Python for DS: Uncertainty Quantification for Machine Learning with Application to Financial Time Series

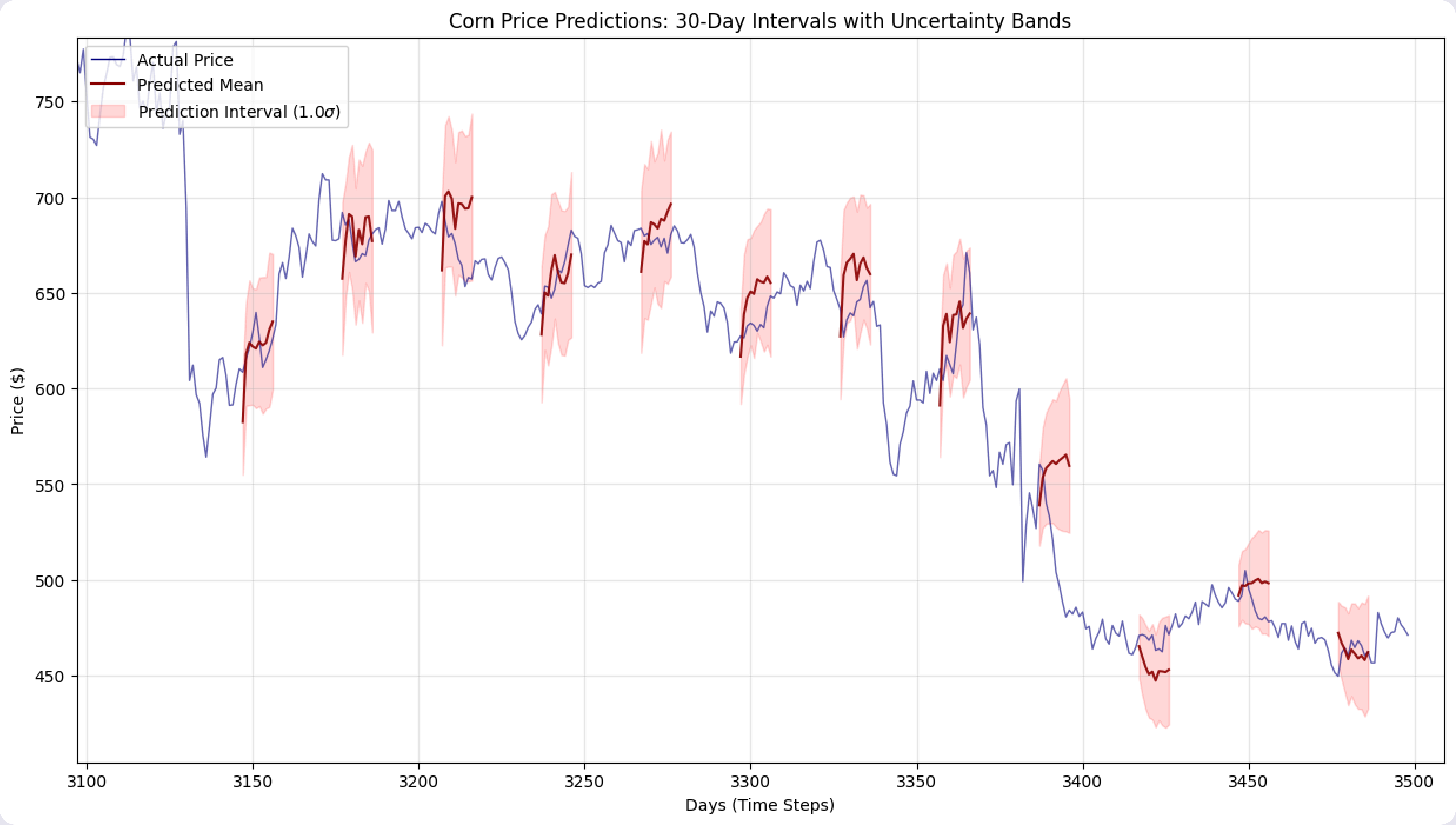
Kota OKUDA & Ryunosuke TANAKA

IP Paris

Introduction

- predictive accuracy is essential
- but if it only yields point prediction
- UQ is also important: - next price: 476 USD v.s. 460-520 USD w. proba. 80% - nice thing: never below 460 USD w. proba. 20%

And if you ever saw it…



How to measure the “uncertainty”?

