446	0.0514	0.0524	0.0460	**0.0440**	0.0480
MAPE	1 day	3.5840	9.2107	8.9328	6.7932	5.0918	10.8874	5.2234	3.3121	**3.2282**	3.9072
	7 days	13.4556	18.2455	18.7574	16.2607	13.1425	16.6501	14.0777	12.2991	**12.1936**	14.4643
	14 days	19.8646	23.9450	24.3448	21.6301	17.7806	20.8744	19.2472	17.5705	**17.3186**	18.1345
	30 days	21.6435	24.8841	25.5652	22.6731	18.9188	21.5849	21.6900	19.1657	**18.3518**	19.6172
RMSE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	0.0083	**0.0077**	0.0102
	7 days	0.0415	0.0571	0.0584	0.0491	0.0398	0.0467	0.0433	0.0368	**0.0365**	0.0449
	14 days	0.0587	0.0739	0.0763	0.0616	**0.0520**	0.0588	0.0585	0.0548	0.0529	0.0581
	30 days	0.0704	0.0806	0.0830	0.0714	**0.0612**	0.0670	0.0685	0.0644	0.0625	0.0661

Remember: This is **zero-shot** (Chronos) versus **dedicated** deep-learning models

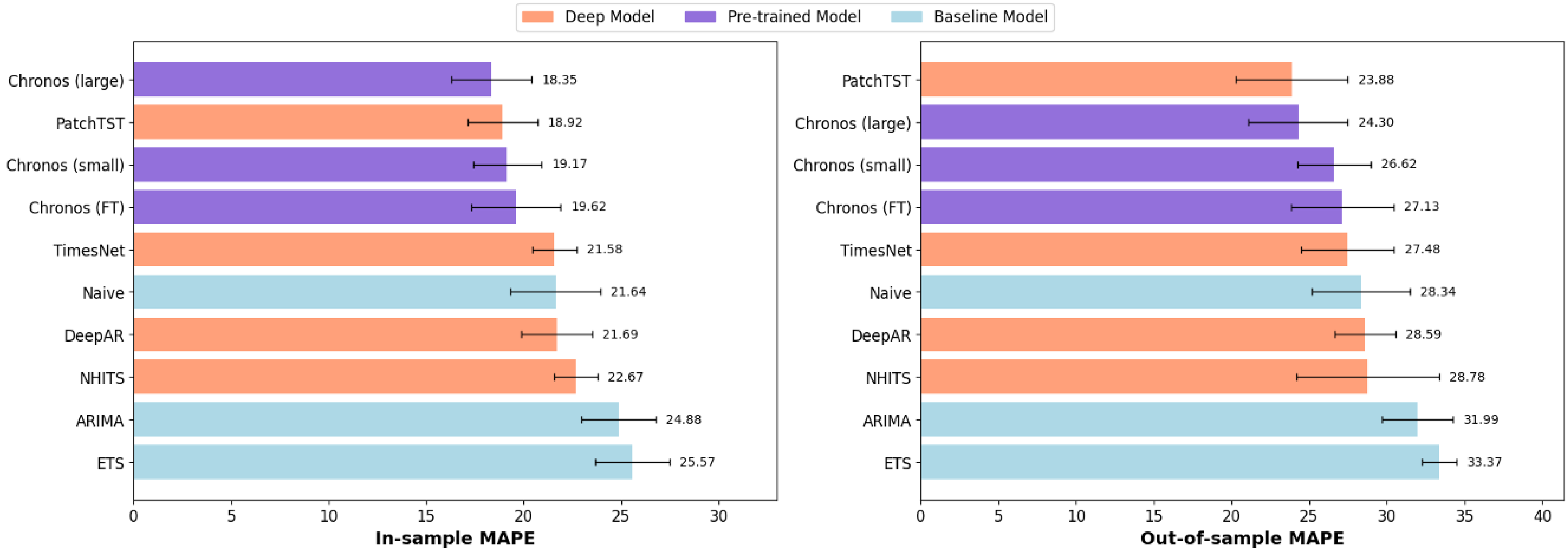
Results: Accuracy of in-sample forecasting

When looking at the **MAPE** and a **30-day** forecasting horizon:



Results: Accuracy of in-sample forecasting

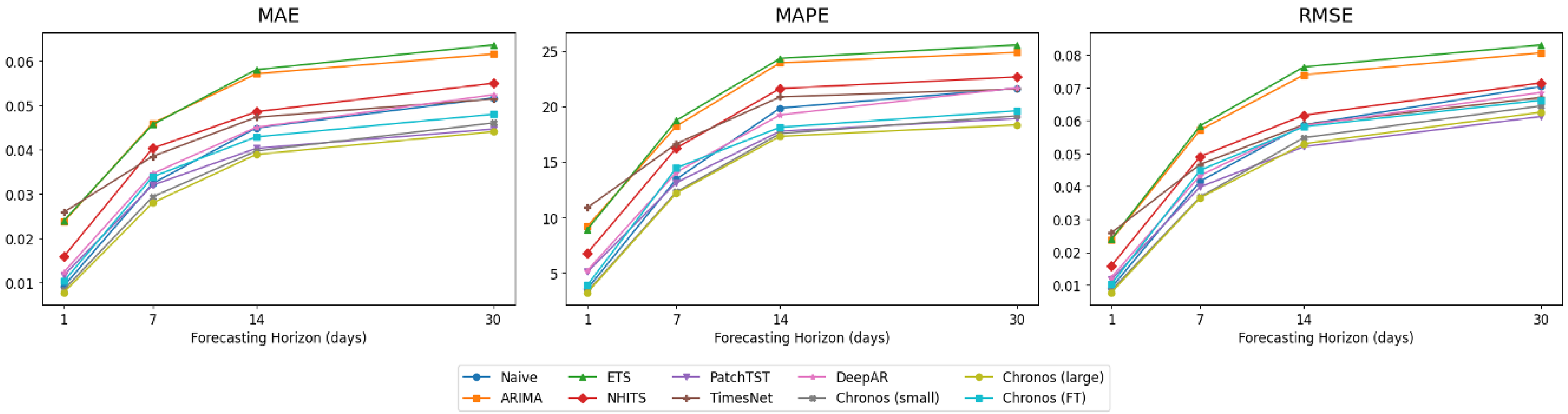
When comparing in-sample performance vs out-sample performance (**MAPE**, **30-day horizon**):



Interestingly: Performance of statistical and zero-shot models decreases, indicating "more difficult" batch

