Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book

Petter Kolm
NYU Courant

petter.kolm@nyu.edu
linkedin.com/in/petterkolm

Joint work with Jeremy Turiel and Nicholas Westray

Northfield's Newport Summer Research Seminar
June 14, 2023

Petter Kolm

Clinical Full Professor, NYU Courant
Director of the M.S. in Mathematics in Finance Program

Petter is the Director of the Mathematics in Finance Master's program and a Clinical Professor of Mathematics at the Courant Institute of Mathematical Sciences, New York University. In this role he interacts with major financial institutions such as investment banks, financial service providers, insurance companies and hedge funds. Petter worked in the Quantitative Strategies group at Goldman Sachs Asset Management developing proprietary investment strategies, portfolio and risk analytics in equities, fixed income and commodities.

Petter was awarded "Quant of the Year" in 2021 by Portfolio Management Research (PMR) and Journal of Portfolio Management (JPM) for his contributions to the field of quantitative portfolio theory. Petter is a frequent speaker, panelist and moderator at academic and industry conferences and events. He is a member of the editorial boards of the International Journal of Portfolio Analysis and Management (IJPAM), Journal of Financial Data Science (JFDS), Journal of Investment Strategies (JoIS), and Journal of Portfolio Management (JPM). Petter is an Advisory Board Member of Alternative Data Group (ADG), AISignals and Operations in Trading (Aisot), Betterment (one of the largest robo-advisors) and Volatility and Risk Institute at NYU Stern. He is also on the Board of Directors of the International Association for Quantitative Finance (IAQF) and Scientific Advisory Board Member of the Artificial Intelligence Finance Institute (AIFI).

Petter is the co-author of several well-known finance books including, Financial Modeling of the Equity Market: From CAPM to Cointegration (Wiley, 2006); Trends in Quantitative Finance (CFA Research Institute, 2006); Robust Portfolio Management and Optimization (Wiley, 2007); and Quantitative Equity Investing: Techniques and Strategies (Wiley, 2010). Financial Modeling of the Equity Markets was among the "Top 10 Technical Books" selected by Financial Engineering News in 2006.

As a consultant and expert witness, Petter has provided his services in areas including alternative data, data science, econometrics, forecasting models, high frequency trading, machine learning, portfolio optimization with transaction costs, quantitative and systematic trading, risk management, robo-advisory, smart beta strategies, trading strategies, transaction costs, and tax-aware investing.

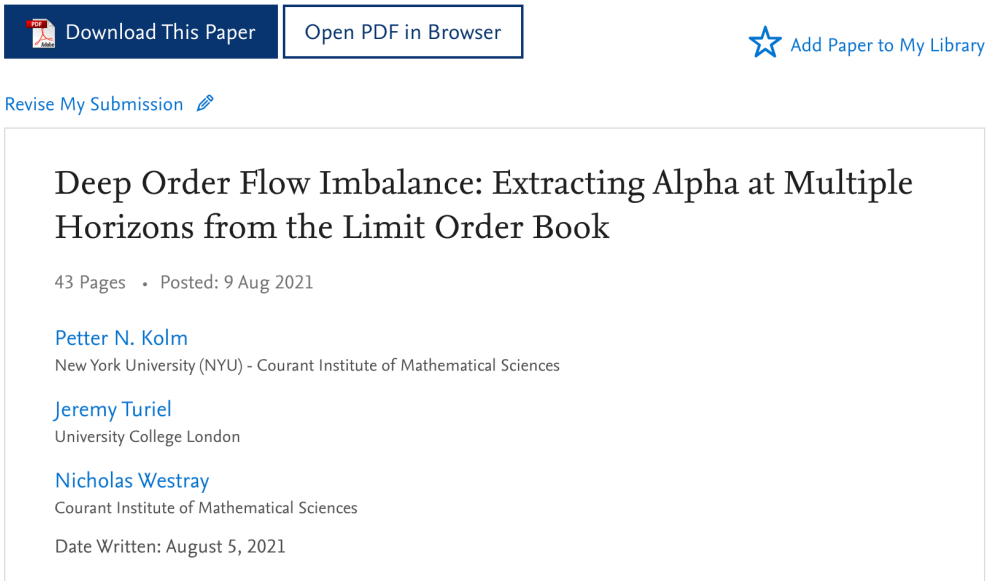
He holds a Ph.D. in Mathematics from Yale University; an M.Phil. in Applied Mathematics from the Royal Institute of Technology, Stockholm, Sweden; and an M.S. in Mathematics from ETH Zurich, Switzerland.

The M.S. in Mathematics in Finance at NYU Courant

- Visit our website
- Follow us on LinkedIn
- Hire our students

Introduction & Motivation

Our Articles Related to This Talk



- Kolm, Turiel, and Westray (2021), "Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book"

  Available on SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3900141

- Kolm and Westray (2023), "A Bayesian Approach to Analyzing Information Content of Cross-Sectional and Multilevel Order Flow Imbalance" *(In preparation)*

Systematic Trading in U.S. Equities at High Frequency

- Generating alpha signals (return predictors) is critical in this endeavour for both buy and sell sides
  - Sell Side: Improving performance in next generation trading algorithms
  - Buy Side: Developing and deploying profitable HF strategies
- The holy grail for many of these firms is to take as input a raw limit order book (or a collection of order books) and produce high frequency price/return forecasts