- Look-back (Context Window):
  - How many past days are used as input? (cf. dimension. Consider only weekly, monthly or even seasonal effects, etc.)
- Look-ahead (multi-step forecasting):
  - how many days into the future are we predicting?
- Data Amount:
  - How much historical data is used for training?
- Model Uncertainty (dropout):
  - how many neurons (parameters) are stably contributed to the prediction?

MC Dropout for the uncertainty in models (1)

Algorithm 2: MC Dropout Inference with LSTM: complete version

Input : Input sequence $X = (x_1, \ldots, x_T)$, Pre-trained Parameters $\theta = \{W, b\}$,
Dropout rate $p \in [0, 1)$, Number of MC samples K (e.g., 100)

Output: Predictive Mean μ_Y, Predictive Uncertainty σ_Y^2

for $k \leftarrow 1$ **to** K **do**
  $z_x^{(k)} \sim \text{Bernoulli}(1 - p)$ `// Input mask`
  $z_h^{(k)} \sim \text{Bernoulli}(1 - p)$ `// Recurrent mask`
  Initialize states h_0, C_0
  for $t \leftarrow 1$ **to** T **do**
    $\tilde{x}_t \leftarrow x_t \odot z_x^{(k)}$
    $\tilde{h}_{t-1} \leftarrow h_{t-1} \odot z_h^{(k)}$
    $v_{\text{in}} \leftarrow [\tilde{h}_{t-1}, \tilde{x}_t]$
    $f_t \leftarrow \sigma(W_f \cdot v_{\text{in}} + b_f)$
    $i_t \leftarrow \sigma(W_i \cdot v_{\text{in}} + b_i)$
    $\tilde{C}_t \leftarrow \tanh(W_C \cdot v_{\text{in}} + b_C)$
    $o_t \leftarrow \sigma(W_o \cdot v_{\text{in}} + b_o)$
    $C_t \leftarrow f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
    $h_t \leftarrow o_t \odot \tanh(C_t)$
  $y^{(k)} \leftarrow \text{OutputLayer}(h_T)$

$\mu_Y \leftarrow \frac{1}{K} \sum_{k=1}^{K} y^{(k)}$
$\sigma_Y^2 \leftarrow \frac{1}{K} \sum_{k=1}^{K} (y^{(k)} - \mu_Y)^2$
return μ_Y, σ_Y^2

MC Dropout for the uncertainty in models (2)

- cf. [Gal and Ghahramani, 2016]
- disactivation or masking of the parameters w. proba. p
- mean and variance over K different masked models
- intuition: perturbation of the models by masking some parameters. If the variance is large, the masked parameters are essential for the prediction.