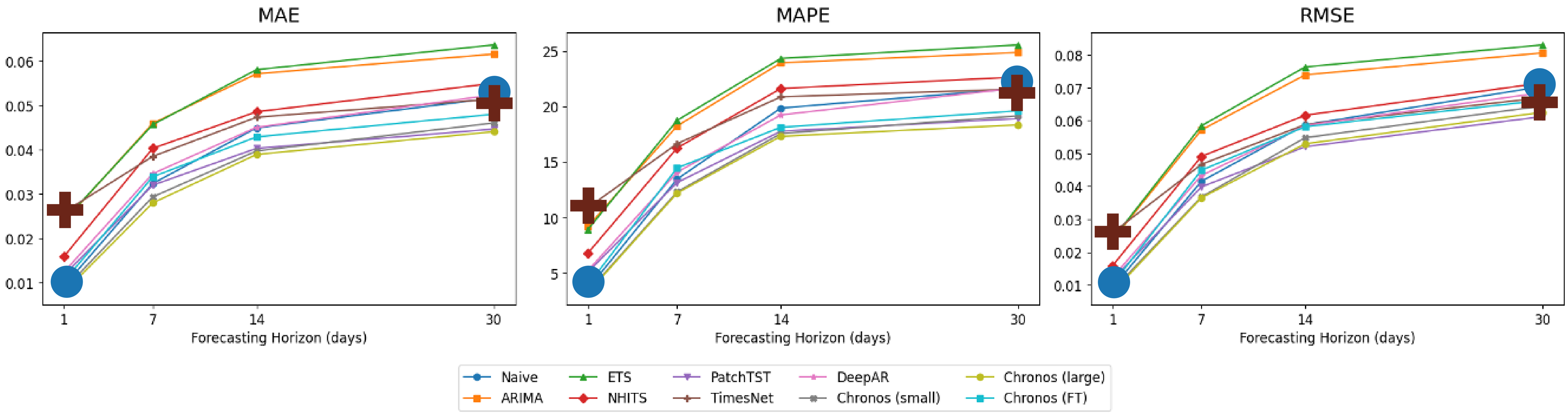
Results: Accuracy of in-sample forecasting

When looking at all metrics over the four forecasting horizons:



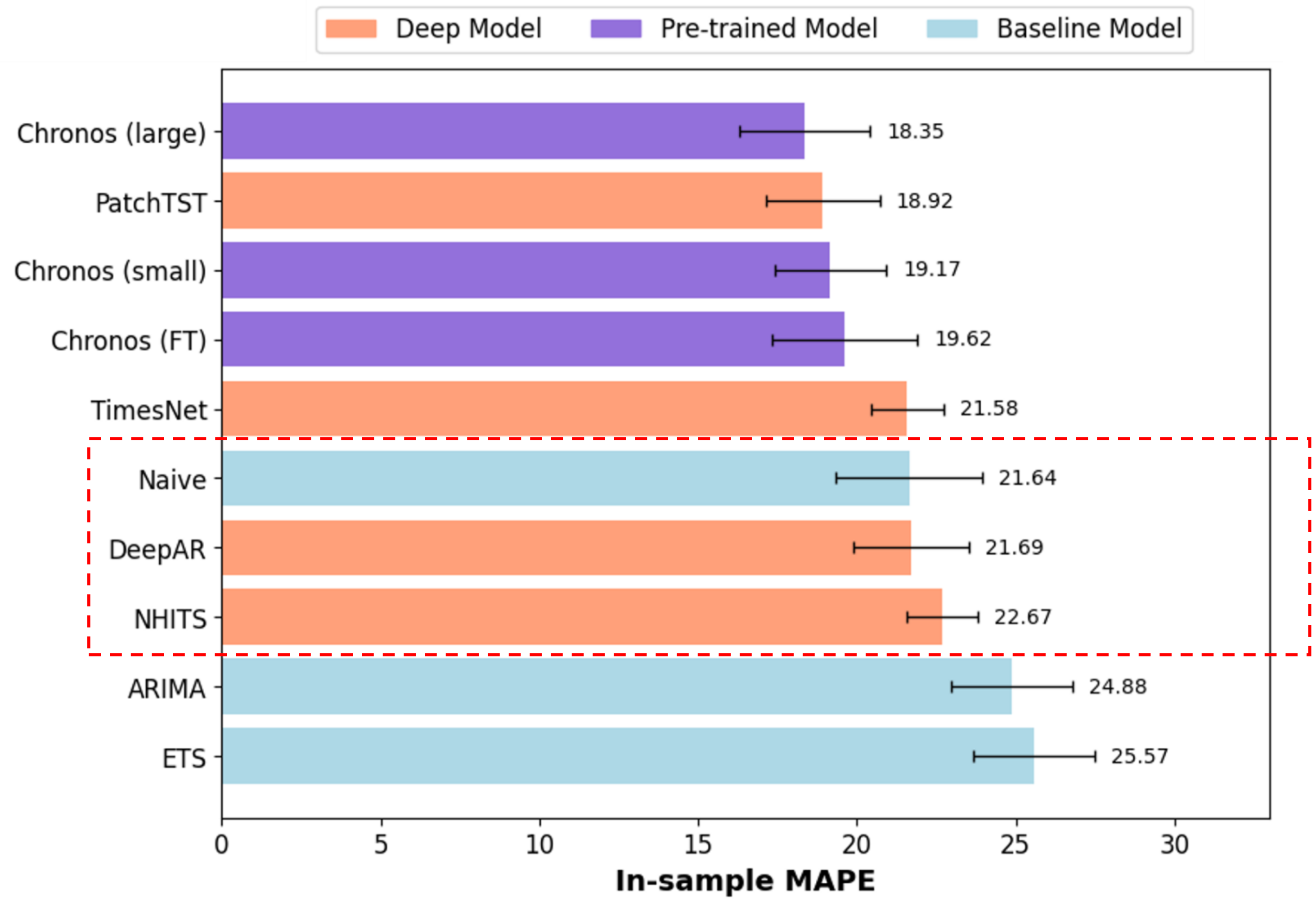
Results: Accuracy of in-sample forecasting

When looking at all metrics over the four forecasting horizons:



Notice: The **Naïve** ● model starts well, but its performance deteriorates quicker than other models (✚)

