MATH70125 Market Microstructure Coursework

An analysis of "Deep Learning Meets Queue-Reactive: A Framework for Realistic Limit Order Book Simulation"

Dominic Rushton CID: 06019954
Charles Favell CID: 01851680
Mustafa Hamdani CID: 06021930

March 19th 2025

Contents

1 Introduction

Limit Order Books (LOBs) are the primary mechanism for price discovery and trade execution across major global exchanges, and have become able to accommodate order flow at a very high frequency over the last several decades. Understanding and effectively modeling their dynamics is therefore essential for developing and testing trading strategies, assessing market stability, and analysing price impact.

In this report we analyse Hamza Bodor and Laurent Carlier's research article titled 'Deep Learning Meets Queue-Reactive: A Framework for Realistic Limit Order Book Simulation'. The paper introduces and explores several LOB models, starting with an overview of the Queue-Reactive (QR) model, introduced in 2014 (Huang, Lehalle, and Rosenbaum 2014) which has become widely popular in LOB modeling due to its tractability and effectiveness. Extensions of the QR model, namely the Size Aware Queue-Reactive model (SAQR) and Deep Queue-Reactive model (DQR) are also introduced and their applications and limitations are discussed.

The paper then presents the Multidimensional Deep Queue-Reactive (MDQR) model, which further extends the QR framework by implementing a neural network architecture to learn complex relationships between price levels based on the overall state of the order book. Using data from the Bund futures market, Bodor and Carlier demonstrate the strengths of the MDQR in reproducing important market properties and stylised facts of market microstructure.

In this report, we summarise some of the paper's main findings and attempt to replicate key aspects of the models described with simulated LOB data.

2 Simulating the Limit Order Book

To implement the LOB models described in the paper, we require a large amount of high frequency LOB data from a large tick instrument (an instrument that tends to trade at a bid-ask spread of one tick.) Given the difficulty and cost involved in procuring such data, we instead simulate the data. We use the Queue Reactive Model as a baseline model and include additional features including time varying intensities, exogenous events and queue level excitations.

Our approach consists of simulating order arrivals in the first 5 queues on both the bid and ask side of the LOB as a Poisson point process using pre-defined intensities (drawing on results in Huang, Lehalle, and Rosenbaum 2014). We model the arrival of limit, cancellation and market orders via a transition matrix, where the intensities of the 4 central queues have a dependence structure so that we can capture the dynamics that are typical of a true LOB. Using the intensities of order arrivals across the entire LOB for a given state we sample a waiting time to the next event from exponential density

$$f(x;\Lambda) = \Lambda e^{-\Lambda x}$$

where, given queues q and events η

$$\Lambda = \sum_{q=1}^{10} \sum_{\eta=1}^{3} \lambda^{q,\eta}$$

given that we have 10 total queues and 3 types of events that can occur. We then again sample from the normalized intensities to determine the event type and queue for the arriving order.

The reference price p_{ref} shifts with a fixed probability when one of the two central queues becomes empty. We make the assumption that all orders across all queues are the same size, a simplifying assumption that will not significantly impact the results in the paper that we seek to replicate.

In addition to this standard QR model, we introduce self-exciting Hawkes components to order arrival intensities using the kernel

$$\phi(t) = \sum_{s<t} \alpha e^{-\beta(t-s)}$$

where α and β control the magnitude and decay rate of the excitation. This makes it more likely for orders of the same type to occur consecutively, which is a stylized fact observed in empirical order book data. We also ensure that $\alpha < \beta$ so that this kernel is stable over time.

In addition, we introduce intraday time varying intensities using a sinusoidal time factor to capture the fact that trading volumes are highest at the beginning and end of the trading day. Finally, we incorporate a fixed probability of a price shock occurring at any moment in time along with a full redraw of the order book from its latent distribution, to capture the impact of exogenous shocks (i.e. news or other new information) influencing the market suddenly.

3 Independent Queue Models

The paper begins by providing an overview of 3 separate models which model the LOB by considering each queue independently.

3.1 Queue-Reactive Model

The QR model represents the LOB as a set of queues organised around a reference price p_{ref}, which is typically the midprice p_{mid} (the arithmetic average of the best available bid and ask price). Each queue represents the volume of orders at a given price and queue sizes are influenced by limit orders, cancellations and market orders.

More formally, the LOB is modelled as a $2K$ dimensional vector, where K is the number of price levels on each side of the book. Then p_{ref} divides the book into bid side queues $[Q_{-i} : i = 1, \ldots, K]$ and ask side queues: $[Q_i : i = 1, \ldots, K]$. A sequence $\mathcal{E}$ of events which evolves the LOB's queues is formulated as follows. $\mathcal{E} = \{e_k\}_{k=1}^{N}$, where each event e_k is characterized by its type $\eta_k \in \{L, C, M\}$ (limit, cancel, or market), the queue size q_k just before the event, and the time interval Δt_k since the previous event.