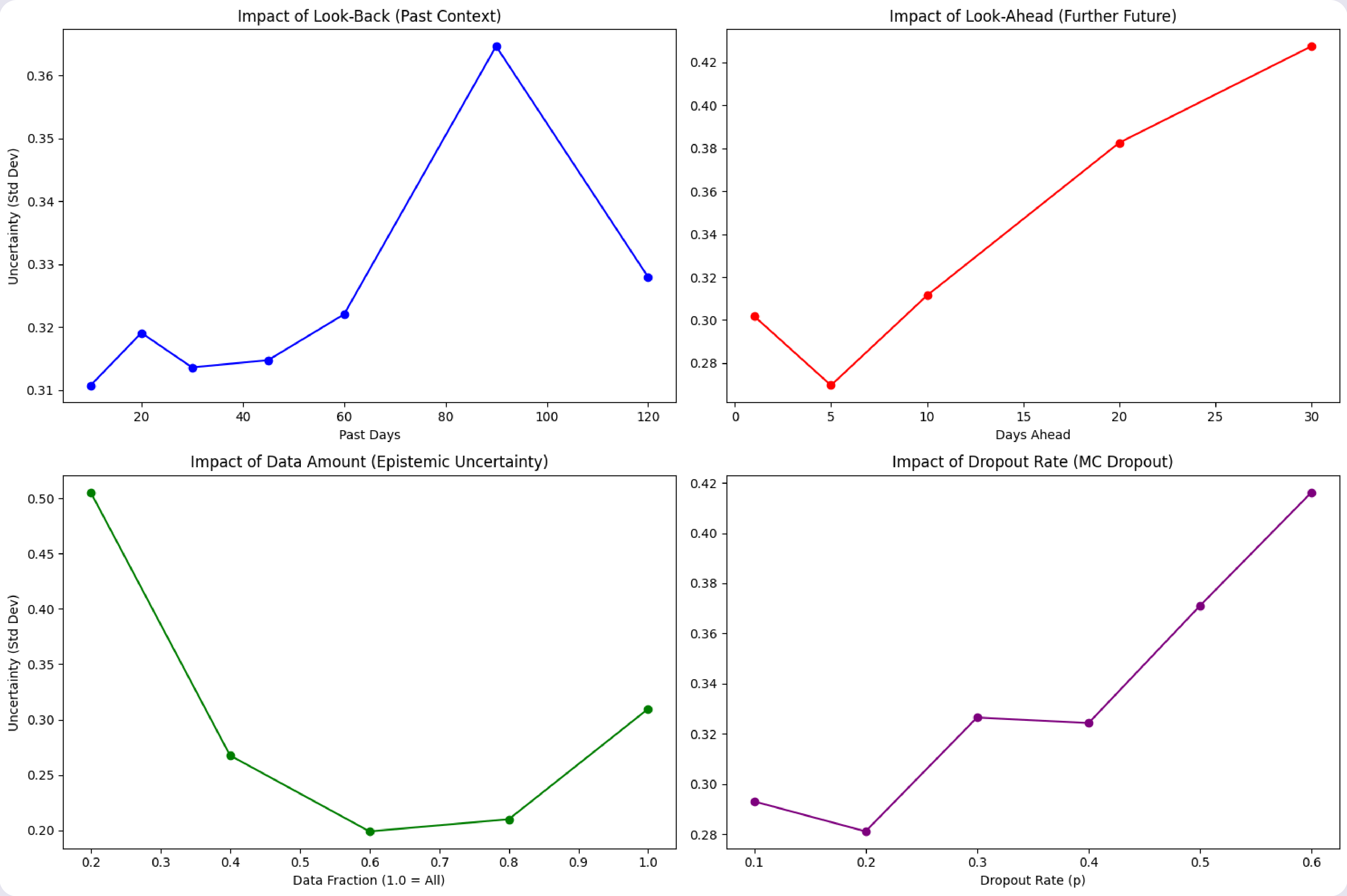
Data

- Dataset: Daily closing price series of Corn Futures via Yahoo Finance API.
- Period: 1 Jan, 2015 – 1 Jan, 2024 ($\simeq$ 9 years).
- Frequency: Daily.
- Other features: momentum, trend, high/low price of the day from the same source.

Model

- LSTM with Dropout layers ($p = 0.3$) applied after each recurrent (weight) layer.
- $K = 20$ passes of LSTM (i.e., mean of model distributions with 20 sample sizes)

Results: 4 Uncertainties (1)