Signal Generation

- This type of alpha generation is *very challenging* for a number of reasons
  - Enormous amounts of data (TB & PB sizes)
  - Specialist infrastructure required to store, process and analyze
  - Data is noisy, non-stationary and fat tailed
  - Field is extremely competitive, every single one of your competitors is trying to do the same thing with the same data
- Current state of the art is to employ quants to extract and handcraft features using expert domain knowledge which then become high value IP

The Bitter Lesson

> *The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation.*
>
> *. . . Researchers seek to leverage their human knowledge of the domain [to improve performance], but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to.*

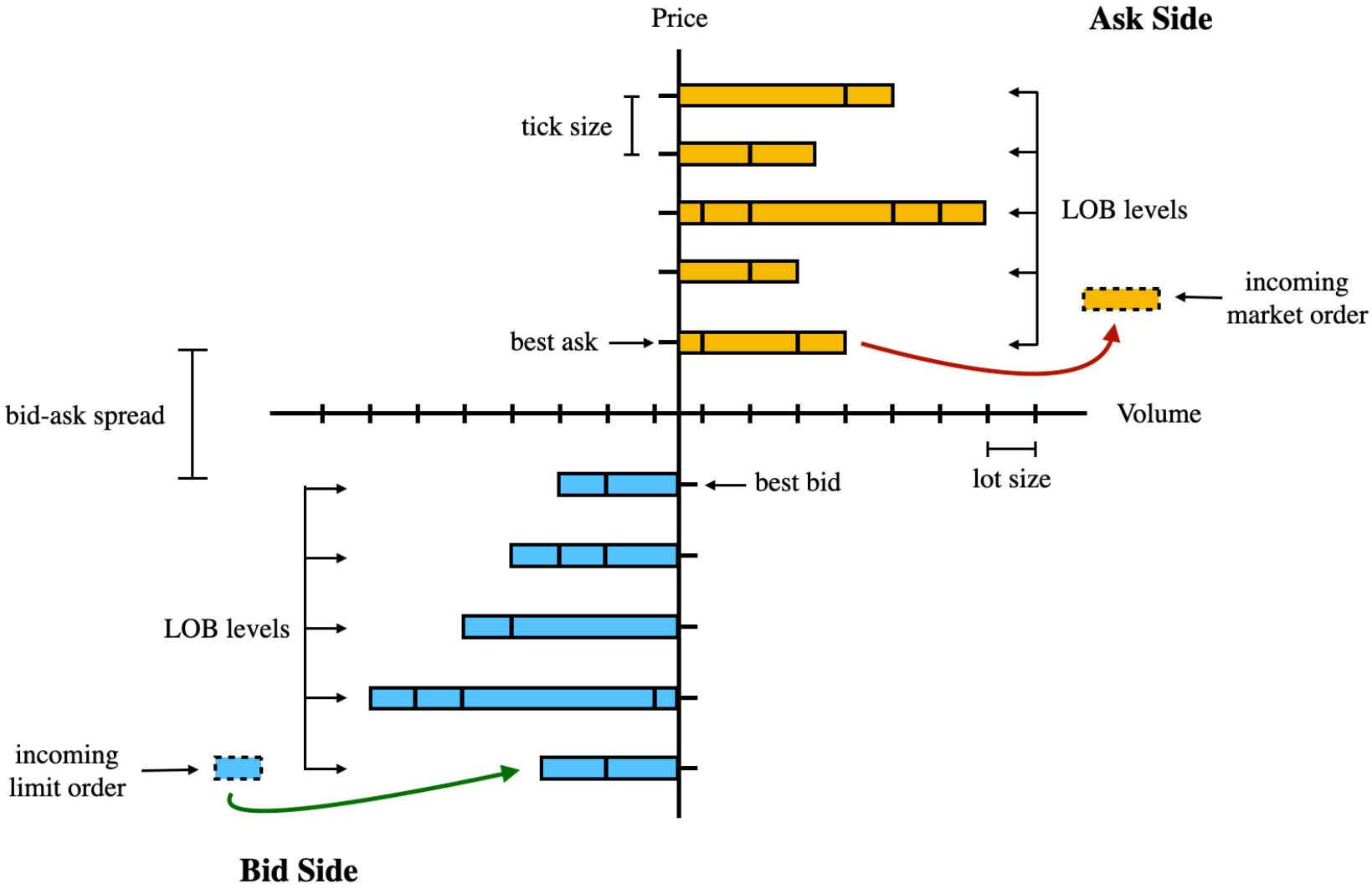
Rich Sutton - The Bitter Lesson, March 2019[1]

Examples of the Bitter Lesson

- Chess: Methods that defeated Kasparov were based on massive deep search, not the special structure of Chess
- Go: 20 years later, leveraging learning from self play (to discover value functions) provided the edge as this allowed massive computation to be brought to bear
- Computer Vision: Early methods searched for edges, cylinders etc. This has all been discarded in favour of modern Convolutional Neural Networks (CNNs)
- Natural Language Applications: Methods based on linguistics, knowledge of words/phonemes are also no longer used. Modern approaches are all deep learning/statistically based (BERT, FinBERT)

Return Forecasting

Review: Limit Order Book (LOB)



General Problem Statement

- Focus on the top 10 (non-zero) levels of the LOB and define the vector

$$x_t := (a_t^1, v_t^{1,a}, b_t^1, v_t^{1,b}, \ldots, a_t^{10}, v_t^{10,a}, b_t^{10}, v_t^{10,b})^\top \in \mathbb{R}^{40} \quad (1)$$