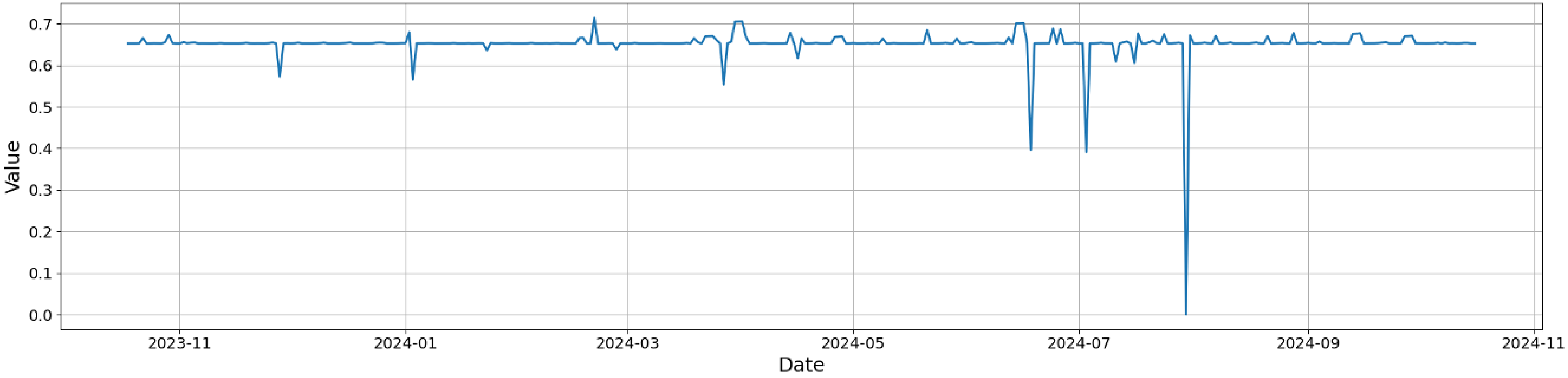
Discussion: Differences in characteristics of time series



Naïve model outperforming dedicated deep-learners?

Discussion: Differences in characteristics of time series

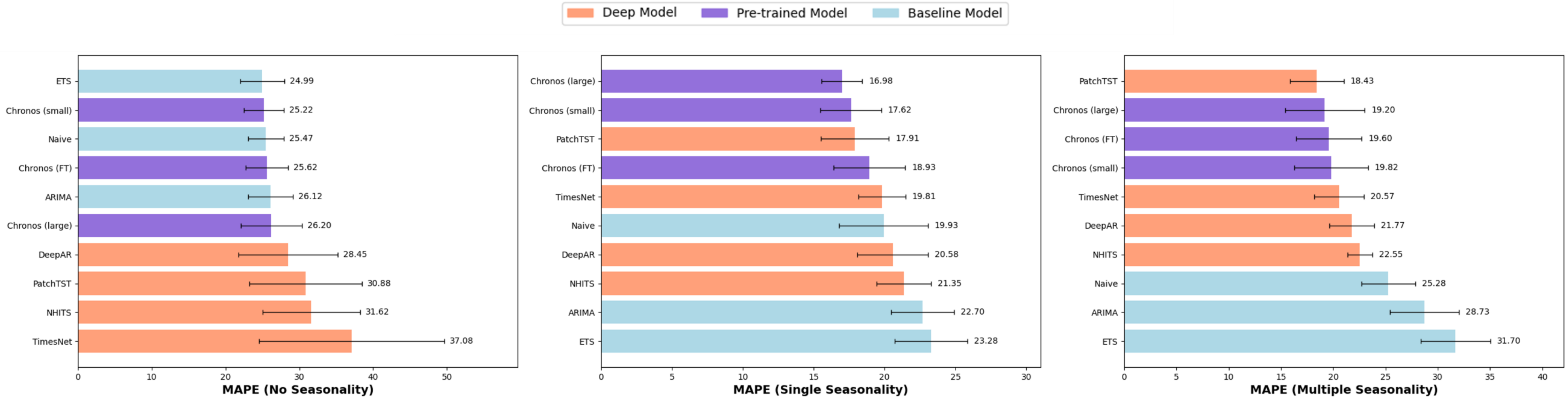
Time series where **Naïve** model performs **extremely** well:



First careful conclusion: Different time series require different models (ensemble?)

Discussion: Differences in characteristics of time series

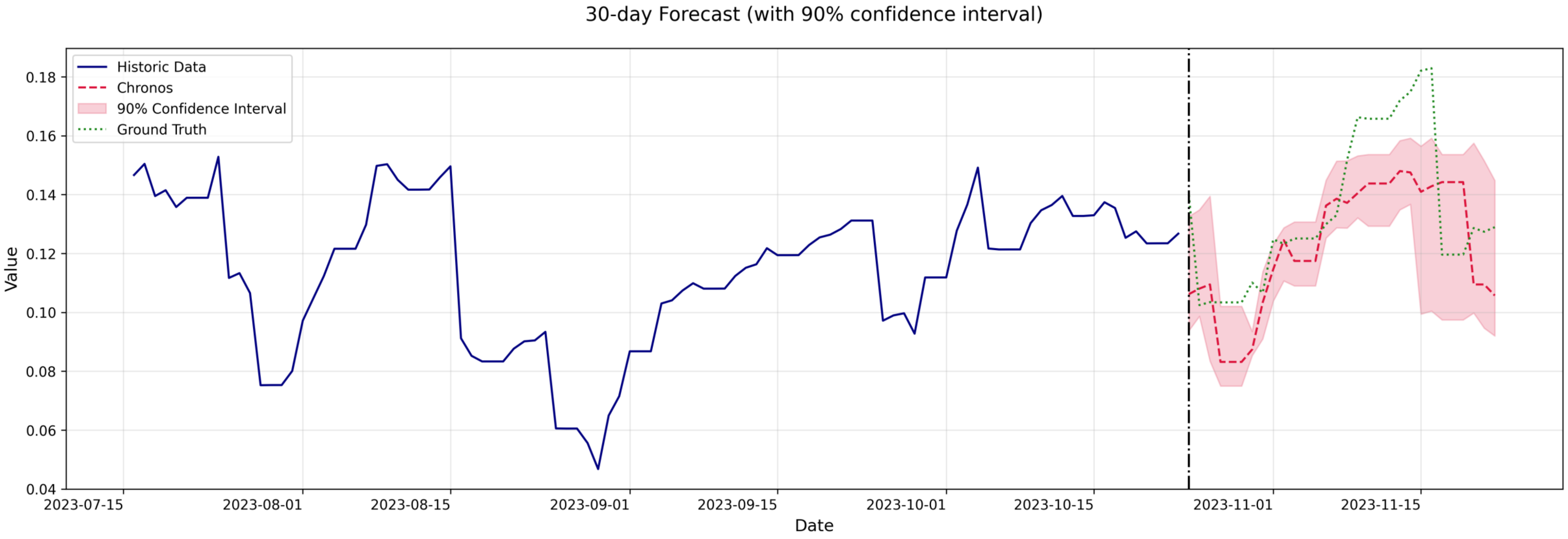
Comparing performance on time series with different types of **seasonality**:



First careful conclusion: Different time series require different models (ensemble?)