A differentiating aspect of the model is its introduction of intensity functions $\lambda_\eta(q)$ for each event type η, which determine the arrival rates of different orders based on the current queue size, incorporating the empirical observation that observable LOB conditions impact market participant behaviour. These intensities sum to a global intensity:

$$\Lambda(q) = \lambda^L(q) + \lambda^C(q) + \lambda^M(q).$$

The model's likelihood function is then:

$$\mathcal{L}(\{\lambda^\eta\}|\mathcal{E}) = \prod_{k=1}^{K} e^{-\Lambda(q_k)\Delta t_k} \lambda^{\eta_k}(q_k).$$

The maximiser of this likelihood has the form:

$$\hat{\lambda}^\eta(n) = \frac{\#\{e_k \in \mathcal{E} \mid \eta_k = \eta, q_k = n\}}{\#\{e_k \in \mathcal{E} \mid q_k = n\}} \cdot \left(\frac{1}{\#\{k \mid q_k = n\}} \sum_{\{k|q_k=n\}} \Delta t_k \right)^{-1}.$$

3.2 Size Adjusted Queue-Reactive Model

The SAQR model, introduced by Bodor and Carlier 2024 extends the QR model by modifying the intensity functions to include order sizes, specifically:

$$\hat{\lambda}^{\eta,s}(n) = \frac{\#\{e_k \in \mathcal{E} \mid \eta_k = \eta, s_k = s, q_k = n\}}{\#\{e_k \in \mathcal{E} \mid q_k = n\}} \cdot \left(\frac{1}{\#\{k \mid q_k = n\}} \sum_{\{k|q_k=n\}} \Delta t_k \right)^{-1}$$

The QR and SAQR use only queue sizes to model event arrivals, constraining their ability to reproduce certain stylised facts. These limitations motivate the use of deep learning techniques to incorporate more complex market features.

3.3 Deep Queue-Reactive Model

The DQR introduces a neural net that is able to learn intensities based on a flexible state vector x_k, which can incorporate any desired information about the order book state. Thus, it can be viewed as a generalization of the QR model, which takes only the queue size q_k as an input. In our implementation, we seek to replicate the paper's results by capturing the influence of intraday seasonality and event excitation on order arrival intensities.

With the more general state vector x_k as an input, the model parameterises the intensity functions through neural networks $\lambda_\eta^\theta(x_k)$, where θ represents the network parameters.

The model's likelihood function is then:

$$l(\lambda_\theta \mid \mathcal{E}) = \sum_{k=1}^{N} \log \lambda_\theta^{\eta_k}(x_k) - \Lambda_\theta(x_k)\Delta t_k$$

where

$$\Lambda_\theta(x_k) = \sum_{\eta \in \{L,C,M\}} \lambda_\theta^\eta(x_k).$$

We minimize the negative log-likelihood to calibrate the model:

$$\text{loss}(\theta \mid \mathcal{E}) = -l(\lambda_\theta \mid \mathcal{E}) = \sum_{k=1}^{N} \Lambda_\theta(x_k)\Delta t_k - \log \lambda_\theta^{\eta_k}(x_k).$$

This extension allows the DQR to capture complex market dynamics, some of which will be subsequently discussed.

3.4 DQR Results

3.4.1 Baseline Model

We implement the DQR following the model architecture used in the paper: a multi-layer perceptron with 2 hidden layers consisting of 64 and 32 nodes with hyperbolic tangent activations. This set-up results in computational efficiency while providing stable convergence to the empirical intensities across queues. However, the model described in the paper incorporates batch normalization layers between successive dense layers. Through experimentation we found that with a variety of momentum and scaling parameters for the batch normalization layers, the trained model exhibited a noticeably better fit when batch normalization was not utilized.

We first train the DQR model where the input x_k consists only of the current queue size $x_k = [q_k]$, and compare this with the intensities implied by the standard Queue Reactive model. For computational efficiency and reduced training time, we utilise 3 days of simulated data. The bias of the model will be reduced if a larger dataset is used (as a higher degree of generalization will be needed), however this amount of data was sufficient for demonstrating the efficacy of the model.

