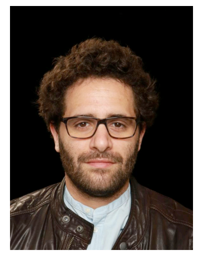
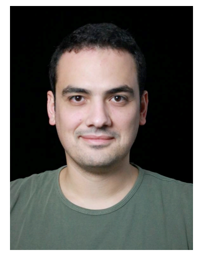
CHRONOS

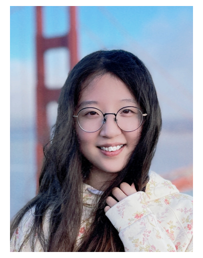
Learning the Language of Time Series

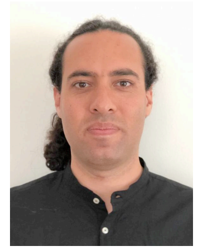
Abdul Fatir Ansari *

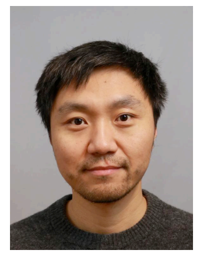
Applied Scientist, AWS

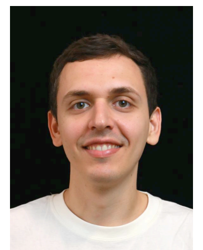

Lorenzo Stella*

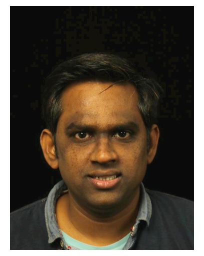

Caner Turkmen


Xiyuan Zhang


Pedro Mercado


Huibin Shen


Oleksandr Shchur


Syama Rangapuram


Sebastian Arango


Shubham Kapoor

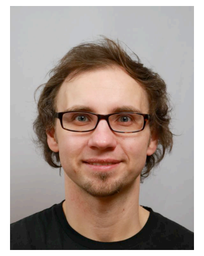

Jasper Zschiegner


Danielle Robinson


Andrew Wilson


Kari Torkkola


Michael Mahoney

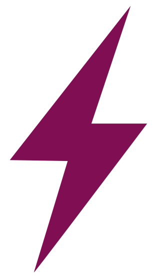

Michael B.-Schneider


Bernie Wang

* Equal Contribution

What is Forecasting?

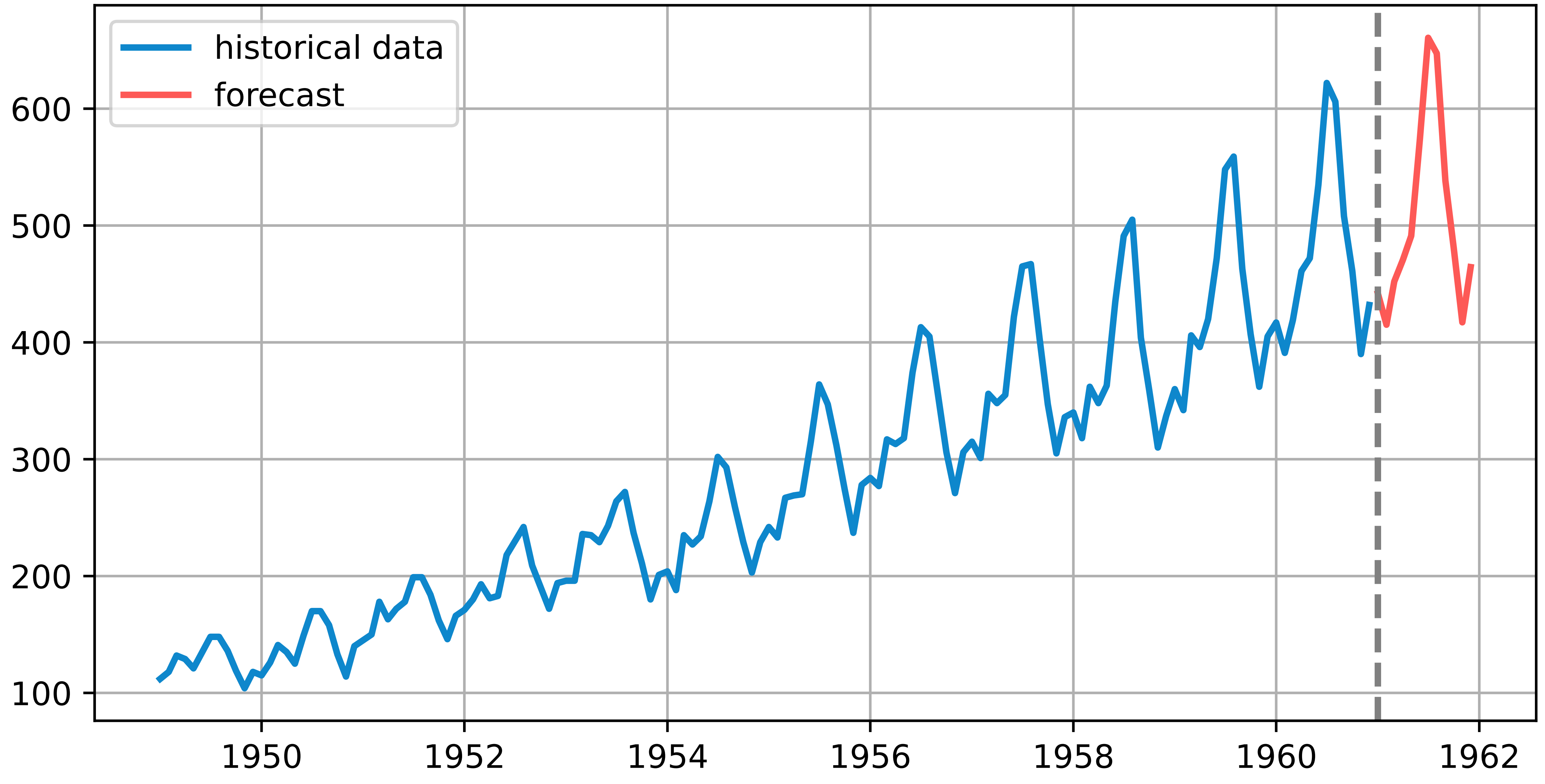
Predict the future behavior of a time series given its past



Energy

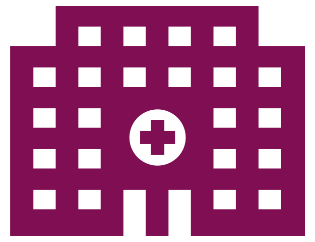
Finance

Weather

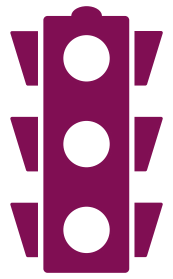


Monthly US Airline Passengers from 1949 to 1960

Retail



Healthcare



Traffic

$$f : (x_1, x_2, \ldots, x_T) \rightarrow \hat{x}_{T+1}, \hat{x}_{T+2}, \ldots, \hat{x}_{T+h}$$

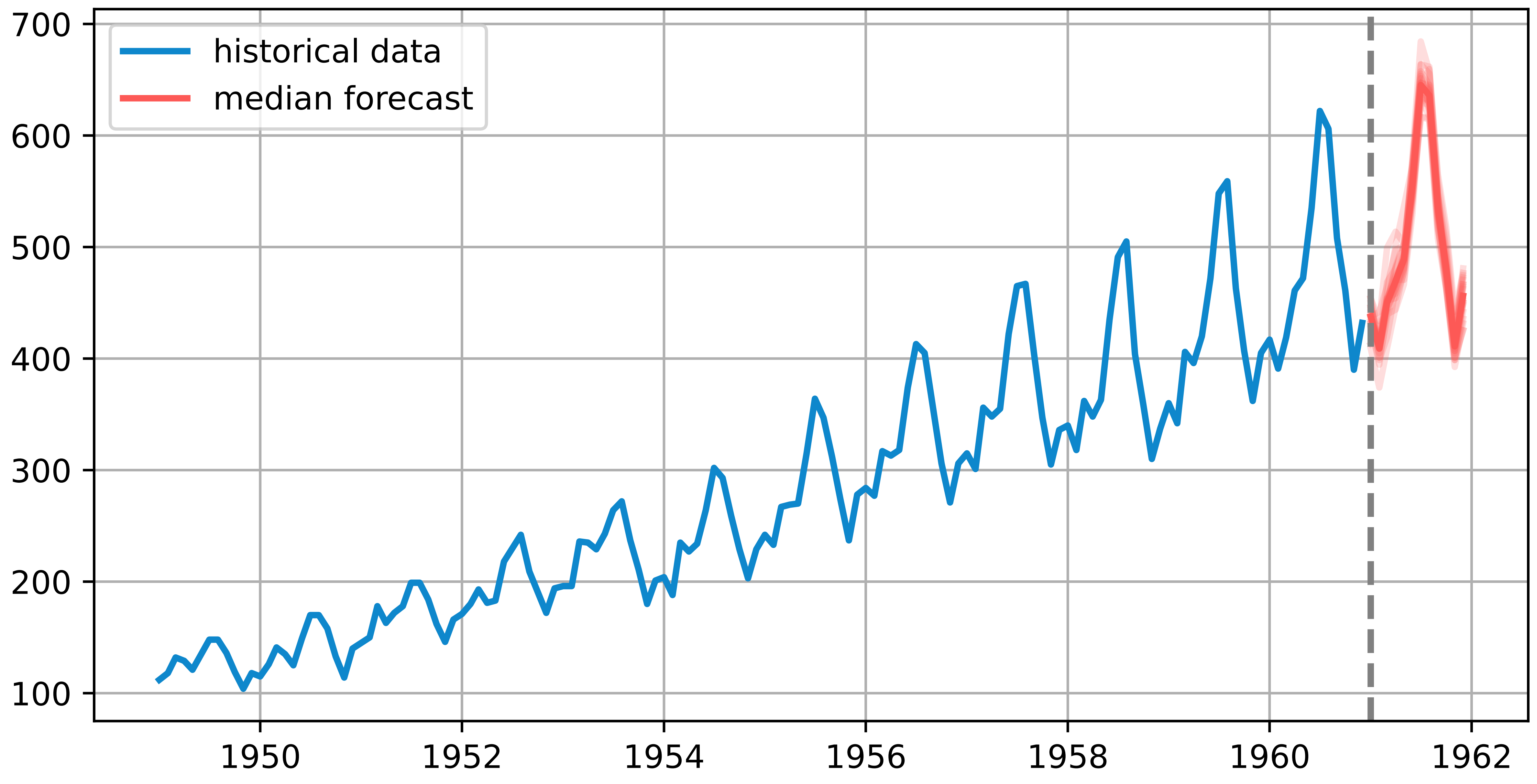
What is Forecasting?

Predict the future behavior of a time series given its past

Energy

Finance

Weather



Monthly US Airline Passengers from 1949 to 1960

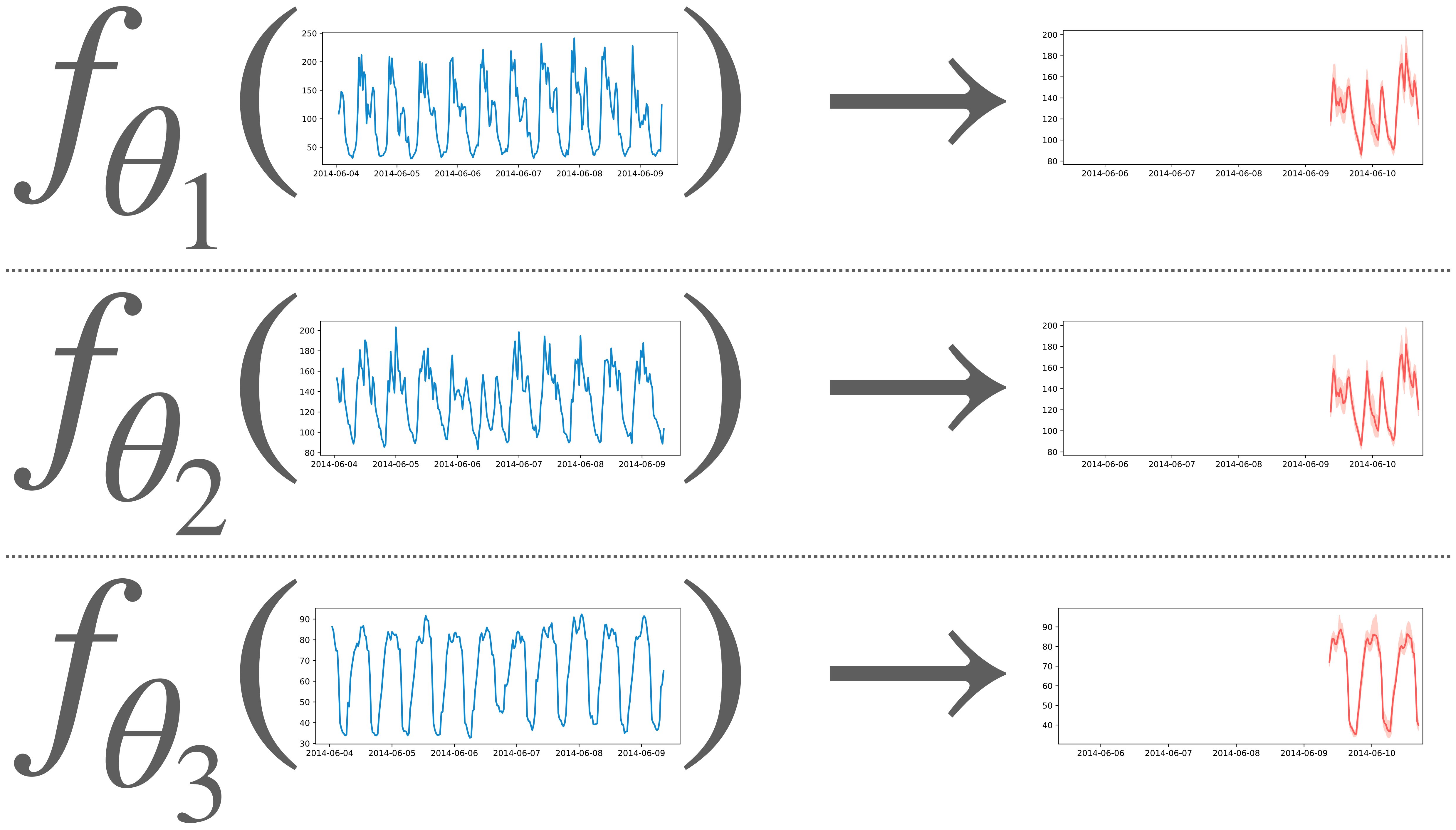
Retail

Healthcare

Traffic

$$p(x_{T+1}, x_{T+2}, \ldots, x_{T+h} \mid x_1, x_2, \ldots, x_T)$$

Local Models



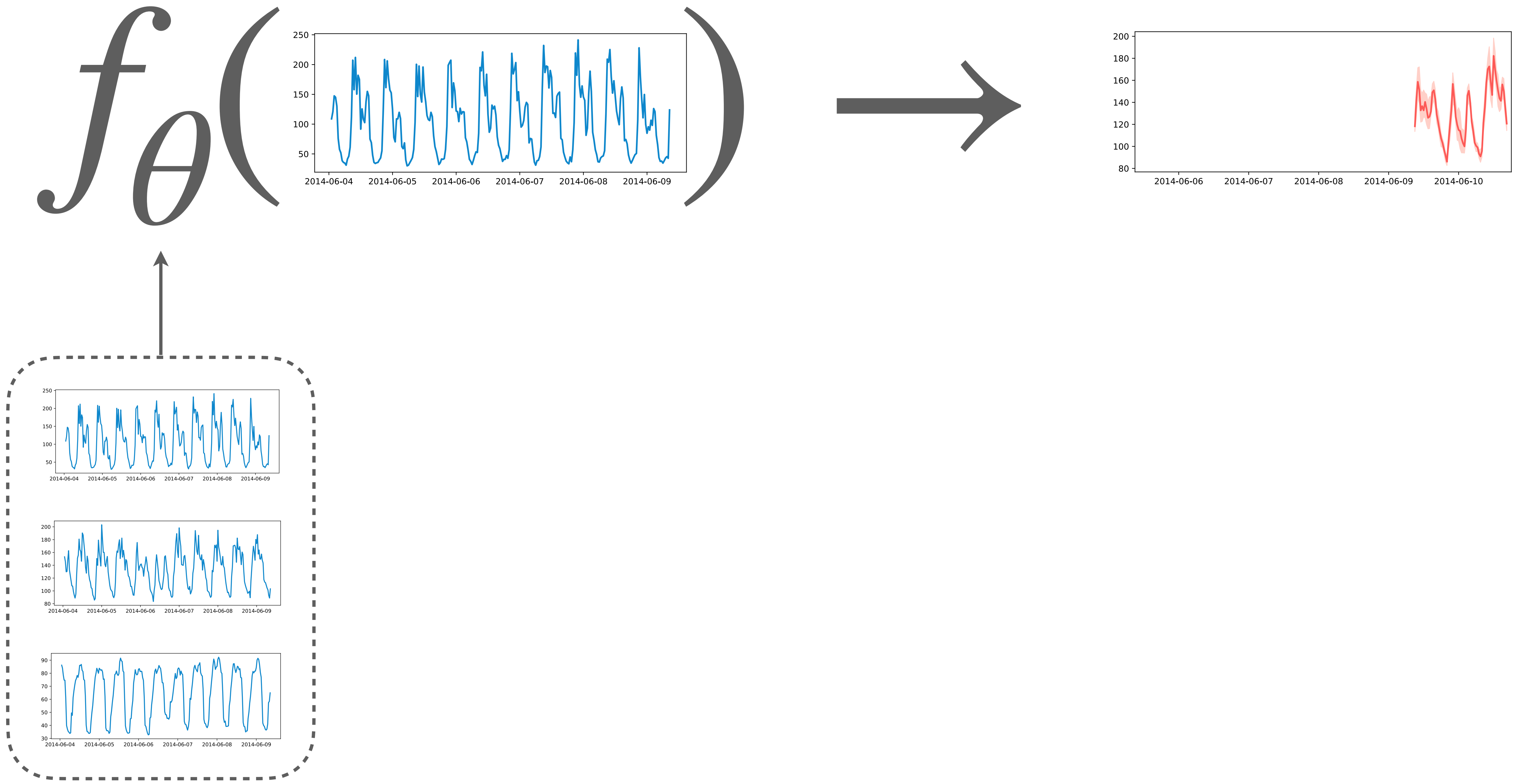
- Fit a **separate** model for each **individual** time series
- Examples: ETS, ARIMA, Theta

✔ *strong baselines (esp. limited data)*
✔ *often interpretable*
✘ *low modeling flexibility*
✘ *slow inference*

Classical statistical models *typically* belong to this category.

Januschowski, Tim, et al. "Criteria for classifying forecasting methods." *International Journal of Forecasting* 36.1 (2020)

Global Models



Training Corpus of
Related Time Series

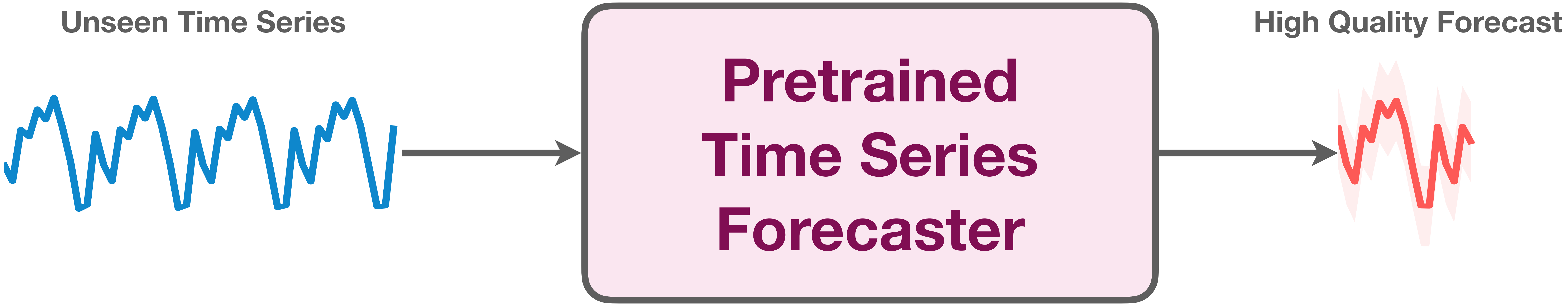
- Fit a **single** model for each **dataset** or task
- Examples: DeepAR, TFT, PatchTST
  - ✔ *high flexibility*
  - ✔ *fast inference*
  - ✘ *slow training*
  - ✘ *data hungry*

Deep learning models *typically* belong to this category.

