Innovations in Time Series Forecasting: DeepAR[1] and iTransformer[2]

Exploring Key Technologies and Applications

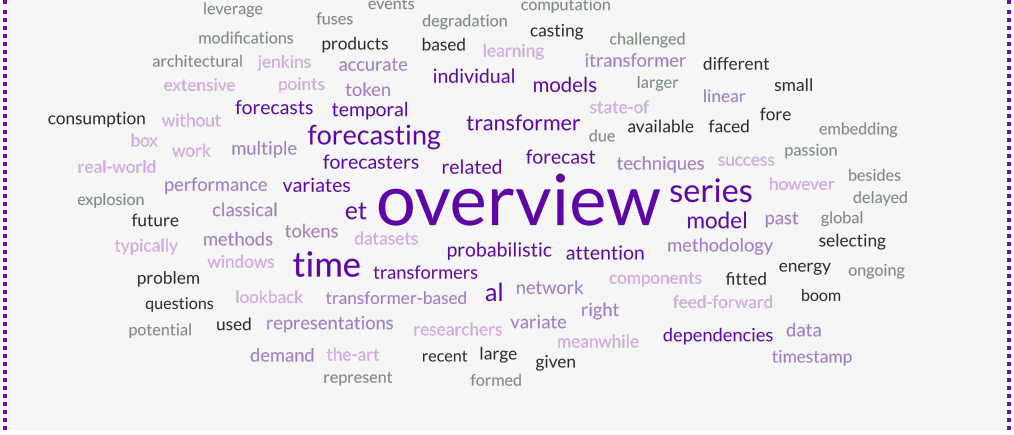
Presented by: Tobias Becher

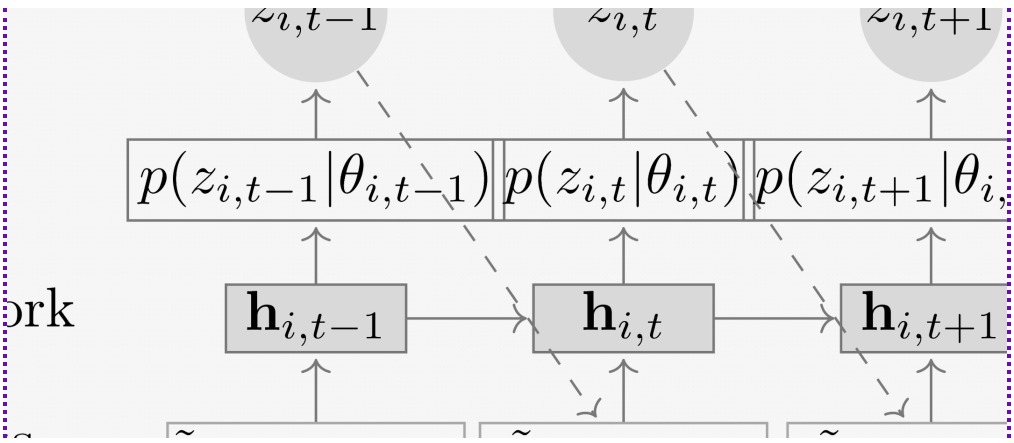
From: Technische Universität Berlin

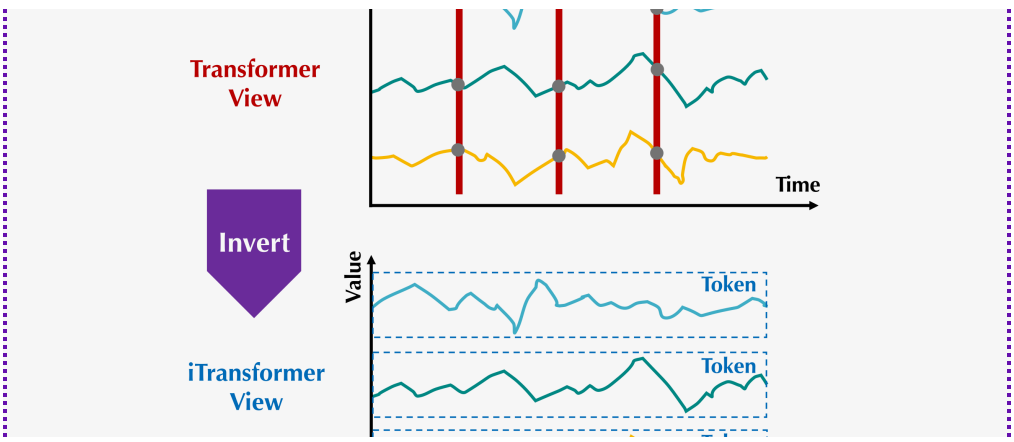
At: Tsinghua University

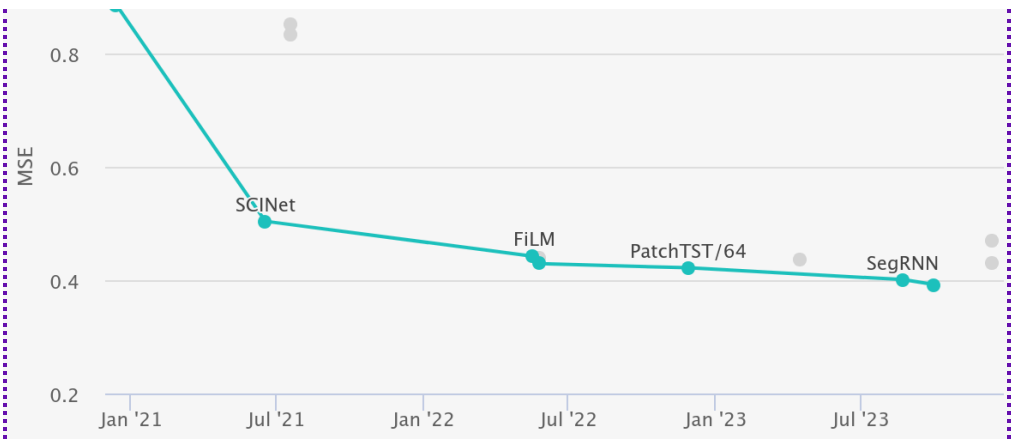
Date: 2024-05-15

[1] David Salinas, et al.. 2020. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" ISSN:01692070.
[2] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.


leverage
fuses
events
computation
degradation
modifications
products
based
learning
casting
challenged
architectural
jenkins
accurate
itransformer
different
extensive
points
token
individual
models
larger
linear
small
consumption
without
forecasts
temporal
state-of
fore
box
work
multiple
forecasting
transformer
due
available
faced
embedding
real-world
forecasters
related
forecast
techniques
success
passion
explosion
performance
variates
overview
series
however
besides
future
classical
et
model
past
delayed
typically
methods
tokens
datasets
probabilistic
attention
methodology
global
selecting
problem
windows
time
transformers
components
fitted
energy
ongoing
questions
lookback
transformer-based
al
network
feed-forward
boom
potential
used
representations
researchers
variate
right
dependencies
data
demand
the-art
recent
large
meanwhile
given
timestamp
represent
formed


$p(z_{i,t-1}|\theta_{i,t-1})$
$p(z_{i,t}|\theta_{i,t})$
$p(z_{i,t+1}|\theta_{i,}$
$\mathbf{h}_{i,t-1}$
$\mathbf{h}_{i,t}$
$\mathbf{h}_{i,t+1}$


