RESEARCH-ARTICLE

State Dependent Parallel Neural Hawkes Process for Limit Order Book Event Stream Prediction and Simulation

ZIJIAN SHI, University of Bristol, Bristol, U.K.

JOHN CARTLIDGE, University of Bristol, Bristol, U.K.

Open Access Support provided by:

University of Bristol

Published: 14 August 2022

Citation in BibTeX format

KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 14 - 18, 2022
Washington DC, USA

Conference Sponsors:
SIGMOD
SIGKDD

https://doi.org/10.1145/3534678.3539462
ISBN: 9781450393850

State Dependent Parallel Neural Hawkes Process for Limit Order Book Event Stream Prediction and Simulation

Zijian Shi
zijian.shi@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

John Cartlidge
john.cartlidge@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

ABSTRACT

The majority of trading in financial markets is executed through a limit order book (LOB). The LOB is an event-based continuously-updating system that records contemporaneous demand ('bids' to buy) and supply ('asks' to sell) for a financial asset. Following recent successes in the literature that combine stochastic point processes with neural networks to model event stream patterns, we propose a novel state-dependent parallel neural Hawkes process to predict LOB events and simulate realistic LOB data. The model is characterized by: (1) separate intensity rate modelling for each event type through a parallel structure of continuous time LSTM units; and (2) an event-state interaction mechanism that improves prediction accuracy and enables efficient sampling of the event-state stream. We first demonstrate the superiority of the proposed model over traditional stochastic or deep learning models for predicting event type and time of a real world LOB dataset. Using stochastic point sampling from a well trained model, we then develop a realistic deep learning-based LOB simulator that exhibits multiple stylized facts found in real LOB data.

CCS CONCEPTS

• **Mathematics of computing → Time series analysis**; • **Computing methodologies → Neural networks**; • **Applied computing → Economics**.

KEYWORDS

limit order book, neural point process, financial market simulation

ACM Reference Format:
Zijian Shi and John Cartlidge. 2022. State Dependent Parallel Neural Hawkes Process for Limit Order Book Event Stream Prediction and Simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539462

1 INTRODUCTION

Most contemporary financial markets adopt the continuous double auction (CDA) mechanism to determine the price of a certain asset. Buyers and sellers interact via a physical or digital venue by submitting bid orders (i.e., orders to buy) and ask orders (i.e., orders to sell), which contain price and volume information, into an order queuing system called the limit order book (LOB). A matching engine then matches orders into transactions, and the resulting time series of transaction prices form an asset's price on the micro level. The information involved in such a process can be concisely expressed as an irregularly sampled time series of event streams posted by market participants, together with an event-based continuously-updating LOB that stores all unexecuted orders left in the market. The accumulation of order streams determines the evolution of the LOB. Therefore, gaining insight into order stream patterns helps us better understand and predict LOB behaviour.

Empirical evidence indicates that the arrival of different types of events in a LOB market can be fitted with certain stochastic point processes, in which intensity functions are used to control the frequency of the arrival of events in a unit time interval. In a naïve setting, the intensity rates can be set as history irrelevant (e.g. a Poisson process [7]); while in a more advanced and realistic setting, can be set as history dependent (e.g. a Hawkes process [19]). For instance, in a multivariate Hawkes process the intensity rates of different types of events interact, in the sense that the intensity rate of a particular type of event not only depends on its own history but also on the arrival history of other types of events. Such a setting introduces more complex interaction dynamics among events, allowing us to grasp the pattern of event streams instead of rigidly allocating a fixed intensity rate to each event type. Stochastic point processes are one of the mainstream tools for modelling LOB dynamics.

Inspired by the recent progress made in the field of deep learning, a neural point process was proposed to model a stochastic point process using neural networks [21, 32]. Such a combination fuses the merits of good explainability of traditional stochastic models, and the improved fitting ability of neural networks. Such approaches have achieved state-of-the-art performance in predicting medical visits [9], online purchasing behaviours [13], and social media behaviours [33]. By categorizing events types in a LOB market according to the order type (submission or cancellation) and the side it is affiliated with (ask or bid), neural point process family models can be easily adapted to learn the patterns embedded in order streams. Another unique characteristic of modelling the LOB event stream in this manner is that it provides us with a validated way to stochastically sample (using the thinning algorithm [24]) the most likely next event type and time according to the evolution of the intensity function. Further, by combining the iteratively generated events with affiliated price and volume information sampled from empirical distributions, an alternative method of LOB simulation can be achieved.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539462

Figure 1: A LOB of four price levels evolving with time. White and blue boxes indicate the top level and deeper levels of the LOB. Grey boxes indicate market events stream. Orange box indicates trade records.

This paper presents the first systematic attempt to utilize neural point process family models in LOB market event stream prediction. The paper further extends the topic from event type prediction to realistic LOB data simulation, to enrich current methodologies in generating simulated LOB data. The main contributions are:

(1) The parallel continuous time long-short memory (PCT-LSTM) module is proposed to model the intensity rates of different types of events in LOB markets. In its core, K (the number of event types) continuous time long-short memory (CT-LSTM) units are stacked in parallel to encode the whole event stream sequence into separate exponentially decaying latent states. Unlike the original neural Hawkes process, which uses a single CT-LSTM unit for all events, the parameters that control the evolution of latent states are tailored to each event type and are no longer shared.
(2) An event-state interaction mechanism is introduced, in which the current market state indicator influences the evolution of intensity rates and, in turn, the type of each event arrival influences the state transition probability. This mechanism is shown to enhance prediction accuracy for event type and time, while also allowing efficient sampling of the event state stream for stream simulation.
(3) We use the ability to predict the upcoming event to build a realistic LOB market simulator. By implementing the thinning algorithm, the minimum time until the next event arrival and the most likely event type at that time are stochastically generated from a model that has been trained on historical event stream patterns. We are then able to simulate an event stream of arbitrary length by iteratively incorporating new predictions as input to the next iteration; and using empirical distributions to generate each order's price and volume. Experiments confirm that the simulated LOB exhibits several stylized facts that exist in real LOB data.