- For each stock and time t, fix a horizon h and consider the time series regression problem

$$r_{t,t+h} = g(x_t, x_{t-1}, \ldots, x_{t-W}) + \varepsilon_t \quad (2)$$

- $r_{t,t+h}$ are the forward returns (hereafter r_t)
- The function g is a neural network
- W denotes the length of the lookback window (typically 100)

Overview: Neural Networks

- The simplest Neural Network is best thought of as being made of sequential applications (layers) of a linear transform and an element wise non-linearity

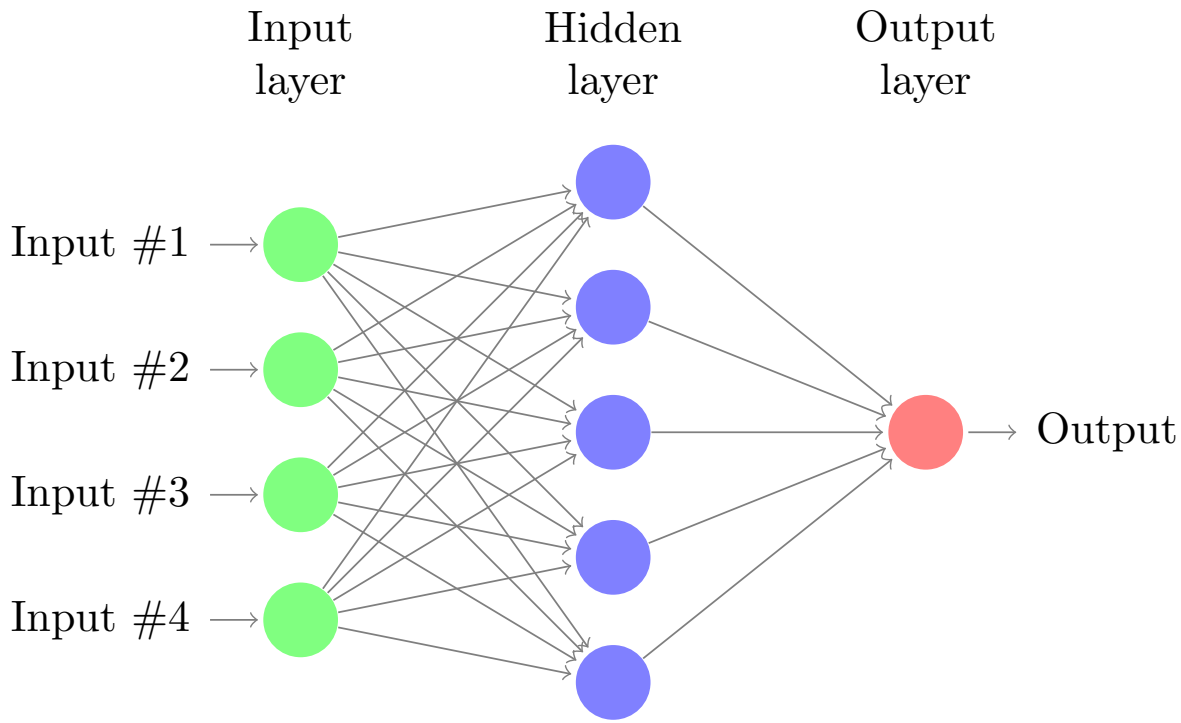


Figure 1: Simple Multilayer Perceptron.

Overview: Multilayer Perceptron (MLP)

- More generally, when we have $L \geq 1$ hidden layers, each with $N_l \in \mathbb{N}$ neurons. If $x \equiv h_0 \in \mathbb{R}^{N_0}$, is the input then an MLP with L hidden layers takes the form

$$h^{(l)} = f\left(W^{(l)} h^{(l-1)} + b^{(l)}\right), \ l = 1, 2, \ldots, L,$$
$$y = W^{(L+1)} h^{(L)} + b^{(L+1)}$$

- The function f is called the activation (commonly chosen as $f(x) = \max\{x, 0\}$, aka ReLU)
- Can be easily trained using backpropagation (chain rule), and there are software packages (Tensorflow, PyTorch) which do this for you
- They are a universal approximator (Hornik, 1991)

Overview: Recurrent Neural Networks (RNNs)

- MLP is great for function approximation but has no notion of order. Recurrent neural networks were designed for sequence tasks (NLP, time series forecasting etc.)
- This time there is an input at each time x_t together with a hidden state h_t

$$h_t = f_h(x_t, h_{t-1})$$
$$y_t = f_o(h_t)$$

- The non-linear functions f_h, f_o handle the recurrent transformation and the output transformation
- Trained via *backpropagation through time* (BPTT) where the network is unrolled. Issues with vanishing/exploding gradients
- Important architecture avoiding this is the Long Short-Term Memory Network (LSTM) (Hochreiter and Schmidhuber, 1997)

Overview: Convolutional Neural Networks (CNNs)

- Originally introduced in computer vision, specifically image classification
- The layers apply (multiple) kernels to locally average parts of the image, then apply a non-linearity