Transformer View
Invert
iTransformer View
Time
Value
Token
Token


0.8
0.6
0.4
0.2
MSE
SCINet
FiLM
PatchTST/64
SegRNN
Jan '21
Jul '21
Jan '22
Jul '22
Jan '23
Jul '23

SUMMARY

Datasets Overview

Name	Description	Number of Time Series	Time Steps	Used By	Reference
Parts	Monthly sales of different items by a US automobile company	1046	50	DeepAR	Seeger et al., 2016
Electricity	Hourly electricity consumptions of 370 customers	370	Hourly	DeepAR	Yu et al., 2016
EC	Weekly item sales from Amazon	534,884	Weekly	DeepAR	Private
Traffic	Hourly occupancy rates of 963 car lanes in San Francisco bay area freeways	963	Hourly	DeepAR, iTransformer	Yu et al., 2016
ECL	Hourly electricity consumption data of 321 clients	321	Hourly	iTransformer	Wu et al., 2021
ETT	Electricity transformer telemetry data, various recording intervals	7, 7, 7, 7	Hourly, every 15 minutes	iTransformer	Li et al., 2021
Exchange	Daily exchange rates data panel from 8 countries	8	Daily	iTransformer	Wu et al., 2021
Weather	Meteorological factors, 21 types collected every 10 minutes	21	Every 10 minutes	iTransformer	Wu et al., 2021
Solar-Energy	Solar power production data of 137 PV plants	137	Every 10 minutes	iTransformer	Lai et al., 2018
PEMS	Public traffic network data from California	358, 307, 883, 170	Every 5 minutes	iTransformer	Liu et al., 2022a
Market	Minute-sampled server load data of Alipay transactions	285 to 759 variates	Minute-sampled	iTransformer	Wu et al., 2023

Tab. 1: Time-Series Forecasting Datasets.

Introducing DeepAR: Probabilistic Forecasting

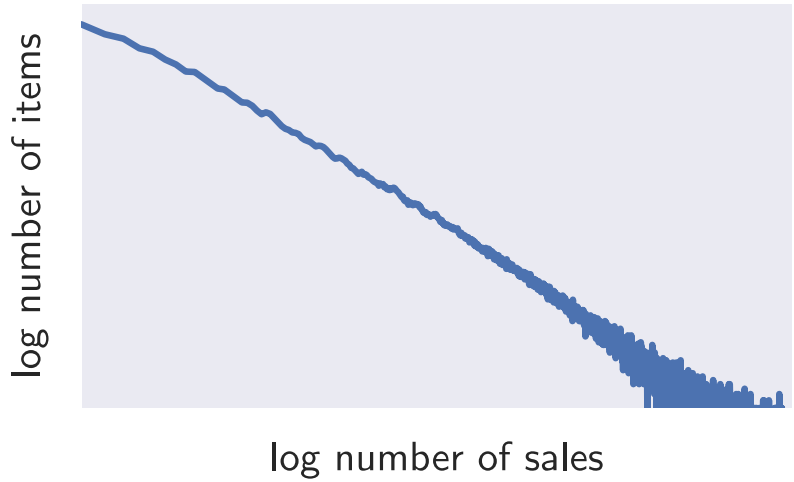
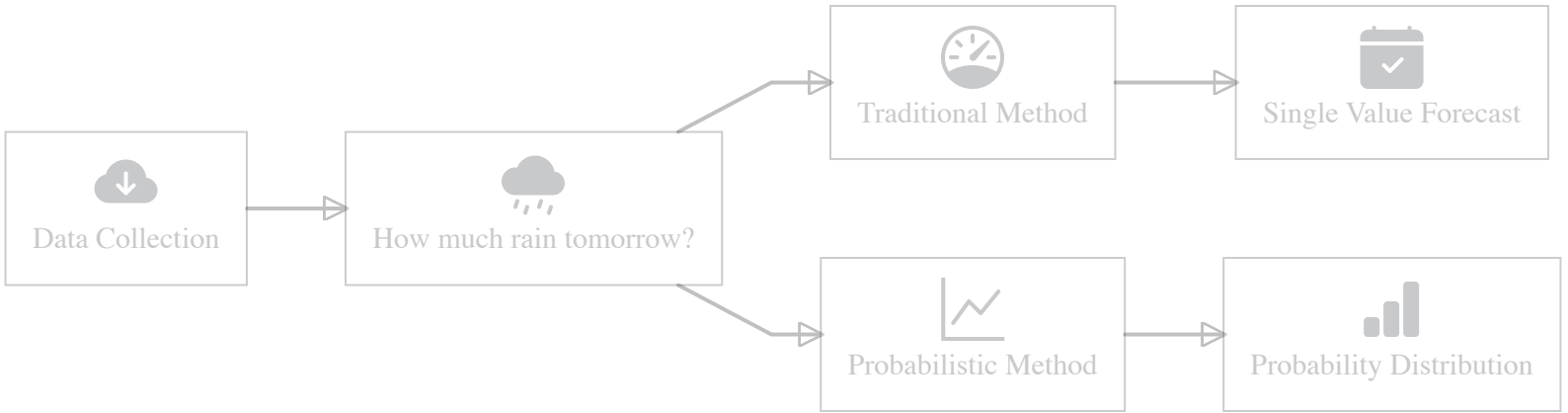


Fig. 1: Histogram of average sales per item (log-log scale)[1].

Motivation & Problem

- Prevalence of Skewed Distributions
  - Skewed sales data in large-scale applications (e.g., Amazon).
  - Challenge in capturing sales variability across products.
- Difficulties with Existing Forecasting Models
  - Current methods falter with scale-free distributions.
  - Inaccurate predictions for items with varying sales magnitudes.
- Necessity for Robust Forecasting Methods
  - Essential for managing high variability and skewness.
  - Traditional methods like normalization less effective.

[1] David Salinas, et al.. 2020. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" ISSN:01692070.


Data Collection
How much rain tomorrow?
Traditional Method
Single Value Forecast
Probabilistic Method
Probability Distribution

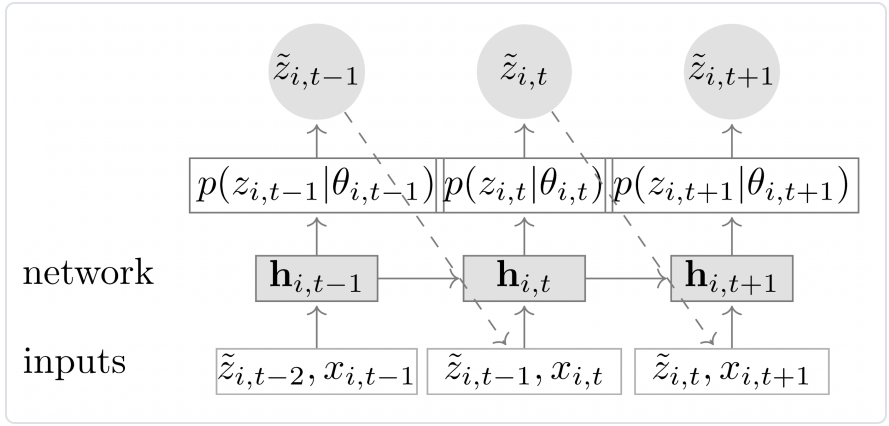


Fig. 2: DeepAR Model Architecture Diagram[1].