Januschowski, Tim, et al. "Criteria for classifying forecasting methods." *International Journal of Forecasting* 36.1 (2020)

Wouldn't it be great if …

Time Series datasets have diverse **patterns**, **frequencies**, **history lengths**, **prediction horizons**, **missing values**, … → *long trial-and-error development cycles*



… we could develop a pretrained forecaster that performs well on unseen time series tasks?

Existing LLM-based Approaches

Text-based Prompting

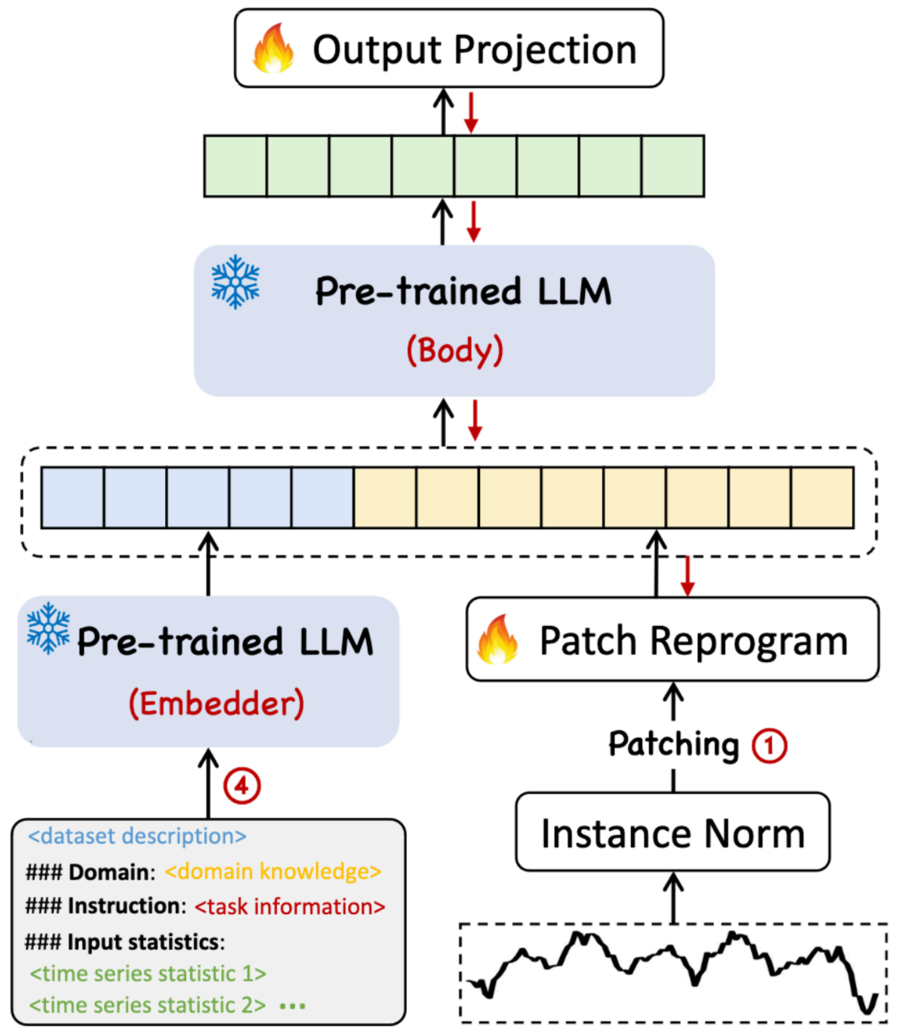
Fine-tuning Pretrained LLMs

Context: From August 16, 2019, Friday to August 30, 2019, Friday, the average temperature of region 110 was 78, 81, 83, 84, 84, 82, 83, 78, 77, 77, 74, 77, 78, 73, 76 degree on each day.
Question: What is the temperature going to be on August 31, 2019, Saturday?
Answer: The temperature will be 78 degree.

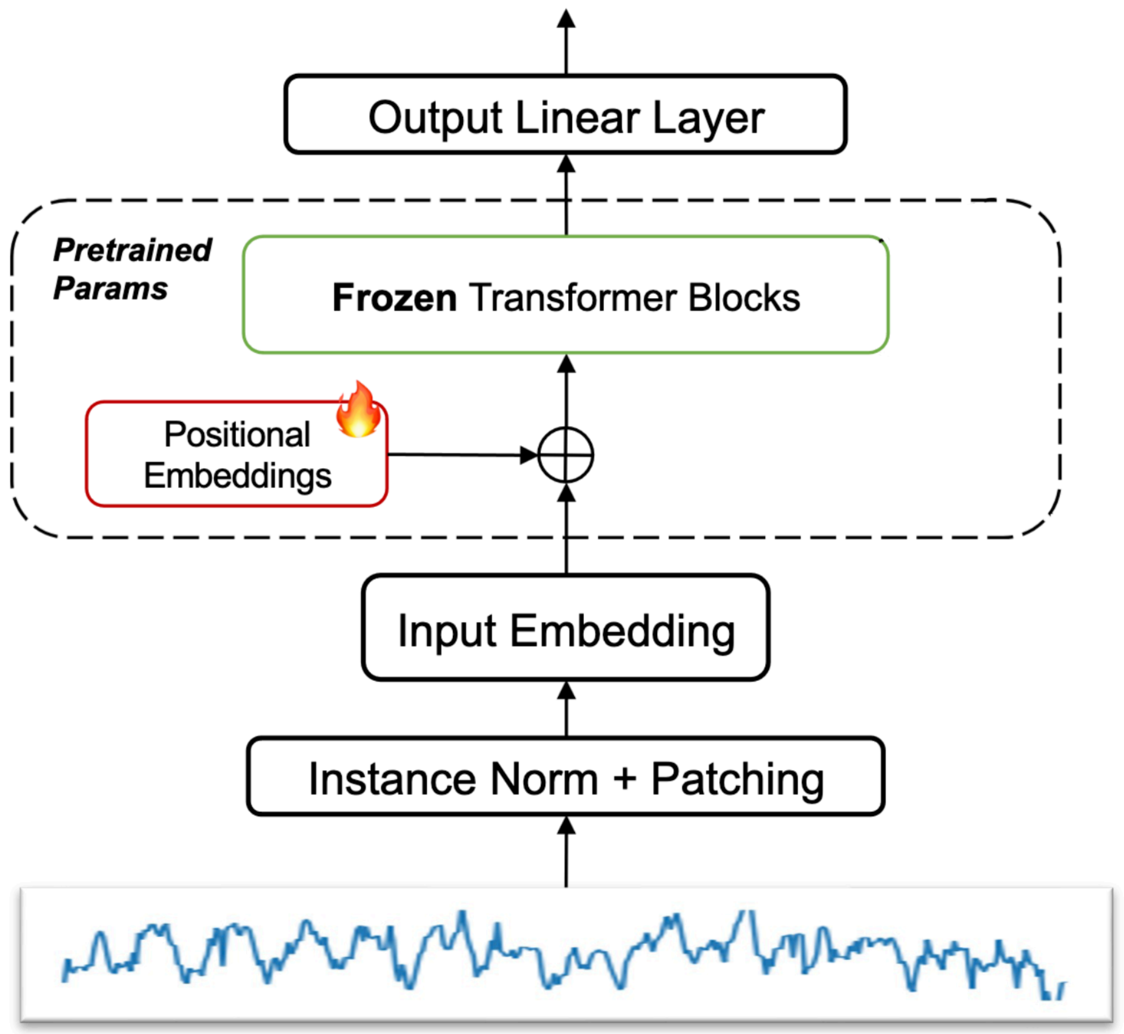
PromptCast

0.123, 1.23, 12.3, 123.0 → " 1 2 , 1 2 3 , 1 2 3 0 , 1 2 3 0 0"

LLMTime



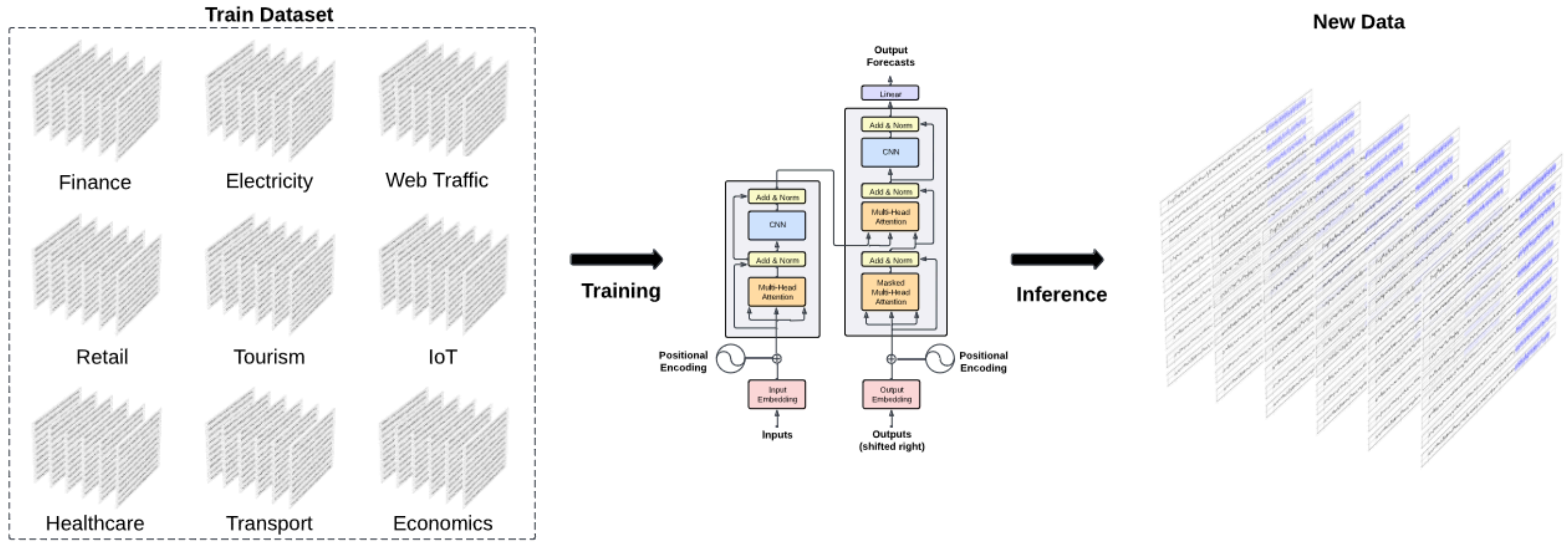
TimeLLM



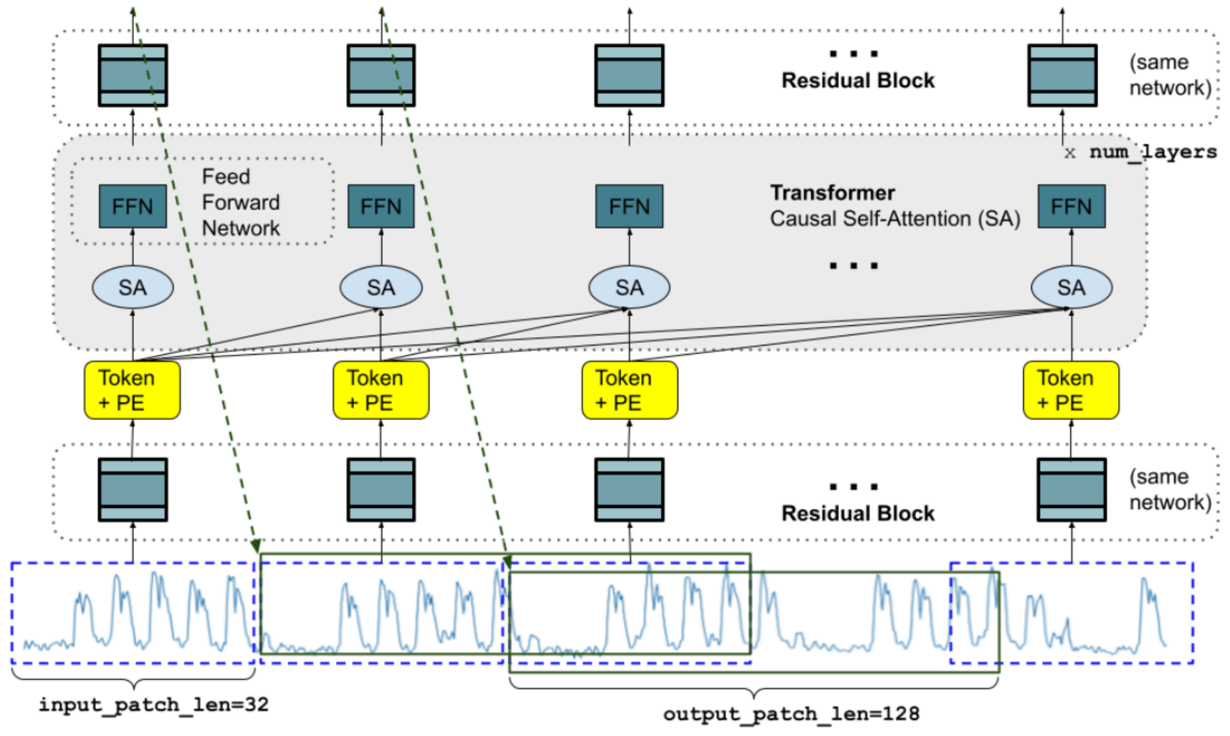
GPT4TS

Gruver, Nate, et al. "Large language models are zero-shot time series forecasters." *NeurIPS* (2023).
Xue, Hao, and Flora D. Salim. "PromptCast: A new prompt-based learning paradigm for time series forecasting." (2023).
Jin, Ming, et al. "Time-LLM: Time series forecasting by reprogramming large language models." *ICLR* (2023).
Zhou, Tian, et al. "One fits all: Power general time series analysis by pretrained lm." *NeurIPS* (2023).

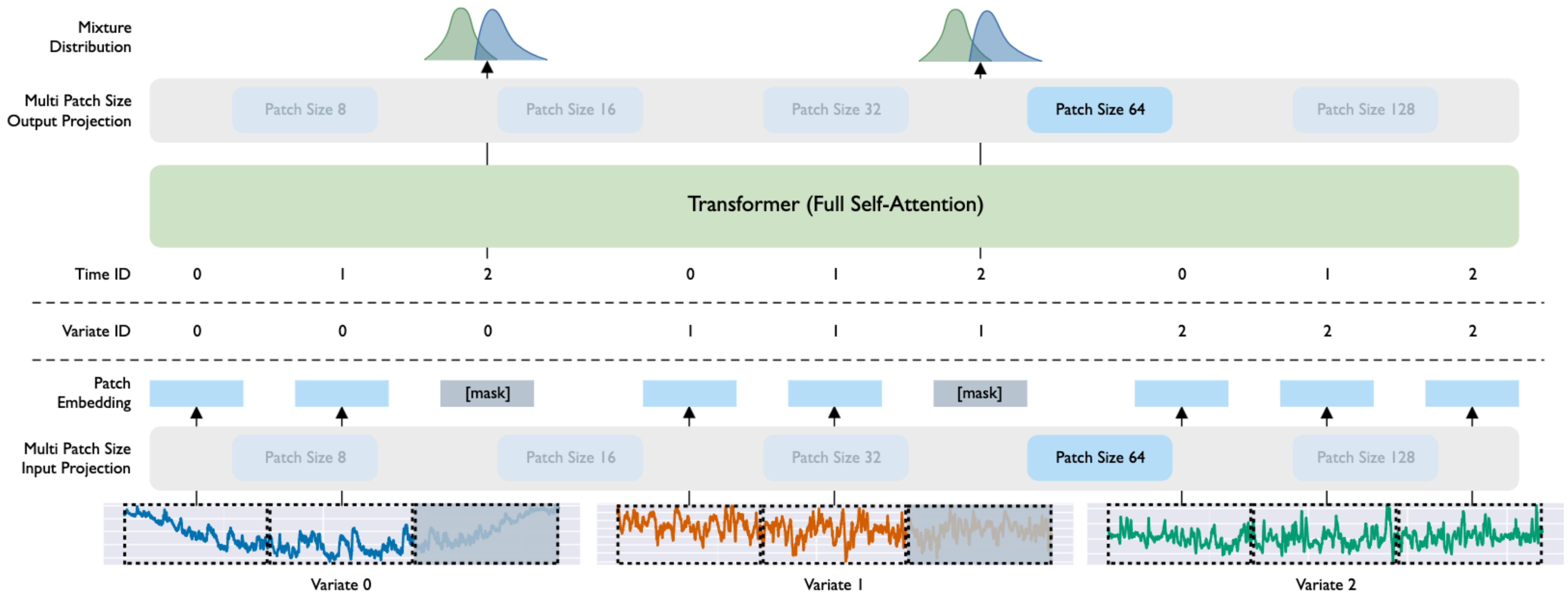
Recent Pretrained Time Series Models



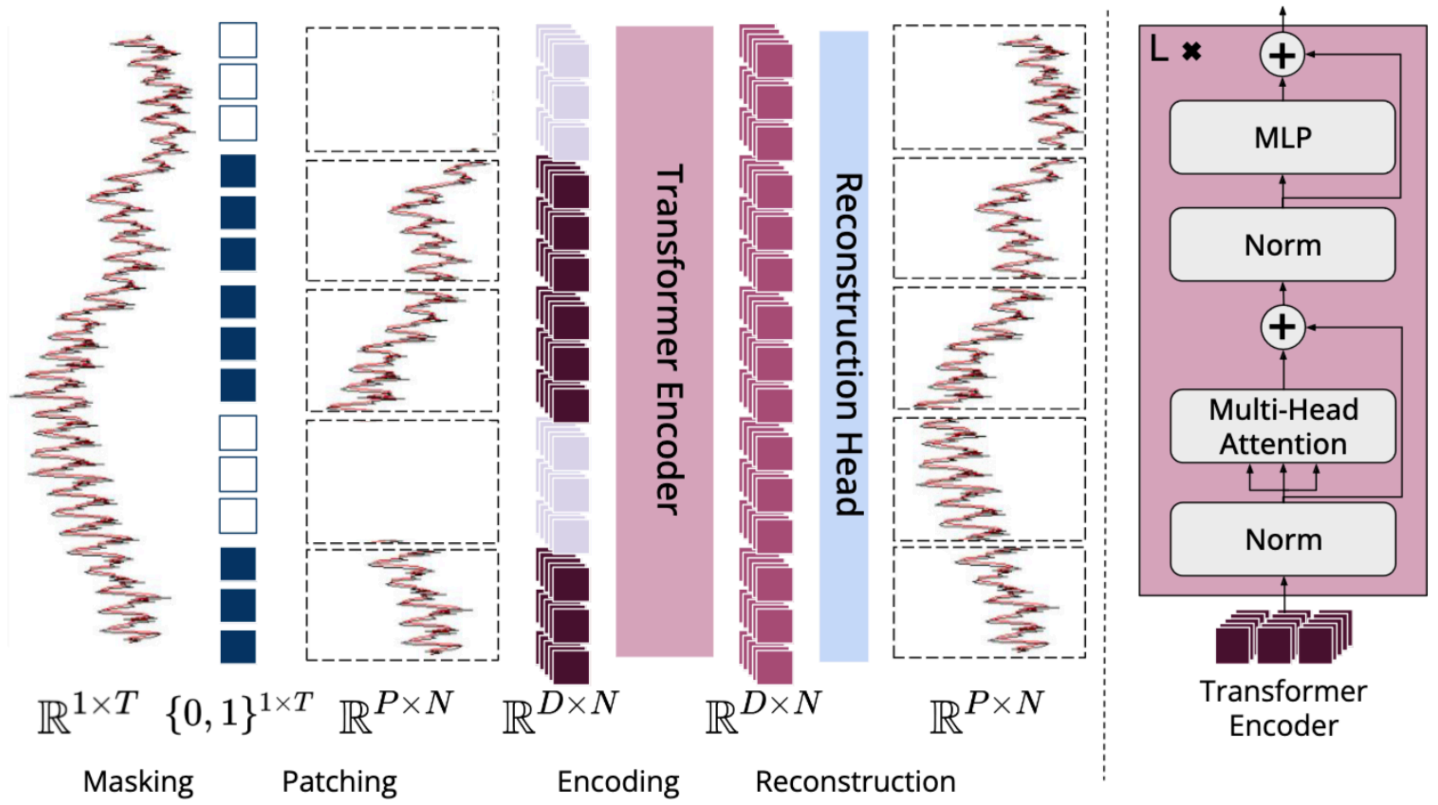
TimeGPT (Nixtla)



TimesFM (Google)



Moirai (Salesforce)



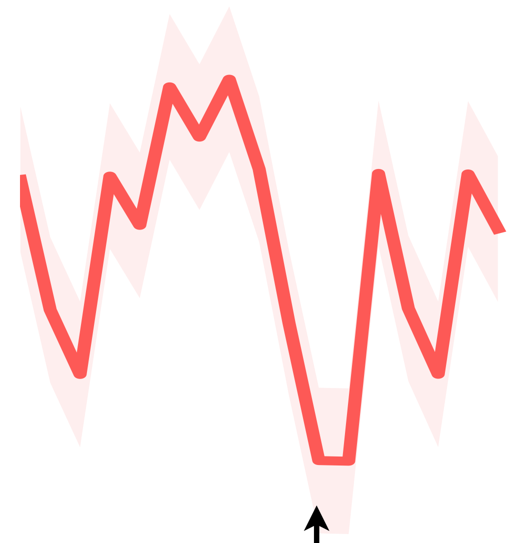
MOMENT (CMU)

Garza, Azul, and Max Mergenthaler-Canseco. "TimeGPT-1." *arXiv preprint arXiv:2310.03589* (2023).
Das, Abhimanyu, et al. "A decoder-only foundation model for time-series forecasting." arXiv preprint arXiv:2310.10688 (2023).
Woo, Gerald, et al. "Unified training of universal time series forecasting transformers." arXiv preprint arXiv:2402.02592 (2024).
Goswami, Mononito, et al. "Moment: A family of open time-series foundation models." arXiv preprint arXiv:2402.03885 (2024).