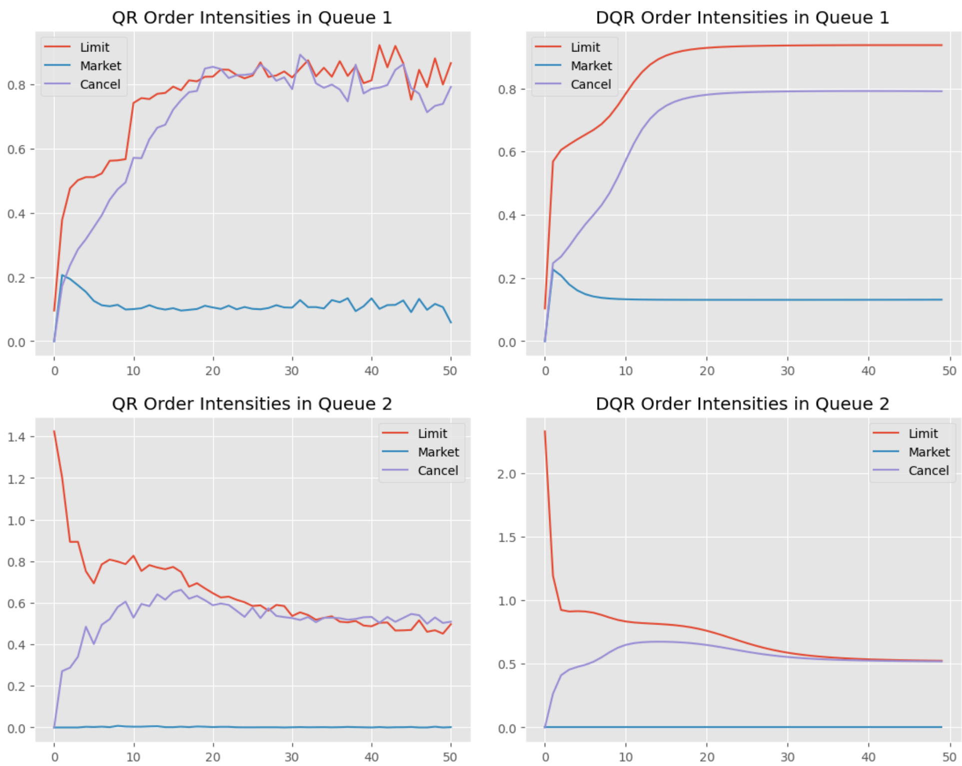

Figure 1: Empirical Intensities and DQR Intensities in Queue 1 and 2

From Figure 1, we can see that while the QR implied intensities exhibit noise (which would smooth out as the sampling period is increased), the DQR model trained on the same input data is able to learn the general shape of arrival intensities without significant overfit.

3.4.2 Excitation between Events

In high-frequency markets, the type of the previous event η_{k-1} often affects both the type and intensity of the next event. Therefore, it is crucial that this dynamic, known as excitation, is incorporated in realistic LOB modeling.

The QR and SAQR do not explicitly model order flow history and so fail to capture excitation. To address this, we include η_{k-1} as a feature in the state space $x_k = [q_k, \eta_{k-1}]$, which allows the DQR to learn how the previous event type in a given queue influences future events that occur in that queue.

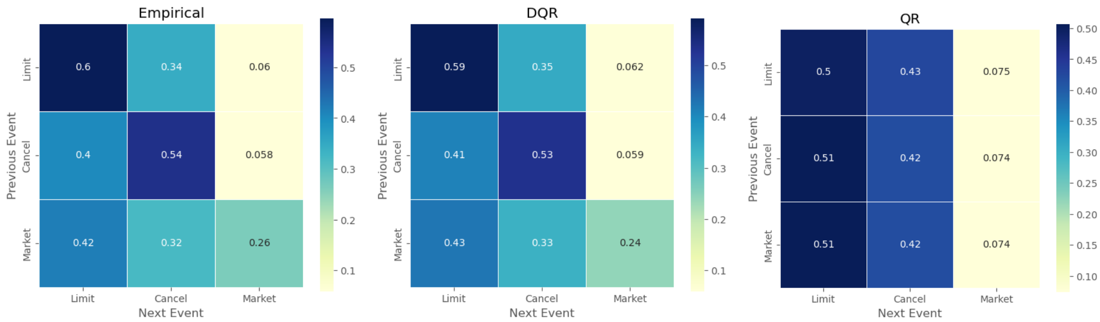

Figure 2: Transition probabilities between events in Queue 1

As shown in Figure 2, the empirically measured transition probabilities show a high degree of excitation between events. We observe the DQR is able to capture this behaviour, although not to the extent of our empirical observations. This aligns well with the result achieved in the paper. On the other hand, the SAQR and QR models produce uniform transition matrices as they fail to account for inter-dependencies between event types. Thus, we can see the large advantage the DQR has compared to the QR model in that additional features can be easily and flexibility added to better learn patterns in the underlying arrival intensities.

3.4.3 Intraday Seasonality

Another drawback of the QR model is its assumption of uniform market activity throughout the trading day. In reality, market activity varies significantly, often with increased activity during opening and closing periods and quieter phases around midday, this is known as intraday seasonality.