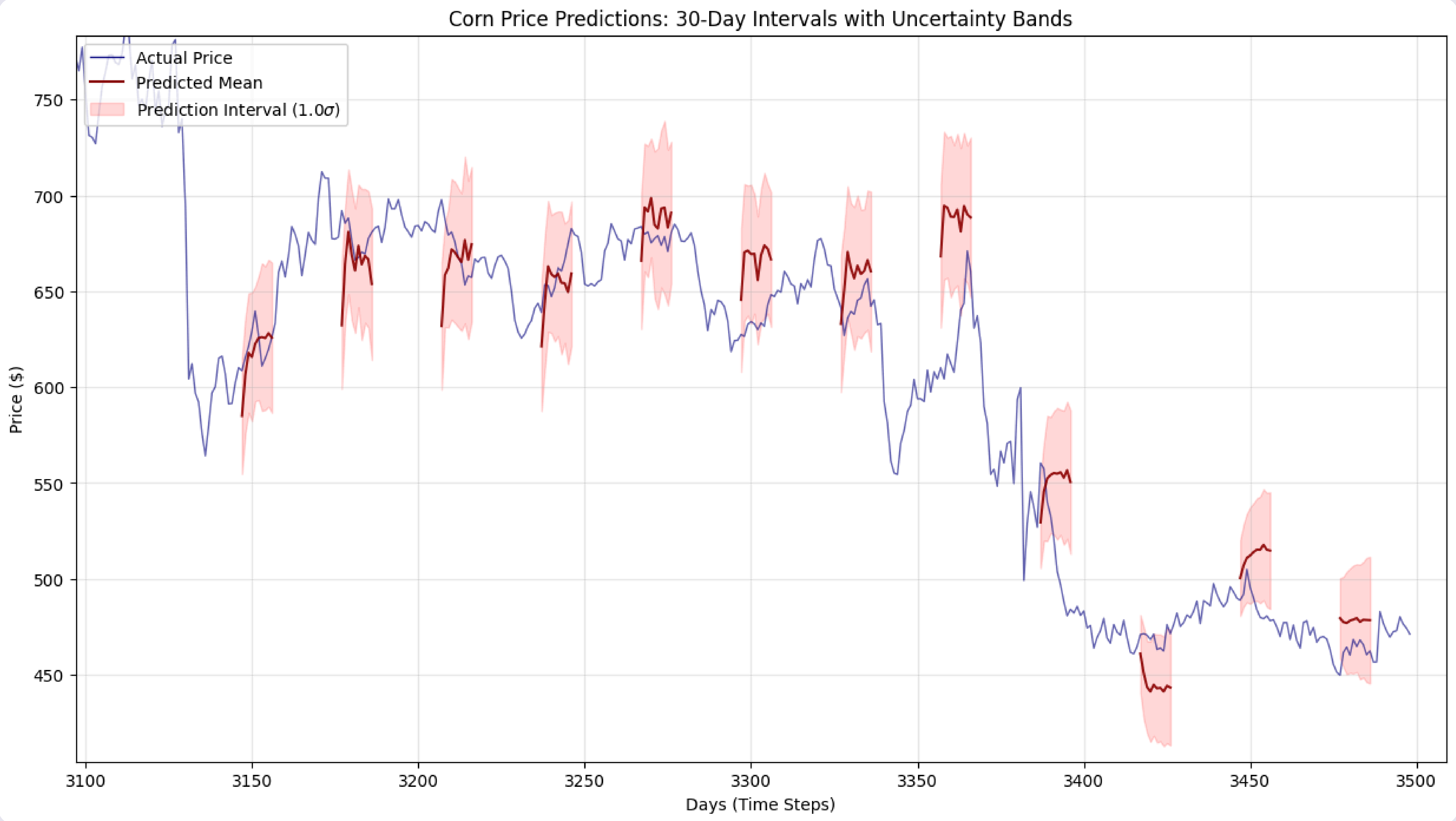
Results: 4 Uncertainties (2)

- Look-back (Context Window):
  - How many past days are used as input? $\longrightarrow$ Variation exists. Too long history may encode outdated noise?
- Look-ahead (multi-step forecasting):
  - how many days into the future are we predicting? $\longrightarrow$ The shorter, the surer: temporal horizon.
- Data Amount:
  - how many neurons (parameters) are stably contributed to the prediction? $\longrightarrow$ The more, the surer. Increasing data fraction significantly reduces an uncertainty as the model parameters converge, but…
- Model Uncertainty:
  - how many neurons are masked? $\longrightarrow$ The less masked, the surer. Increasing model capacity also reduces an uncertainty.