DeepAR Model Specifics

- **Autoregressive RNN:** Leverages LSTM cells to model temporal dependencies using:

$$h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$$

- **Negative Binomial Distribution:** Models count data effectively, discrete distribution to find number of trials needed to achieve a number of successes

$$p_{NB}(z|\mu,\alpha) = \frac{\Gamma(z+\frac{1}{\alpha})}{\Gamma(z+1)\Gamma(\frac{1}{\alpha})} \cdot \left(\frac{1}{1+\alpha\mu}\right)^{\frac{1}{\alpha}} \cdot \left(\frac{\alpha\mu}{1+\alpha\mu}\right)^{z}$$

Symbol	Description
h	Hidden state vector
z	Observed value at time t
x	Covariates or external factors
Θ	Model parameters
μ, α	Parameters of the distributions
Γ	Gamma function used in the distribution

Tab. 2: Symbols used in the DeepAR Model.

[1] David Salinas, et al.. 2020. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" ISSN:01692070.

Gaussian and Negative Binomial Distributions

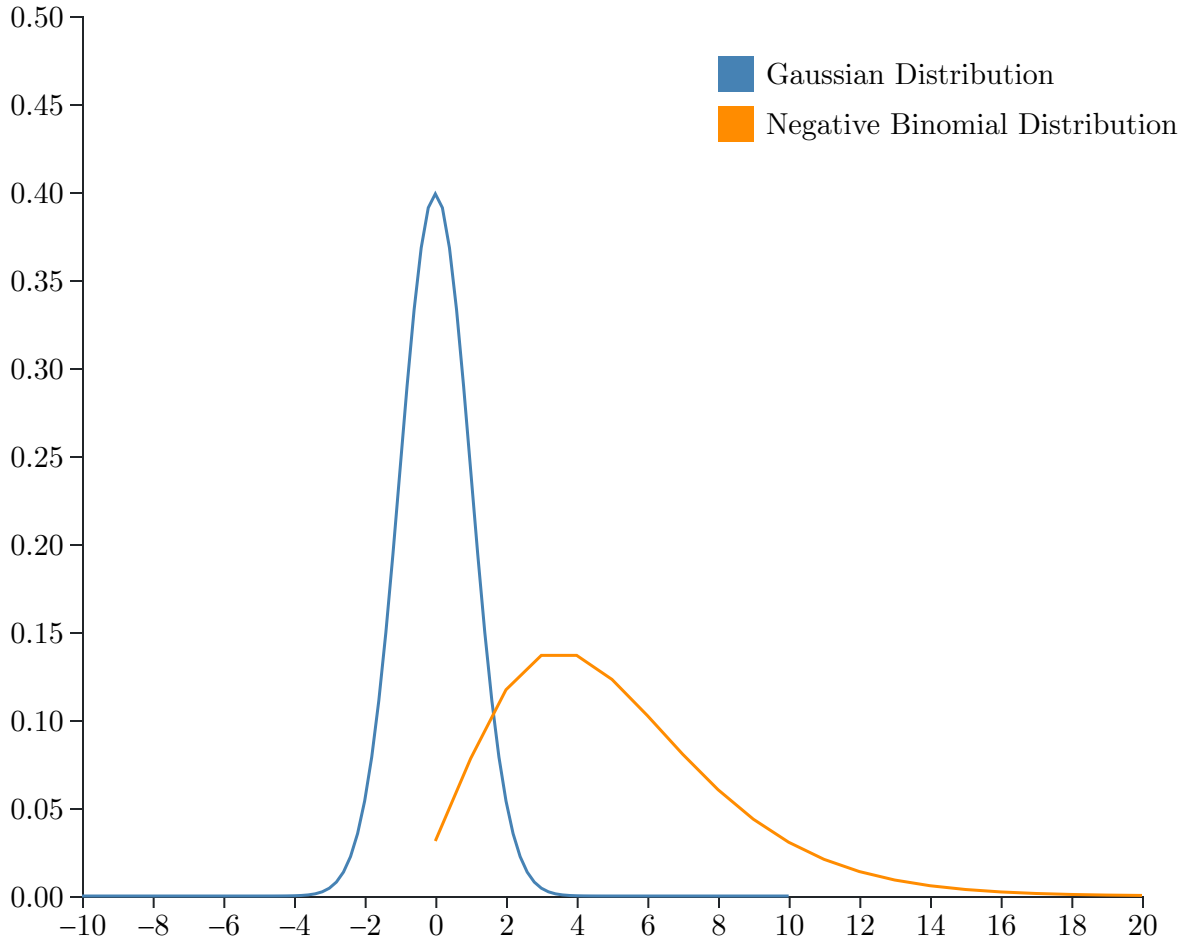
Gaussian Mean μ:
0

Gaussian Variance $\sigma 2$:
1

Negative Binomial Size α:
5

Negative Binomial Probability:
0.5

Reset to Default



DeepAR Training

- **Input Data Preparation:**
  Sliding window approach transforms historical data into training examples, allowing the model to learn from long-term and short-term trends.
- **Loss Function:**
  Maximize log-likelihood for accurate parameter estimation. The loss function is:

$$L = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p(z_{i,t} \mid \theta(h_{i,t})).$$

  where:
  - N is the number of time series.
  - t_0 is the starting point of the prediction window.
  - $p(z_{i,t} \mid \theta)$ is the likelihood function for series i at time t.
  - h represents the state of the recurrent neural network (RNN).
- **Optimization:**
  The Adam optimizer with early stopping ensures efficient learning by minimizing the loss function via stochastic gradient descent.

Scale Handling and Covariates

- **Scale Handling:**
  Scaling normalizes the input for each time series using an item-specific factor:

$$\nu_i = 1 + \frac{1}{t_0} \sum_{t=1}^{t_0} z_{i,t}.$$

Normalization & Weighted sampling

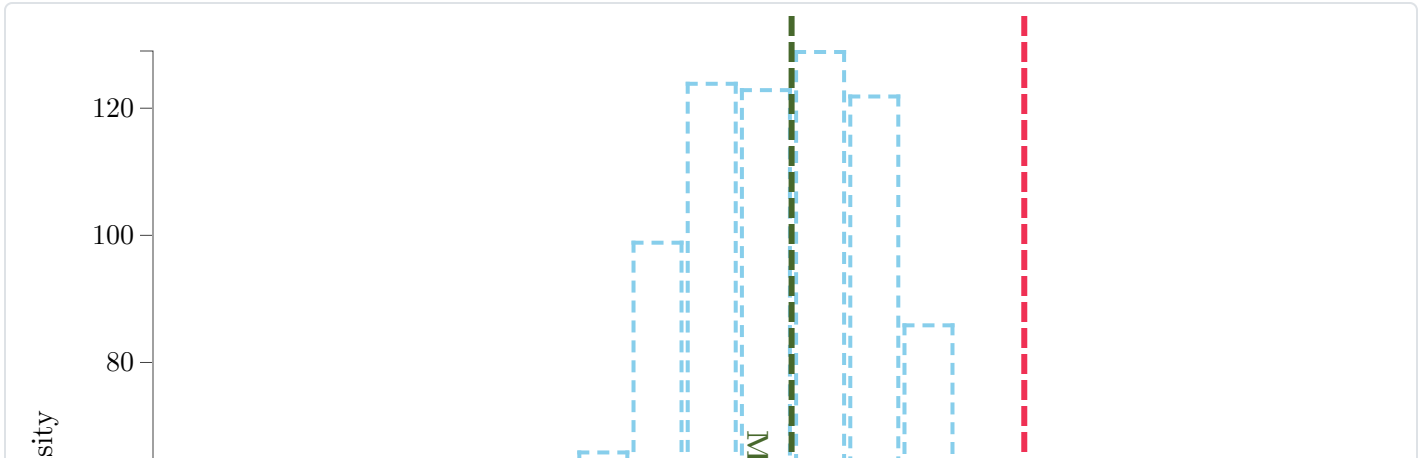
- **Covariates:**
  Covariates provide crucial contextual information to DeepAR. They include:
  - **Item-Specific:** Product categories and item attributes.
  - **Time-Dependent:** Temporal attributes like day-of-week, month-of-year, and special events.