Introducing **CHRONOS**

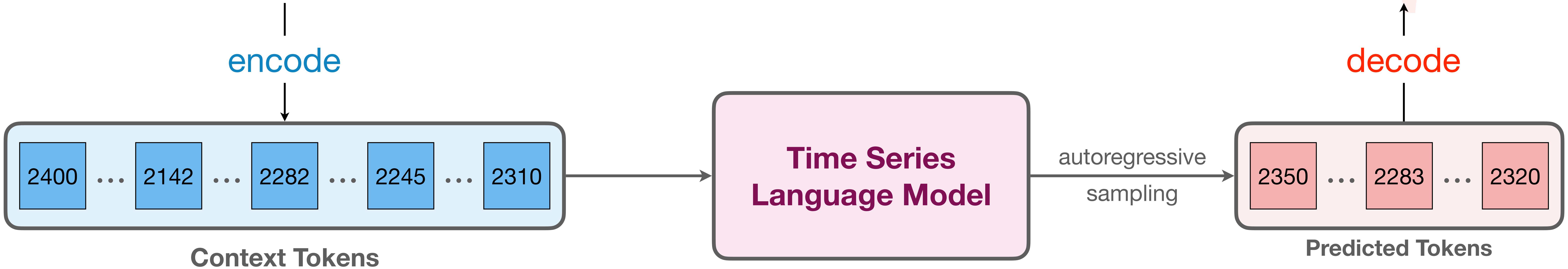
Historical Time Series

Probabilistic Forecast



A language modeling framework for time series data

that encodes time series into discrete tokens
and trains a language model on them



- **Two methods to diversify training data**
  - TSMixup: a data augmentation scheme
  - KernelSynth: a synthetic data generation scheme using Gaussian processes
- **5 model sizes and an extensive evaluation**
  - 15 in-domain datasets
  - 27 zero-shot datasets

Language Modeling and Forecasting

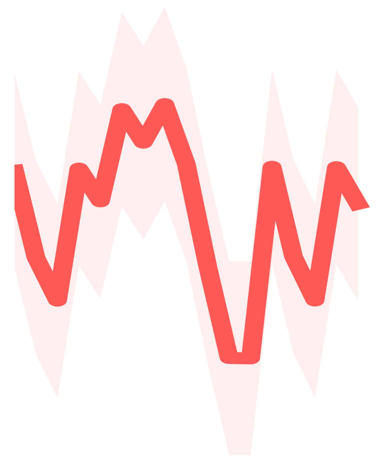
"Three Rings for the Elven-kings under the sky,
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie."

Language Model

"One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie."

Predict the next sequence of words (tokens)

Time Series Forecaster



Predict future values conditioned on the past

Time Series Tokenization

“Three Rings for the Elven-kings under the sky,
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie.”

↓

Text Tokenizer

↓

“ Three” “ Ring” “s” “ for” “ the” “ El” “ven” “-” “king” “s” …

Text language models have a discrete vocabulary

↓

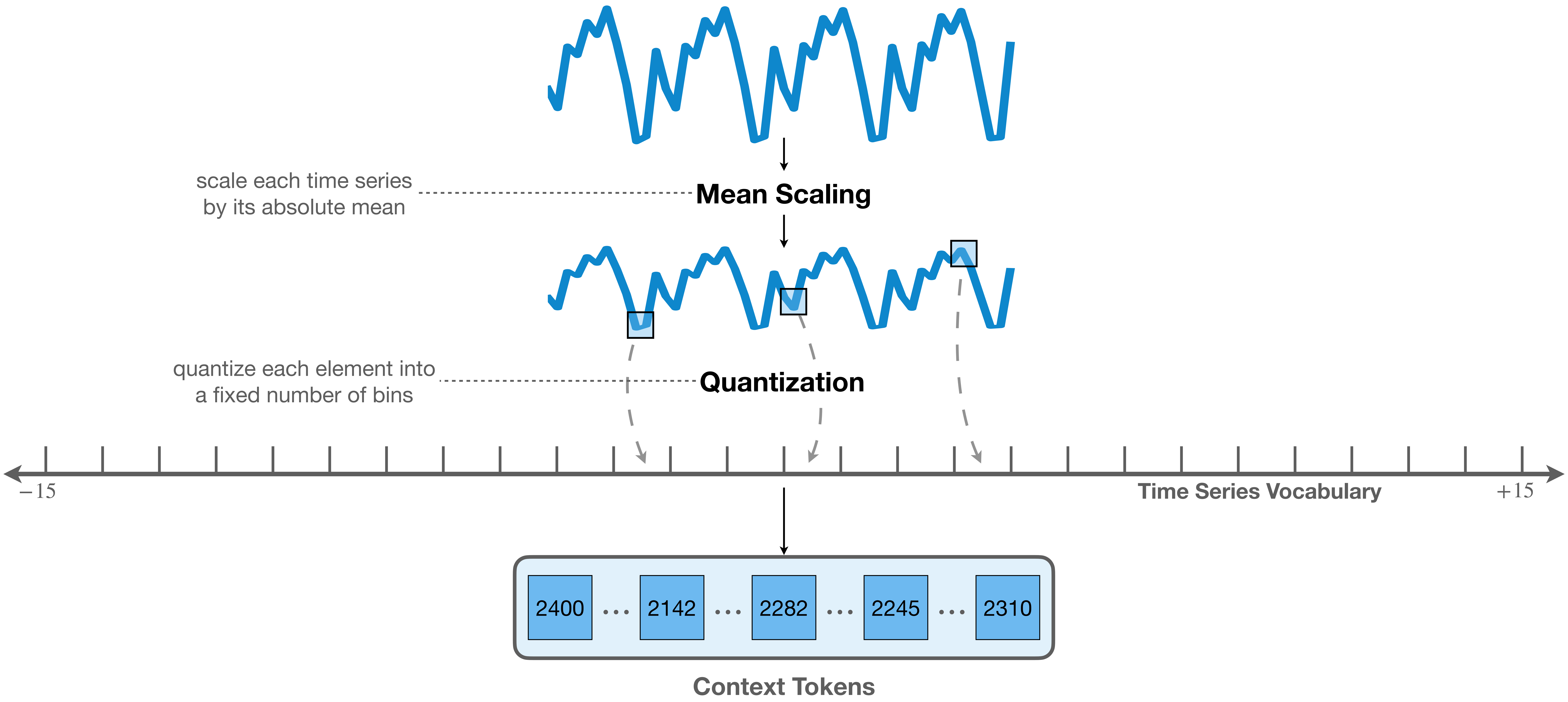
?

↓

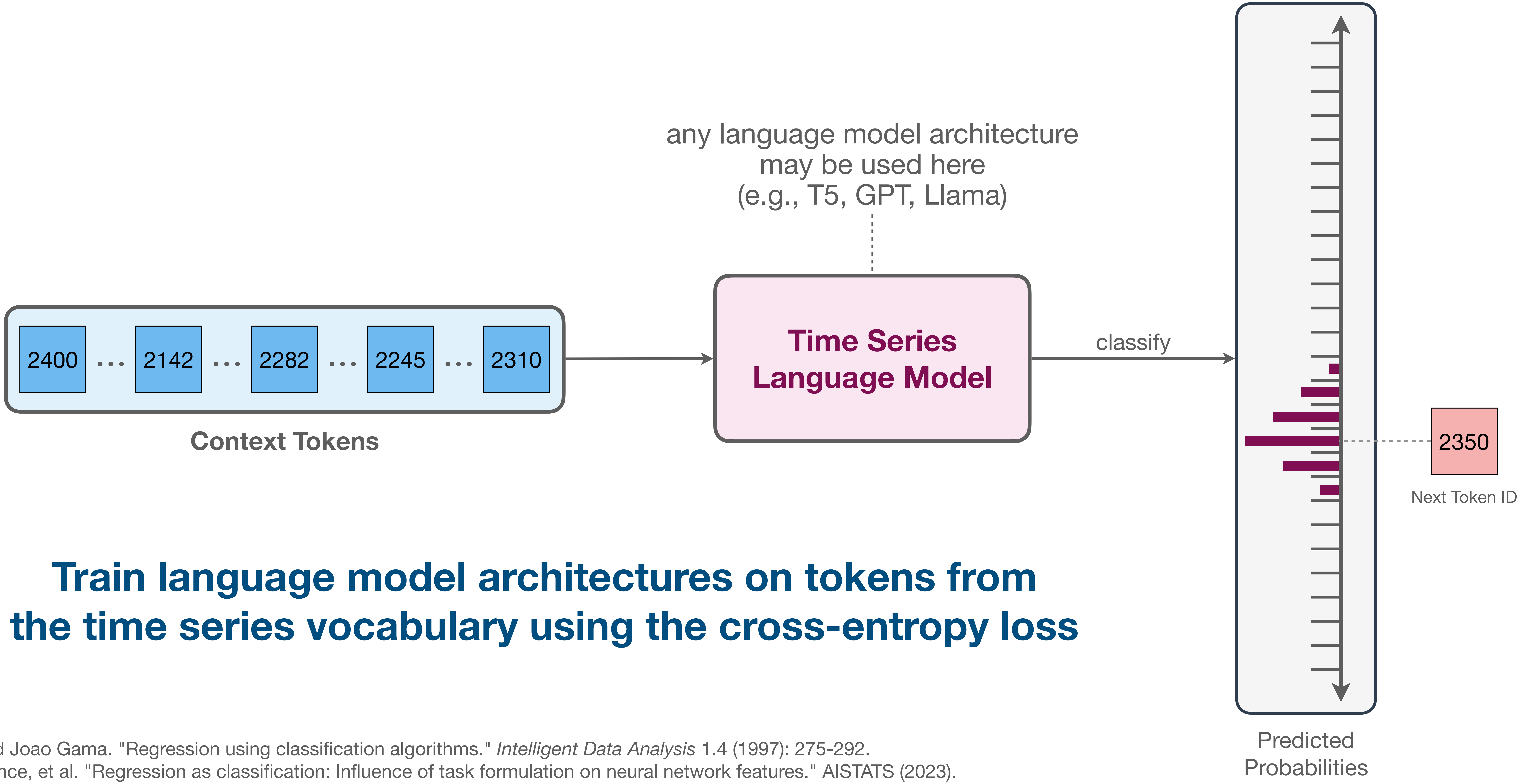
?

Time series are real-valued signals

Time Series Tokenization



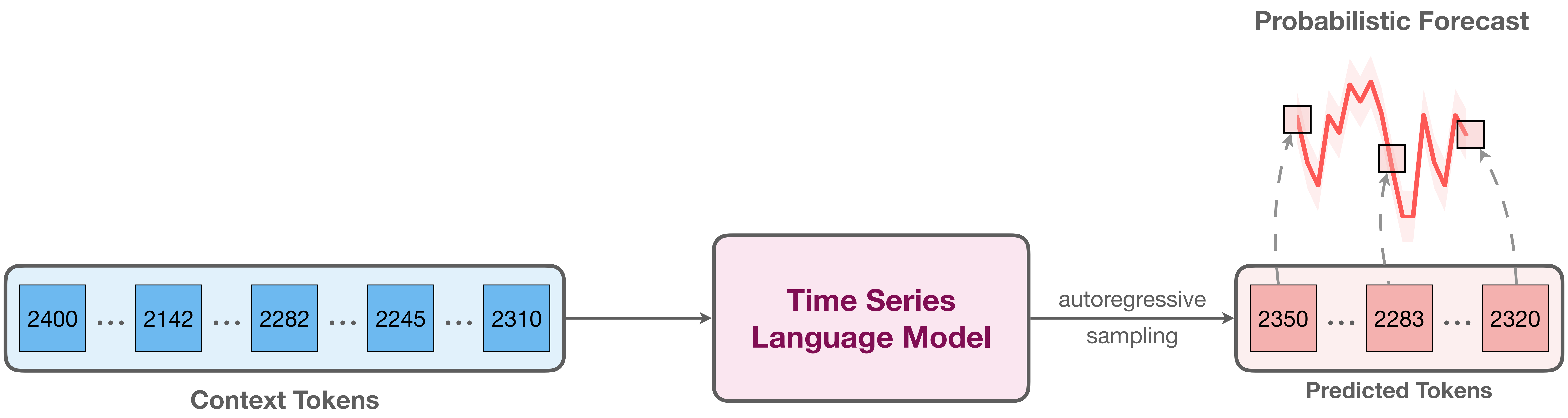
Regression via Classification



Train language model architectures on tokens from the time series vocabulary using the cross-entropy loss

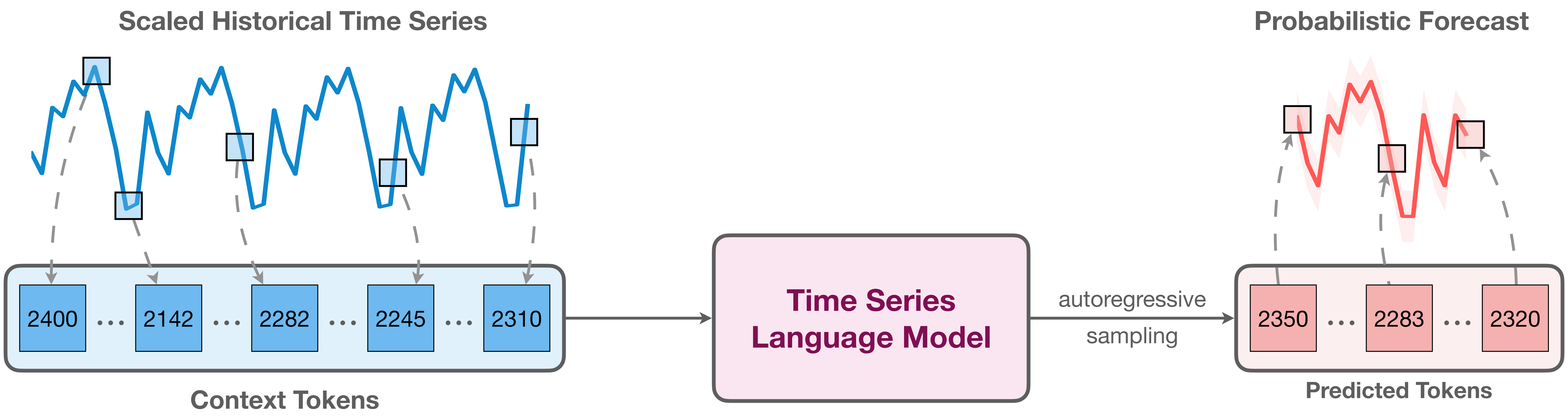
Torgo, Luis, and Joao Gama. "Regression using classification algorithms." *Intelligent Data Analysis* 1.4 (1997): 275-292.
Stewart, Lawrence, et al. "Regression as classification: Influence of task formulation on neural network features." AISTATS (2023).

Sampling



Probabilistic Forecast
2400
2142
2282
2245
2310
Context Tokens
Time Series
Language Model
autoregressive
sampling
2350
2283
2320
Predicted Tokens

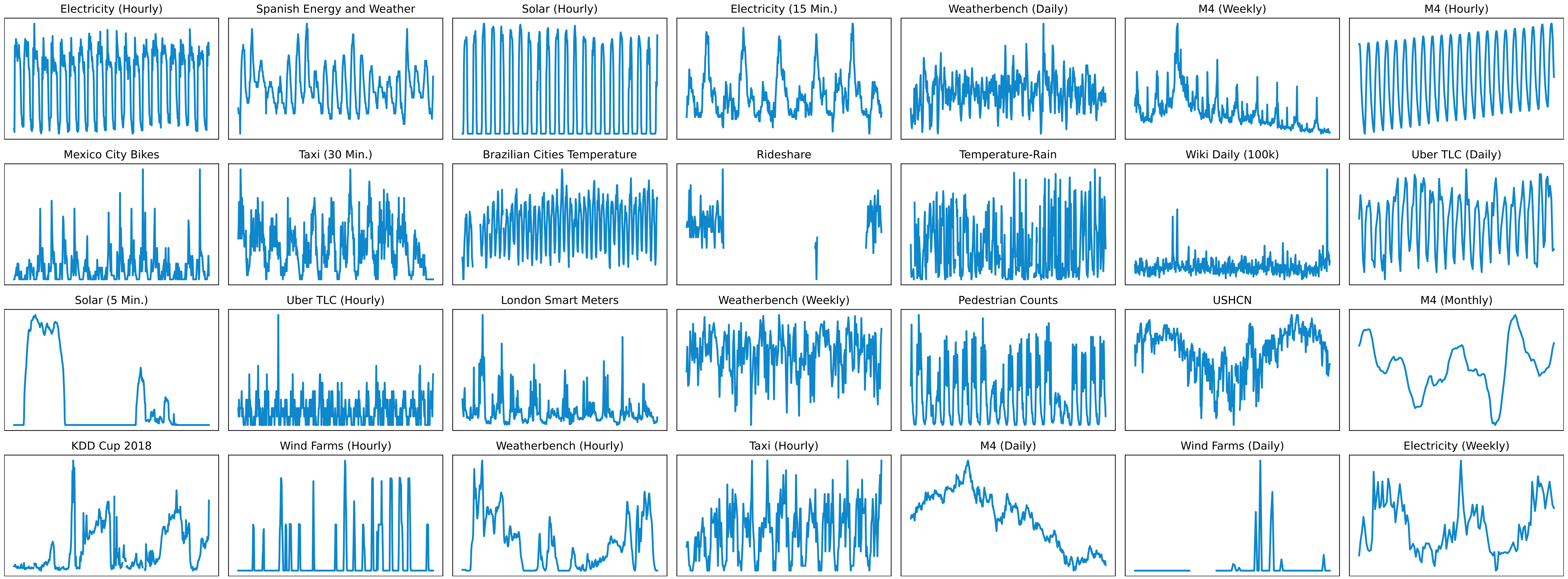
The Complete CHRONOS Framework



- ✔ *simple and intuitive*
- ✔ *probabilistic by design*
- ✔ *requires no changes to the language model architecture or training procedure*