By including temporal information in the DQR model's state space we are able to capture this. Specifically, for each event e_k, we transform x_k into $x_k = [q_k, h_k]$, where h_k represents the current hour of the day and is included as a categorical feature. Per the specifications in the paper, we embed this categorical features into a 2-dimensional representation that is learnable by the model.

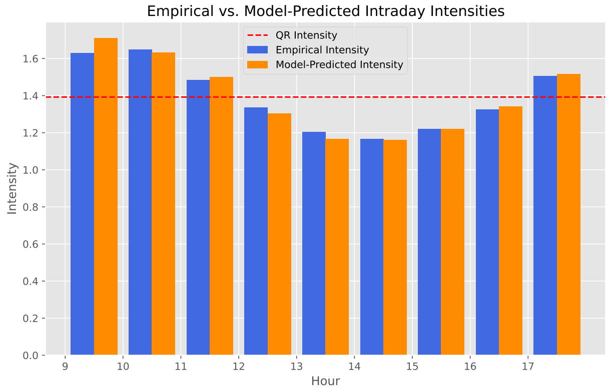

Figure 3: Total Order Intensity in Queue 1 throughout the trading day

Figure 3 demonstrates the DQR's ability to learn market activity patterns across the trading day. We notice that the model predicted intensities are almost identical to the empirical data, which is expected given that the simulated data has intraday seasonality patters driven by a fixed sinusoidal function.

3.4.4 Combined Model and Analysis of Results

We now combine the two aformentioned models so that $x_k = [q_k, \eta_{k-1}, h_k]$. To validate the impact of the DQR's feature enrichment, we simulate additional LOB data and test each model's performance. Figure 4 shows the inclusion of more information into the state space x_k leads to improvements over the vanilla model, which takes in only the current queue size q_k as an input. However, both the margin and consistency of improvement when additional features is added is not fully consistent with the results in the paper.

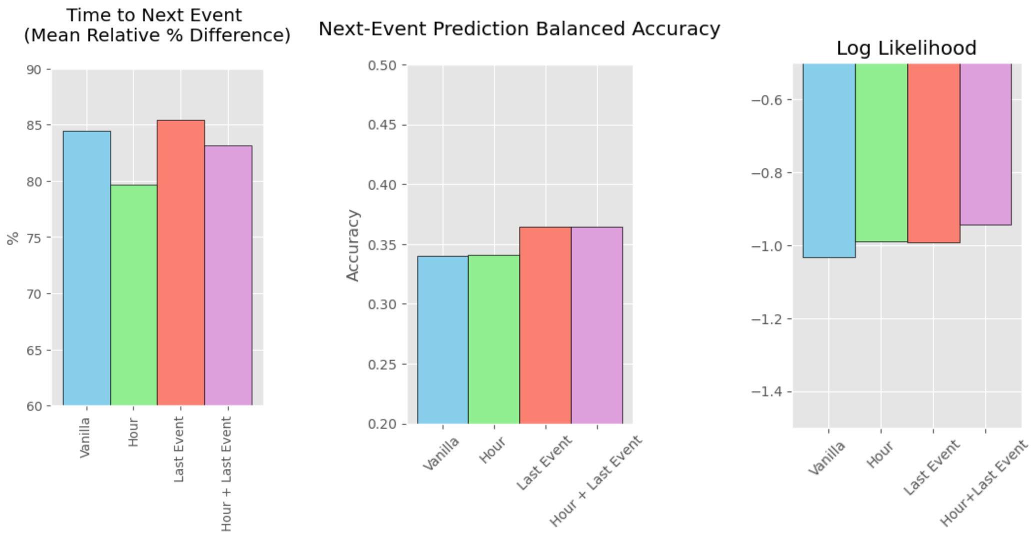

Figure 4: Performance of the DQR models based on input feature space

The left panel of Figure 4 shows the difference in time to next event compared to our simulated data. As we would expect, the models that takes h_k as an input in the feature space exhibit more accurate inter-event times. However, we would expect the model with $x_k = [q_k, \eta_{k-1}, h_k]$ to perform as well as (if not better than) the model with just $x_k = [q_k, h_k]$ given that it has information in addition to just h_k and so it should capture both the seasonality and excitation effects. We plot the balanced accuracy for next event prediction in the center panel of Figure 4. Here we observe that there is a very small difference between the accuracies, but the models that have access to η_{k-1} are consistently more accurate than those without this feature as we would expect.

We believe that, compared to the results of the paper, the relatively small difference in next event prediction across models is driven by the smaller degree of excitation in our simulated data compared to empirical estimates. This is clearly demonstrated when we compare the event transition matrices of our simulation to empirical event transition matrices from the paper.