External effects & Relationships

Fig. 3: Scale-Free Distribution for Time Series Data.

Accuracy Metrics

- **0.5-Risk (Median Quantile Loss):**
  Measures the model's accuracy in predicting the 0.5-quantile (median) value of the data distribution. Quantifies whether predictions fall above or below the observed data points.
- **0.9-Risk (Upper Quantile Loss):**
  Assesses the model's ability to predict the 90th percentile, focusing on the upper quantile of the data distribution.



Baseline Models

- **Croston Method**[1]:
  - Handles intermittent demand forecasting for inventory control.
  - Separately forecasts average demand size and interval between demands.
- **ETS Model**[2]:
  - Automatically selects exponential smoothing models for forecasting.
  - Uses state-space models to capture trend, seasonality, and noise.
- **Snyder's Negative Binomial Autoregressive Model**[3]:
  - Models count data variability with a negative binomial distribution.
  - Optimizes forecasting for slow-moving inventory with intermittent demand.
- **ISSM**[4]:
  - Bayesian state-space model that incorporates covariates for better predictions.
  - Addresses large-scale intermittent demand forecasting.

[1] Rob J. Hyndman and Yeasmin Khandakar. 2008. "Automatic Time Series Forecasting: The Forecast Package for R" ISSN:1548-7660.
[2] Rob Hyndman, et al.. 2008. "Forecasting with Exponential Smoothing: The State Space Approach" arXiv:GSyzox8Lu9YC.
[3] Ralph D. Snyder, et al.. 2012. "Forecasting the Intermittent Demand for Slow-Moving Inventories: A Modelling Approach" ISSN:0169-2070.
[4] Matthias W Seeger, et al.. 2016. "Bayesian Intermittent Demand Forecasting for Large Inventories" [Identifier missing].

Results - Overview

- **Outperformance in 0.5-Risk and 0.9-Risk:**
  DeepAR consistently outperformed baseline models across all datasets, indicating strong forecasting capabilities.
- **Handling Power-Law Datasets:**
  Weighted sampling and scaling enabled DeepAR to handle power-law distributions more effectively than other models.
- **Uncertainty Growth:**
  Learned patterns of non-linear uncertainty growth over time, especially around seasonal peaks.
- **Benchmark Performance:**
  DeepAR outperformed the best baseline model, capturing temporal correlations more effectively.

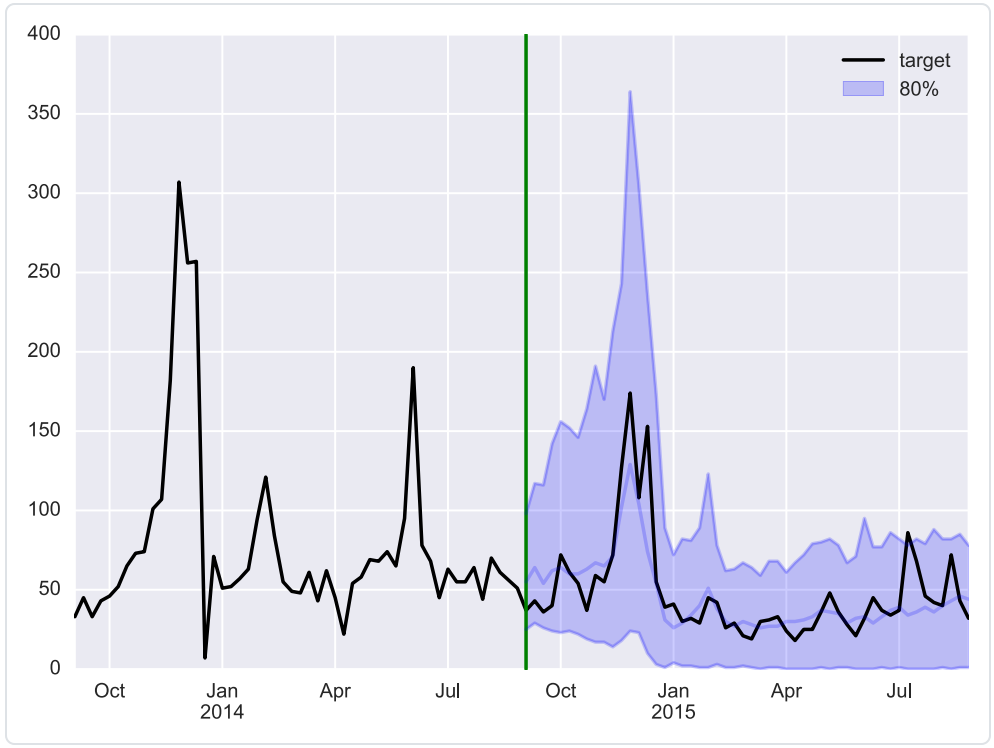


Fig. 4: Example time series of ec. The vertical line separates the conditioning period from the prediction period. The black line shows the true target[1].

[1] David Salinas, et al.. 2020. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" ISSN:01692070.