Analysis: Confidence Interval Reliability

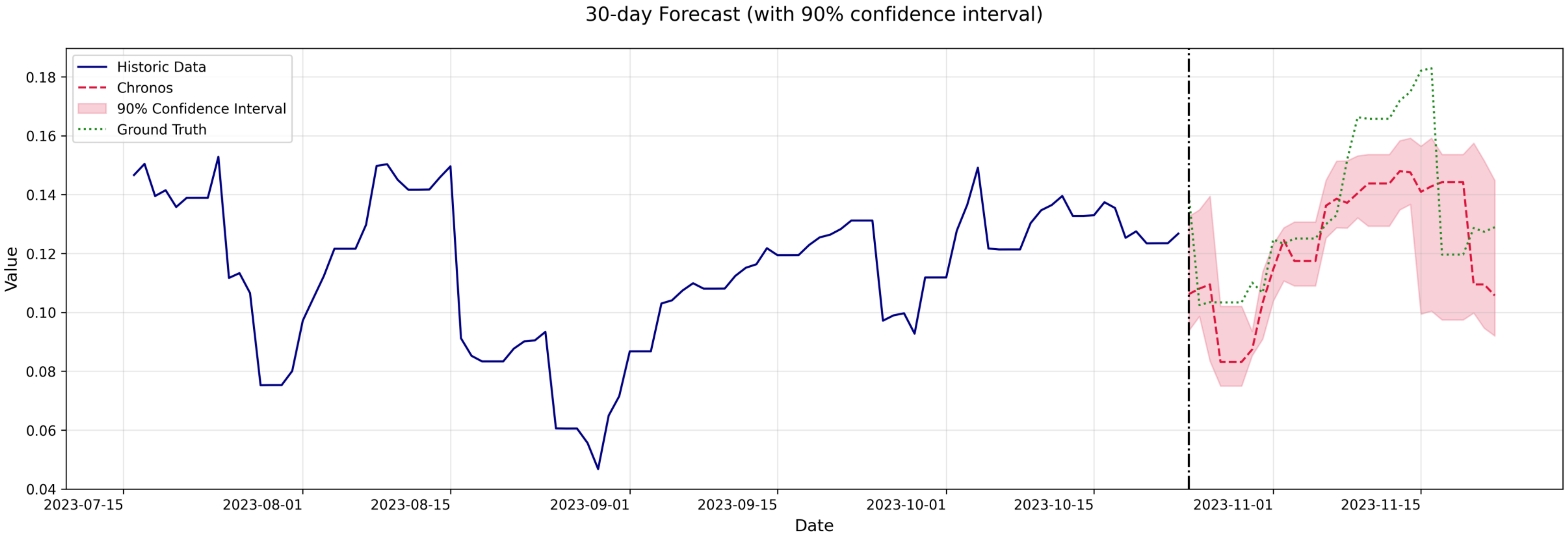
All non-Naive models can make **probabilistic forecasts** (60-, 70-, 80- and 90% confidence intervals):



Analysis: Confidence Interval Reliability

All non-Naive models can make **probabilistic forecasts** (60-, 70-, 80- and 90% confidence intervals):

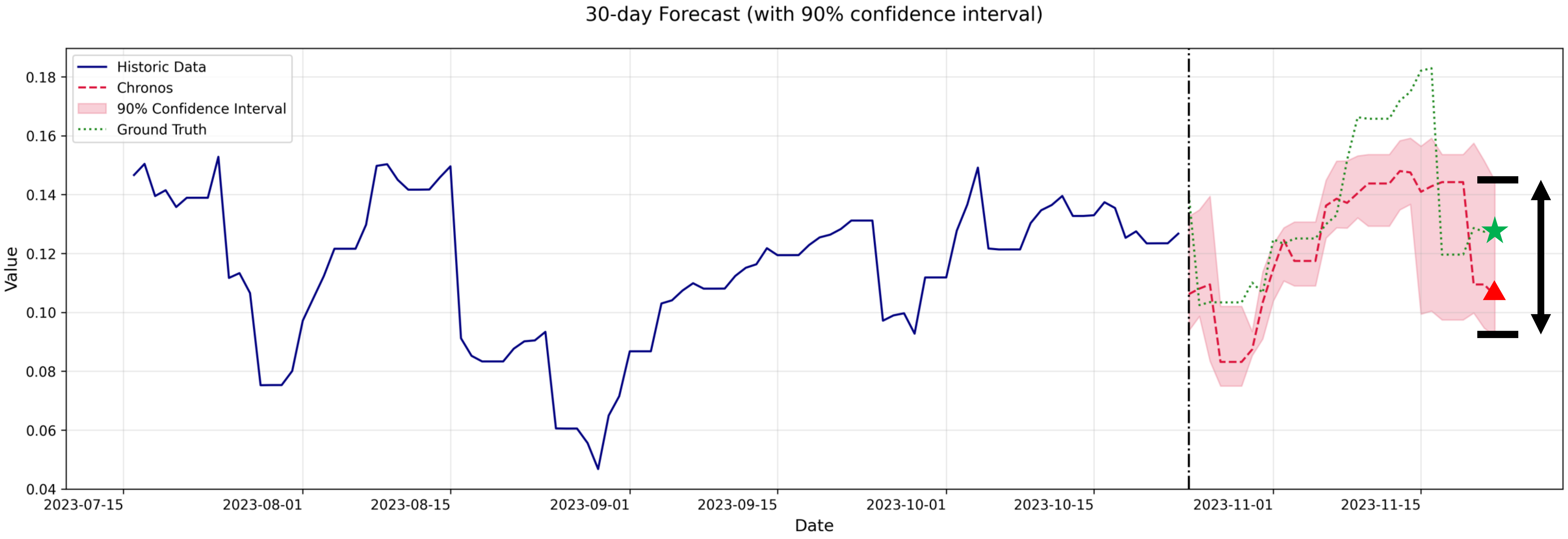
Question: How well *aligned* are these confidence intervals? How *honest*?