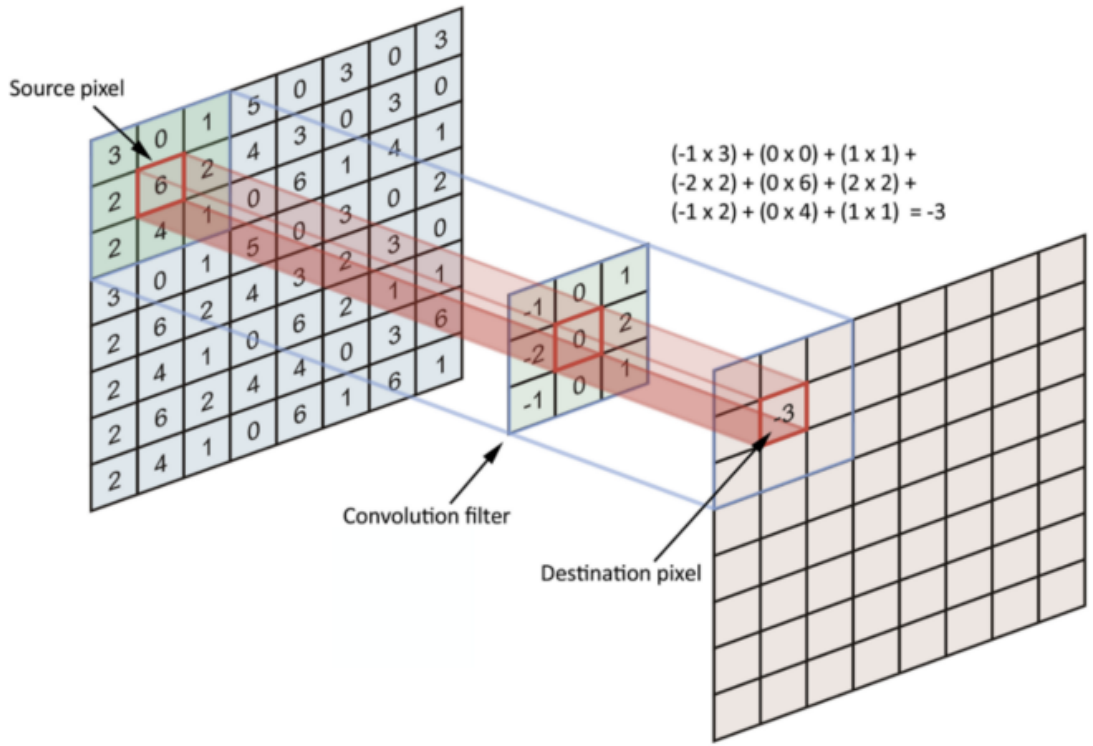


Figure 2: Example Convolution.

Financial CNNs

- We can think of an order book as a matrix, one state per row
- Temporal convolutions are interpreted as non-linear moving averages
- Spatial convolutions aggregate information across the different levels (and sides) of the LOB
- The most well known model (Zhang, Zohren, and Roberts, 2019) sequentially applies spatial and temporal convolutions processing the order book into a series of features
- Convolutional layers are a preprocessing step, the resulting time series is used as the input to an LSTM
- The CNN-LSTM can be trained via BPTT

Related Work

- Due to success of NNs in classification, existing literature reformulates the regression problem as one of classification
- There are three main groups of authors who have focussed on this problem, including
  - Tsantekidis, Passalis, Tefas, Kanniainen, Gabbouj, and Iosifidis (2020) - Focus on 5 Finnish stocks (FI-2010), investigate different architectures
  - Sirignano and Cont (2019) - Focus on S&P 500 with a single architecture, investigate questions of universality
  - Zhang, Zohren, and Roberts (2019) - Focus on 5 LSE stocks, investigate sophisticated CNN-LSTM inception networks
- There are still a lot of open practical questions to investigate

Some Important Questions for the Practitioner

- Should we transform the raw LOB before inputting into the NN?
- What architecture(s) are best?
- Is there a linkage between model predictive performance and stock characteristics / microstructural properties?
- What kind of horizon do these alphas have?

What About Stationarity?

- Recall the original regression formulation

$$r_t = g(x_t, x_{t-1}, \ldots, x_{t-W}) + \varepsilon_t \tag{3}$$

- Statistical properties of model inputs and outputs
  - Prices are not-stationary
  - Volumes might be stationary - evidence mixed
  - Returns generally considered stationary
- Therefore, we have stationary LHS and non-stationary RHS. This seems odd
- Econometric theory would suggest to difference the input when we are interested in performing inference

How to Difference a LOB?

- Given two snapshots of a limit order book, define the following transform

$$\mathrm{bOF}_{t,i} := \begin{cases} v_t^{i,b}, & \text{if } b_t^i > b_{t-1}^i, \\ v_t^{i,b} - v_{t-1}^{i,b}, & \text{if } b_t^i = b_{t-1}^i, \\ -v_{t-1}^{i,b}, & \text{if } b_t^i < b_{t-1}^i \end{cases}$$

$$\mathrm{aOF}_{t,i} := \begin{cases} -v_{t-1}^{i,a}, & \text{if } a_t^i > a_{t-1}^i, \\ v_t^{i,a} - v_{t-1}^{i,a}, & \text{if } a_t^i = a_{t-1}^i, \\ v_t^{i,a}, & \text{if } a_t^i < a_{t-1}^i \end{cases}$$

- If the bid price doesn't move and bid level moves down, that's a negative imbalance (signal). Bid price moves up, that's a positive imbalance. Bid price moves down that's a negative imbalance
- Classical transform taken from the market microstructure literature Cont, Kukanov, and Stoikov (2014)