Results - Uncertainty

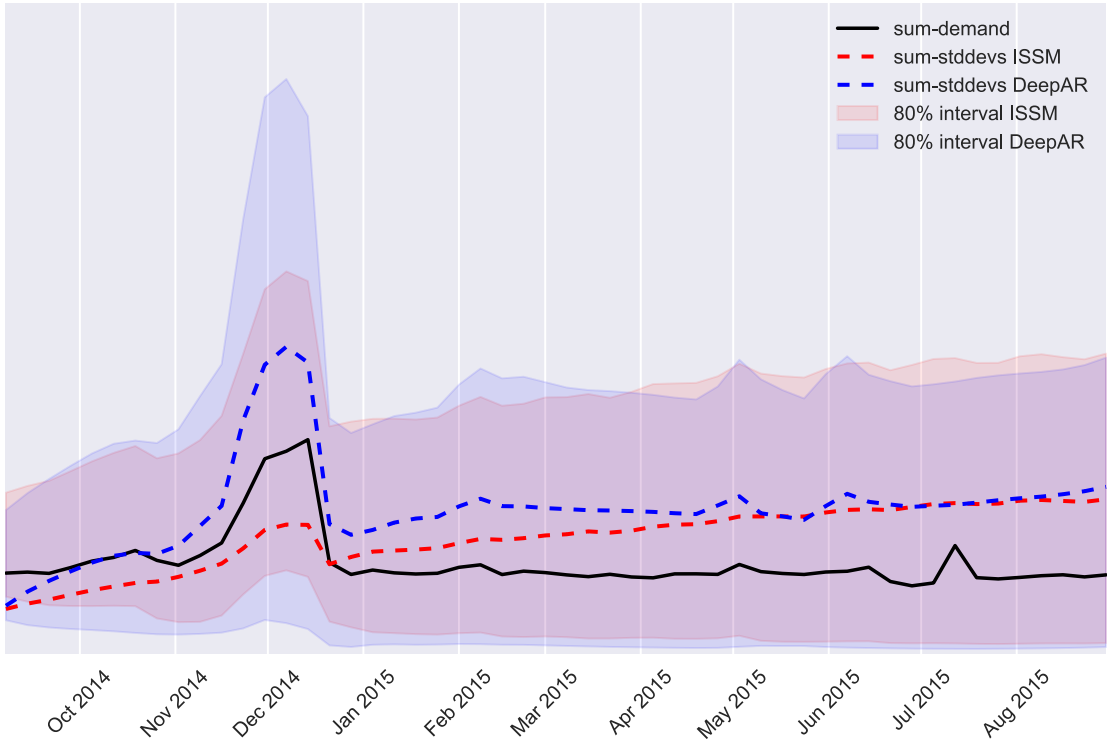
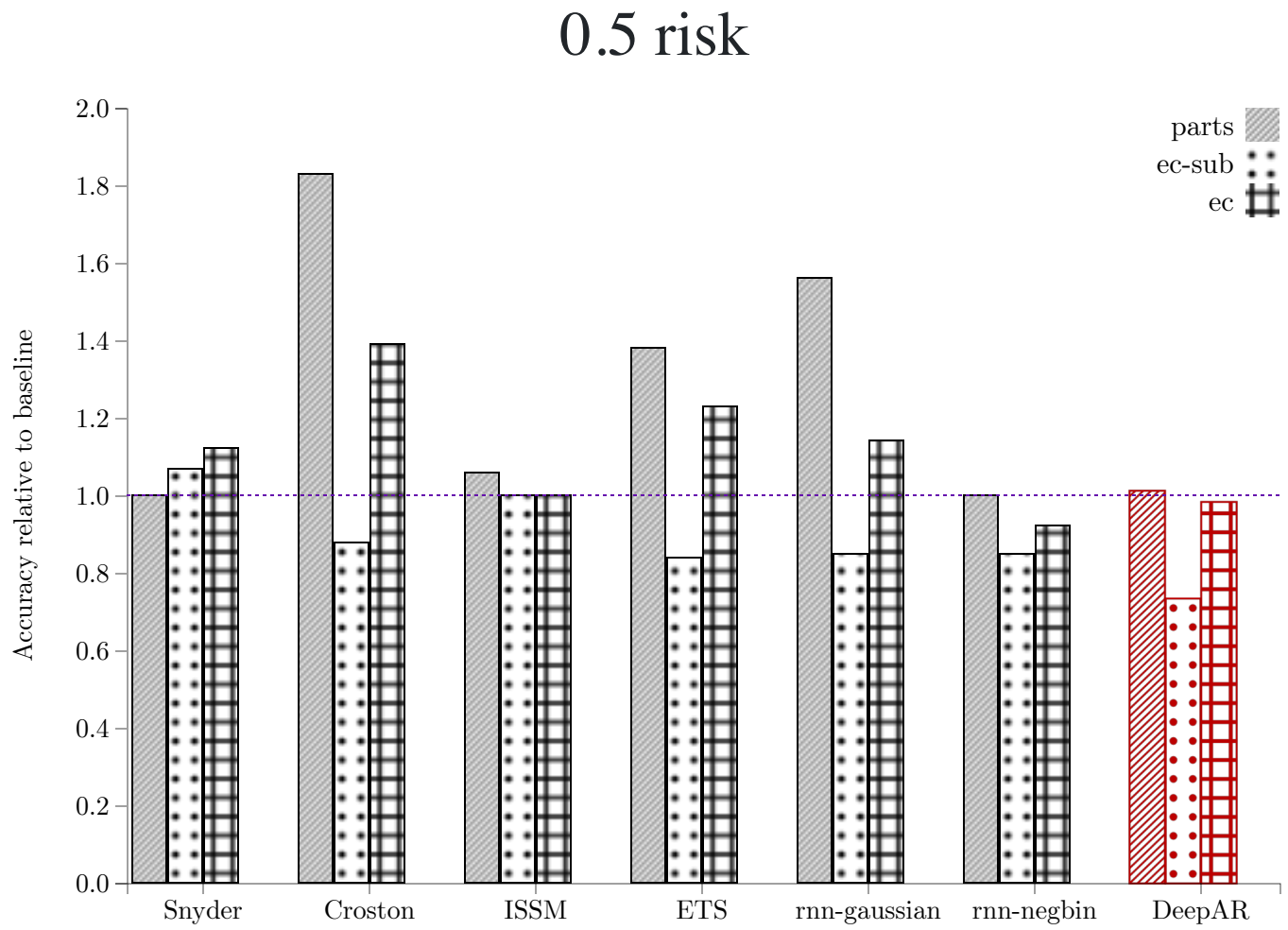


Fig. 5: Uncertainty growth over time for the ISSM and DeepAR models. Unlike the ISSM, which postulates a linear growth of uncertainty, the behavior of the uncertainty is learned from the data, resulting in a non-linear growth with a (plausibly) higher uncertainty around Q4. The aggregate is calculated over the entire ec dataset[1].

[1] David Salinas, et al.. 2020. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks" ISSN:01692070.

Accuracy Metrics



Introducing iTransformer: Inverted Transformers

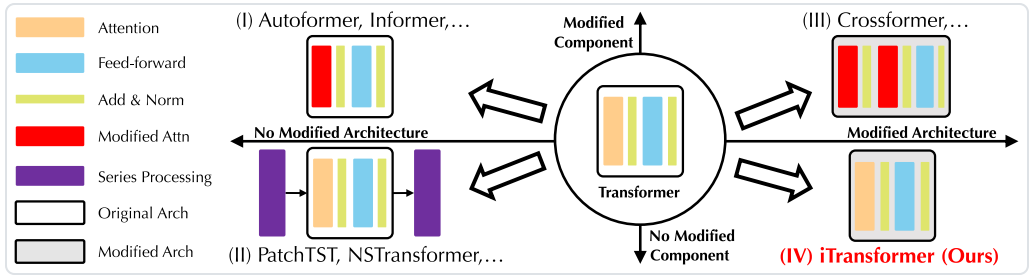


Fig. 6: Performance comparison of different transformer models[1].

Overview & Significance

- Inverted Transformer Architecture
  - Enhanced efficiency for longer sequences
  - Addresses quadratic memory challenge of standard transformers
- Advancements in Computational Efficiency
  - Reduces complexity and memory usage
  - Improves capture of long-term dependencies
- Effectiveness in Forecasting
  - Outperforms benchmarks in diverse forecasting scenarios
  - Sets new standards in forecasting accuracy

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

iTransformer Model Architecture

- **Architecture**
  Encoder-only, leverages Transformer
- **Tokenization**
  Treats time series as independent tokens
- **Self-Attention**
  Captures multivariate correlations efficiently
- **Feed-Forward Networks**
  Transforms series into insightful representations