Finally, the right panel in Figure 4 shows an increasing log-likelihood as more features are added to the model, suggesting a better fit to the data. We note that these log-likelihood gains are minimal compared to those reported in Bodor and Carlier 2025. Again, this is likely driven by the fact that seasonality and excitations effects in the simulated data are not as extreme as true empirical observations.

3.4.5 Summary

These results demonstrate the DQR's improvement on the QR model. We see that the DQR model's effectiveness in replicating market properties is influenced by the choice of features in the state space x_k, as evidenced by $x_k = [q_k, \eta_{k-1}, h_k]$ showing the best general performance across our validation tests. The limited degree of improvement as features are added is likely due to the nature of the simulated data rather than a flaw in the modeling approach. Moreover, our results still encourage future investigation into more comprehensive state spaces in modeling independent queues.

However, it remains that the DQR in this form still treats the queues independently much like the QR/SAQR and does not have the flexibility to model order sizes. It is clear that these limitations hurt the models ability to generate realistic order flow, which motivates an investigation of the Multidimensional Deep Queue-Reactive Model.

4 Multidimensional Deep Queue-Reactive Model

The MDQR extends the QR framework in three ways.

1. It generalises q_k to the state space x_k (as in the DQR).
2. It considers the LOB as a single multidimensional entity, to capture queue dependence.
3. it incorporates order sizes into the probabilistic formulation.

More formally, for a sequence of events $\mathcal{E} = \{e_k\}_{k=1}^{N}$, each event $e_k = (\eta_k, \ell_k, \Delta t_k, s_k, x_k)$ is characterised by:

- $\ell_k \in P = \{-K, \dots, -1, 1, \dots, K\}$: The price level at which the event occurs.
- s_k: The size of the order associated with the event.

With η_k, Δt_k defined as before and x_k a state vector that may include information from multiple levels and sides.
The joint likelihood of the observed event sequence is as follows:

$$\mathcal{L}(\theta \mid \mathcal{E}) = \left(\prod_{k=1}^{N} p(\eta_k, \ell_k, t_k \mid x_k; \theta) \right) \times \left(\prod_{k=1}^{N} p(s_k \mid \eta_k, \ell_k, t_k, x_k; \theta) \right)$$

For the first part of this factorisation, we assume a conditional Poisson structure for event arrivals, as in the QR and DQR models. We define an intensity function $\lambda^{(\eta,\ell)}(x_k;\theta)$ for each event category η and level ℓ. Then the total intensity of observing any event at any level is:

$$\Lambda(x_k;\theta) = \sum_{\eta \in \{L,C,M\}} \sum_{\ell \in P} \lambda^{(\eta,\ell)}(x_k;\theta)$$

The probability of an event of type η_k, at level ℓ_k occurring at time t_k given the state x_k is then:

$$p(\eta_k, \ell_k, t_k \mid x_k; \theta) \propto e^{-\Lambda(x_k;\theta)\Delta t_k} \lambda^{(\eta_k,\ell_k)}(x_k;\theta)$$

For model calibration, we minimise the negative log-likelihood:

$$\ell_\lambda(\theta) = \sum_{k=1}^{N} \Lambda(x_k;\theta)\Delta t_k - \log \lambda^{(\eta_k,\ell_k)}(x_k;\theta)$$

The second component of the factorisation, models the distribution of order sizes conditioned on the event type, level, and state. To specify $p(s_k \mid \eta_k, \ell_k, t_k, x_k; \theta)$ we discretise the order size space into a set of predefined classes or bins. We then train a neural network with input (η_k, ℓ_k, x_k) to estimate the probability distribution over discrete order size classes, optimising parameters by minimizing the cross-entropy loss across all observed events:

$$\ell_s(\theta) = -\sum_{k=1}^{N} \sum_{c=1}^{C} y_{k,c} \log \hat{p}_c(s_k \mid \eta_k, \ell_k, x_k; \theta)$$

where $\hat{p}_c$ denotes the model's predicted probability for class c, $y_{k,c}$ is the one-hot encoded ground truth (1 if s_k belongs to class c, 0 otherwise), and C is the total number of size classes. The probabilities are conditioned on the event characteristics (η_k, ℓ_k) and market state x_k.

A major advantage of the MDQR over the independent queue models discussed earlier is its ability to learn more complex relationships between order arrival intensities and the current state of the order book, including historical order flow features and dependencies between queues.

4.1 Implementing the MDQR

While the MDQR model used in the paper simulates both the intensities and sizes of events arriving into the LOB, we focus here only on the former. Practically this means that we only parameterize one neural network, which learns intensities of all event types across all queues based on our input LOB state vector x_k. Here x_k consists of the following features

Feature	Preprocessing
Queue Sizes	$\log q_k$
Last Event Types	One hot encoding $\rightarrow$ Embedded layer
Hour	One hot encoding $\rightarrow$ Embedded layer
Bid-ask Spread	None
Trade Imbalance	None

where the trade imbalance is measured over the previous 20s, 1 minute, 5 minutes and 15 minutes and is calculated as

$$TI(t) = \frac{V^b(t) - V^a(t)}{V^b(t) + V^a(t)}$$

where V^b, V^a represent the volume of trades on the bid and ask sides of the book over the specified time horizon.