Order Flows & Order Flow Imbalances

- We form *order flow* (OF) by concatenation and *order flow imbalance* (OFI) by subtraction

$$\mathrm{OF}_t := \begin{pmatrix} \mathrm{bOF}_t \\ \mathrm{aOF}_t \end{pmatrix} \in \mathbb{R}^{20},$$

$$\mathrm{OFI}_t := \mathrm{bOF}_t - \mathrm{aOF}_t \in \mathbb{R}^{10}$$

- Both OF and OFI are stationary inputs. In addition, OFI is well known to be an excellent predictor of contemporaneous returns (all time scales)
- These are ideal candidates to explore as alternative RHS in the regression

Our Setup & Contribution

- Recast the problem as standard regression and include multiple horizons in the output

$$r_t := (r_{t,1}, \ldots, r_{t,H})^\top \in \mathbb{R}^H, \quad \text{where } H \geq 1 \tag{4}$$

- We refer to the forecasts as an *alpha term structure* at time t. Our forecasting models take the form

$$r_t = g(x_t, x_{t-1}, \ldots, x_{t-W}) + \varepsilon_t \tag{5}$$

- We are going to
  - Understand the effects of different RHS in the regression
  - Compare multiple architectures across a large set of symbols
  - Explore the relationship between stock characteristics and model predictive power

Data & Model Fitting

Data Description

- We use data for the time period January 1, 2019 through January 31, 2020 (LOBSTER, WRDS)
- Dataset is *very large*, approximately 10TB uncompressed. We need specialist infrastructure to process and store
- Across our universe we have stocks with a few updates per day (EBAY) and stocks with many updates per day (MSFT). Information flows at different rates for different stocks and so horizon cannot be constant in time across stocks

Defining Return Horizons

- We use a stock specific increment defined via

$$\Delta t := \frac{2.34 \cdot 10^7}{N}\,, \tag{6}$$

  where the numerator is the number of milliseconds in a trading day and the denominator N denotes the average number of non-zero tick by tick mid-price returns
- We then define horizons in terms of Δt or *"number of average price changes."* We focus on a short horizons, $\frac{1}{5}k\Delta t\,, k = 1, \ldots, 10$ i.e. between 0 and 2 average price changes
- Insert a fixed latency buffer of 10ms for all intervals to mimic production setting

Model Universe

- ARX - Autoregressive with exogenous features (linear model)

$$r_t = w_0 + \sum_{i=1}^{100} v_i^\top x_{t-i} + \varepsilon_t$$

- MLP - Multilayer Perceptron (4 layers) (Briola, Turiel, and Aste, 2020)
- LSTM - Long Short-Term Memory Network (128 hidden units)
- LSTM-MLP - LSTM (128 hidden units) mapping to an MLP (64 hidden units)
- LSTM (3) - Deep LSTM (150 hidden units) (Sirignano and Cont, 2019)
- CNN-LSTM - State of the art model proposed by Zhang, Zohren, and Roberts (2019)

Model Hyperparameters

Model	Input	Number of layers	Number of parameters$^{(*)}$	Learning rate	Batch size	Training epochs	Early stopping
ARX	OF	1	2.0×10^{3}	10^{-4}	256	50	Yes
CNN-LSTM	OF	27	1.3×10^{5}	10^{-3}	256	50	Yes
LSTM	OF	2	1.0×10^{5}	10^{-5}	256	50	Yes
LSTM (3)	OF	4	4.6×10^{5}	10^{-5}	256	50	Yes
LSTM-MLP	OF	3	8.4×10^{4}	10^{-5}	256	50	Yes
MLP	OF	4	1.3×10^{6}	10^{-5}	256	50	Yes
ARX	LOB	1	4.0×10^{3}	10^{-4}	256	50	Yes
CNN-LSTM	LOB	27	1.4×10^{5}	10^{-3}	256	50	Yes
LSTM	LOB	2	1.1×10^{5}	10^{-5}	256	50	Yes
LSTM (3)	LOB	4	4.7×10^{5}	10^{-5}	256	50	Yes
LSTM-MLP	LOB	3	9.4×10^{4}	10^{-5}	256	50	Yes
MLP	LOB	4	2.3×10^{6}	10^{-5}	256	50	Yes

Table 1: Summary of the inputs and hyperparameters used. $^{(*)}$The number of parameters are approximated to the nearest order of magnitude and truncated for readability.

Training

- To mimic a real life production setting we perform (for each symbol) rolling fits over our time period
- We choose a (1,4,1) configuration
  - The first week is for validation (early stopping)
  - The middle 4 weeks are for training
  - The final week is for out of sample testing
- We then step this forward by 3 weeks (to keep numbers of fits manageable)
- Overall we have 12 x 115 x 18 (models x stocks x time periods) NNs to fit
- We apply winsorization and Z-scoring to all independent and dependent variables used in the regressions

Network Fitting

- Training deep learning models at this scale would likely be impossible without the use of GPUs which offer $\sim 10\times$ speedup
- Our computations are performed on the NYU Greene[2] and Hudson[3] high performance computing environments. Greene has 32K CPU cores, 332 NVIDIA GPUs and 145TB of RAM distributed over 568 nodes, Hudson has 960 CPU cores, 160 AMD GPUs and 10TB of RAM distributed across 20 nodes
- Code is written in Python and NNs are trained using Tensorflow 2.3.1 and Keras (ADAM optimizer)
- For LSTM/CNN-LSTM-based models, we use a single GPU, either NVIDIA Quadro RTX8000 (48GB), an NVIDIA V100 (32GB) or an AMD MI-50 (32GB).
- Time to train a single model varies in the range of 10-60 minutes, depending on model, stock and GPU

Results

Order Flow Imbalance or Limit Order Book?