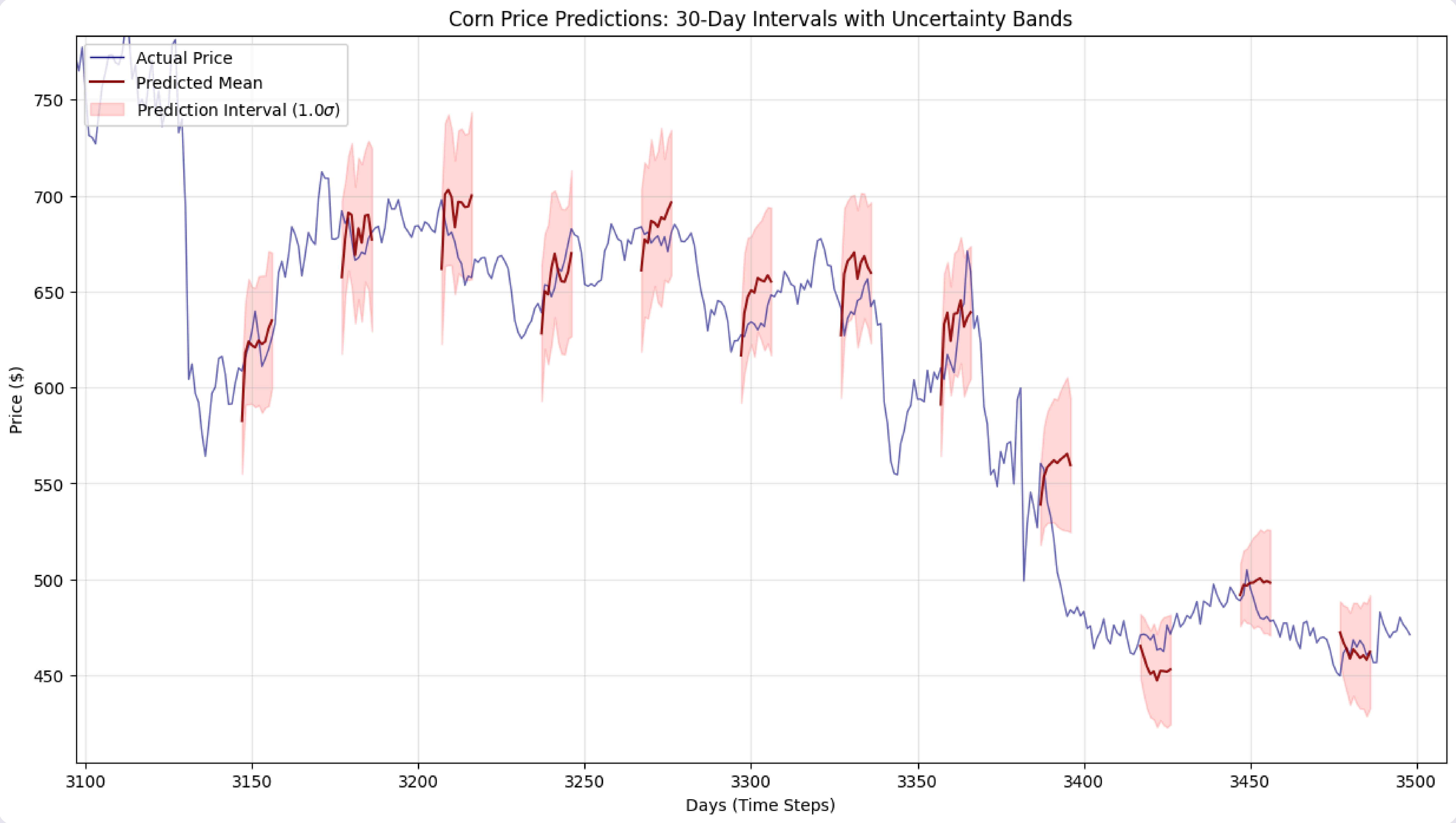
UQ over the test-time: preliminary

- We fix:
    - look-ahead: 10 days
- different look-back days: {20, 60, 120}

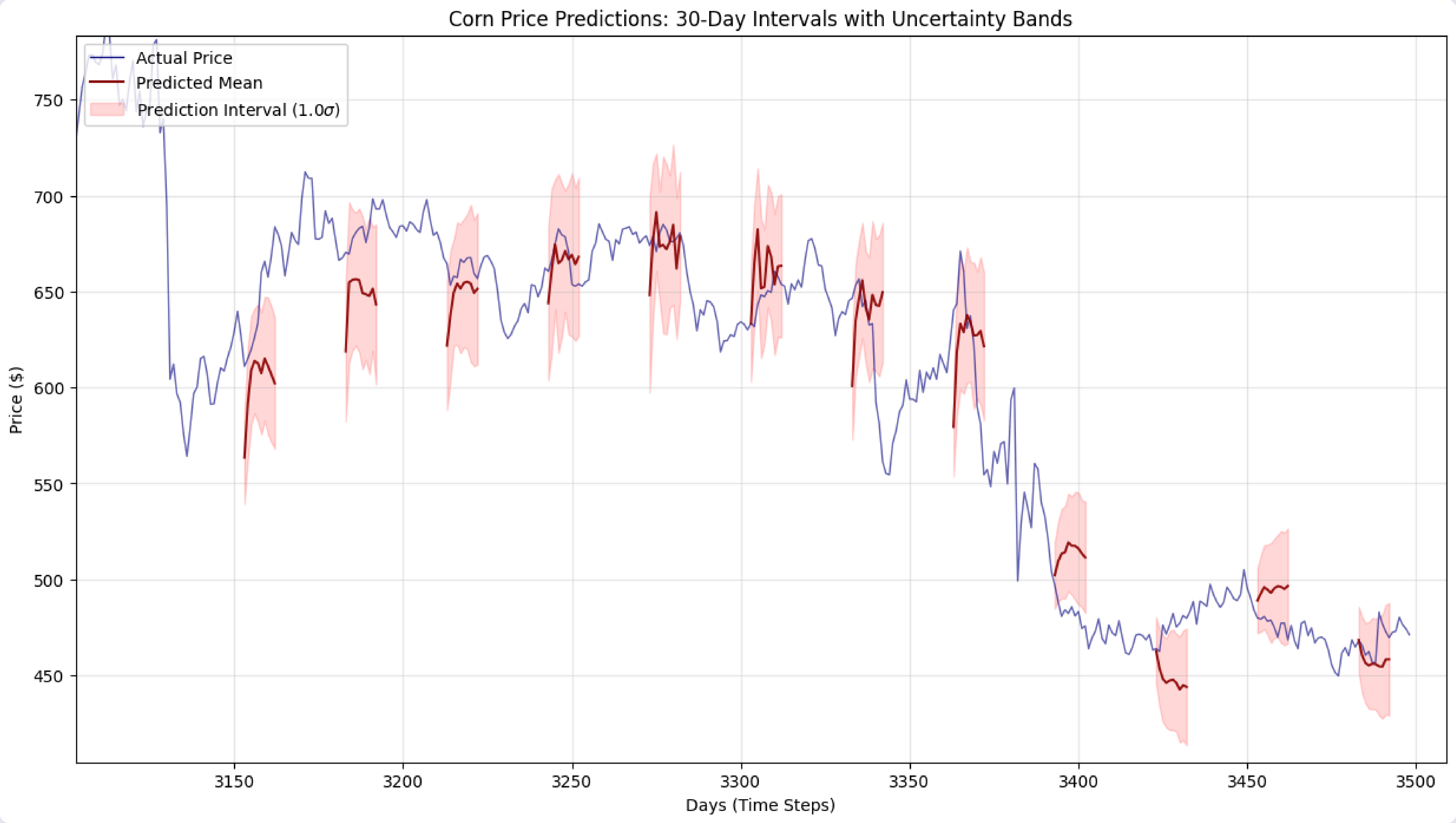
UQ over the test-time: 20 look-back days



UQ over the test-time: 60 look-back days



UQ over the test-time: 120 look-back days



UQ over the test-time

- Intervals can capture the sudden shock in some models (not for 20 days).
- The worse model has larger intervals.
- But each model might have a strength?
  - (e.g., 120 look-back days for long-term effects?)
  - Future work: model averaging according to local structure?