After experimentation with Bayesian Optimization for the tuning of model hyper-parameters, we ultimately follow the architecture implemented in the paper: a multi-layer perceptron with 2 hidden layers containing 256 and 64 neurons and hyperbolic tangent activation functions. This architecture was most consistently able to produce intensities across all 10 queues while exhibiting stable convergence behaviour.

We now replicate results in the paper which test the MDQR's ability to learn inter-queue dynamics. For this purpose, additional LOB data is simulated using the intensity functions learned by the MDQR from our original LOB simulation. We note here that the process of generating predictions from the model was computationally intensive, and so we were unable to replicate the length of the simulations used in the paper.

4.2 Capturing Inter-Queue Dependence

The MDQR's feature space means that it should be able to capture the relationships that exist between different queues across the order book.

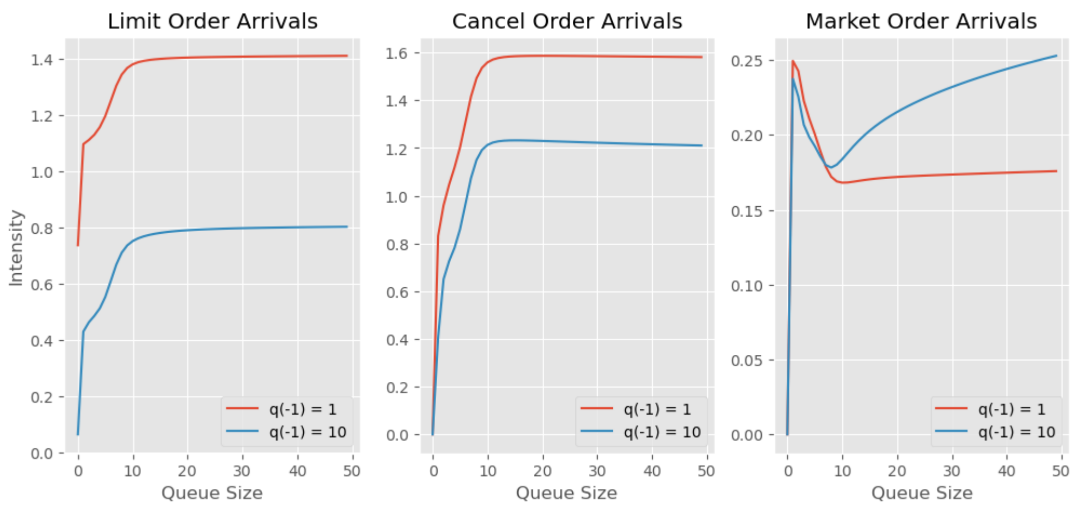

Figure 5: MDQR intensities in the first ask based on size of the first bid queue

Figure 5 clearly demonstrates this capability, as we observe a distinct difference in the behaviour of order arrivals at the first ask level depending on the value of the first bid level. Further, these learned intensities aligns well with the intensity functions that were parameterized as inputs to the empirical LOB data that we simulated.

4.3 Event Excitation and Frequency

A principal motivation for relaxing the queue independence assumption of the DQR is to better learn excitation patterns between different queues, particularly on the bid and ask side. Figure 6 displays transition matrices for events in Q_{-1} and Q_1. As observable especially along the main diagonal, the MDQR closely matches the empirical matrix, demonstrating its able to learn excitation probabilites across multiple queues.

