

# Deep Factors for Forecasting

Yuyang Wang, Alex Smola, Danielle C. Maddix,  
Jan Gasthaus, Dean Foster, Tim Januschowski

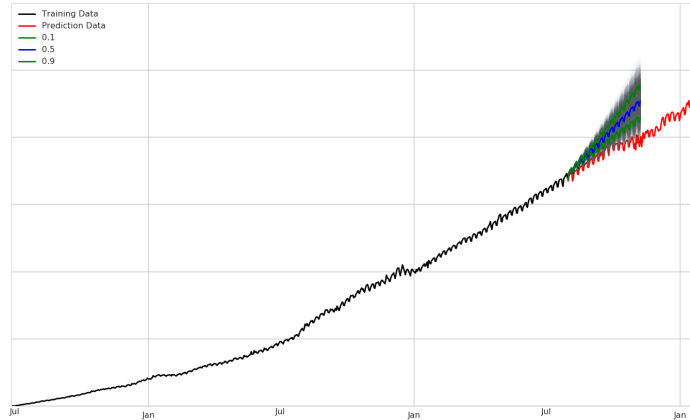
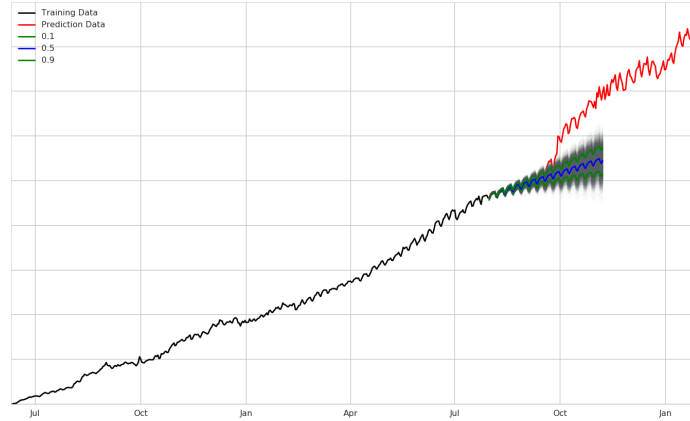