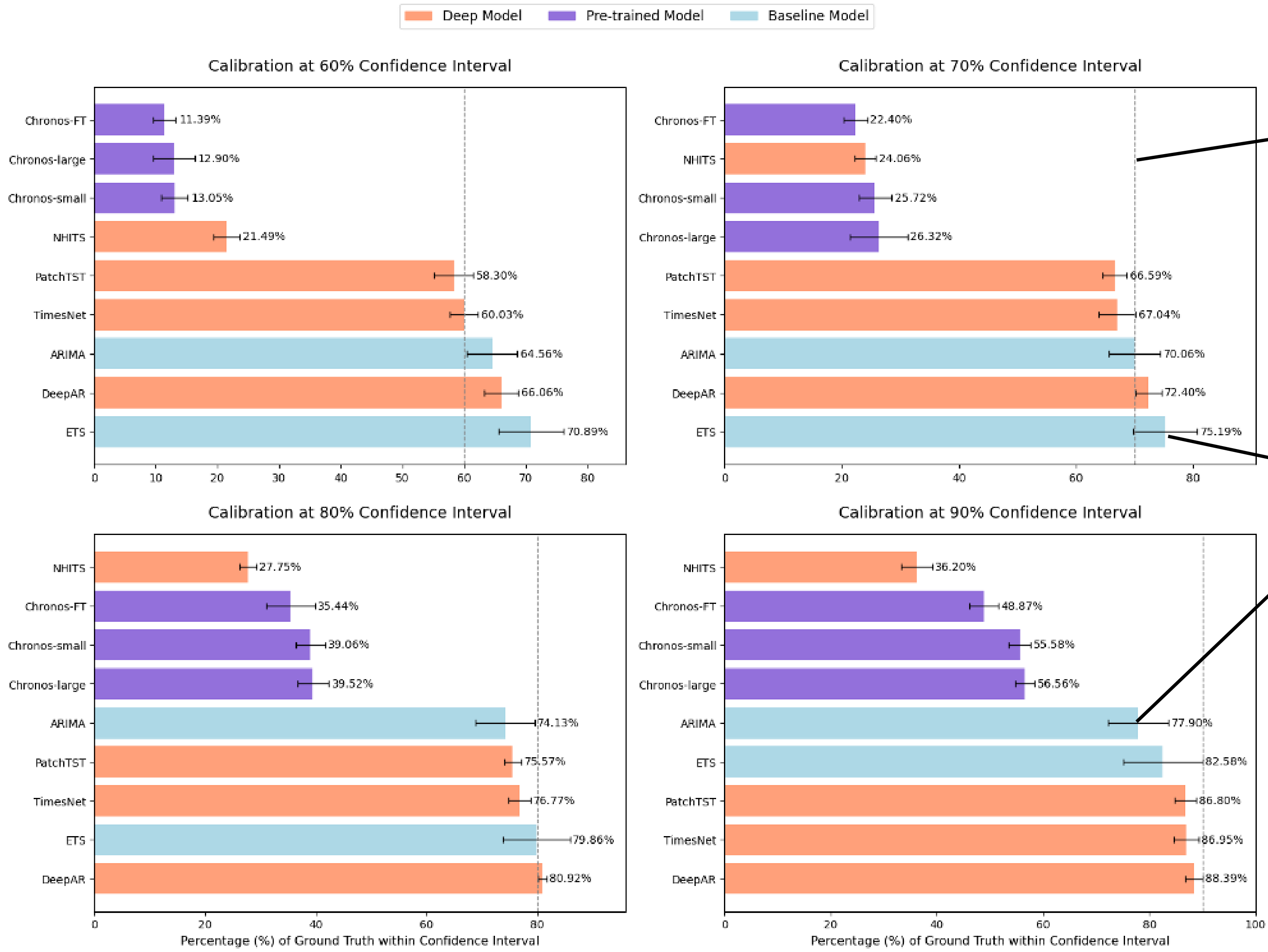
Analysis: Confidence Interval Reliability

All non-Naive models can make **probabilistic forecasts** (60-, 70-, 80- and 90% confidence intervals):

Question: How well *aligned* are these confidence intervals? How *honest*?



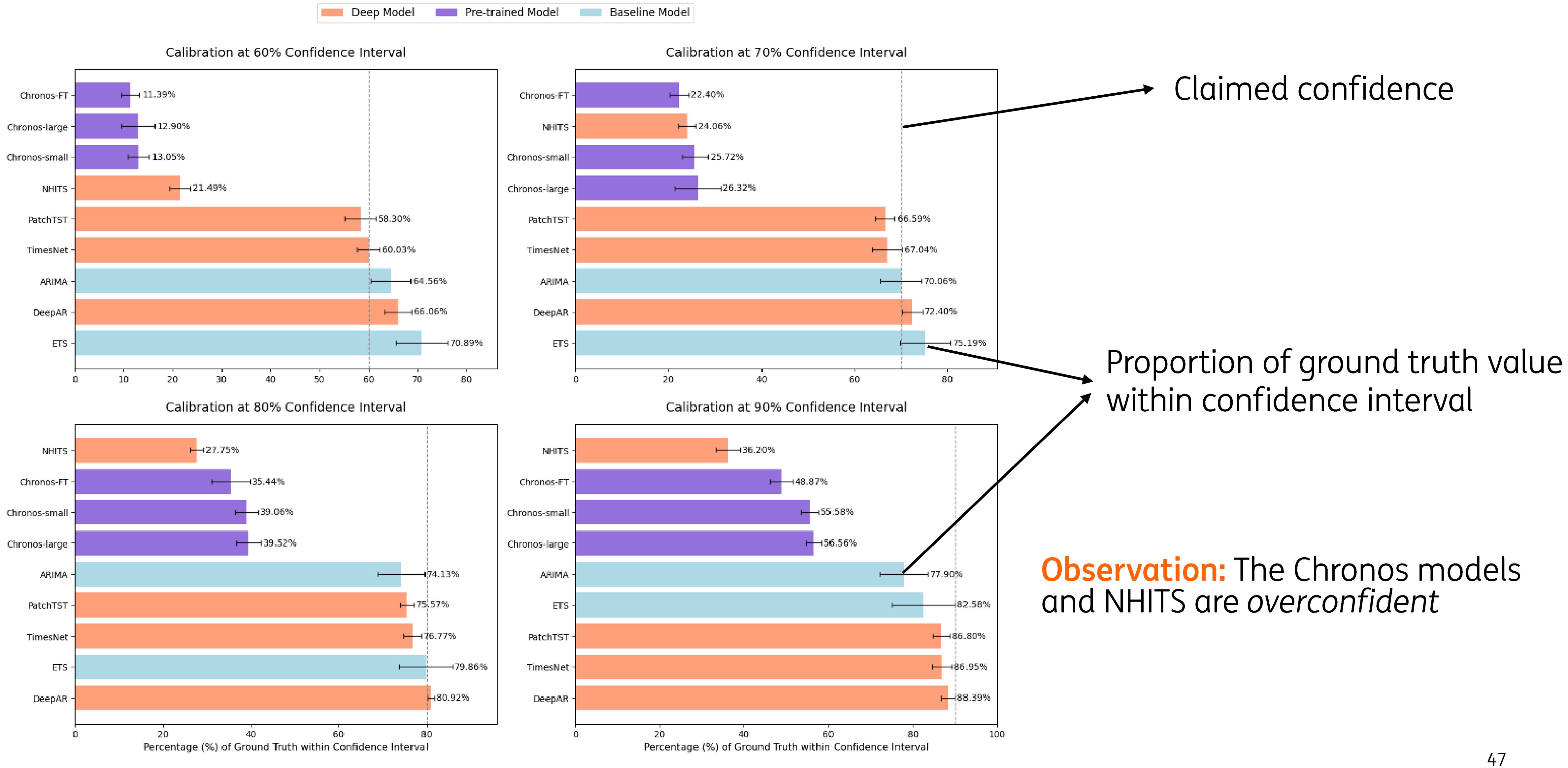
Analysis: Confidence Interval Reliability