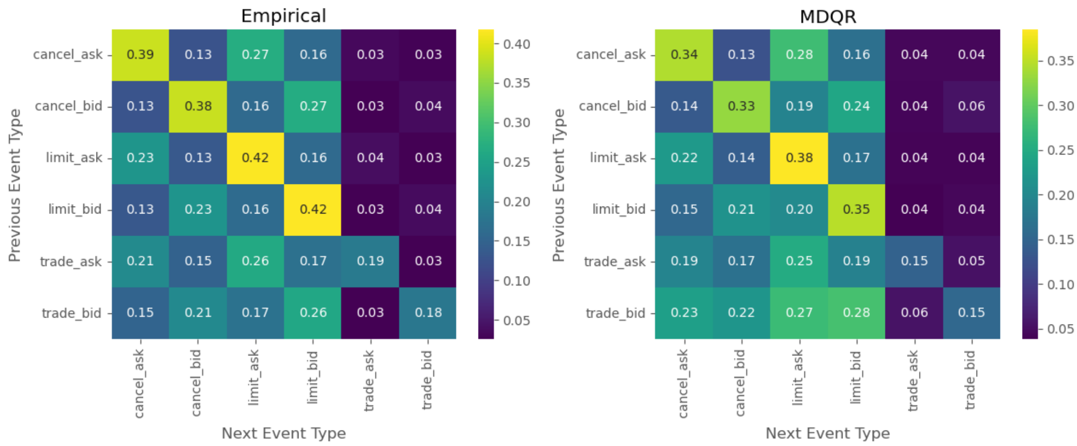

Figure 6: Empirical and MDQR Transition Matrices at the Best Price Level

Another important feature in order book models is the number of orders that arrive over a fixed time window. While real order books exhibit a large spread in the arrival count (i.e. there may be a large decrease in order arrivals due to news or other events), our empirical LOB simulation shows a much tighter clustering of event arrival counts around the median, which is to expected given that it adapts the QR model.

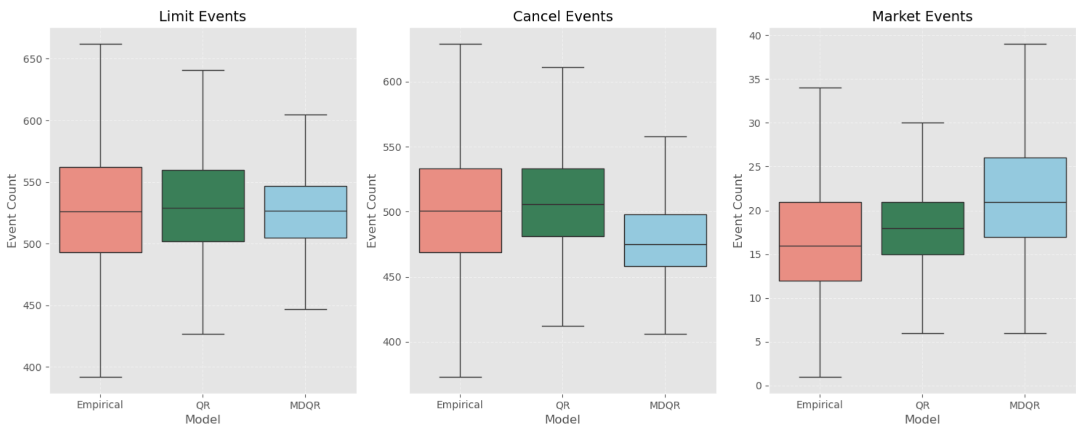

Figure 7: Frequency of each event type in a 5 minute lookback window sampled over the full simulation period

Figure 7 shows that, as expected, both the QR model and MDQR model provide a reasonable approximation to order event counts for each type of event. The QR model exhibits a slightly tighter grouping around its median, which aligns with the fact that there are no variations in event intensities based on the order book state. However, the MDQR does not capture the range of values that we would expect, particularly given its ability to replicate excitation behaviour in the LOB. We believe that with a larger sample of MDQR simulated data, the box plots would likely more closely replicate the behavior of the empirical data.

4.4 Queue Size Distribution

We now examine the degree to which the MDQR model is able to replicate the queue size distribution of our simulated data. Given that we parameterize our empirical order book simulation with a fixed event size, we believe that the MDQR should exhibit only a small improvement over the vanilla QR model.