Amazon AI



# Time Series Prediction ... at Amazon

weekly units  
shipped  
and **forecast**  
years ahead



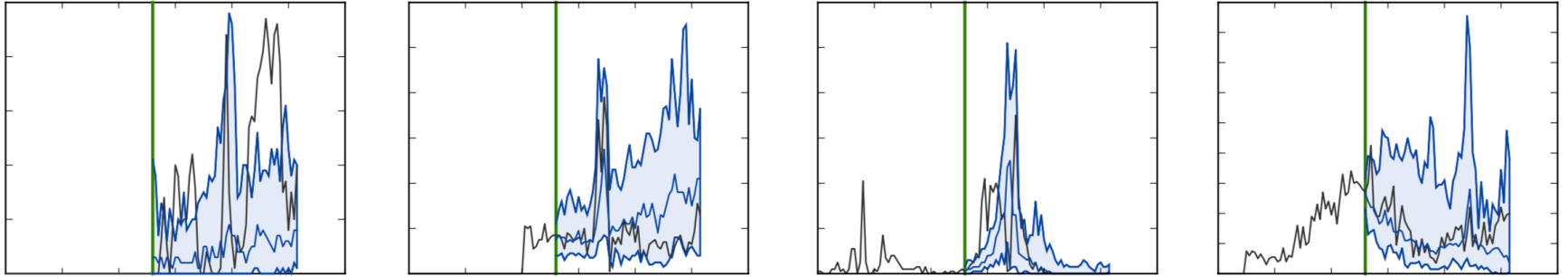
servers  
**forecast**  
and **used**

Capacity  
planning

Market  
entry

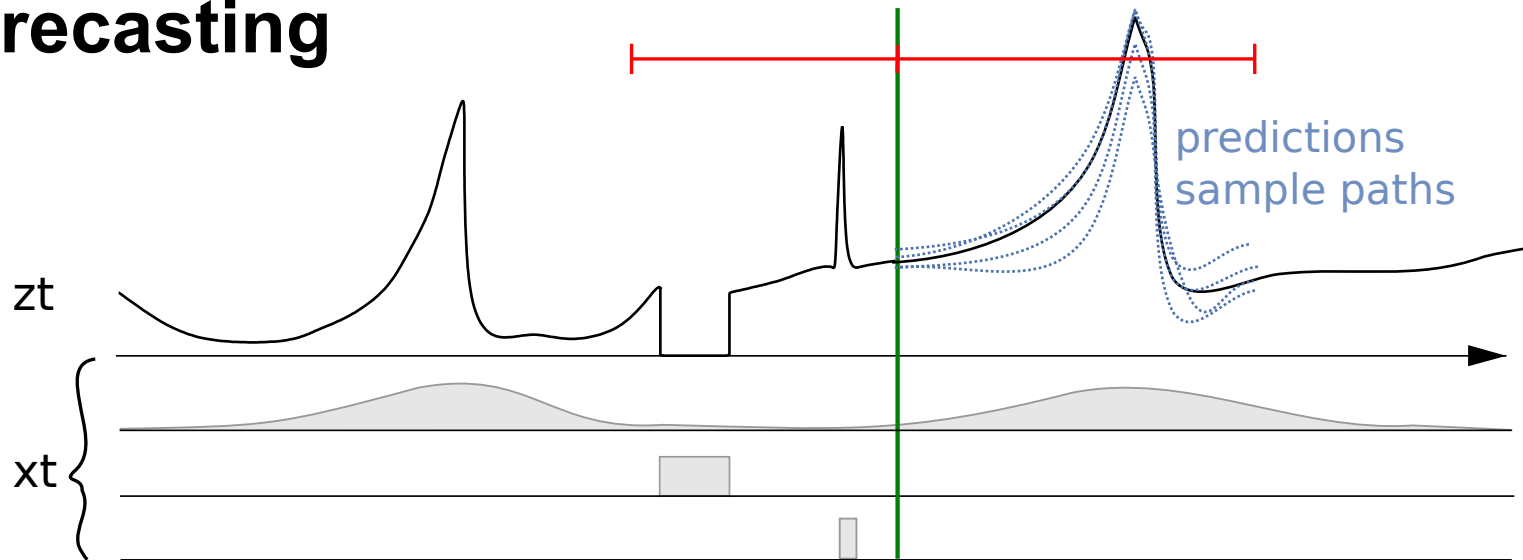
Topology  
Planning

# Time Series Prediction ... at Amazon



- Predict demand for each product available at Amazon
- Problem
  - How many items to order
  - Where to order
  - When to mark down (ugly sweaters after Christmas)

# Forecasting



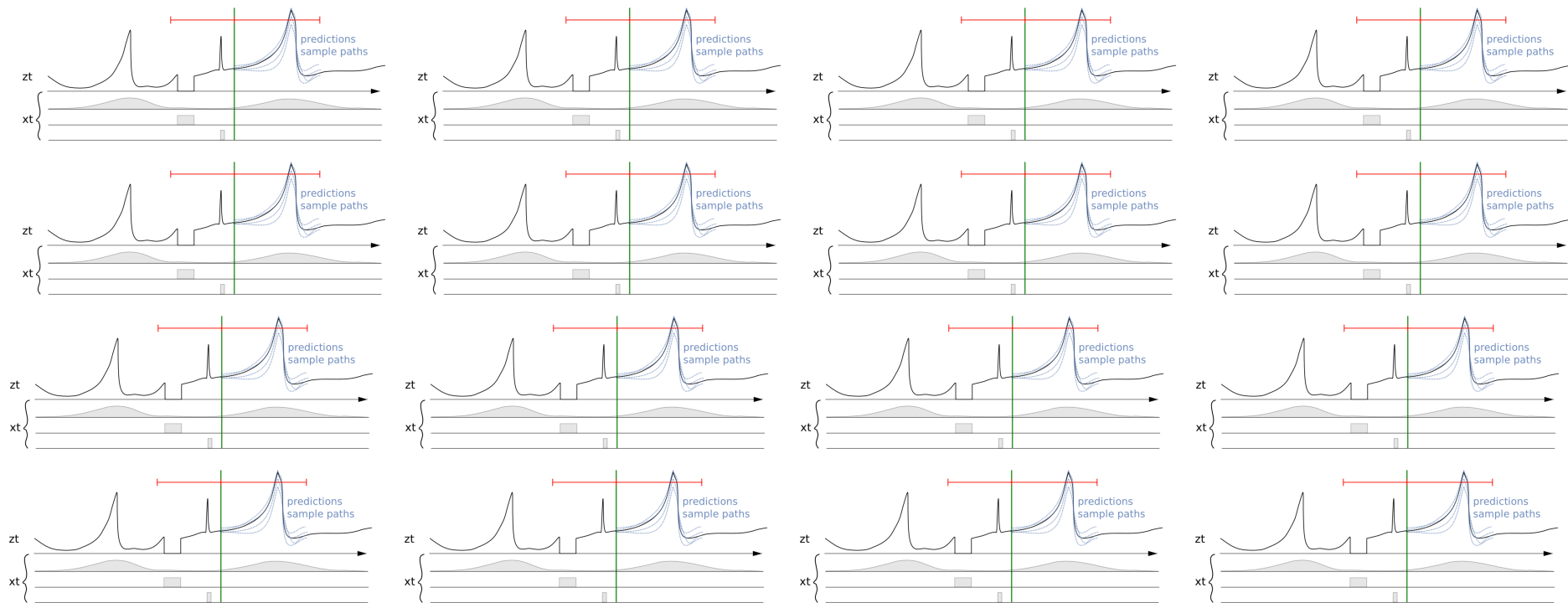
Estimate future observations (univariate case)