Claimed confidence

Proportion of ground truth value within confidence interval

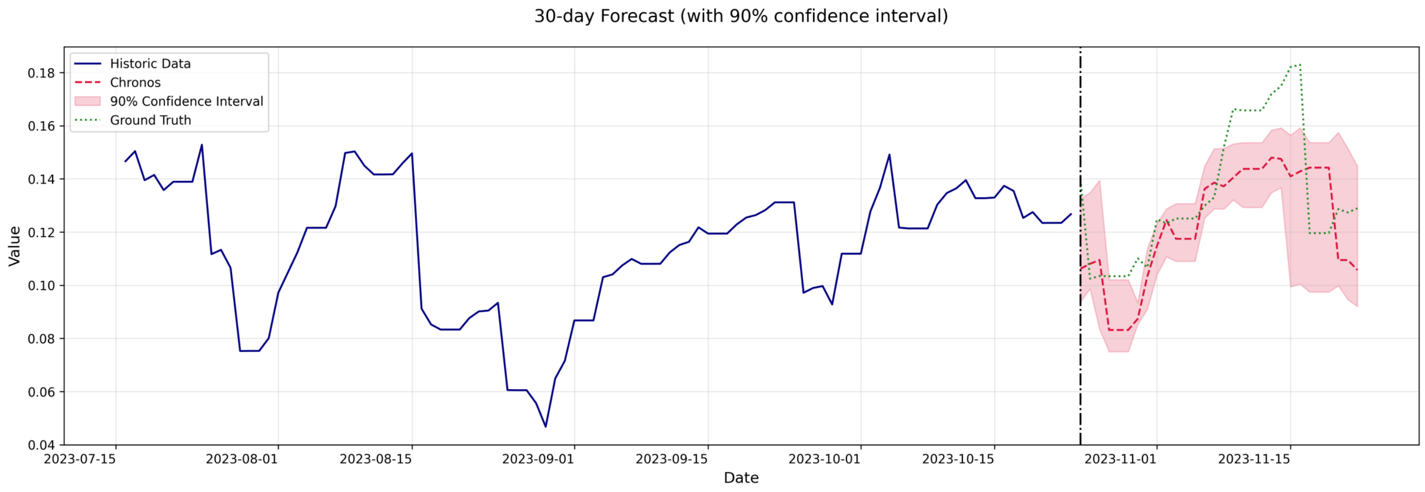
Analysis: Confidence Interval Reliability



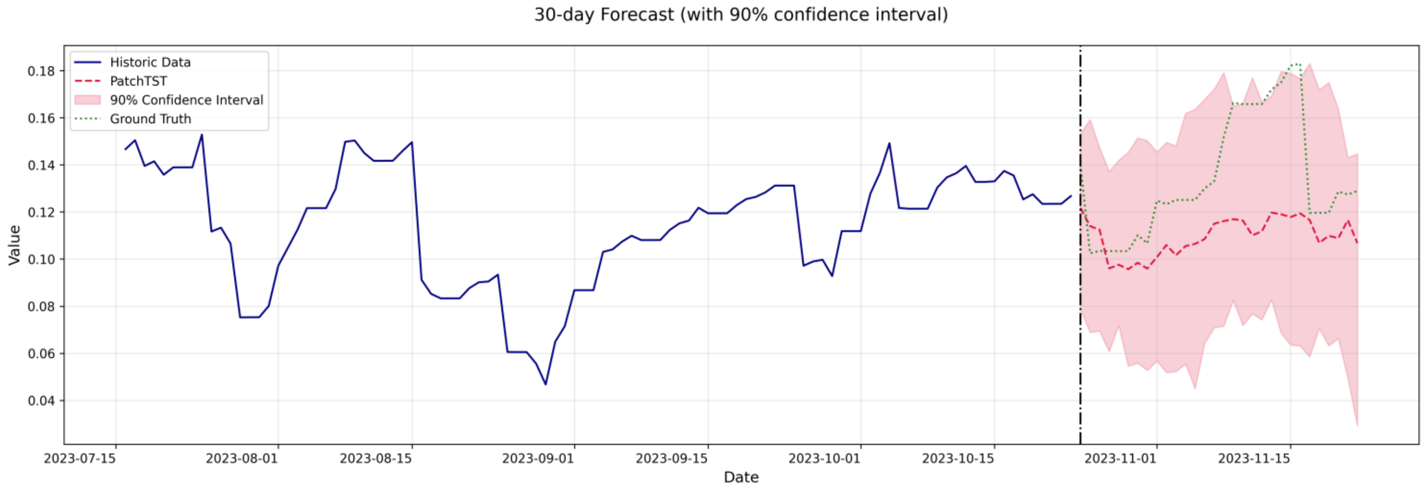
Observation: The Chronos models and NHITS are *overconfident*