Training Datasets

Various Domains and Frequencies



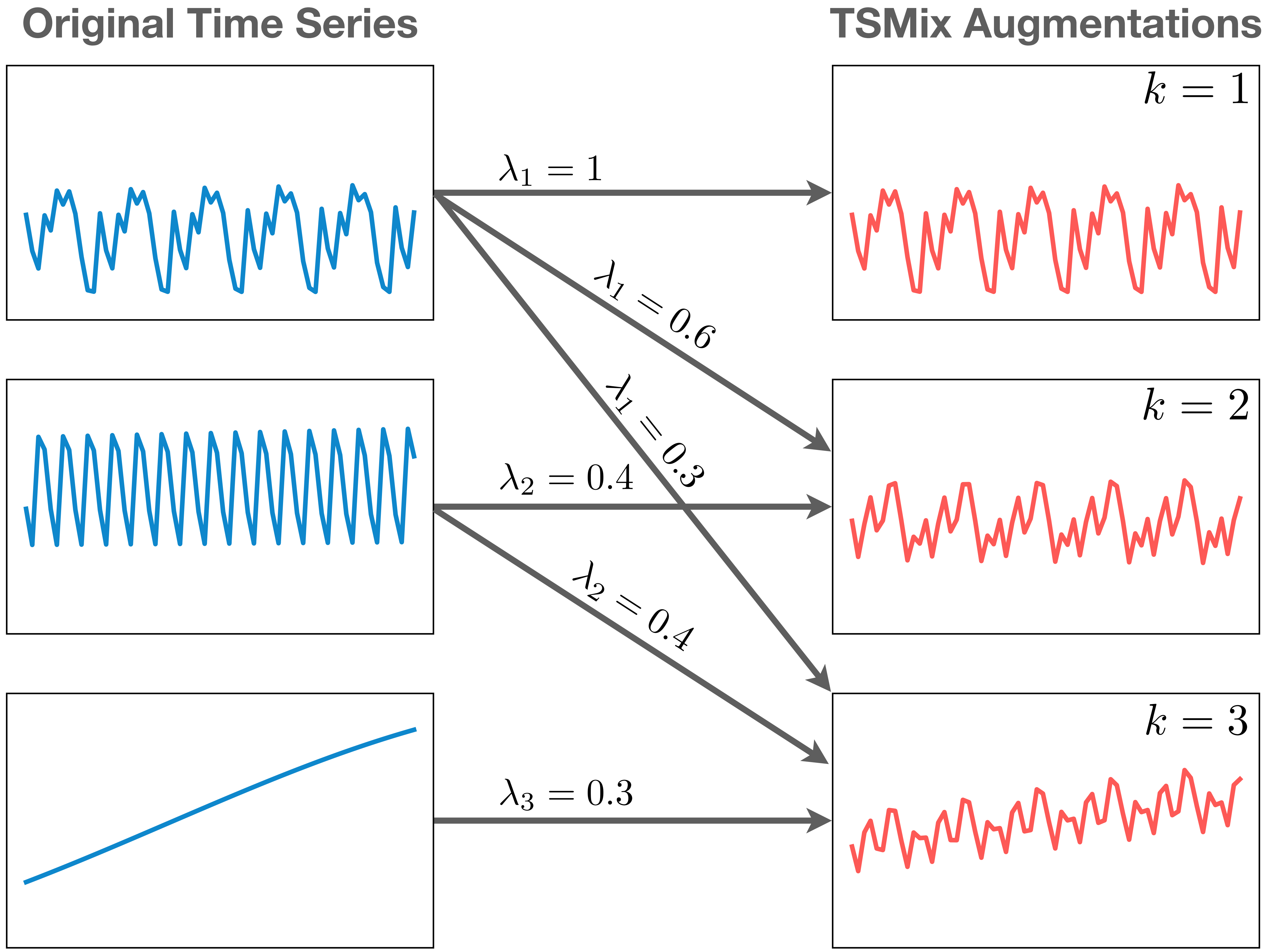
28 Datasets

890K Time Series

84B Observations

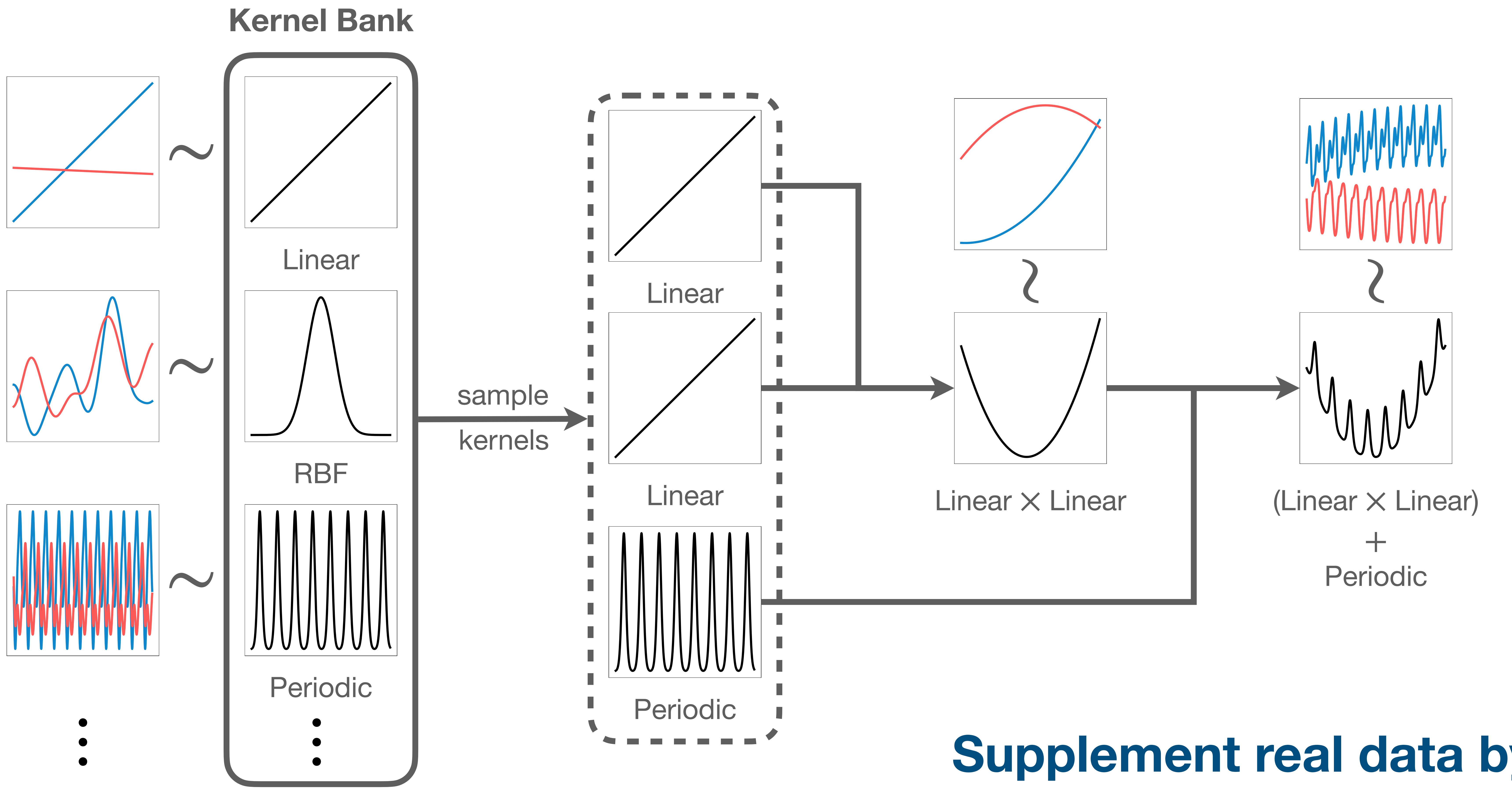
TSMixup Augmentations

- **Sample $K \sim \{1,2,3\}$ time series**
  - $y_1, \ldots, y_K \sim$ dataset bank
- **Sample weights**
  - $\lambda_1, \ldots, \lambda_K \sim \mathrm{Dirichlet}(\alpha)$
- **Combine time series**
  - $y = \sum_{i=1}^{K} \lambda_i y_i$



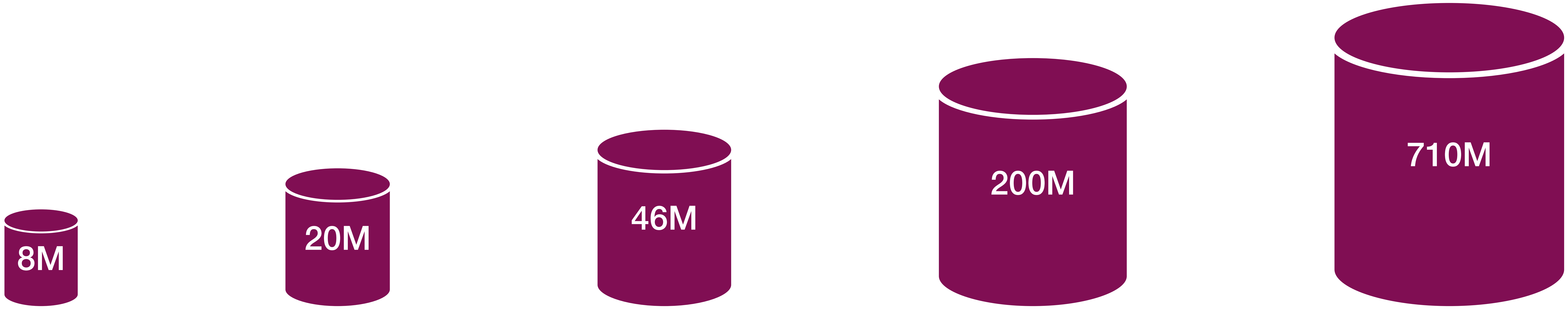
Improve pattern diversity by mixing time series from different datasets

KernelSynth: Synthetic Data Generation



Supplement real data by generating synthetic time series from Gaussian processes

Chronos Variants



Chronos-T5 (Tiny) Chronos-T5 (Mini) Chronos-T5 (Small) Chronos-T5 (Base) Chronos-T5 (Large)

Based on the T5 Encoder-Decoder Architecture

Baselines

Pretrained Models

single model used across all tasks

- LLMTime
- ForecastPFN
- LagLlama
- Moirai
- TimesFM

Task-specific Models

separate model trained/fine-tuned for each task

- PatchTST
- DeepAR
- WaveNet
- TFT
- DLinear
- NBEATS
- NHiTS
- GPT4TS

Local Models

separate model for each time series

- Naive
- SeasonalNaive
- AutoETS
- AutoTheta
- AutoARIMA

Evaluation Metrics

Weighted Quantile Loss (WQL) ↓

evaluates the quality of **probabilistic forecasts**

$$\text{wQL} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \text{wQL}[q]$$

$$\mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}.$$

Mean Absolute Scaled Error (MASE) ↓

evaluates the quality of **point forecasts**

$$\text{MASE} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{H} \frac{\sum_{h=1}^{H} |y_{i,T+h} - \hat{y}_{i,T+h}|}{\sum_{t=1}^{T-s} |y_{i,t+s} - y_{i,t}|}$$

N: number of time series in the dataset
H: forecast horizon
q: quantile level
s: season length
y: ground truth
$\hat{y}$: prediction

Aggregated Relative Score ↓

scale the score by the score of a baseline model and take the **geometric mean** across datasets

Benchmarks

Benchmark I

15 in-domain datasets **for CHRONOS**

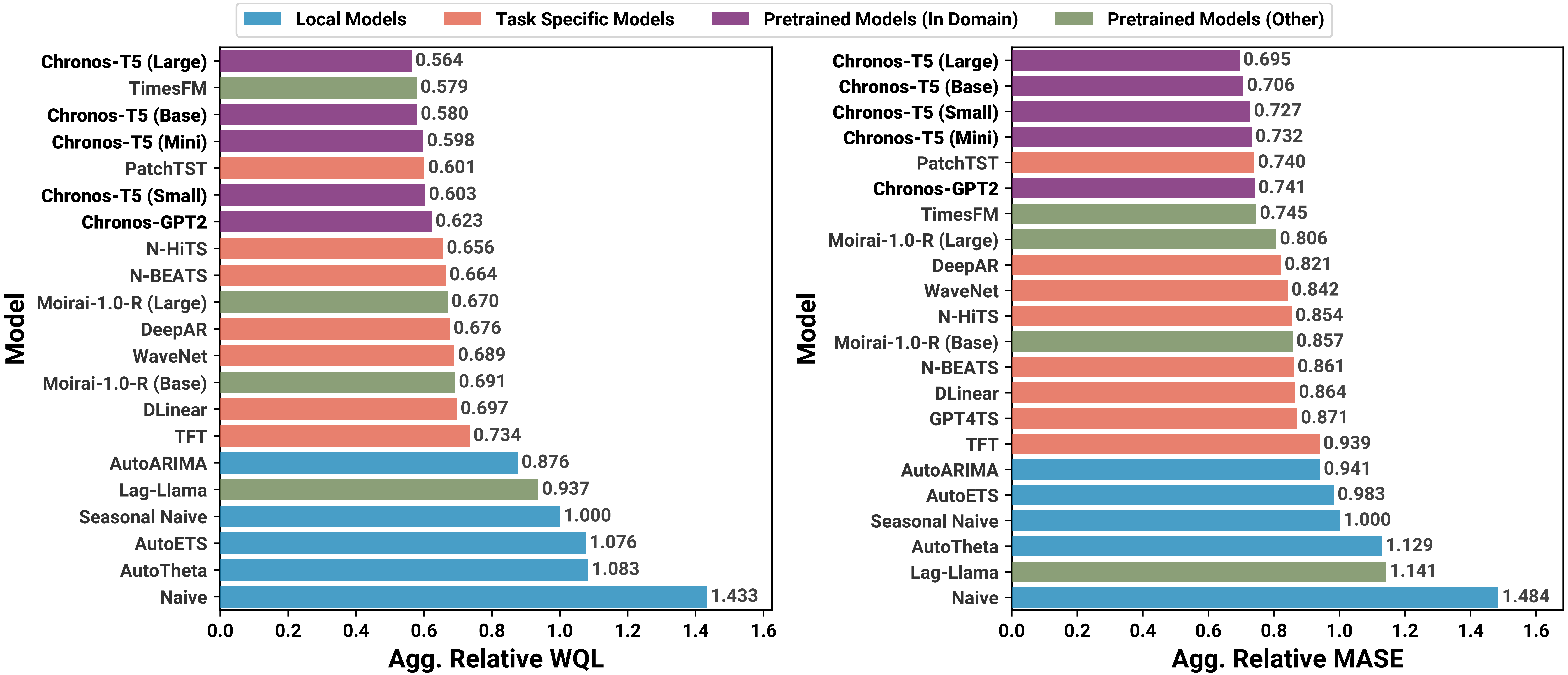
- Electricity (15 Min.)
- Electricity (Hourly)
- Electricity (Weekly)
- KDD Cup 2018
- M4 (Daily)
- M4 (Hourly)
- M4 (Monthly)
- M4 (Weekly)
- Pedestrian Counts
- Taxi (30 Min.)
- Uber TLC (Hourly)
- Uber TLC (Daily)
- Rideshare
- Temperature-Rain
- London Smart Meters

Benchmark II

27 zero-shot datasets **for CHRONOS**

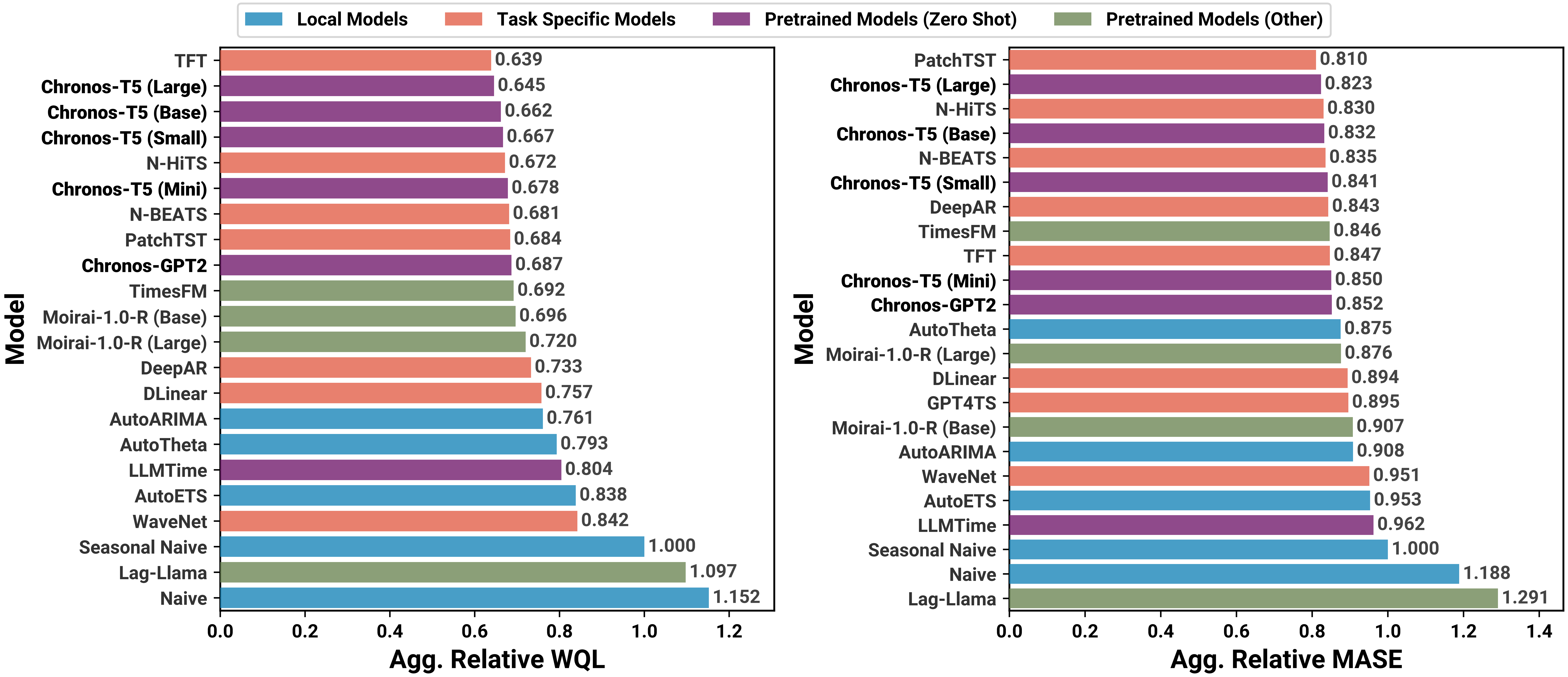
- Australian Electricity
- Car Parts
- CIF 2016
- Covid Deaths
- Dominick
- ERCOT Load
- ETT (15 Min.)
- ETT (Hourly)
- Exchange Rate
- FRED-MD
- Hospital
- M1 (Quarterly)
- M1 (Monthly)
- M1 (Yearly)
- M3 (Monthly)
- M3 (Quarterly)
- M3 (Yearly)
- M4 (Quarterly)
- M4 (Yearly)
- M5
- NN5 (Daily)
- NN5 (Weekly)
- Tourism (Monthly)
- Tourism (Quarterly)
- Tourism (Yearly)
- Traffic
- Weather

Benchmark I: In-domain Results



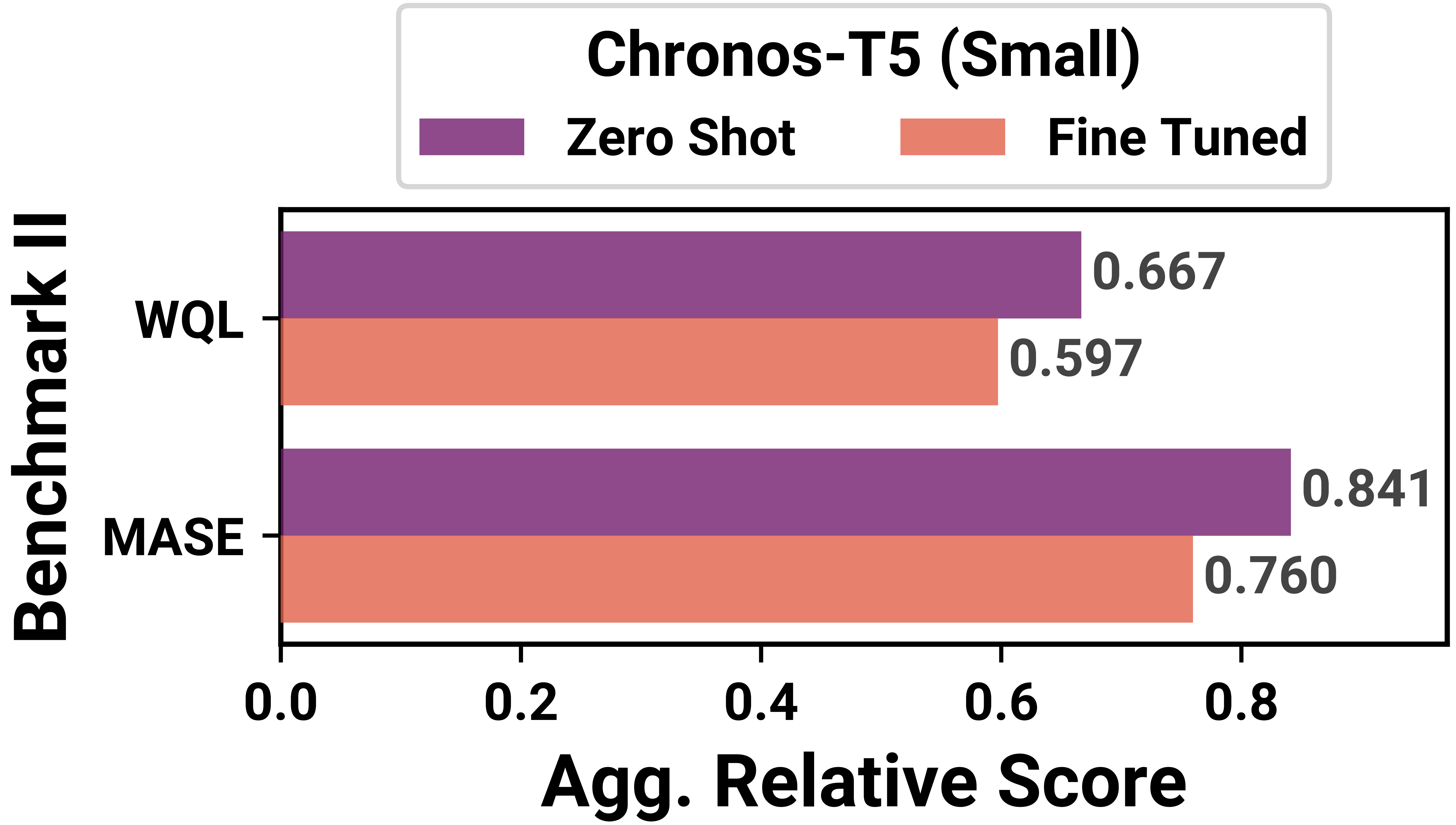
In-domain: 15 datasets that were part of the training corpus of **CHRONOS**