$$p(z_{t+1} | (x_1, z_1), \dots, (x_t, z_t), x_{t+1})$$

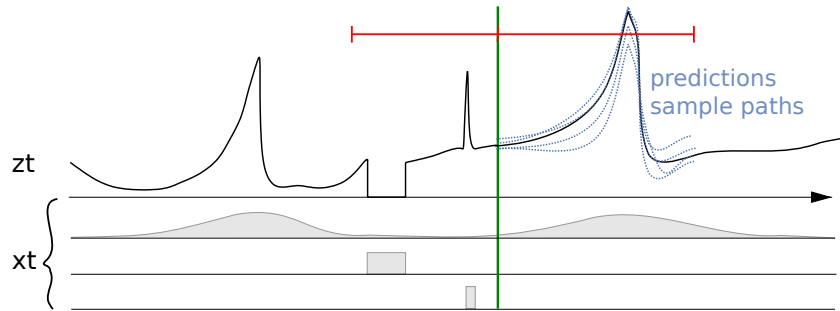
Make optimal decisions

$$\operatorname{argmin}_a \mathbb{E}_{z_{t+1} | \text{past}} [l(a, z_{t+1}, \text{past})]$$

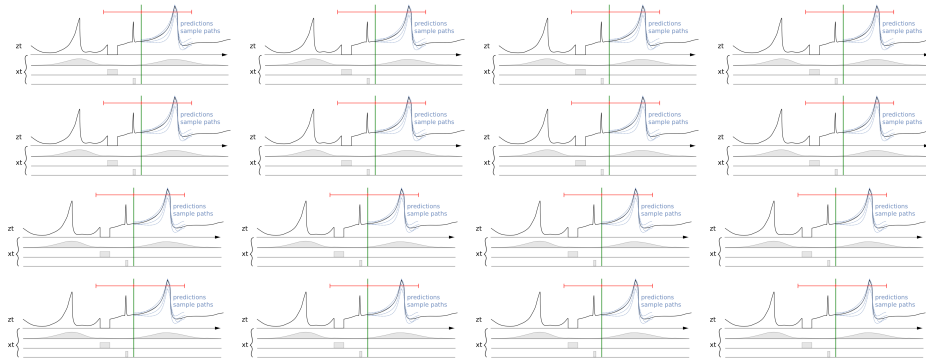
# But in reality ...