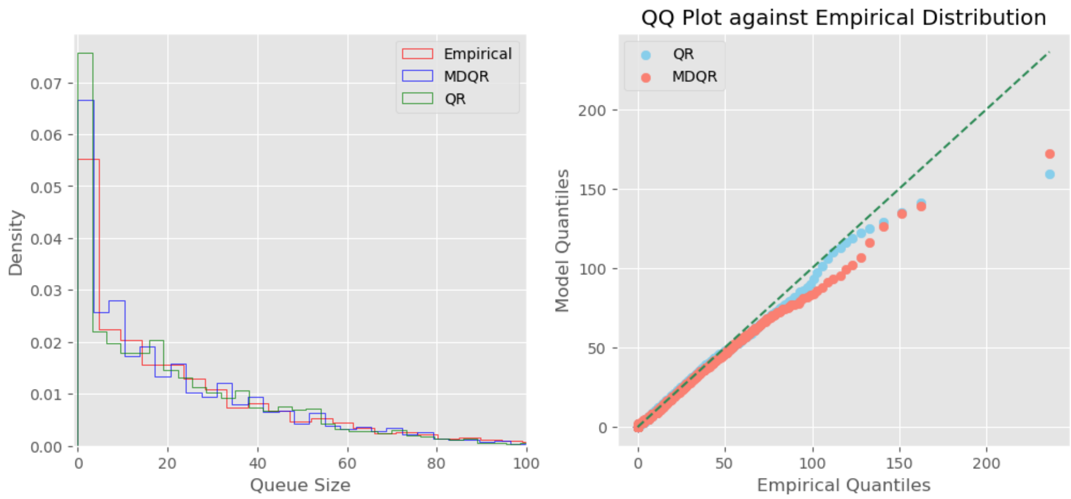

Figure 8: Density and QQ plots of queue size distributions

From the density plot in Figure 8, we see that both the MDQR and QR models provide a good approximation to the empirically observed queue size distribution. The MDQR potentially fits the empirical data slightly better for small queue sizes, likely due to order arrival/departure ratios being more closely aligned as queue sizes vary. However, from the QQ plot we see that any improvement is very minor as for smaller queue sizes the quantiles are well aligned across all models.

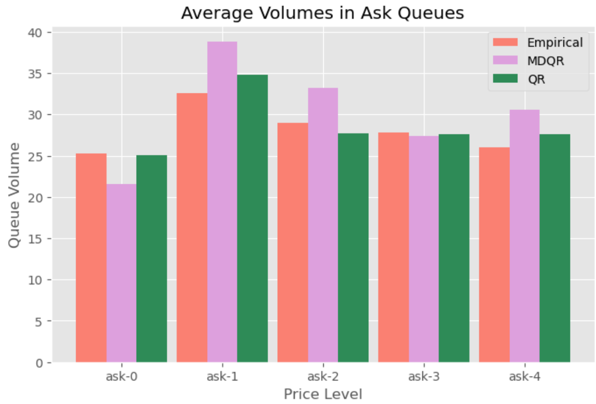

Figure 9: Average queue size on the ask side

Figure 9 compares average queue size across multiple levels on the ask side. Once again both the MDQR and QR models are able to reasonably approximate the behavior observed in our simulated order book data.

4.5 Summary and Additional Results

The above results concerning excitation, queue size distribution and event count demonstrate the MDQR's consistent success in modeling inter-queue patterns seen in LOB data.

In addition to these results, the paper tests the MDQR's computational efficiency and its ability to model price impact and returns distributions. Given the nature of our simulated data and lack of access to computational resources, we opt not to replicate these results, yet it should be noted that the model performs well in these areas. By incorporating the aforementioned Trade Imbalance feature the paper finds that the MDQR is able to accurately model the square-root law of market impact (Bucci et al. 2019) and also captures the characteristic heavier tails of historical returns as compared to simpler independent queue models. Regarding computational efficiency, the paper claims that the MDQR has an impressively low inference time compared to competing techniques for simulating order book data, making it suitable for reinforcement learning applications in the future where it may be leveraged to produce training data to develop automated trading strategies.

5 Conclusion

Hamza Bodor and Laurent Carlier's research article outlines several LOB models and clearly demonstrates each model's improved ability to capture behavior of the Limit Order Book, and with the MDQR, replicate key stylized facts and complex market properties. The MDQR also incorporates flexible state space x_k, used here with the trade imbalance feature, which can be easily be extended to capture further information about the order book state. This encourages future research into more comprehensive feature spaces to improve the models predictive performance. Potential examples include other seasonal factors such holidays that influence market behaviour, or a metric that captures the fact that orders tend to arrive in clusters. Further extensions could include modeling order arrivals across positively/negatively correlated assets to produce simulations with broader applications.

However, the paper relies on highly liquid Bund futures market data to train its neural networks. There's little indication gives little of the MDQR's performance when trained with data from other, potentially less liquid, large-tick assets which might exhibit significantly more irregularity in order flows. Additionally, testing the MDQR on small-tick assets, with distinct market characteristics will help evaluate its broader applicability.

References

Bodor, Hamza and Laurent Carlier (2024). *A Novel Approach to Queue-Reactive Models: The Importance of Order Sizes*. arXiv: 2405.18594 [q-fin.TR]. URL: https://arxiv.org/abs/2405.18594.

— (2025). *Deep Learning Meets Queue-Reactive: A Framework for Realistic Limit Order Book Simulation*. arXiv: 2501.08822 [q-fin.TR]. URL: https://arxiv.org/abs/2501.08822.

Bucci, Frédéric et al. (Mar. 2019). "Crossover from Linear to Square-Root Market Impact". In: *Physical Review Letters* 122.10. ISSN: 1079-7114. DOI: 10.1103/physrevlett.122.108302. URL: http://dx.doi.org/10.1103/PhysRevLett.122.108302.

Huang, Weibing, Charles-Albert Lehalle, and Mathieu Rosenbaum (2014). *Simulating and analyzing order book data: The queue-reactive model*. arXiv: 1312.0563 [q-fin.TR]. URL: https://arxiv.org/abs/1312.0563.