Discussion: Confidence Interval Reliability

We can take a deeper dive into the confidence intervals sizes of each model



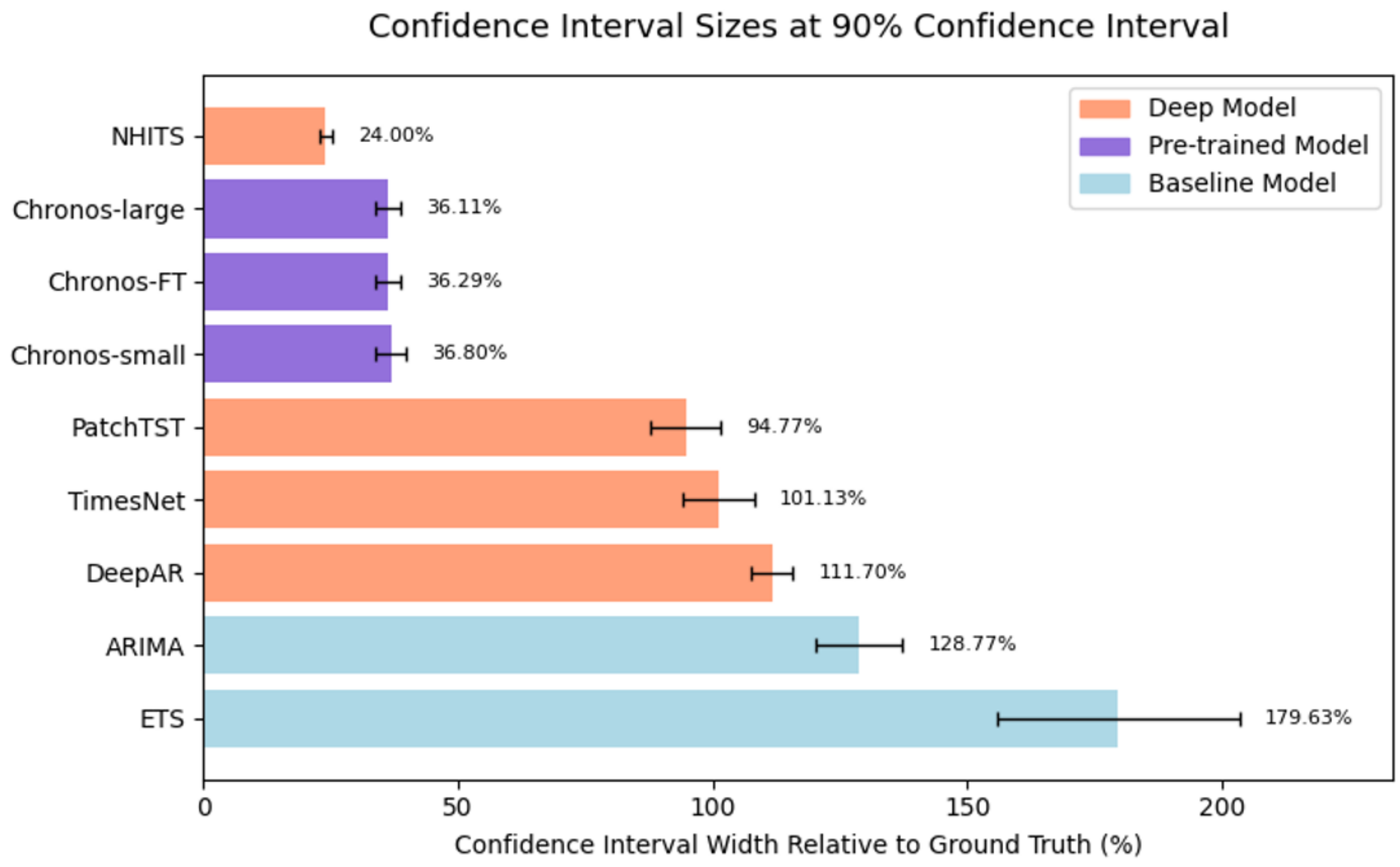
Chronos (Finetuned)



PatchTST

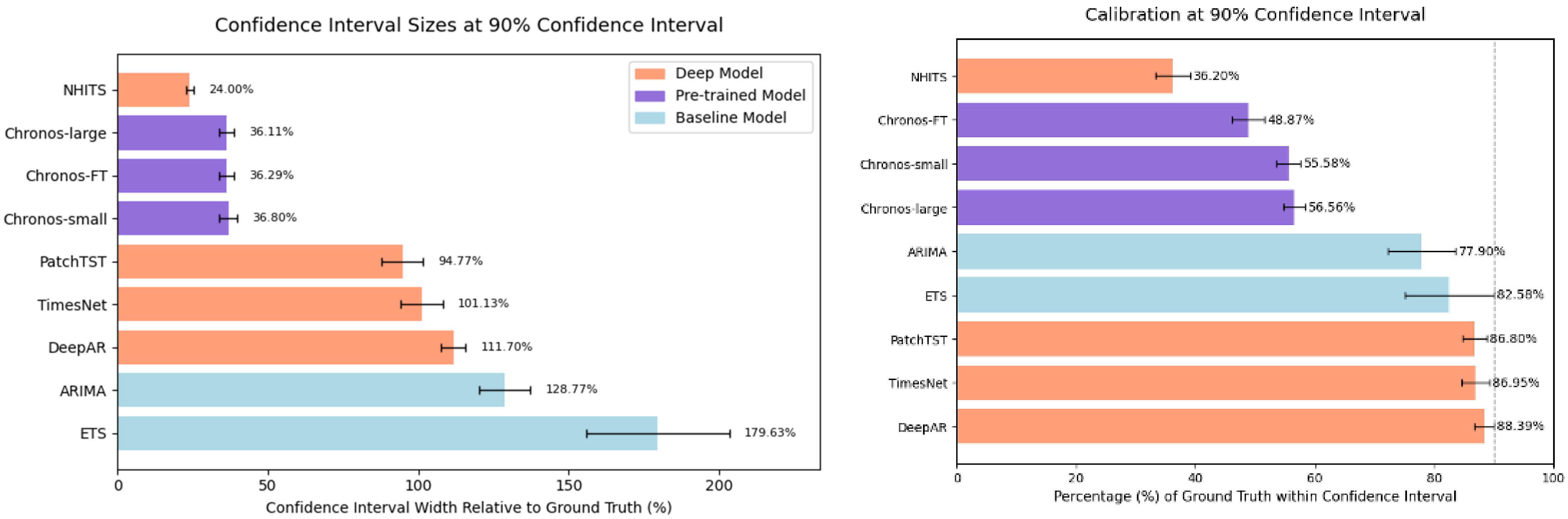
Discussion: Confidence Interval Reliability

We can take a deeper dive into the confidence intervals sizes of each model



Discussion: Confidence Interval Reliability

We can take a deeper dive into the confidence intervals sizes of each model



Conclusion:

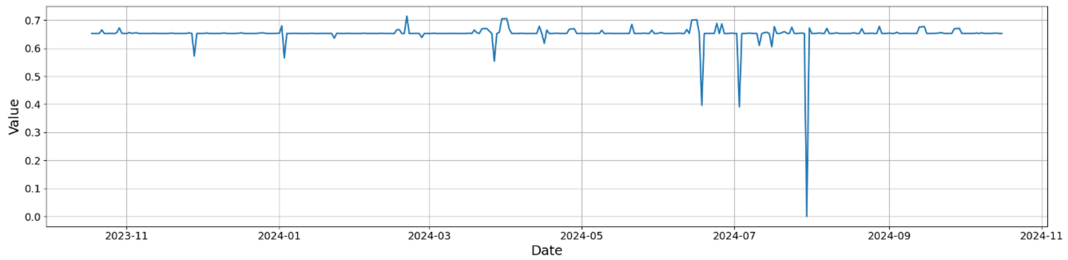
- **Foundation Models** for time series can **outperform** dedicated **deep learners** and **statistical models** by using zero-shot forecasting