# Two old ideas



- Model each time series locally and individually
  - Easy to add more
  - Simple models
  - Doesn't work so well



- Model all time series jointly and globally
  - Works better
  - Impossible to add more
  - Complex model

# ... make a good one

- **Local model**

- Reads from global model
- Updates local state

- **Global model**

- Nonlinear backbone
- Nonparametric

- **Theorem (deFinetti for time series)**

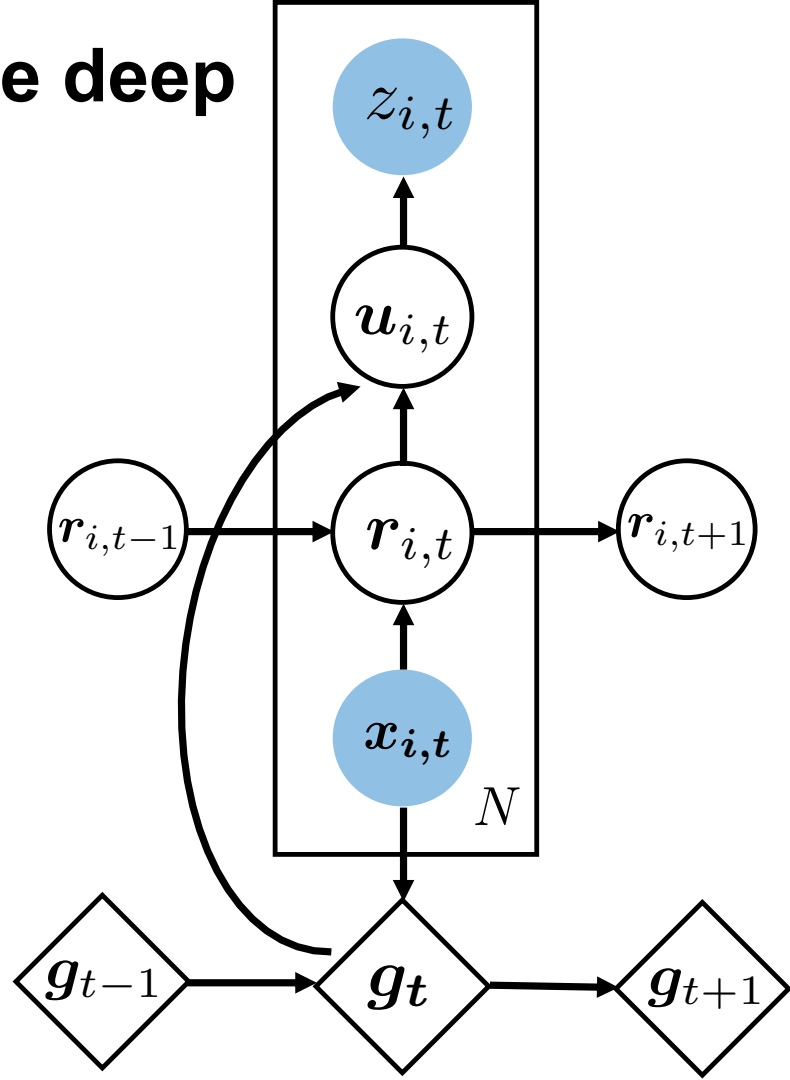
For an **exchangeable** distribution over time series the joint distribution can be written as a local/global model.

$$p(x_{1,\dots,T}^i \text{ for } i \in \{1,\dots,N\}) = \int dg \prod_{t=1}^T p(g_t | g_{t-1}, \dots, g_1) \prod_{i=1}^N p(x_t^i | x_{t-1}^i, \dots, x_t^i, g_t, \dots, g_1)$$

- Corresponding result for trees, too (via Tree-deFinetti)