Benchmark II: Zero-shot Results



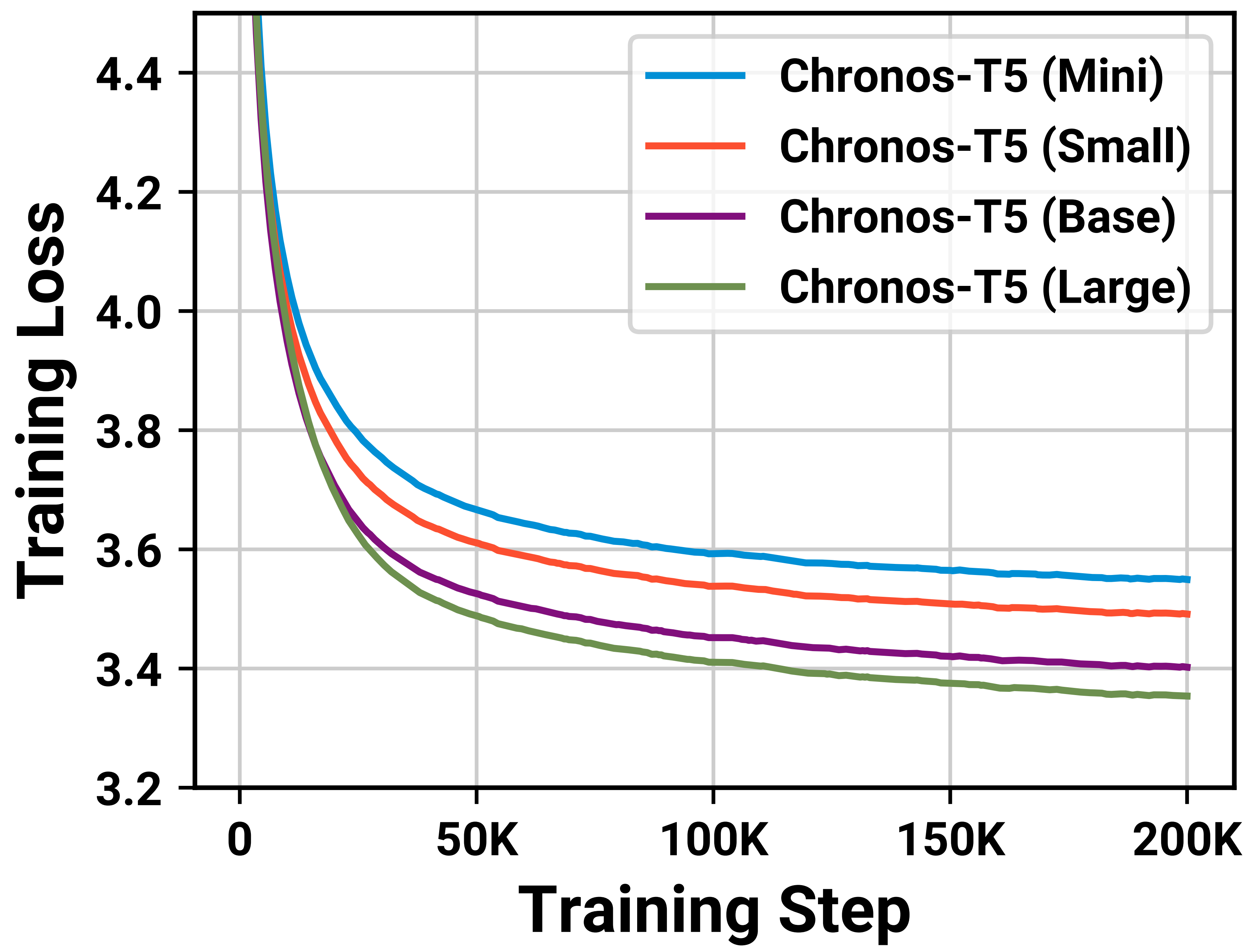
Zero-shot: 27 datasets not seen by **CHRONOS** during training

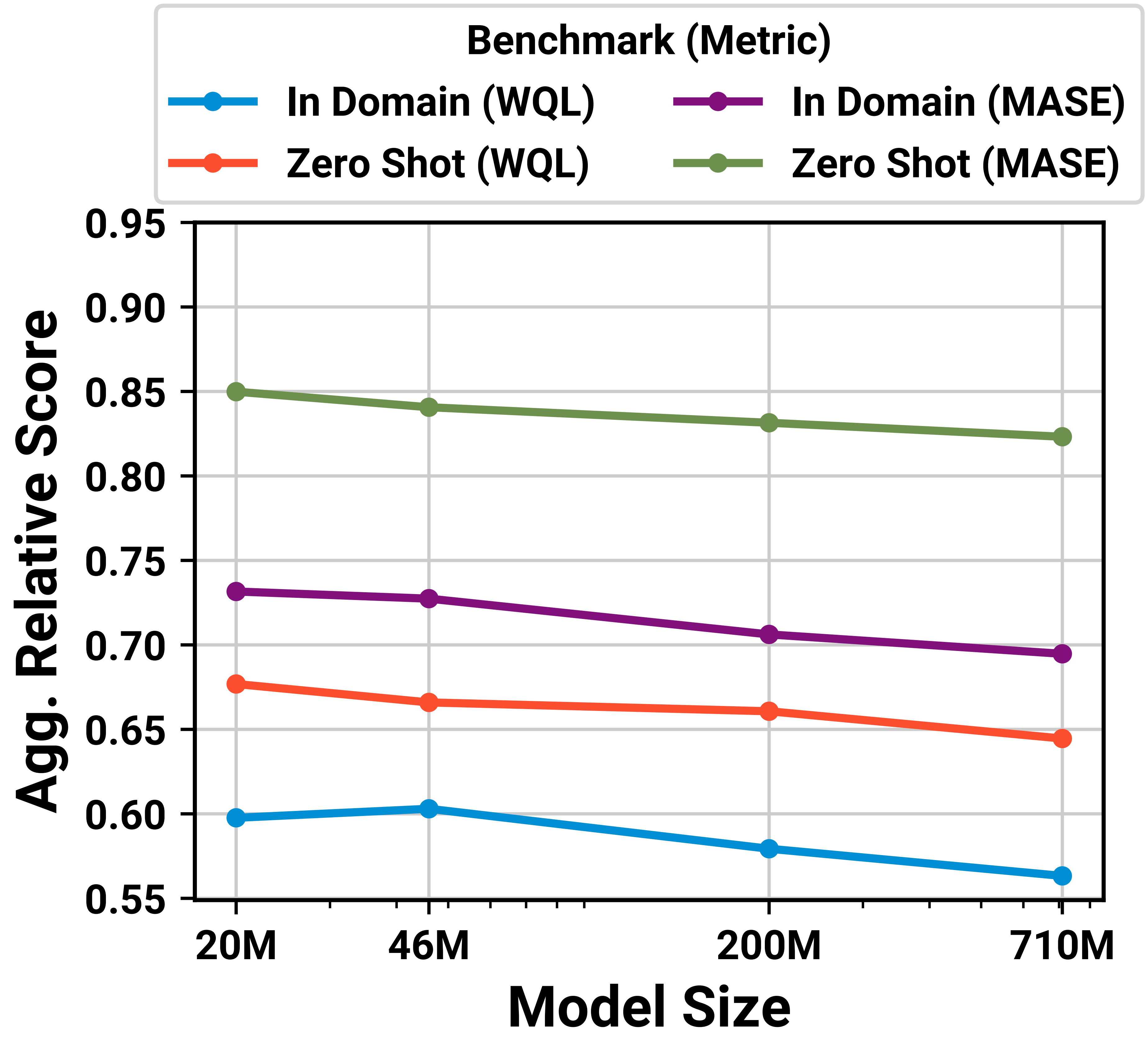
Benchmark II: Fine-tuned Results



dataset-agnostic lightweight fine-tuning for 1000 gradient steps

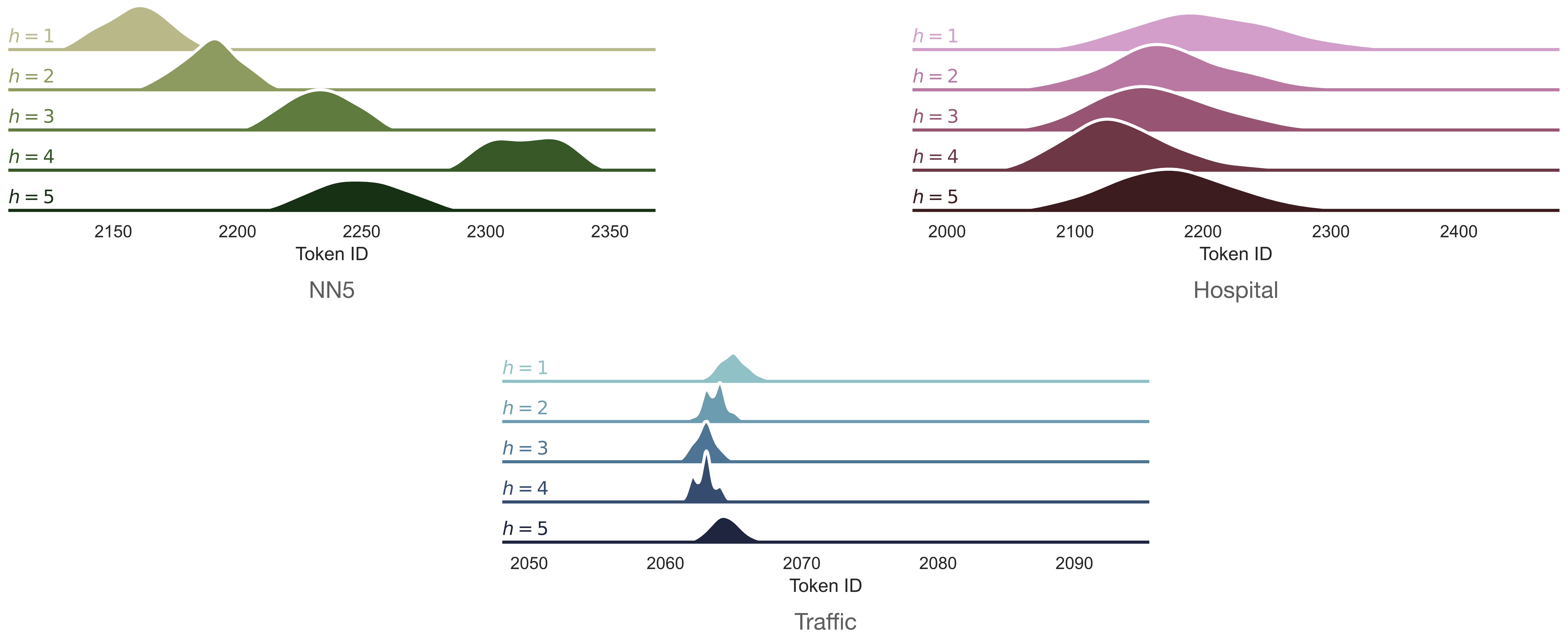
Are larger models better?





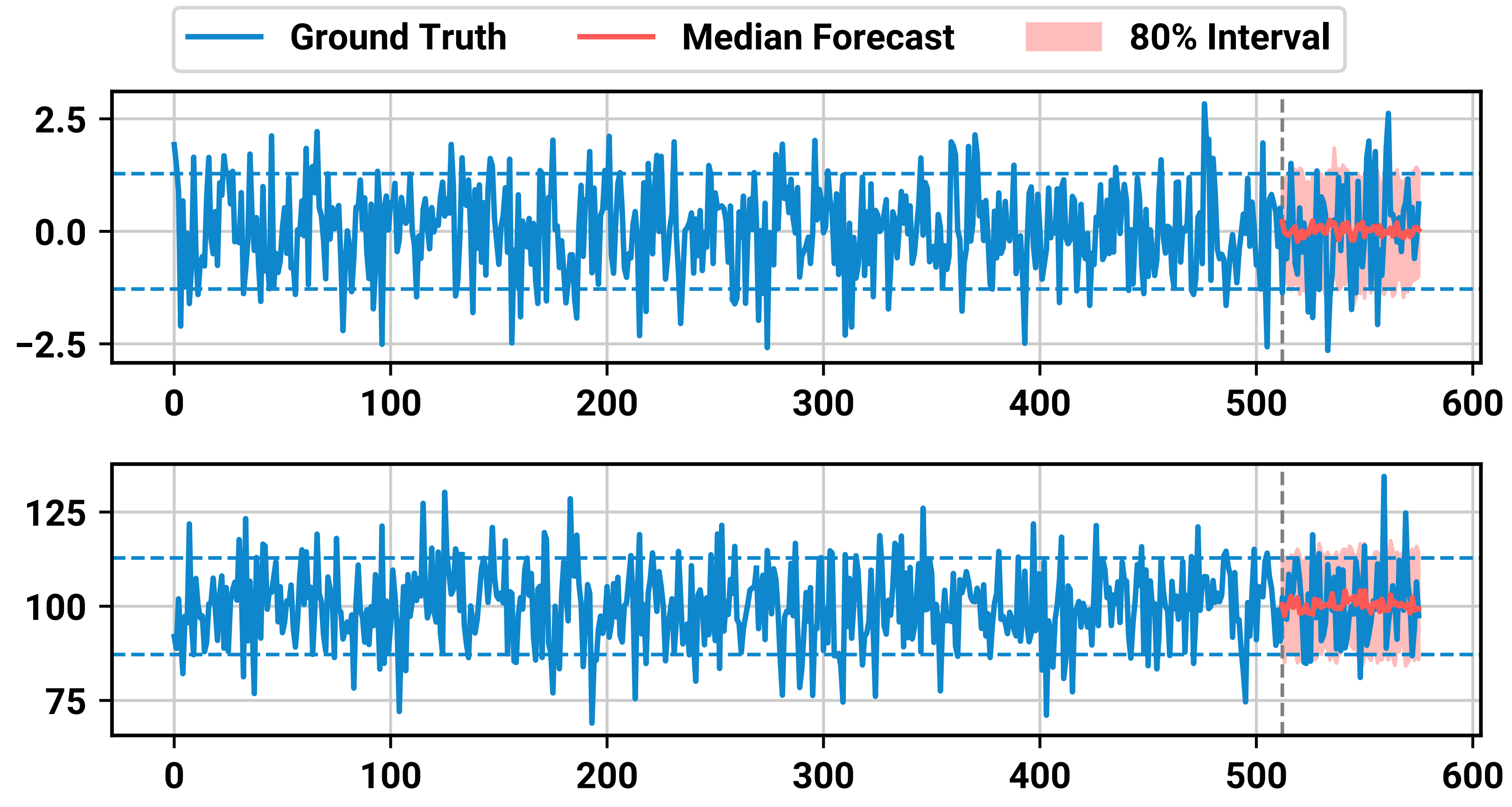
Qualitative Analysis

Token Distribution of First 5 Predictions Steps for 3 Time Series

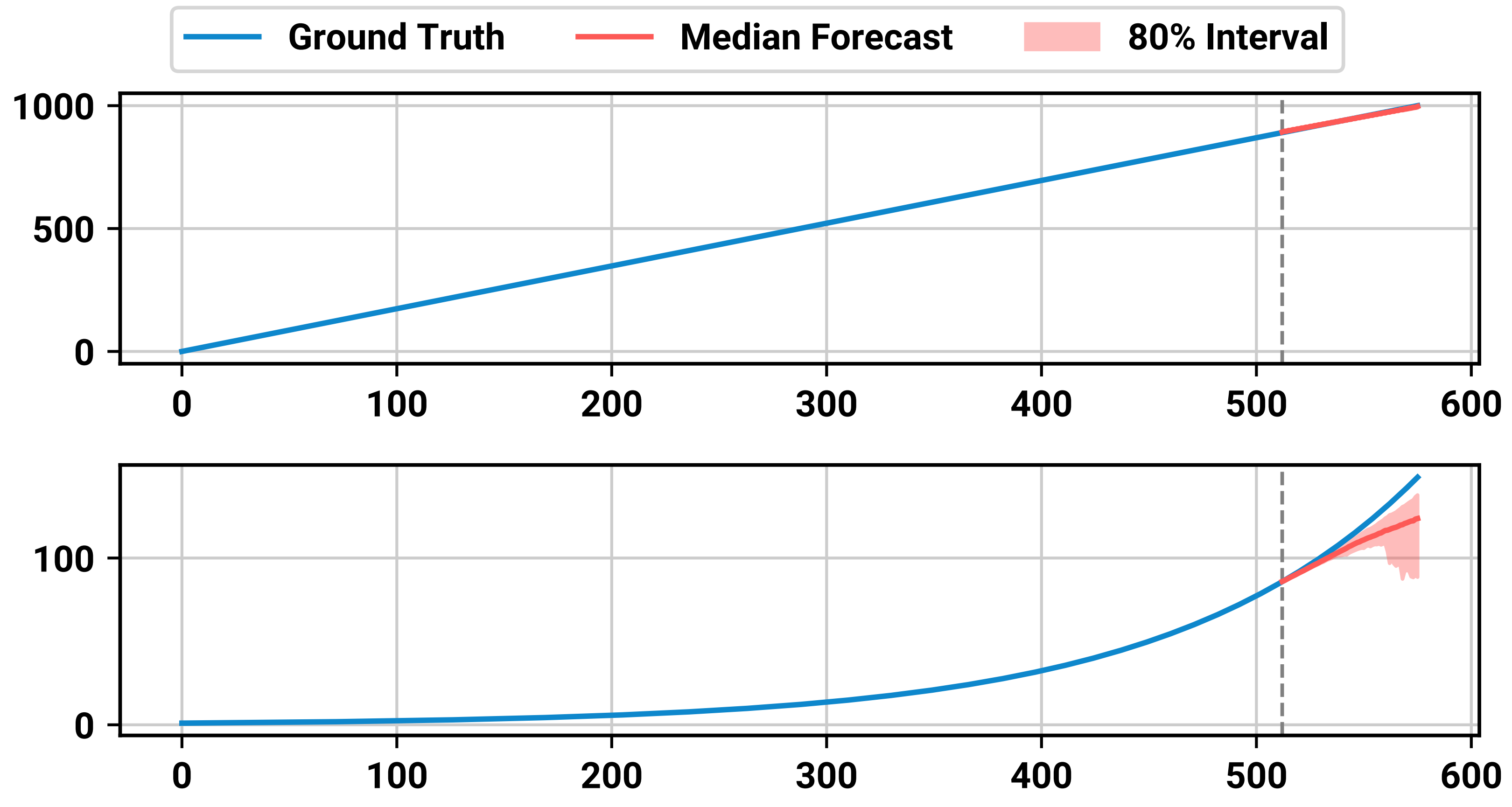


Chronos learns the metric space structure directly from data!