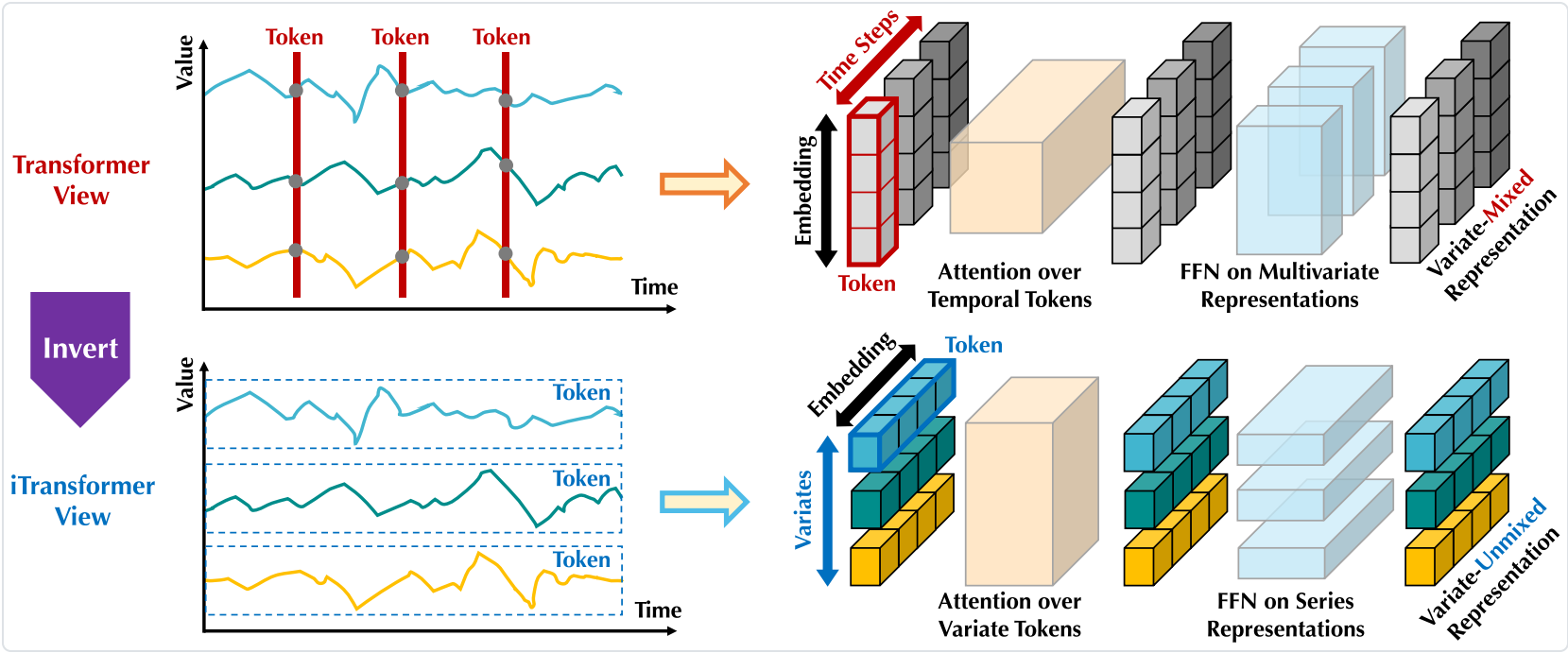


Fig. 7: Diagram of iTransformer architecture highlighting input handling through self-attention and feed-forward layers to output[1].

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

Detailed Architecture of iTransformer

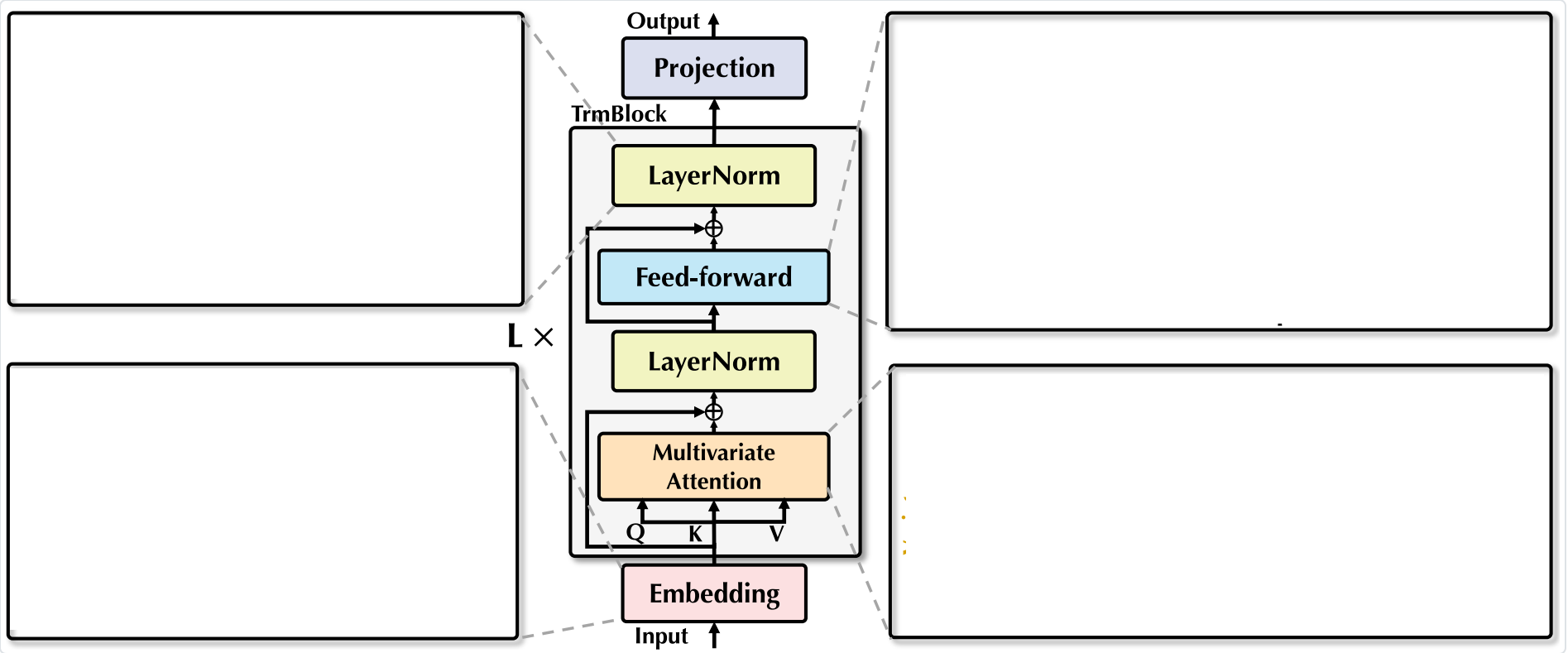


Fig. : Detailed architecture of iTransformer showing the specific components and their interconnections.

iTransformer Model Foundation

No special requirements are necessary for Transformer variants, enabling a range of efficient attention mechanisms.