# Latent variable autoregressive deep

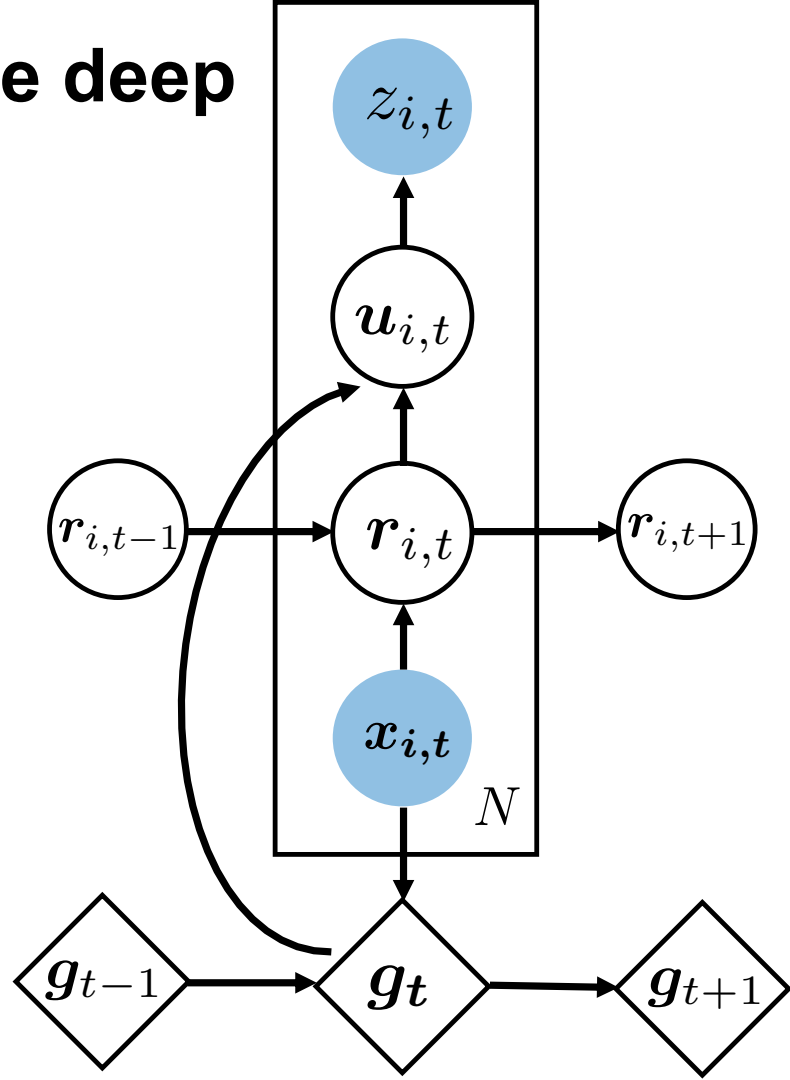
- **Global model**
  - Nonlinear backbone
  - Nonparametric
- **Local model**
  - Reads from global model
  - Updates local state
- Works well in practice





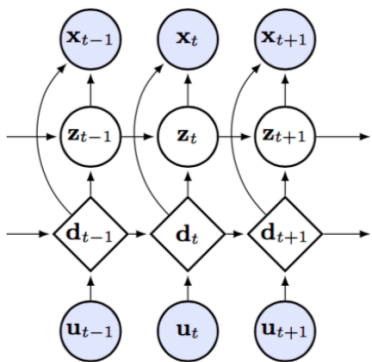
# Latent variable autoregressive deep

- **Global model**
  - Nonlinear backbone
  - Nonparametric (e.g. LSTM) to handle complex dependence
- **Local model**
  - Reads from global model
  - Use LDS / Gaussian Process for speed and uncertainty characterization
- **Inference**
  - Exact for Gaussian
  - VAE for other likelihoods

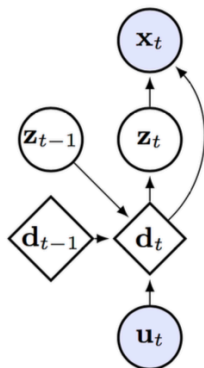


# We are not the first to realize this ...

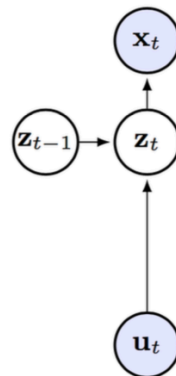
SRNN [Fraccaro et al., 2016]



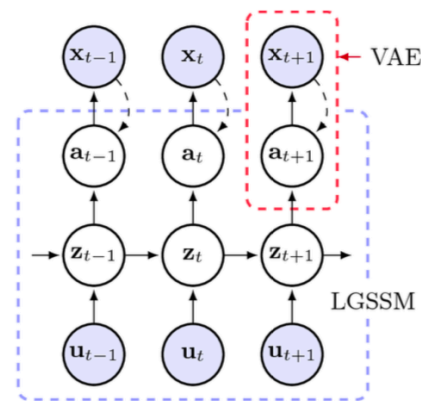
VRNN/SSL/LSTM-LDA [Chung et al., 2015; Zaheer et al., 2017; Zheng et al., 2017]