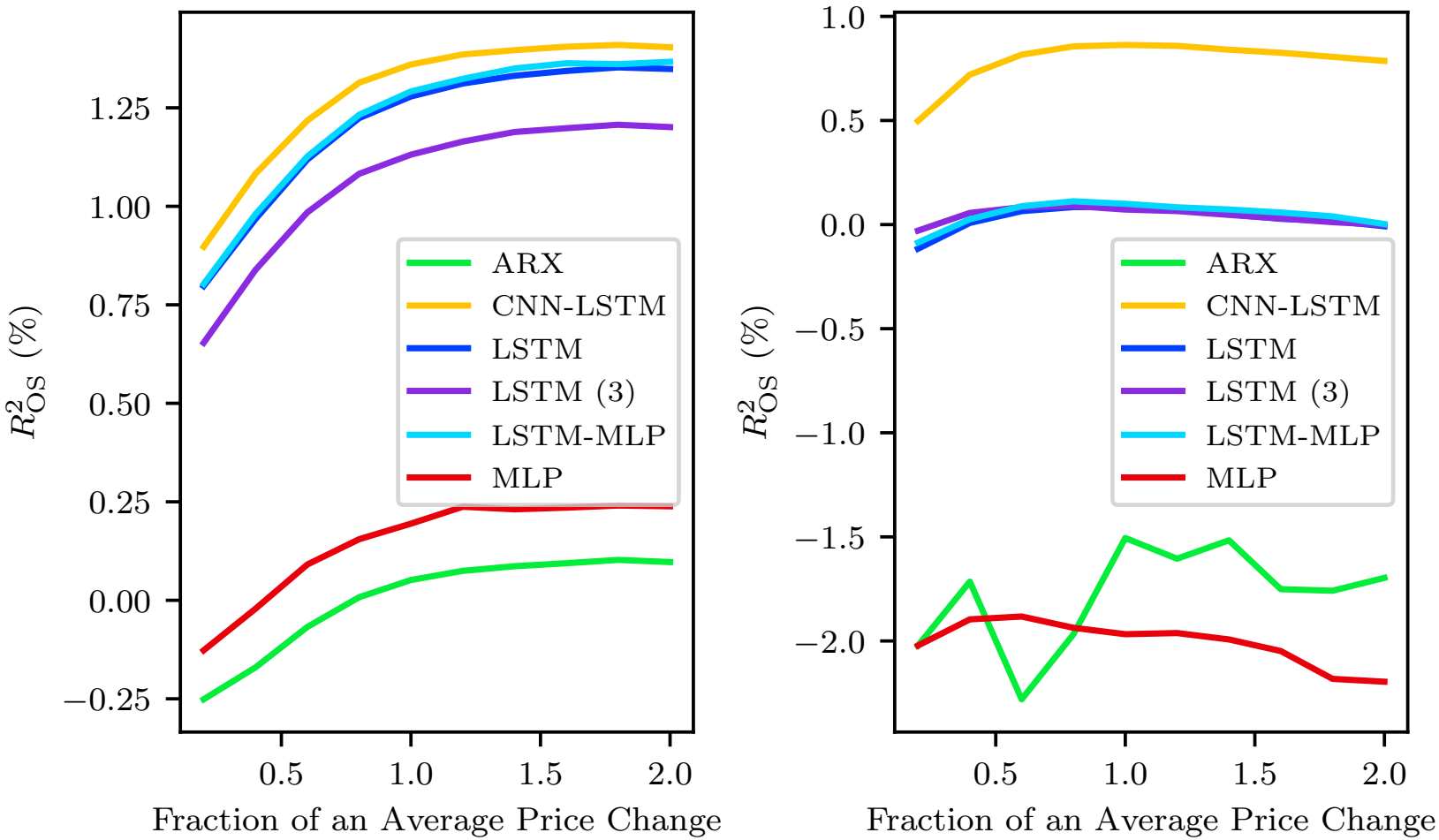


Figure 3: Left Panel OF. Right Panel LOB.

Discussion

- Recall the key questions we set out to address
  - Understand the effects of different RHS in the regression
  - Compare multiple architectures across a large set of symbols
- OF input is clearly better than LOB - stationarity of inputs is important
- LSTM based models outperform non-LSTM models
- Depth/CNN layers do not seem to outperform plain LSTM after converting to OF
- Significant alpha at all horizons for the OF models. Small R^2 but high profitability due to the short horizons

Stock Characteristics

- We have seen that a regular LSTM model with OF input is an excellent (non-complex) model
- Use the following stock characteristics to study model performance
  - Tick Size - Fraction of time the spread is equal to \$0.01 (large tick stocks approximately 1)
  - Log Updates - Log Number of updates/day
  - Log Trades - Log Number of trades/day
  - Log Price Chg - Log Number of price changes/day
- All numbers computed per stock by averaging across the time period

Characteristics Correlation

Figure 4: Dependence Structure of Cross Sectional Characteristics.