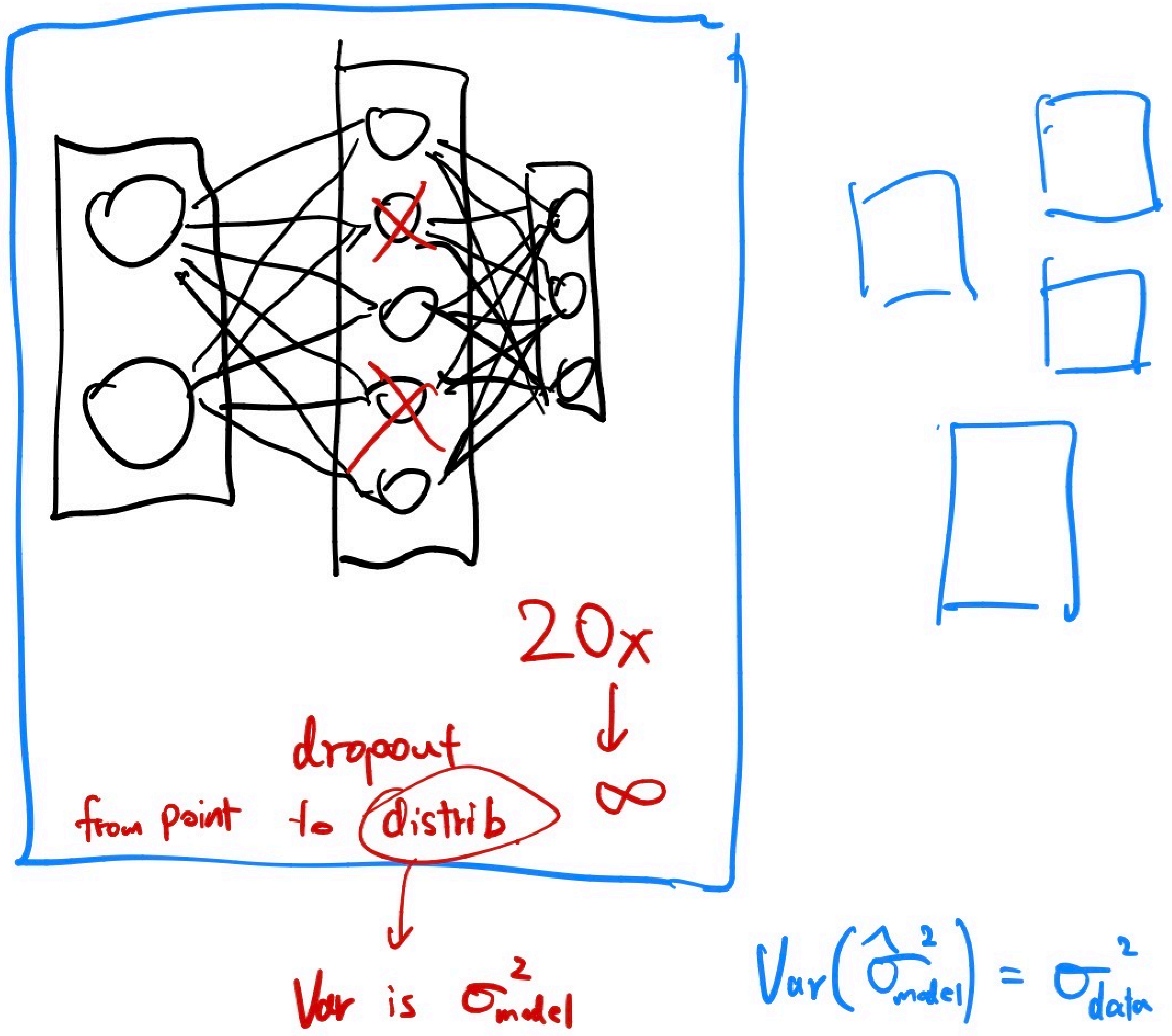
Decomposing UQ: preliminary (1)

- Decomposition Metric: The total uncertainty σ_{total} is decomposed into three components:

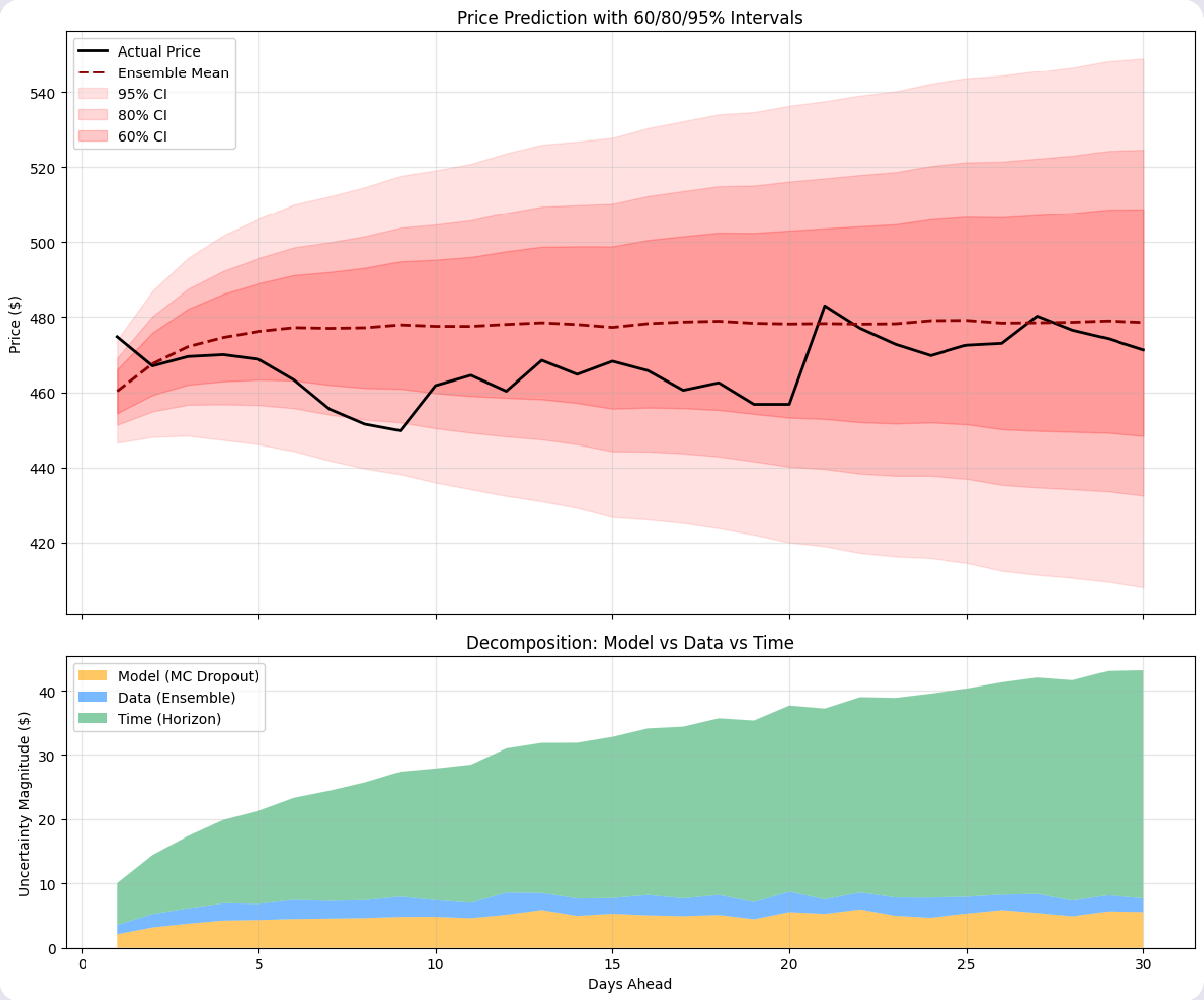
$$\sigma_{\text{total}} = \sqrt{\sigma^2_{\text{model}} + \sigma^2_{\text{data}} + \sigma^2_{\text{time}}}$$

  - σ^2_{model}: variance within MC dropout samples
    - **can be reduced** by e.g., model improvement or data acquisition
  - σ^2_{data}: Mean squared error between models.
    - **cannot be reduced** with the increasing sample size.
  - σ^2_{time}: accounts for the temporal diffusion over the look-ahead horizon.

Decomposing UQ: preliminary (2)



Decomposing UQ



- σ^2_{time} is dominant;
- σ^2_{model} is a bit large;
  - there still be some room for model improvement

Summary & Comment

- Financial time series: point prediction not useful?
- UQ with intervals for more flexible decision making

- Computational burden: 100 times of training LSTM
- Model averaging over 20/60/120 look-back days?

- (KO) good to learn UQ methods other than Bayesian methods.
- (RT) As future work, it would be interesting to study whether these effects on uncertainty differ across commodities.