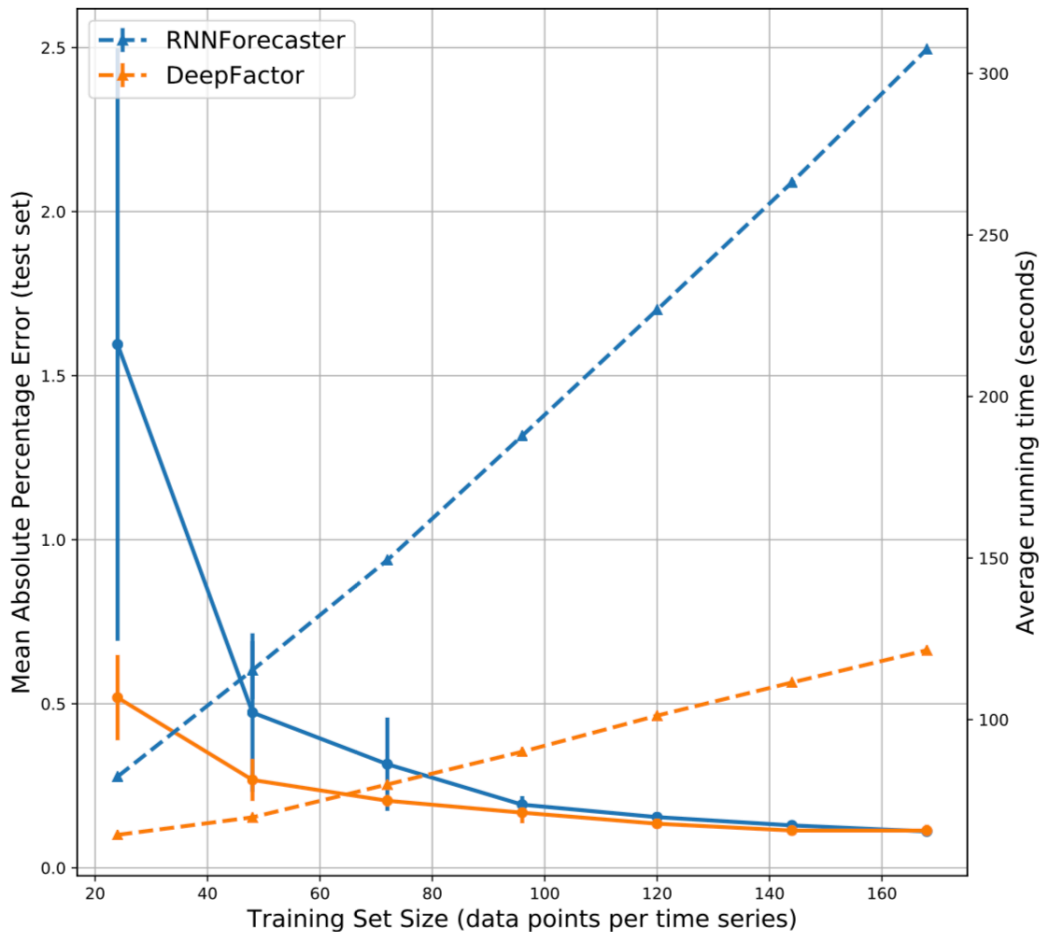
DMM [Krishnan et al., 2017, 2015]



KVAE/DVBF [Fraccaro et al., 2017; Karl et al., 2017]