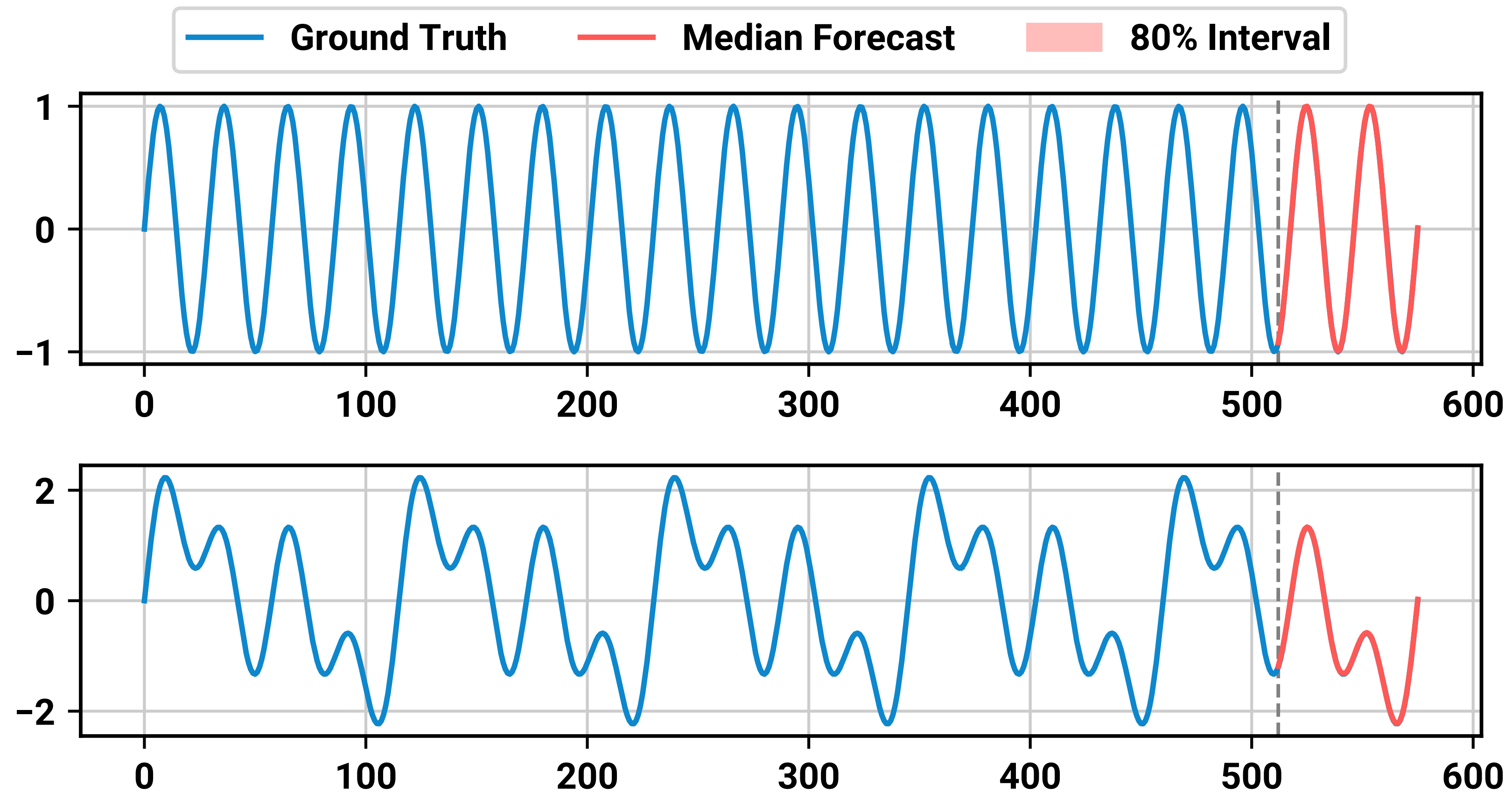
Qualitative Analysis



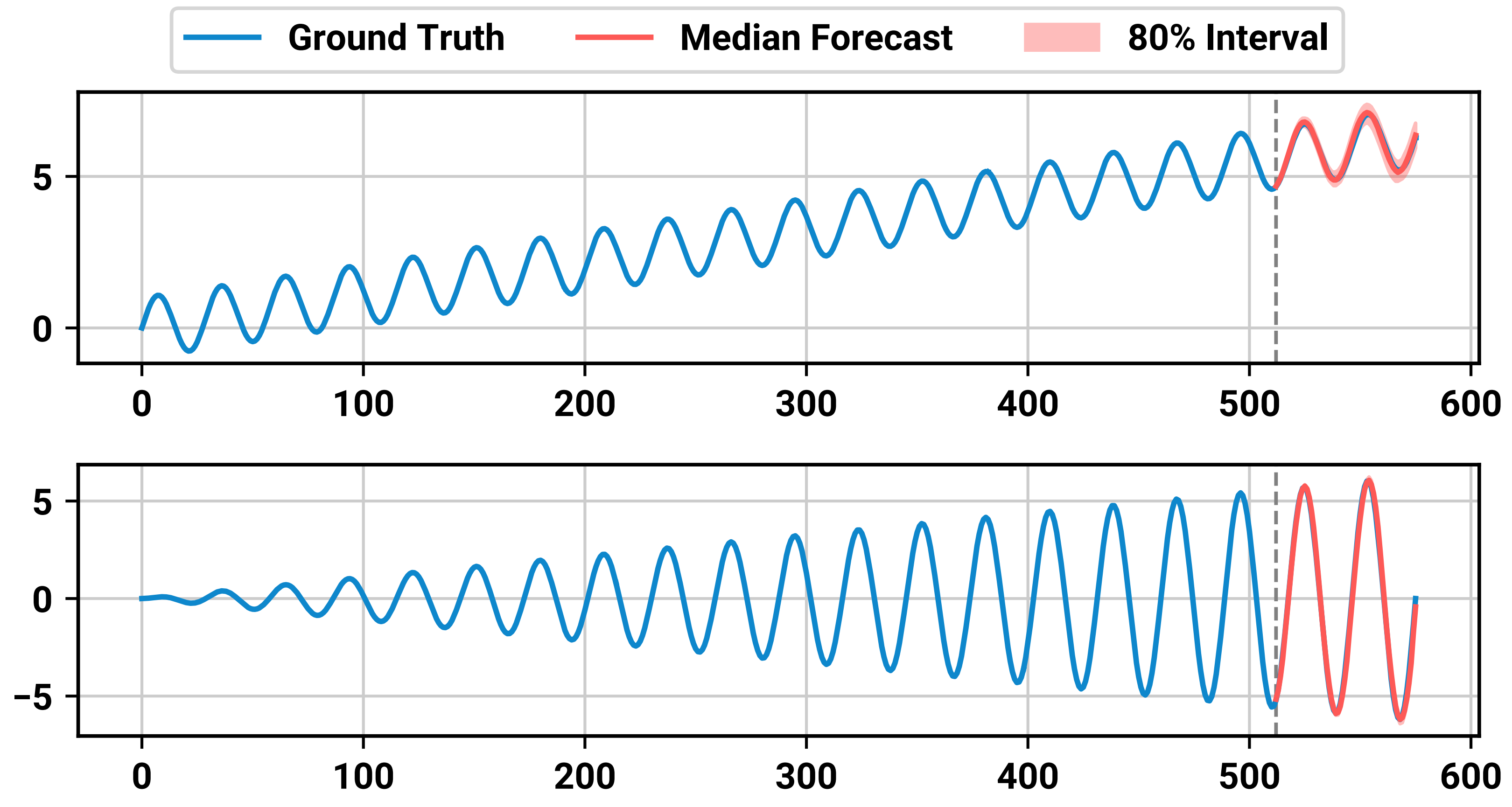
White Noise



Trends

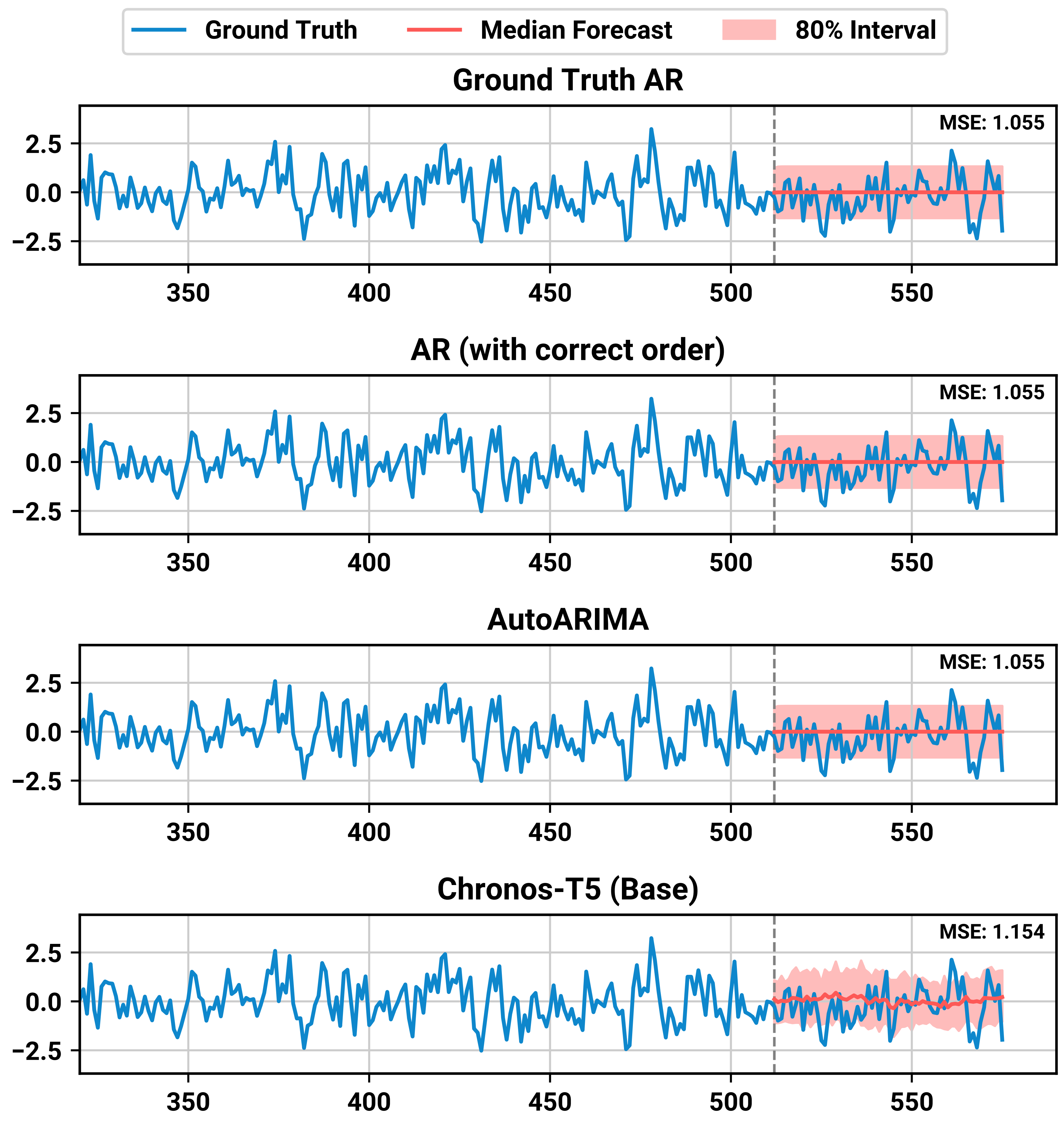


Seasonalities

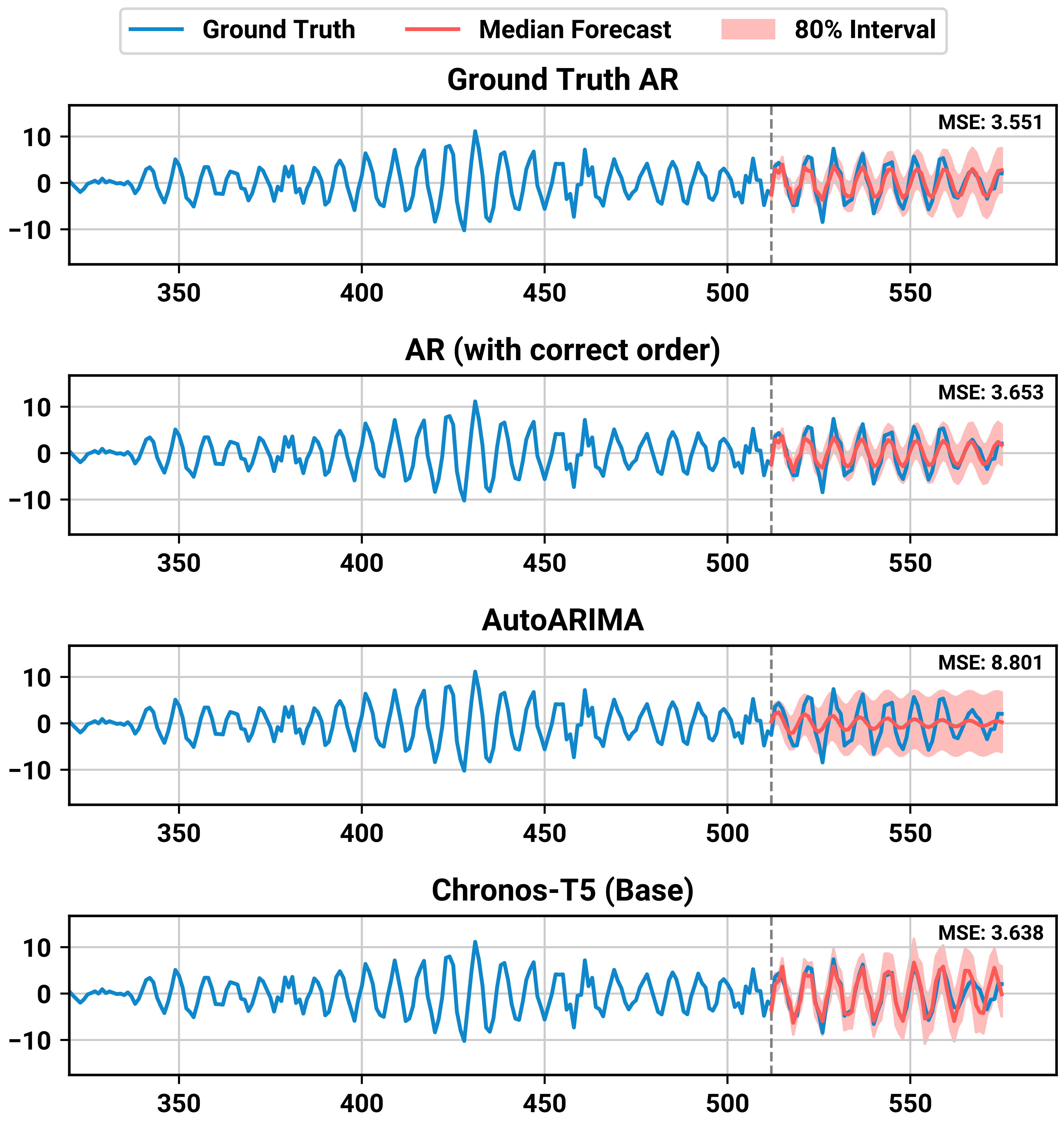


Combined Patterns

Qualitative Analysis

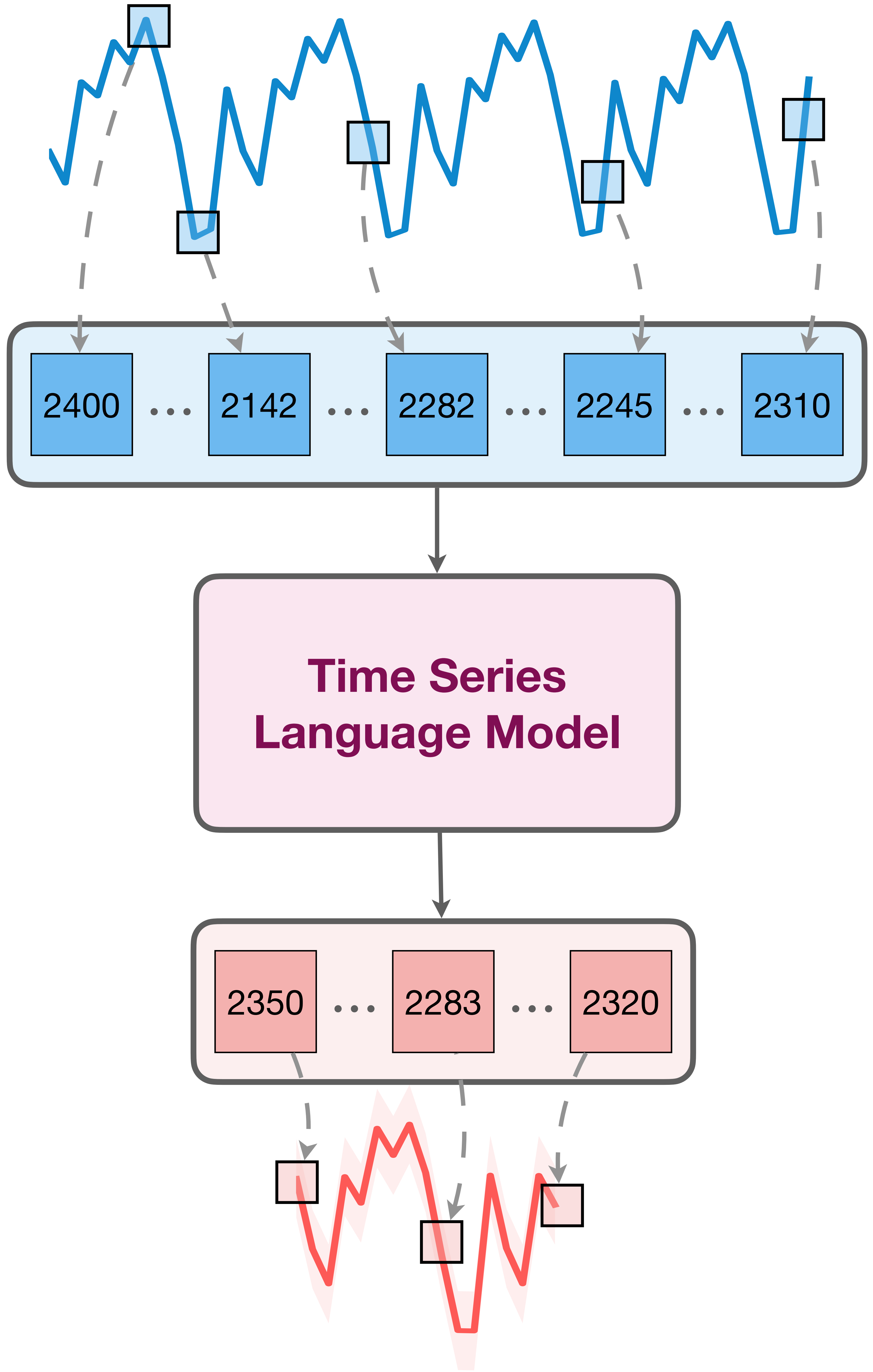


AR(1) Process



AR(4) Process

Summary



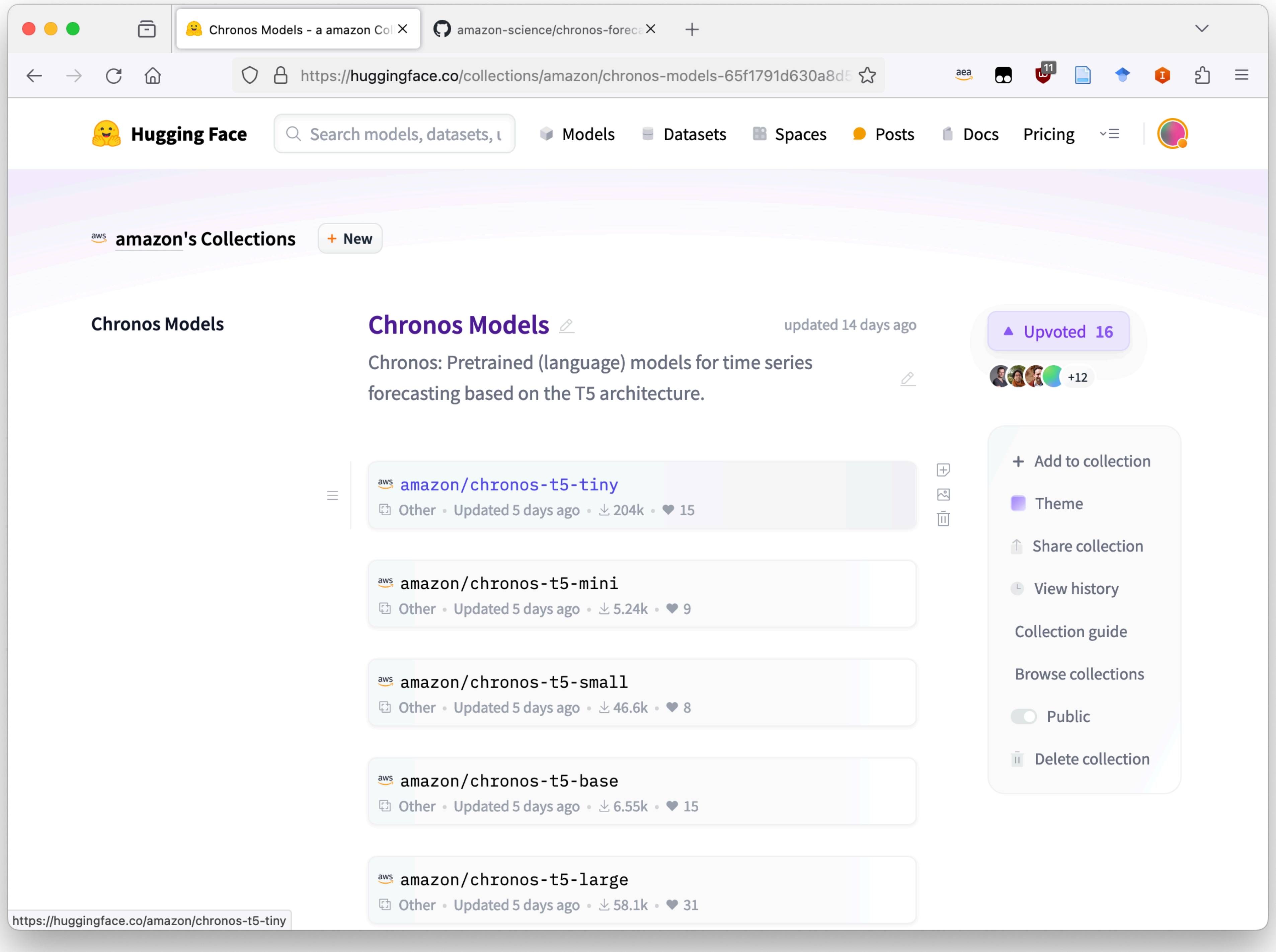
CHRONOS
A language modeling framework for time series data that encodes time series into discrete tokens via scaling and quantization