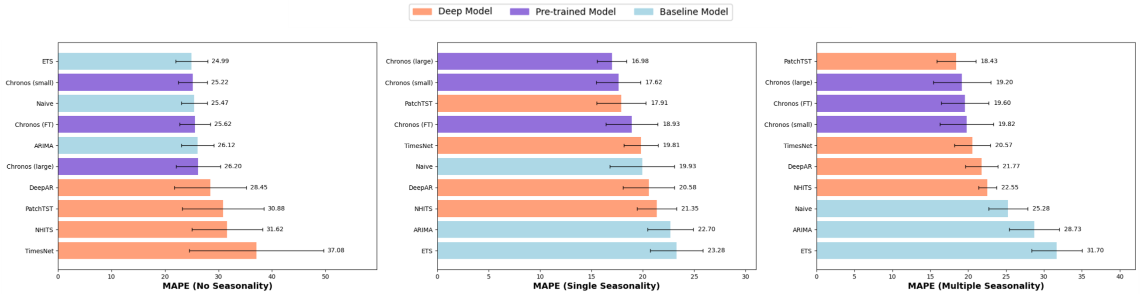
		Statistical			Deep Learning				Foundation Models		
Metric	Horizon	Naive	ARIMA	ETS	NHITS	PatchTST	TimesNet	DeepAR	Chronos-S	Chronos-L	Chronos-FT
MAE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	0.0083	**0.0077**	0.0102
	7 days	0.0323	0.0459	0.0456	0.0403	0.0320	0.0385	0.0346	0.0293	**0.0280**	0.0338
	14 days	0.0449	0.0571	0.0580	0.0485	0.0403	0.0473	0.0450	0.0397	**0.0389**	0.0429
	30 days	0.0517	0.0616	0.0636	0.0550	0.0446	0.0514	0.0524	0.0460	**0.0440**	0.0480
MAPE	1 day	3.5840	9.2107	8.9328	6.7932	5.0918	10.8874	5.2234	3.3121	**3.2282**	3.9072
	7 days	13.4556	18.2455	18.7574	16.2607	13.1425	16.6501	14.0777	12.2991	**12.1936**	14.4643
	14 days	19.8646	23.9450	24.3448	21.6301	17.7806	20.8744	19.2472	17.5705	**17.3186**	18.1345
	30 days	21.6435	24.8841	25.5652	22.6731	18.9188	21.5849	21.6900	19.1657	**18.3518**	19.6172
RMSE	1 day	0.0091	0.0237	0.0239	0.0158	0.0116	0.0258	0.0123	0.0083	**0.0077**	0.0102
	7 days	0.0415	0.0571	0.0584	0.0491	0.0398	0.0467	0.0433	0.0368	**0.0365**	0.0449
	14 days	0.0587	0.0739	0.0763	0.0616	**0.0520**	0.0588	0.0585	0.0548	0.0529	0.0581
	30 days	0.0704	0.0806	0.0830	0.0714	**0.0612**	0.0670	0.0685	0.0644	0.0625	0.0661

Conclusion:

- **Foundation Models** for time series can **outperform** dedicated **deep learners** and **statistical models** by using zero-shot forecasting

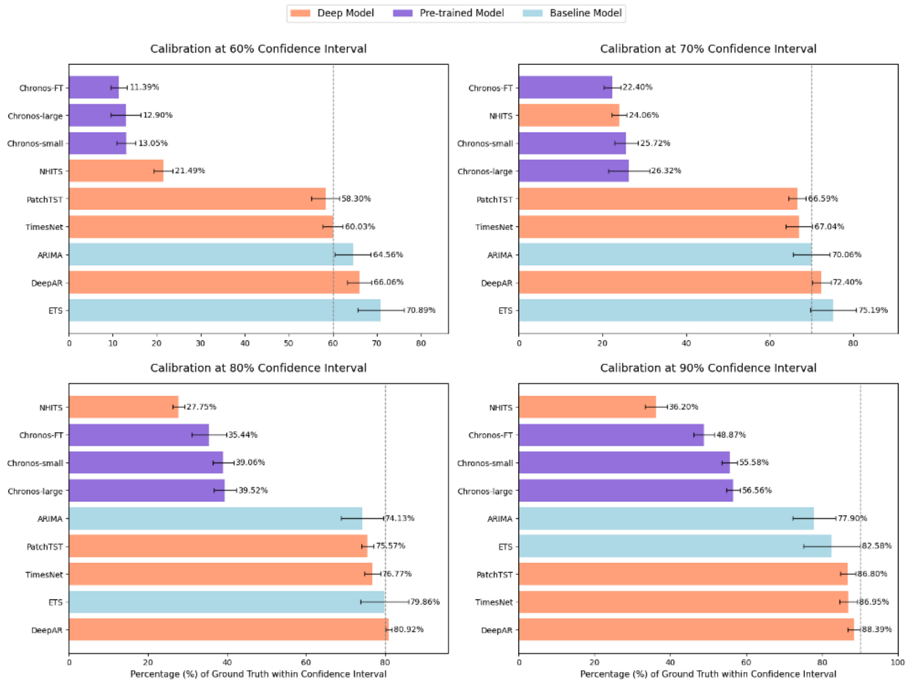
- There are many different time series each with their own characteristics. A one-fits-all model is difficult to achieve





Conclusion:

- **Foundation Models** for time series can **outperform** dedicated **deep learners** and **statistical models** by using zero-shot forecasting
- There are many different time series each with their own characteristics. A one-fits-all model is difficult to achieve
- The probabilistic output of the foundation model showed large inconsistencies, inviting **further research** into **honesty** and **alignment**



Conclusion:

- **Foundation Models** for time series can **outperform** dedicated **deep learners** and **statistical models** by using zero-shot forecasting
- There are many different time series each with their own characteristics. LLM'
- The probabilistic output of the foundation model showed large inconsistencies, inviting **further research** into **honesty** and **alignment**
- Working on a dedicated **Forecasting** repository. Not public yet, see: github.com/didiermerk

Thank you!

do your thing

References:

[1] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters, 2024. URL https://arxiv.org/abs/ 2310.07820.
[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

do your thing