Core Model

Predicting future series for specific variate:

$$\hat{Y}_{:,n} = \text{Projection}(h_n^L) \quad \text{where} \quad h_n^0 = \text{Embedding}(X_{:,n})$$

Where:

- $H^{l+1} = \text{TrmBlock}(H^l)$ for $l = 0, \ldots, L-1$
- $H = \{h_1, \ldots, h_N\} \in \mathbb{R}^{N \times D}$
- Embedding and Projection are MLPs
- Variate tokens interact via self-attention within each TrmBlock

Training and Efficiency

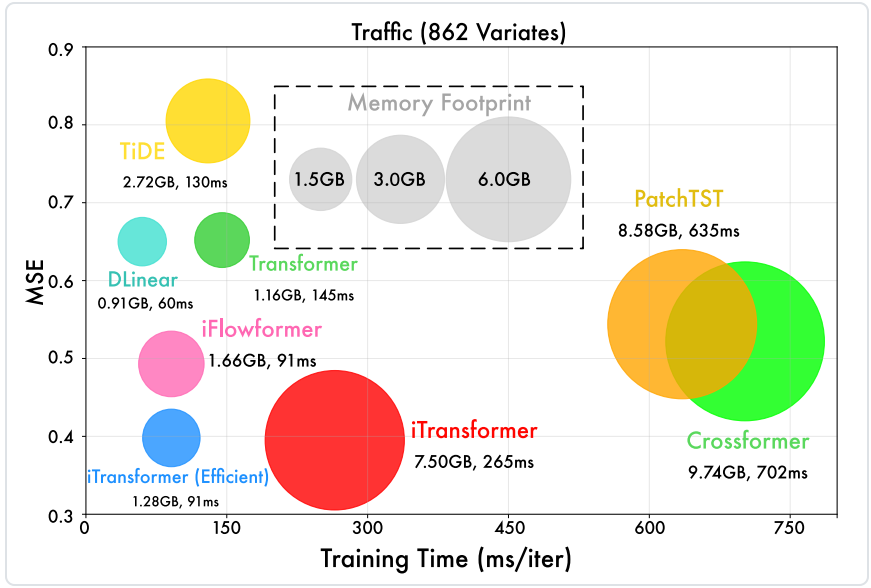


Fig. 8: Model is trained on arbitrary numbers of variates, allowing flexibility from training to inference phases[1].

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

iTransformer Detailed Results

- **Top Forecasting Accuracy:**
  - Reduces error rate by up to 34% compared to baselines
  - Particularly effective in complex datasets like PEMS-BAY and TRAFFIC
- **Ablation Study Highlights:**
  - Robust across different setups
  - Best performance with variate attention and temporal FFN
- **Outstanding Performance Metrics:**
  - Outperforms standard models by up to 60.2% in MSE
  - Superior performance across diverse datasets and forecasting horizons
- **Model Analysis and Innovations:**
  - Inverted self-attention enhances accuracy
  - Temporal FFN leads to significant performance gains

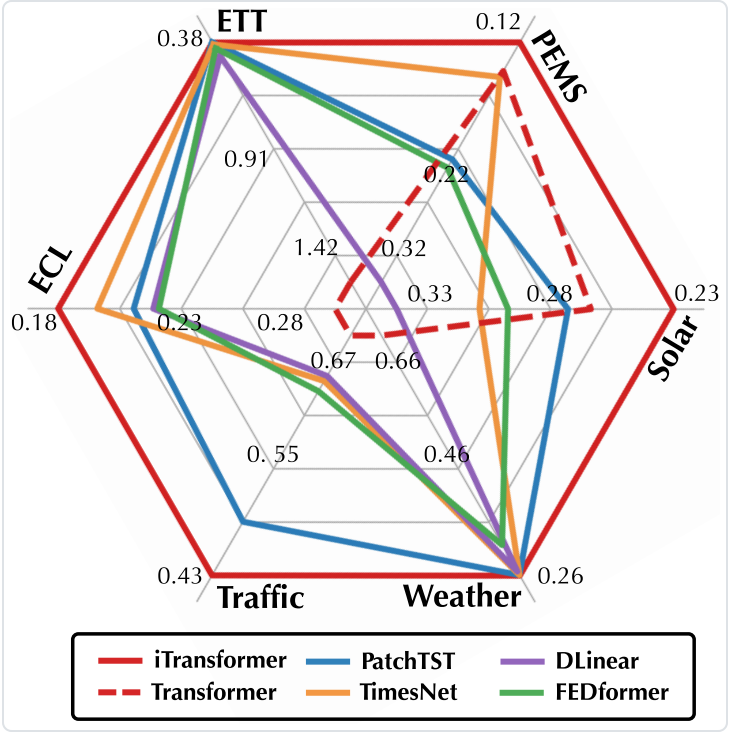


Fig. 9: General performance of iTransformer across different metrics[1].

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

iTransformer: In-depth Analysis of Forecasting Enhancements

- **Variate Generalization:**
  - Enhances predictive robustness with only up to 0.6% deviation
  - Improves forecast accuracy by 12% using cross-variational learning from up to 30 data streams

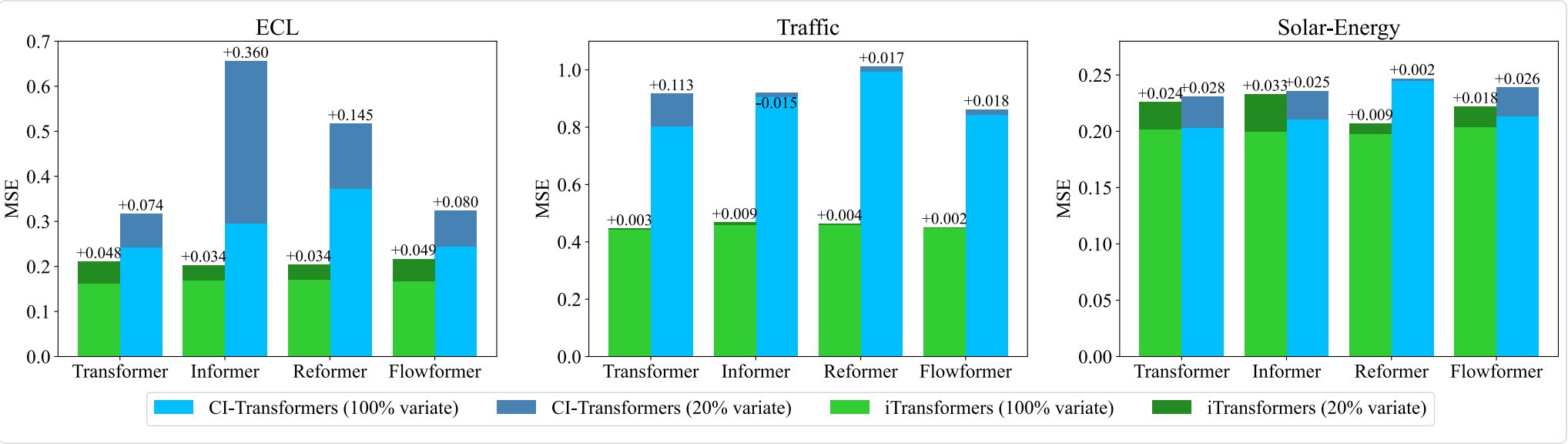


Fig. 10: Visualizing iTransformer's variate generalization capabilities[1].