TSMixup
A mixup-based augmentation scheme for time series

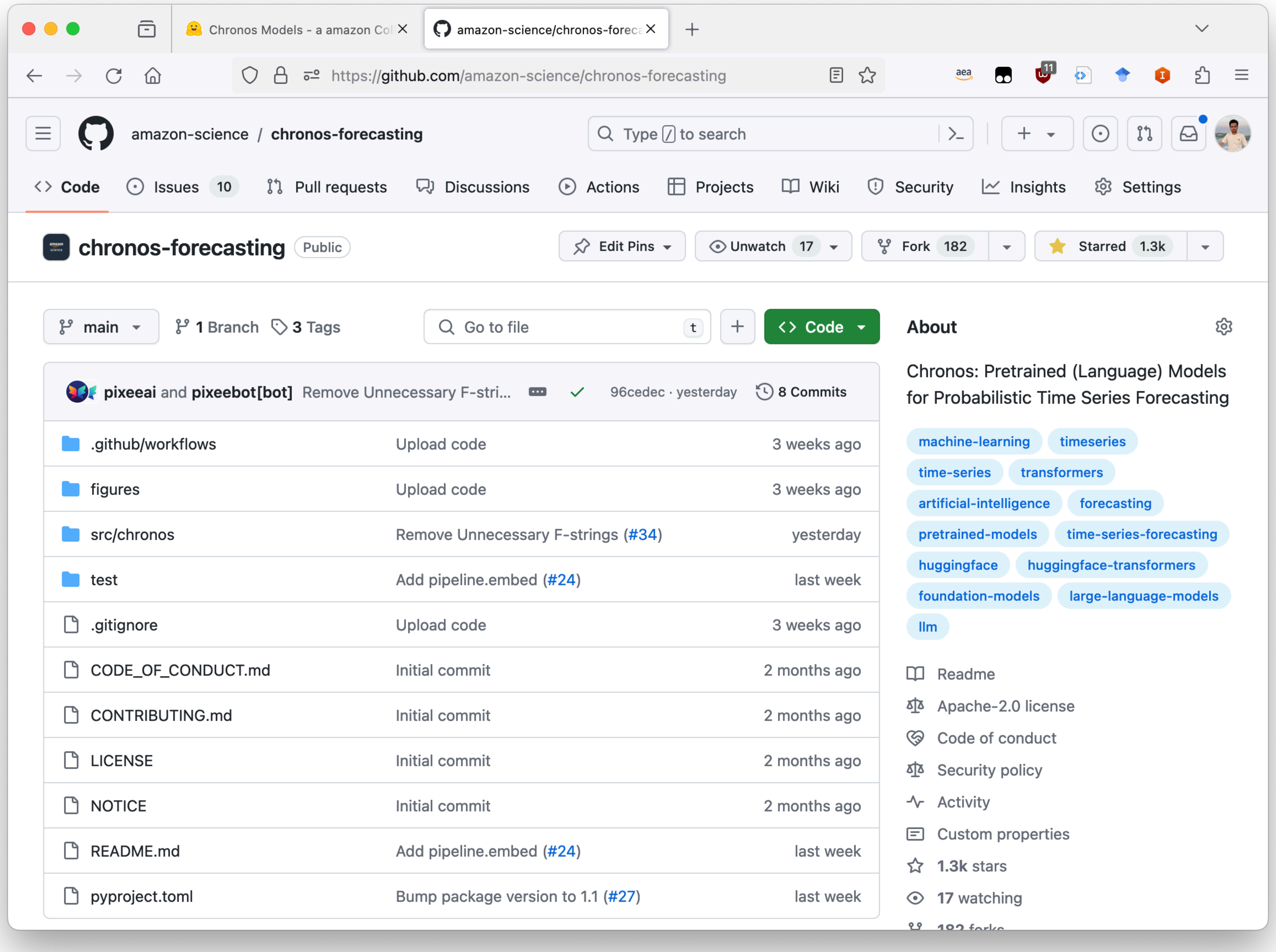
KernelSynth
A synthetic data generation scheme that combines random Gaussian process kernels to generate time series

Practical Models
5 practically usable models with excellent in-domain performance and zero-shot performance on par with *trained* models

Models Available on HuggingFace🤗



Downloaded 100MM+ times from HuggingFace🤗



Starred 2.4K times on Github

Models Available on HuggingFace🤗

chronos.py

```python
import torch
from chronos import ChronosPipeline

pipeline = ChronosPipeline.from_pretrained(
    "amazon/chronos-t5-base",
    device_map="cuda",
    torch_dtype=torch.bfloat16,
)
forecast = pipeline.predict(input_time_series, prediction_length=24)
```

Generate Forecasts in Two Lines of Code

Better Modeling & Inference

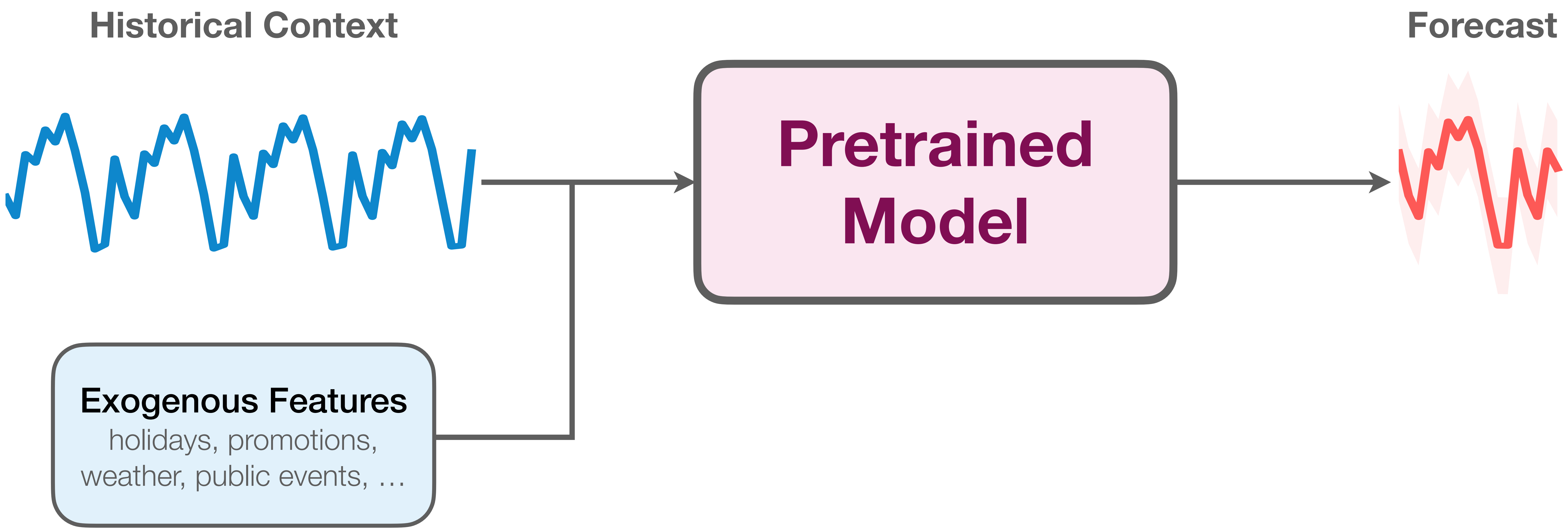
- Longer context and prediction horizons (e.g., to support high frequency data)
- Better time series tokenization/representation
- Better objective function
- Direct multi-token prediction
- Borrow methods from NLP for faster inference
  - optimized CUDA kernels
  - quantization
  - speculative/lookahead decoding
- Improve forecast quality during inference

Higher Quality Data & Benchmarks

- How to quantify the quality and diversity of time series data?
- What's the best recipe to mix data from different domains?
- How to improve existing benchmarks and build new ones?
- Is synthetic data all you need?

Beyond Univariate Forecasting

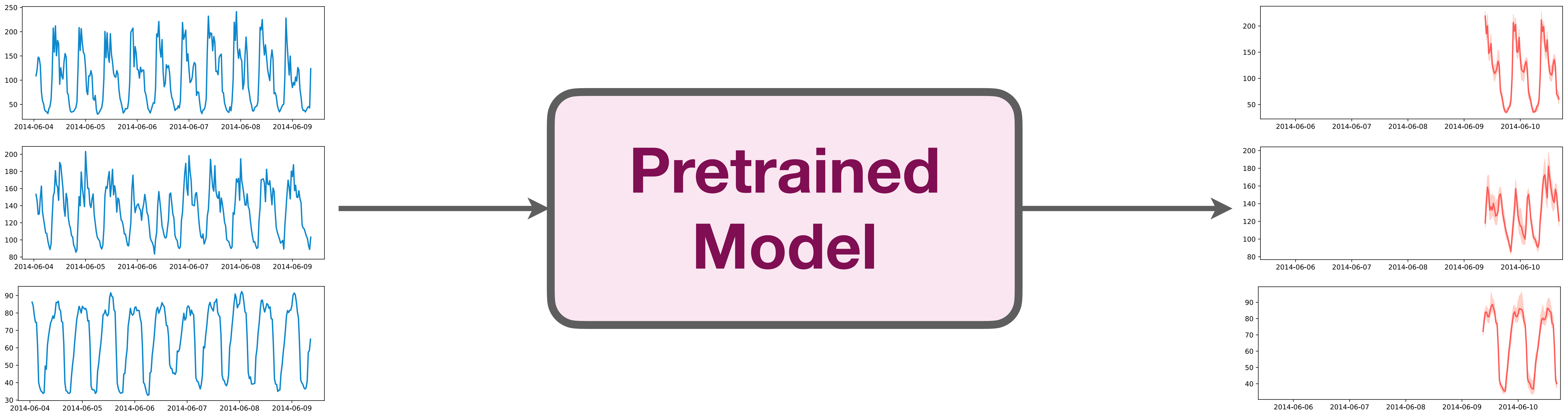
Forecasting with covariates: Covariates provide critical **exogenous information** relevant for accurate forecasting.



Challenge: The number and types of covariates are not known apriori. We need **in-context learning** for time series.

Beyond Univariate Forecasting

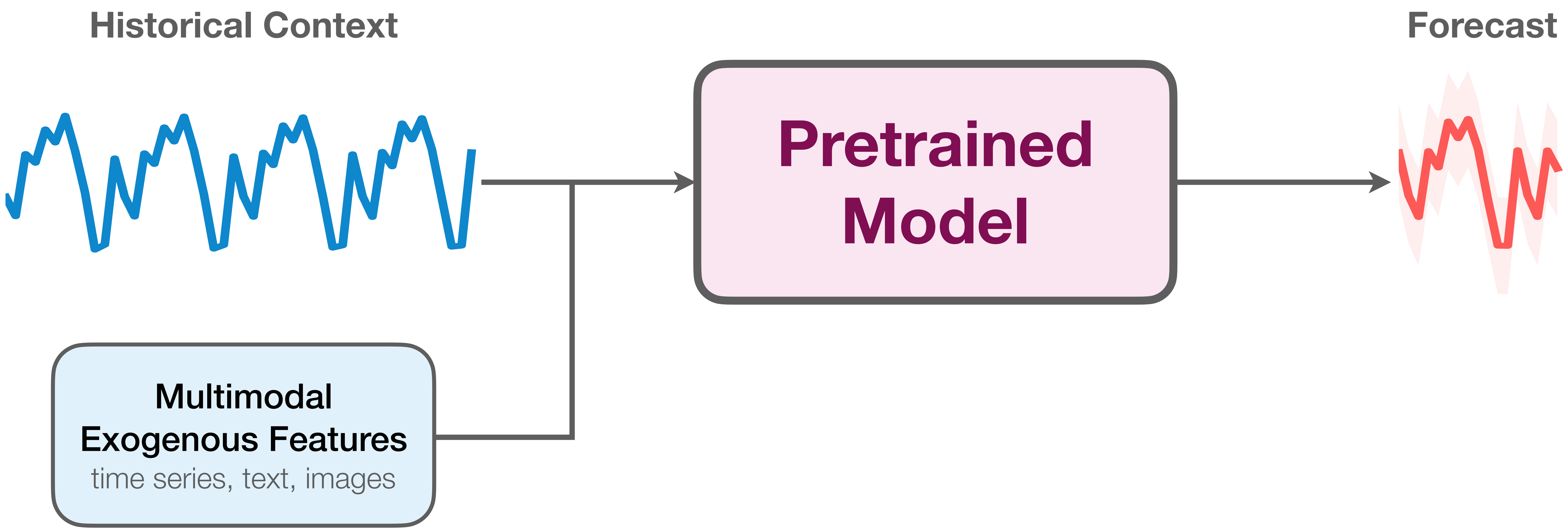
Multivariate Forecasting: joint modeling of multi-dimensional time series.



Challenge: The number of dimensions and their joint interactions is not known apriori.

Beyond Univariate Forecasting

Multimodal Forecasting: using exogenous information from other modalities (e.g., text and images) to improve forecast accuracy.



Challenge: Large scale multimodal time series data and tasks not available in the public domain.

Make a funny, relatable meme for the last slide of a talk on time series forecasting.

Generate

Questions?

Resources

Chronos: Learning the Language of Time Series

Abdul Fatir Ansari[1*], Lorenzo Stella[1*], Caner Turkmen[1], Xiyuan Zhang[2†], Pedro Mercado[1], Huibin Shen[1], Oleksandr Shchur[1], Syama Sundar Rangapuram[1], Sebastian Pineda Arango[3‡], Shubham Kapoor[1], Jasper Zschiegner, Danielle C. Maddix[1], Michael W. Mahoney[4], Kari Torkkola[4], Andrew Gordon Wilson[1], Michael Bohlke-Schneider[1], Yuyang Wang[1]
{ansarnd,stellalo}@amazon.com
[1]Amazon Web Services, [2]UC San Diego, [3]University of Freiburg, [4]Amazon Supply Chain Optimization Technologies

Abstract

We introduce CHRONOS, a simple yet effective framework for pretrained probabilistic time series models. CHRONOS tokenizes time series values using scaling and quantization into a fixed vocabulary and trains existing transformer-based language model architectures on these tokenized time series via the cross-entropy loss. We pretrained CHRONOS models based on the T5 family (ranging from 20M to 710M parameters) on a large collection of publicly available datasets, complemented by a synthetic dataset that we generated via Gaussian processes to improve generalization. In a comprehensive benchmark consisting of 42 datasets, and comprising both classical local models and deep learning methods, we show that CHRONOS models: (a) significantly outperform other methods on datasets that were part of the training corpus; and (b) have comparable and occasionally superior *zero-shot* performance on new datasets, relative to methods that were *trained specifically on them*. Our results demonstrate that CHRONOS models can leverage time series data from diverse domains to improve zero-shot accuracy on unseen forecasting tasks, positioning pretrained models as a viable tool to greatly simplify forecasting pipelines.

https://arxiv.org/abs/2403.07815

amazon-science/
chronos-forecasting

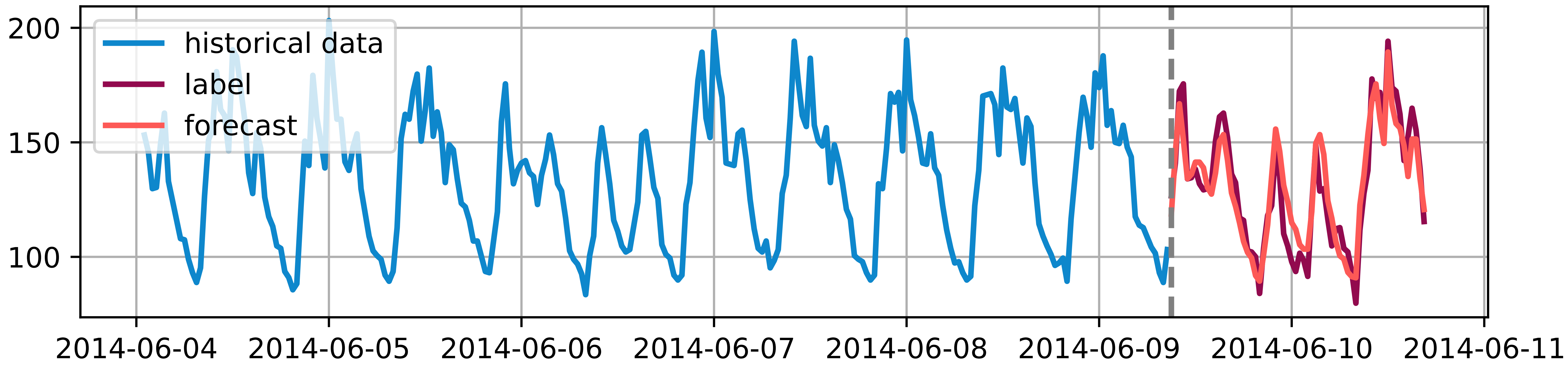
amazon science

Chronos: Pretrained (Language) Models for Probabilistic Time Series Forecasting

8 Contributors | 12 Used by | 6 Discussions | 2k Stars | 211 Forks

Models and code on Github

Evaluating Point Forecasts



Absolute Error: $e_t = |x_t - \hat{x}_t|$

Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{h}\sum_{t=T+1}^{T+h} e_t$$

scale-dependent error

Mean Absolute Percentage Error

$$\mathrm{MAPE} = \frac{1}{h}\sum_{t=T+1}^{T+h} \frac{e_t}{|x_t|}$$

percentage error

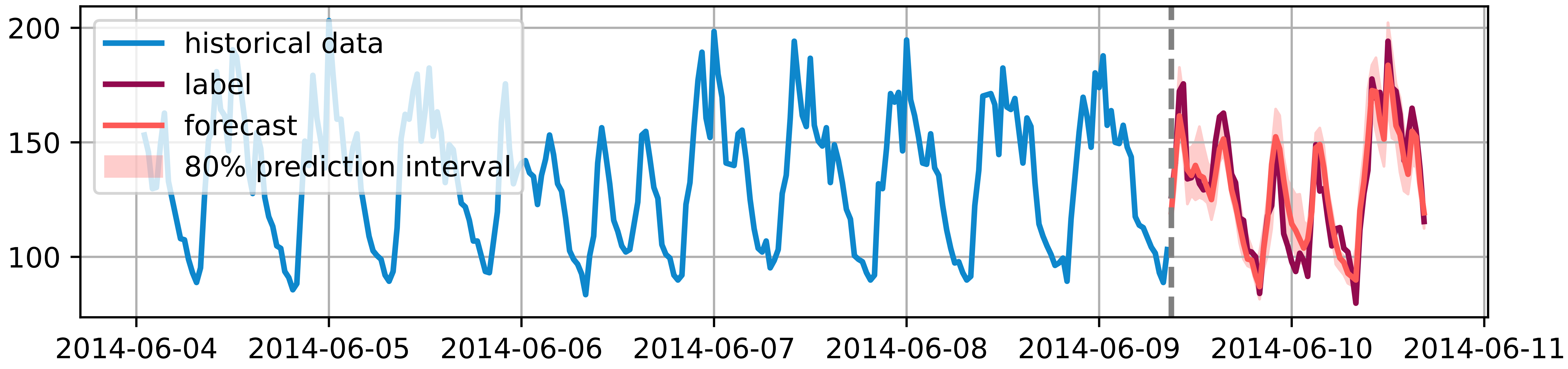
Mean Absolute Scaled Error

$$\mathrm{MASE} = \frac{1}{h}\frac{\sum_{t=T+1}^{T+h} e_t}{\sum_{t=1}^{T-s} |x_{t+s} - x_t|}$$