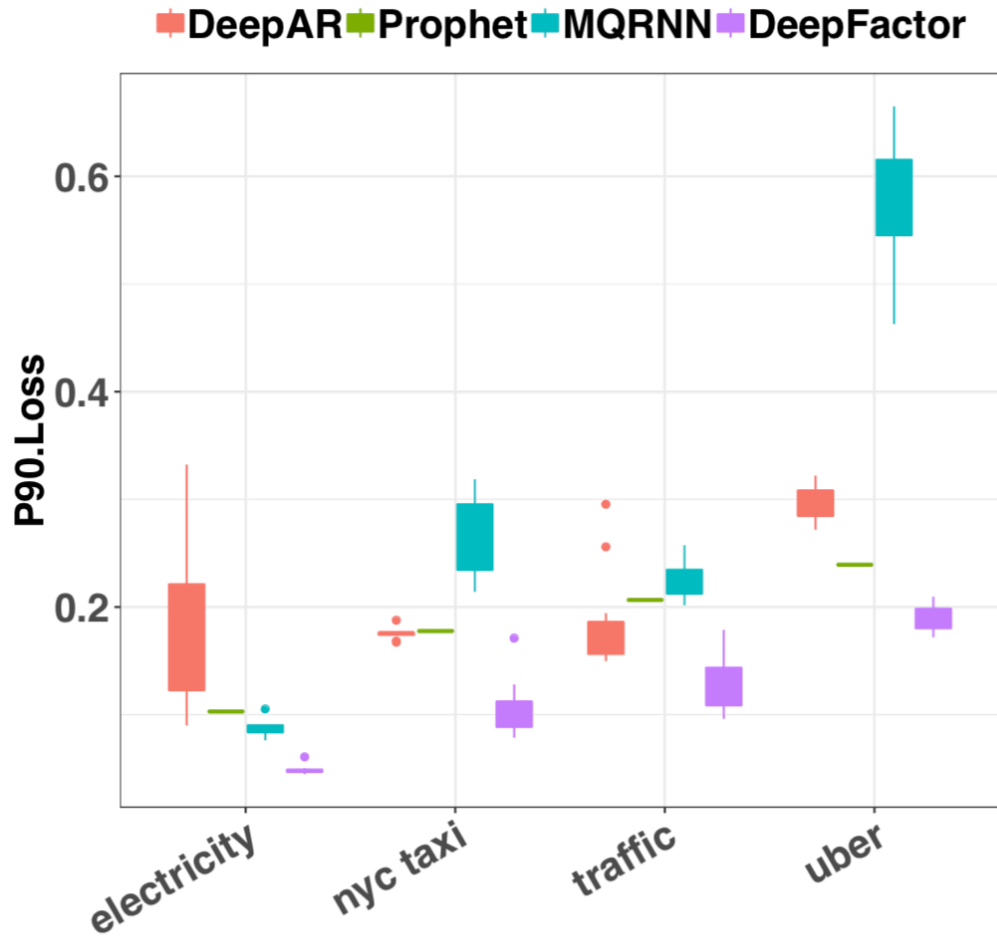
# Faster, stronger, better (than previous work)



When compared to pure RNN forecaster

- Lower uncertainty
- Faster
- More accurate

# Faster, stronger, better (than previous work)



When compared to SOTA (DeepAR and MQ-RNN)

- Less variance
- More accurate

# To be available in GluonTS: [gluon-ts.mxnet.io](http://gluon-ts.mxnet.io)

GluonTS - Probabilistic Time Series Modeling

Install API Community Contribute GitHub

## GluonTS - Probabilistic Time Series Modeling

Gluon Time Series (GluonTS) is the Gluon toolkit for probabilistic time series modeling, focusing on deep learning-based models.

GluonTS provides utilities for loading and iterating over time series datasets, state of the art models ready to be trained, and building blocks to define your own models and quickly experiment with different solutions. With GluonTS you can:

- Train and evaluate any of the built-in models on your own data, and quickly come up with a solution for your time series tasks.
- Use the provided abstractions and building blocks to create custom time series models, and rapidly benchmark them against baseline algorithms.

### Get Started: A Quick Example

Here is a simple time series example with GluonTS for predicting Twitter volume with DeepAR.

(You can click the play button below to run this example.)

```
main.py - run repl.it
1 from gluonts.dataset import common
2 from gluonts.model import deepar
3
4 import pandas as pd
5
6 url = "https://raw.githubusercontent.com/numenta/NAB/master/data/realTweets/Twitter_volume_AMZN.csv"
7 df = pd.read_csv(url, header=0, index_col=0)
8 data = common.ListDataset({"start": df.index[0],
9                             "target": df.value[:"2015-04-05 00:00:00"]},
10                            freq="5min")
11
12 estimator = deepar.DeepAREstimator(freq="5min", prediction_length=12)
```

Python 3.6.1 (default, Dec 2015, 13:05:11)  
[GCC 4.8.2] on linux

Installation

[gluon-ts.mxnet.io](http://gluon-ts.mxnet.io)

## Featuring NN-based Forecasting models

- DeepAR [Valentin et. al., 2017]
- MQ-DNN [Wen et. al., 2017]
- Deep State Space Models [Rangapuram et. al., 2018]
- Spline Regression RNN [Gasthaus et. al., 2019]
- GPs, KF, LDS
- More ...

**Come to our talk at  
the Time Series  
Workshop on Friday!**