Methodology

- We fix the model to be (LSTM, OF), average across horizons and out of sample data points
- We are left with a single R^2_{OS} per stock (115 points)
- These are plotted against the stock characteristics in a scatter
- Results are almost identical for (CNN-LSTM, OF)

Cross Sectional Performance

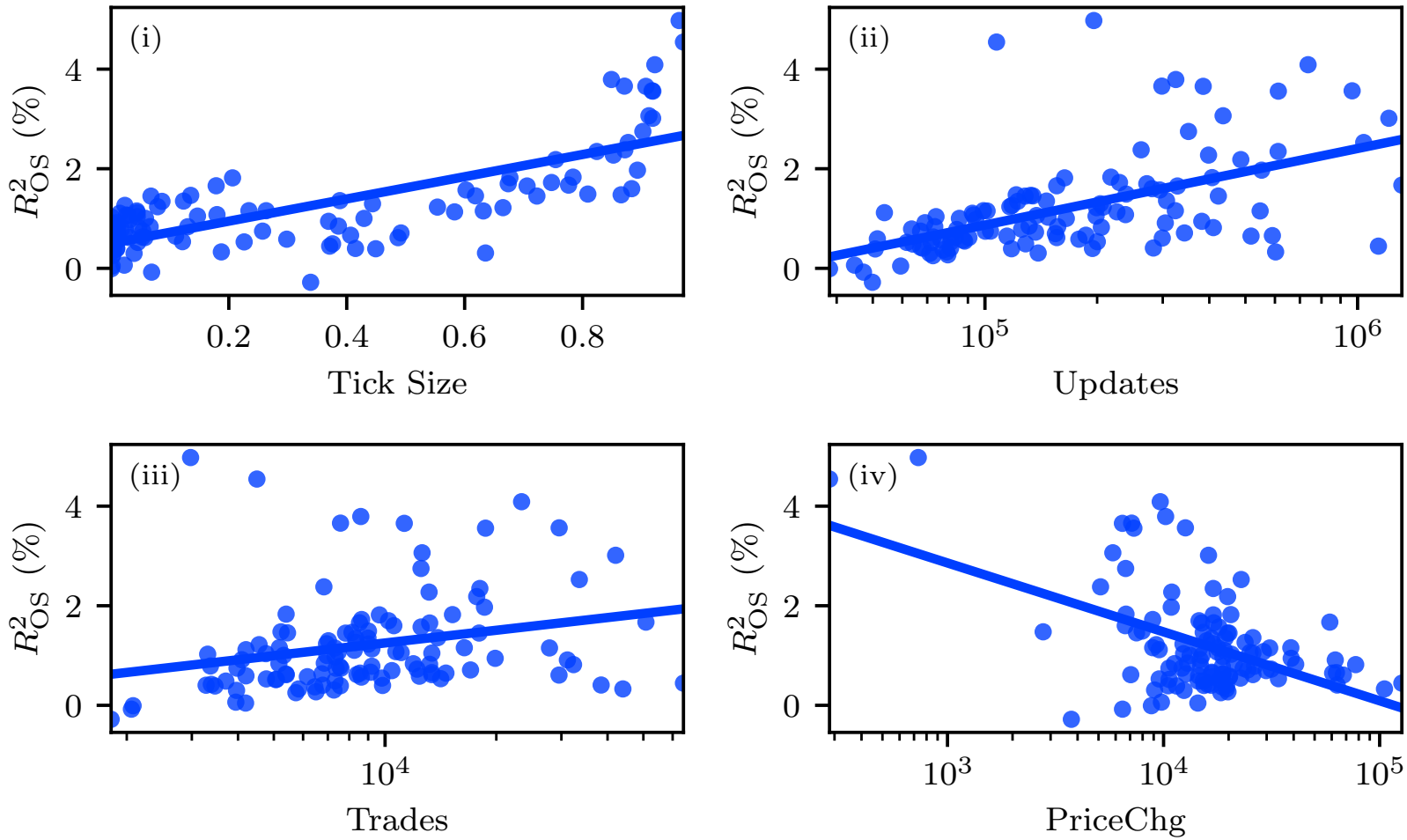


Figure 5: Cross Sectional Performance - (LSTM, OF) Model.