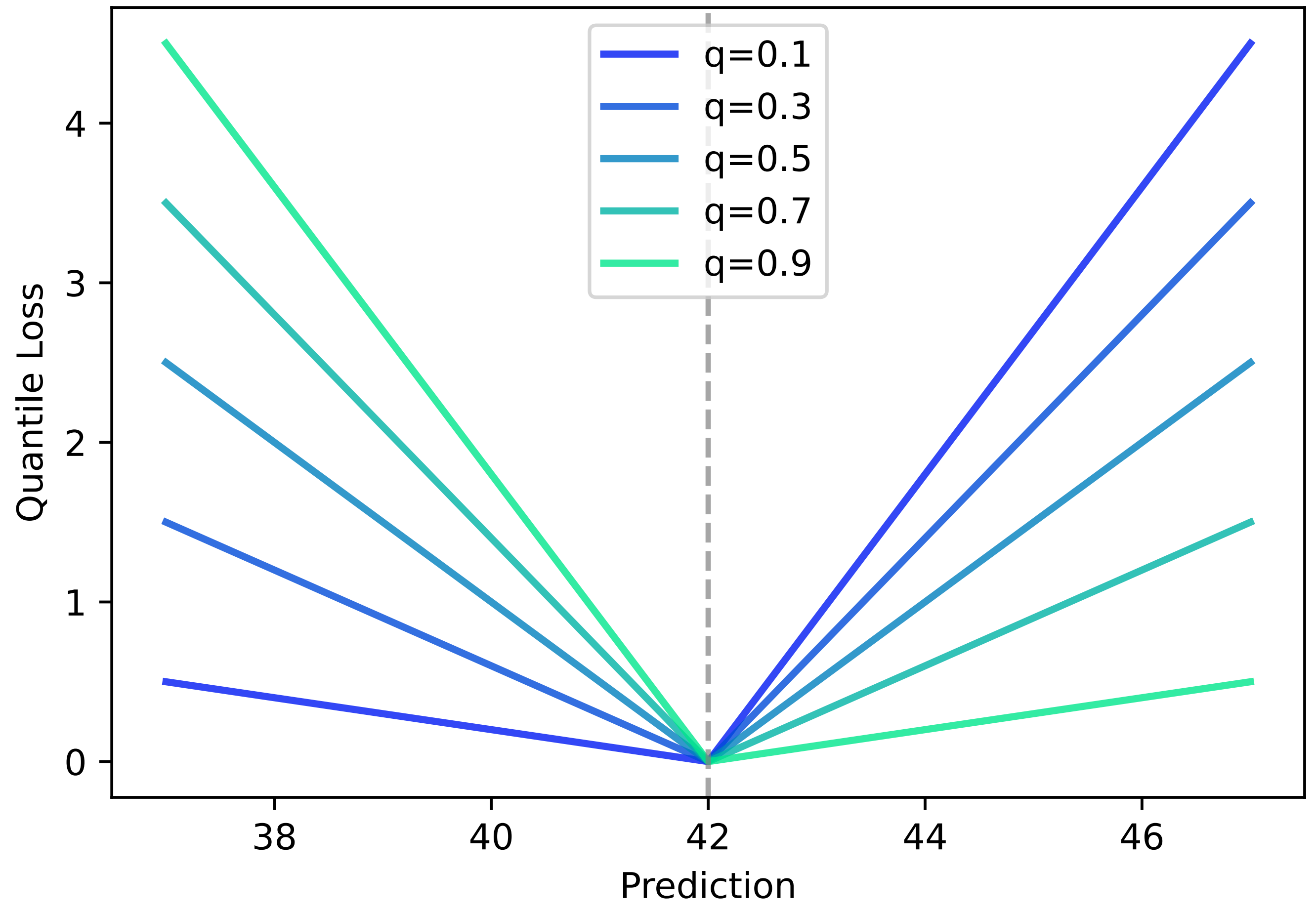
“scale-free” error

Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.

Evaluating Probabilistic Forecasts



Quantile Loss: $e_t^q = \begin{cases} q \cdot (x_t - \hat{x}_t^q) & \text{if } \hat{x}_t^q < x_t \\ (1-q) \cdot (\hat{x}_t^q - x_t) & \text{otherwise} \end{cases}$

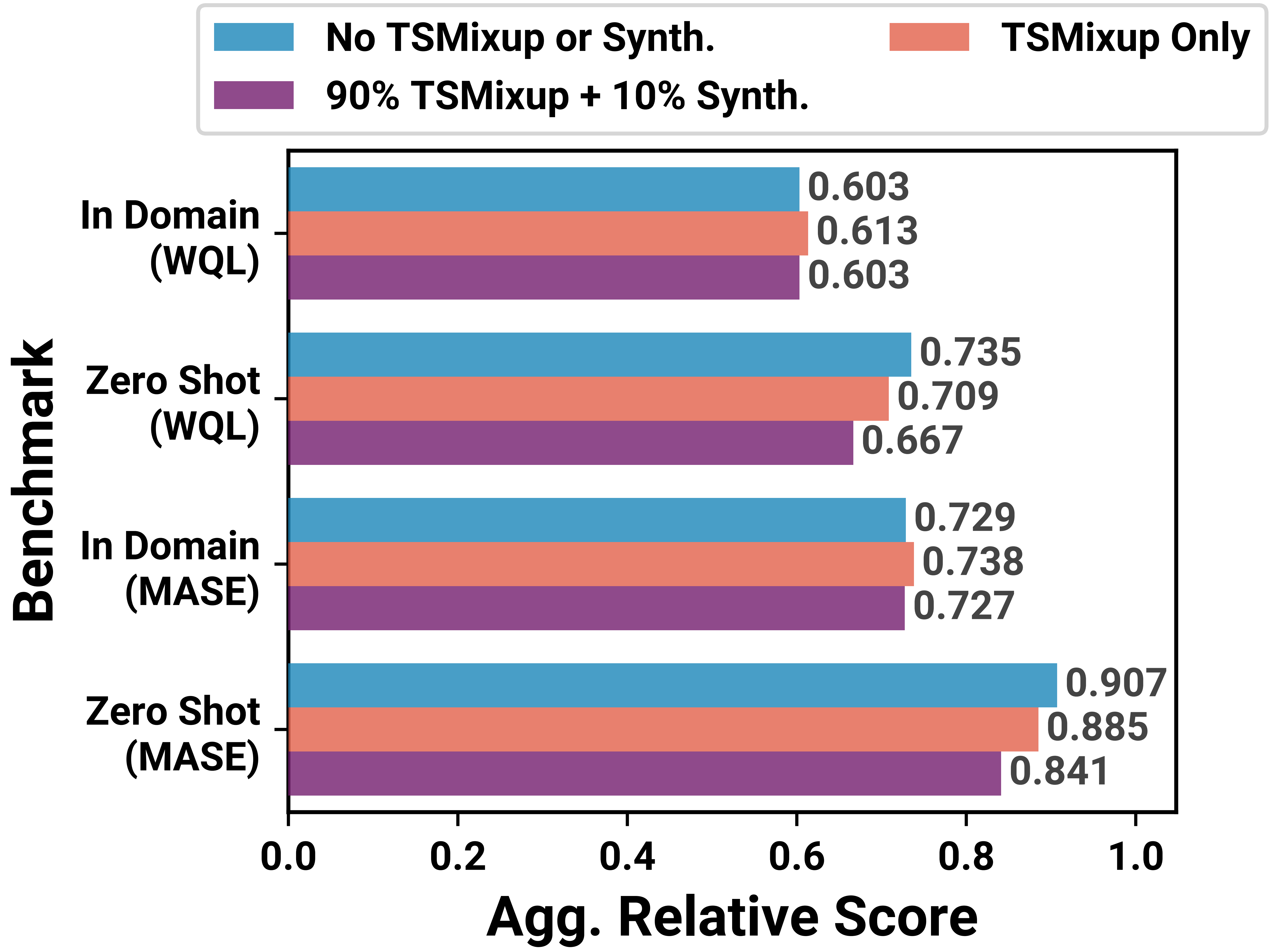


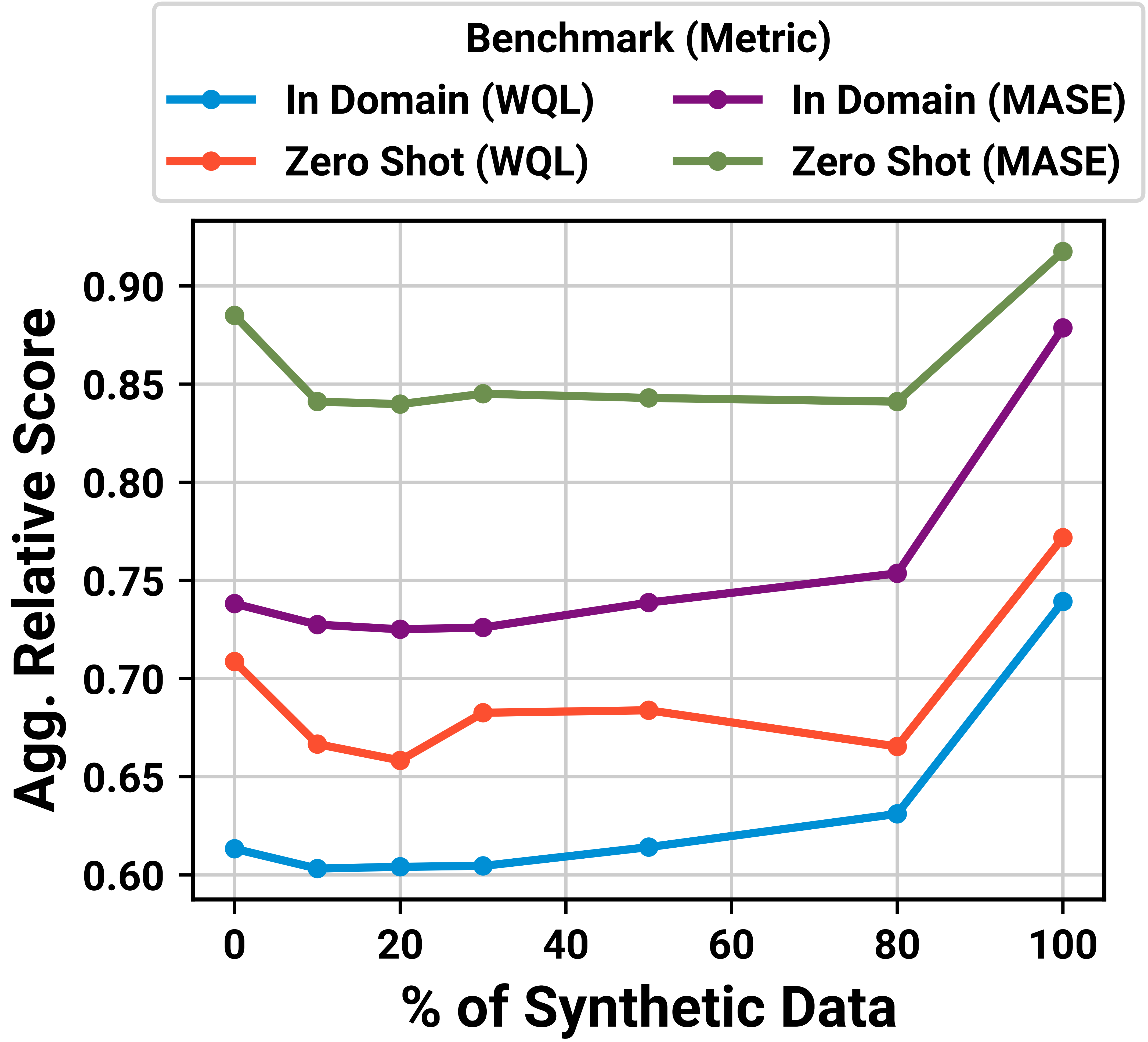
Continuous Ranked Probability Score

$$\text{CRPS} = \frac{1}{h} \sum_{t=T+1}^{T+h} \int_0^1 \text{QuantileLoss}(q)dq$$

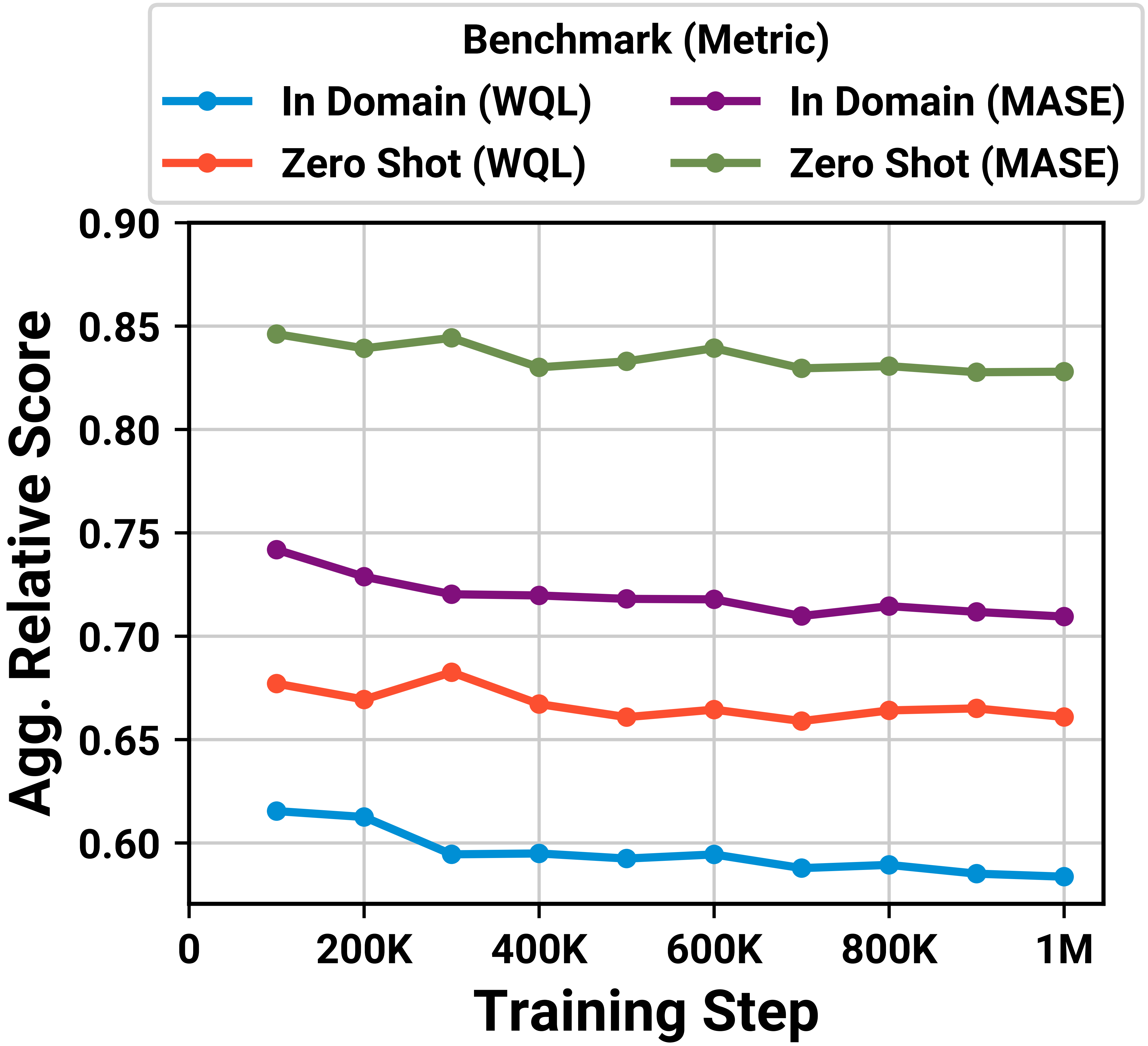
Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.

Do TSMix and KernelSynth help?

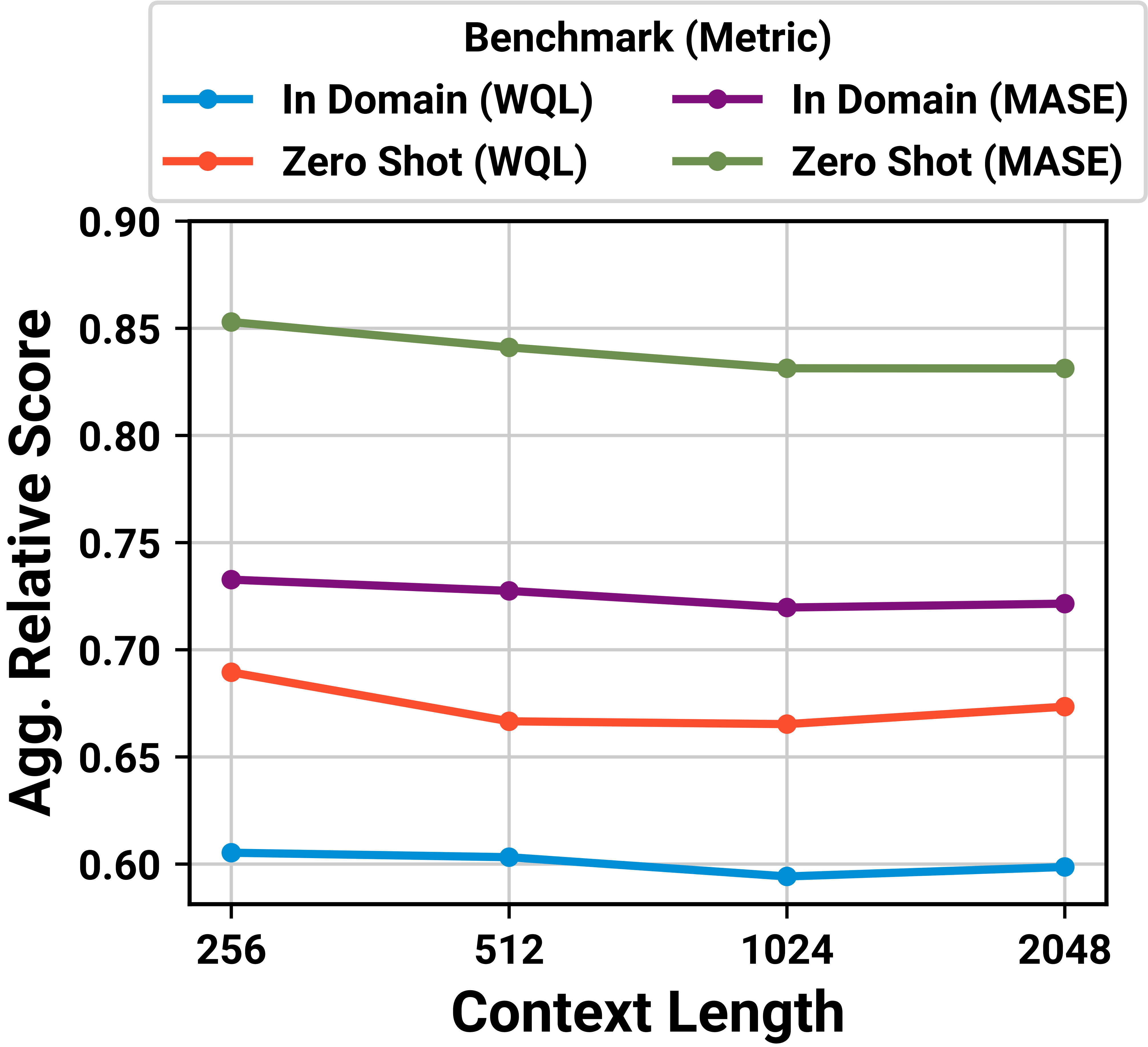




What if you train longer?



What if you increase the context length?



Does LLM initialization help?