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

iTransformer: In-depth Analysis of Forecasting Enhancements

- **Increasing Lookback Length:**
  - 15% increase in predictive precision with lookback extension to 500 steps
  - Adaptive lookback dynamically adjusts to the complexity of data series

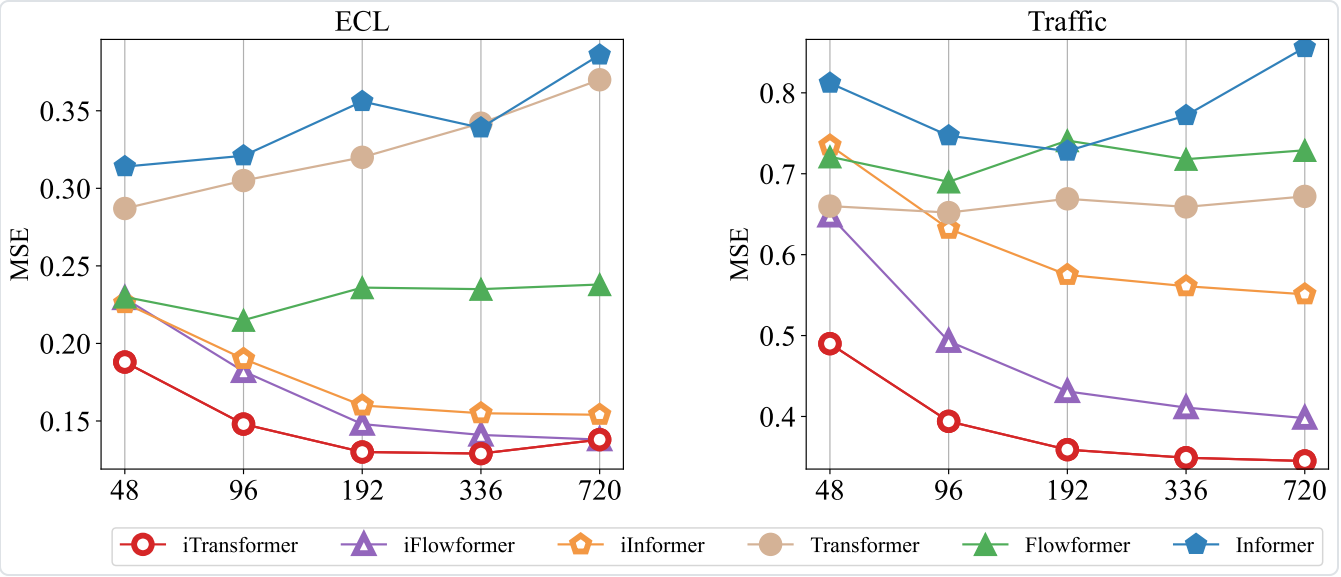


Fig. 11: Visualizing iTransformer's lookback length capabilities[1].

[1] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

Related Work

- **DeepAR (RNN-based, 2020)**
  - **Hyndman et al. (2008)**
    Exponential smoothing and state space models inadequate for intermittent, lumpy demand.
  - **Seeger et al. (2016)**
    Negative binomial distributions, multi-stage likelihoods for zero-inflated, over-dispersed count data.
  - **Chapados (2014)**
    Critique of group-based regularization; leads to individual scale adjustments in DeepAR.
- **iTransformer (Transformer-based, 2024)**
  - **Vaswani et al. (2017)**
    Limitations of Transformer architecture with large lookback windows; motivates iTransformer's inverted dimension approach.
  - **Liu et al. (2020)**
    Favors tokenization while maintaining standard components over altering Transformer architectures.
  - **Kitaev et al. (2020)**
    Reformer's hashing for sequence length management; inspires iTransformer's handling of time series lengths.

State of the Art Comparison

- **Comparison on `Traffic`:**
  - This dataset is the only one commonly tested in both studies
  - iTransformer: **0.428** MSE (converts to 0.65 RMSE)
  - DeepAR: **0.176** MSE (from 0.42 RMSE)
- **Comparison on `ETTh1 (720)`:**
  - iTransformer: **0.503** MSE
- **Performance Discrepancies:**
  - Experimental setup: input sequence, preprocessing, training variations
  - Model implementations: data splits, parameter initialization, training variability

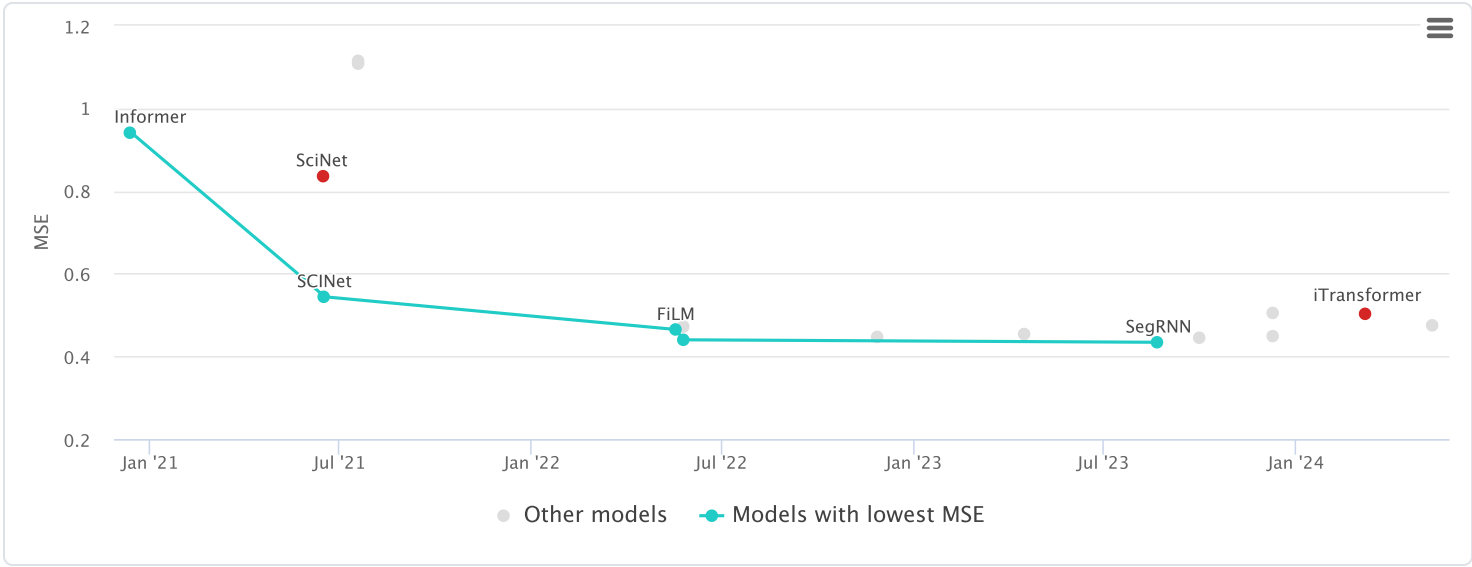


Fig. 12: Comparison of iTransformer with other models on ETTh1 (720) Dataset[1]. Points in red are added from the results published in iTransformer[2].

[1] paperswithcode.com, "Papers with Code - ETTh1 (720) Multivariate Benchmark (Time Series Forecasting)", https://paperswithcode.com/sota/time-series-forecasting-on-etth1-720-1, accessed on 2024-05-13.
[2] Yong Liu, et al.. 2024. "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting" arXiv:2310.06625.

Summary: DeepAR & iTransformer

DeepAR (Autoregressive RNN, 2020)

Contributions

- Probabilistic forecasting with LSTMs
- Global model from large datasets
- Negative binomial distribution for count data

Advantages

- Minimal manual feature engineering
- Calibrated probabilistic forecasts
- Superior accuracy in real-world datasets

Future Directions

- Advanced sampling for better calibration

iTransformer (Inverted Transformer, 2024)

Contributions

- Inverted Transformer architecture
- High-dimensional forecasting
- Variate-centric tokenization

Advantages

- Effective multivariate correlations
- Strong generalization in forecasting
- Adaptable to different data characteristics

Future Directions

- Exploration of large-scale pre-training
- Complex scenarios like real-time forecasting

Thank you! 🙏

Questions? 🤔