Explaining Performance

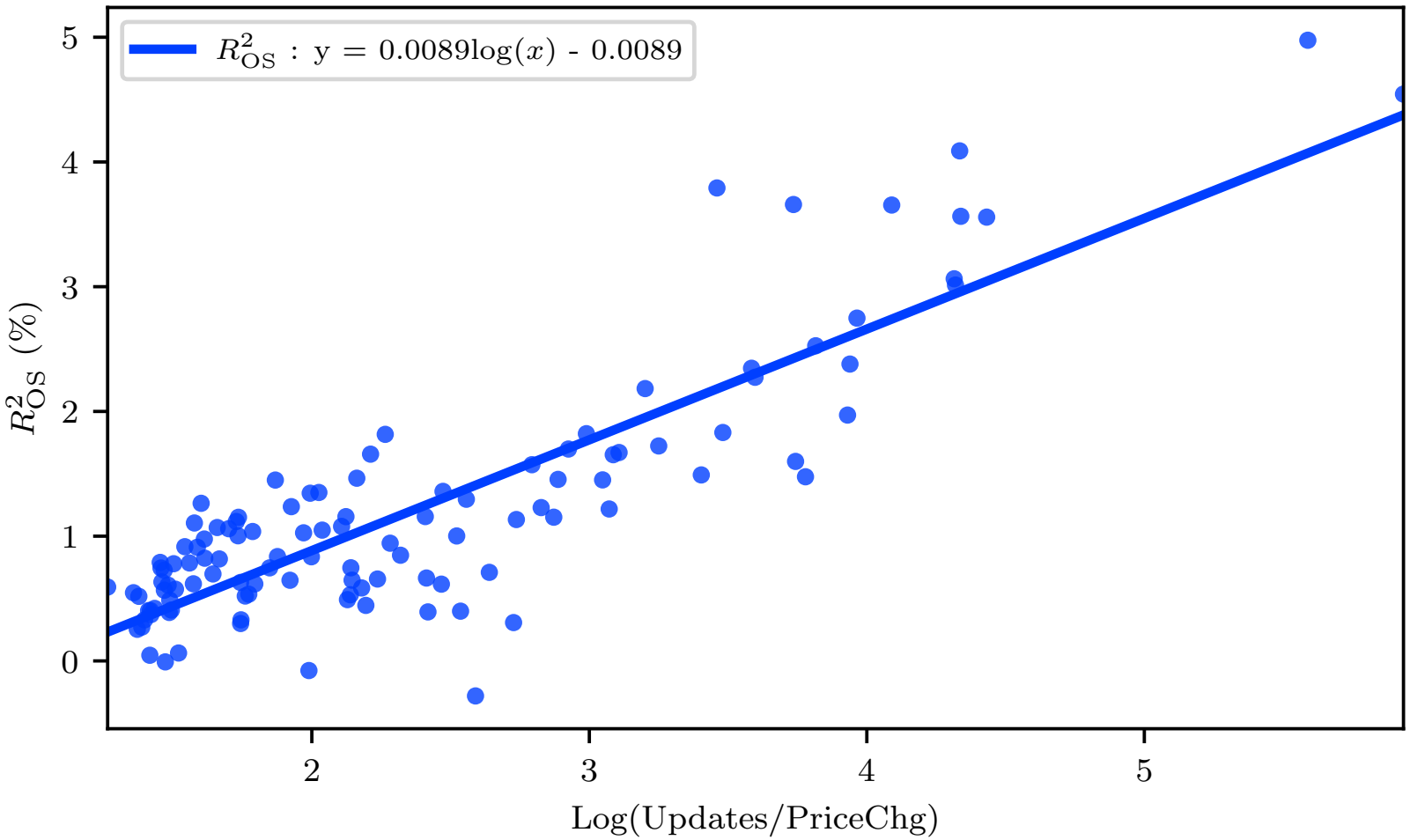


Figure 6: Cross Sectional Performance - Log(Updates/PriceChg).

Discussion

- There are clear dependencies on updates, tick size and trades
- Regression analysis (see article) shows that the best characteristic is in fact a combination, Log(Updates/PriceChg). This explains performance (R^2_{OS}) with an adj.R^2 of 75%
- Why?
  - When Log(Updates/PriceChg) is large, you have a stable order book with lots of updates per price change. This is also a property of large tick stocks (hence the correlation)
  - You have lots of data and complicated patterns forming between time series of imbalances and price changes
  - The LSTM model is able to capture and model this
- We have good out-of-sample performance across the vast proportion of our universe

Additional Results & Robustness Checks

Many additional questions addressed in the article

- How far ahead can we predict returns? (about 2-3 price changes)
- How sensitive are the results to the fixed window length $W = 100$? (interestingly not very)
- What if we use an OFI or volume-only LOB RHS? (removing prices helps but OF is the best)
- Motivation for results in terms of inductive biases - an interesting new concept from the ML literature

Conclusion & Extensions

Conclusion

- We built a framework to evaluate different inputs and deep learning models for return predictions
- We demonstrated that stationarity of the inputs is critical to getting good outcomes
- Our results suggests that universality needs a second consideration as model results have strong microstructural dependencies
- Simple architectures perform as well as complicated ones

Extensions

- Better understand relationship between classification and regression
- How about Bayesian deep nets or other "fancy" models?
- Can we use this setup to perform volume prediction?
- Can we extend what we have to other venues/asset classes (futures)?

Contact

Petter Kolm
NYU Courant

petter.kolm@nyu.edu

https://www.linkedin.com/in/petterkolm

References I

- Briola, Antonio, Jeremy Turiel, and Tomaso Aste (2020). *Deep Learning Modeling Of Limit Order Book: A Comparative Perspective*. arXiv: 2007.07319 [q-fin.TR].
- Cont, Rama, Arseniy Kukanov, and Sasha Stoikov (2014). "The Price Impact Of Order Book Events". In: *Journal Of Financial Econometrics* 12.1, pp. 47–88.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.
- Hornik, Kurt (1991). "Approximation Capabilities Of Multilayer Feedforward Networks". In: *Neural Networks* 4.2, pp. 251–257.
- Kolm, Petter N., Jeremy Turiel, and Nicholas Westray (2021). "Deep Order Flow Imbalance: Extracting Alpha from the Limit Order Book". In: *Working Paper*.
- Kolm, Petter N. and Nicholas Westray (2023). "A Bayesian Approach to Analyzing Information Content of Cross-Sectional and Multilevel Order Flow Imbalance". In preparation.

References II


Sirignano, Justin A. and Rama Cont (2019). "Universal Features Of Price Formation In Financial Markets: Perspectives From Deep Learning". In: *Quantitative Finance* 19.9, pp. 1449–1459.


Tsantekidis, Avraam, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis (2020). "Using Deep Learning For Price Prediction By Exploiting Stationary Limit Order Book Features". In: *Applied Soft Computing* 93, p. 106401.


Zhang, Zihao, Stefan Zohren, and Stephen Roberts (2019). "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books". In: *IEEE Transactions On Signal Processing* 67.11, pp. 3001–3012.