2 BACKGROUND AND RELATED WORKS

2.1 The Limit Order Book

A CDA market is *double* in the sense that market participants are allowed to enter both buy and sell orders. The market being *continuous* means that there is no time interval limit between two consecutive events, and orders can be submitted or cancelled at any time as long as the market is open. An electronic LOB is used to record submitted yet unexecuted orders. As the LOB provides the most detailed demand and supply information in the market, it is deemed as the ultimate microscopic level of description [2].

A *limit* order event $E(t, s, a, v, p)$ specifies a time of event t, a side of event s being bid or ask, an action a being submission or cancellation, a volume v, and a price p. The LOB is updated whenever a new event arrives. The LOB contains a bid list and an ask list, each sorted by price-time priority such that the bid at the front of the bid list (i.e., the *best bid*) has the highest price, $p^{b(1)}$, and the ask at the front of the ask list (i.e., the *best ask*) has the lowest price, $p^{a(1)}$. Best bid $p^{b(1)}$ and best ask $p^{a(1)}$ are termed *quote prices*, and are considered as the *top price* level. The difference between $p^{b(1)}$ and $p^{a(1)}$ is the *bid-ask spread*, and the average of $p^{b(1)}$ and $p^{a(1)}$ is the *mid-price*. We refer to $p^{b(n)}$ and $p^{a(n)}$, where $n \in \{2, 3, ..\}$, as *deep price* levels. When new bid b arrives with price p^b it will execute against the best ask if $p^b \geq p^{a(1)}$, else b will enter the bid-side of the LOB in descending price-ordered position. Likewise, when new ask a arrives with price p^a it will execute against the best bid if $p^a \leq p^{b(1)}$, else a will enter the ask-side of the LOB in ascending price-ordered position. Fig. 1 presents a LOB evolving over time in response to a stream of order arrivals.

2.2 Modelling the LOB

Despite the fact that empirical studies on the LOB are rich, formulating theoretical models that achieve a balance between being analytically tractable and reflective of market realism is challenging [12]. Theoretical LOB models can be summarized into stochastic models and equilibrium models [7]. Stochastic models typically use probabilistic distributions such as independent Poisson processes to describe order flow [29], and stochastic processes such as Markovian queuing systems to model the mechanics of the LOB [7]. In comparison, equilibrium models take a game theoretic perspective of interaction between traders, using subjective utility functions to describe strategy payoffs [25, 26]. Agent-based models (ABMs) are a variant form of equilibrium model in which heterogeneous agents interact with each other and the environment. ABMs offer the ability to include heterogeneity in agents' behaviour patterns, and enable macro-level phenomena to emerge from complex non-linear micro-level interactions [20]. However, ABMs suffer from complications tracing simulation outcomes back to agent parameters, and defined agent behaviours may not be realistic [12].

More recently, deep learning models of the LOB have been introduced. One characteristic study utilized a generative adversarial network to replicate the latent pattern embedded in real order

stream data [18]. The realistic order streams generated can replicate several stylized facts found in real LOB data. The LOB recreation model (LOBRM) proposed in [27, 28] used continuous recurrent neural networks to predict volume information at deep price levels in LOB snapshots, using top level trade and quote data as input. The drawback of LOBRM is that few stochastic characteristics of the market are considered in the modelling process. This lowers the model's explainability, which is a significant criteria in evaluating financial deep learning models [17].

2.3 Neural Point Process

A temporal point process is a stochastic process used to describe a series of event arrivals that occur in continuous time. The dynamics of a point process is described through a intensity function $\lambda(t)$. For an infinitesimal time window $(t, t + dt]$, the rate of event arrivals is represented as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{E(N(t + \Delta t) - N(t))}{\Delta t} \tag{1}$$

in which $N(t)$ is the affiliated counting process that counts total number of arrivals at time t. Various forms of point process have been proposed, to account for an intensity function that is independent, dependent on time, or dependent on event history. For instance, a Hawkes process captures the excitation effect of past events on the intensity rate of current event arrivals [15]. The intensity function in a history dependent Hawkes process with exponential kernel is represented as:

$$\lambda(t) = \mu + \sum_{h:t_h<t} \alpha * exp(-\delta(t - t_h)) \tag{2}$$

in which μ, α, δ respectively indicate mean rate, excitation coefficient, and decaying coefficient.

Neural point processes [31] differ from a traditional temporal point process in that neural networks are used to govern the evolution of event intensity rate through time. One representative model is the neural Hawkes process [21]. In a neural Hawkes model, rather than modelling the additive individual excitation effect that past events lay on the intensity rate [15], the intensity rate during a time interval is directly decoded from the hidden state of a CT-LSTM unit that contains a decaying mechanism. Zhang *et al.* [32] further combined a self-attention mechanism into a neural Hawkes process to explicitly model interactions between events. Neural point processes have achieved improved performance in various application domains compared with traditional stochastic models.

As indicated by empirical LOB studies, event streams in the LOB follow a multivariate Hawkes process in a general sense. This observation provides potential in using state-of-the-art neural point process models to model LOB dynamics. In fact, in [21, 32], performance of neural Hawkes models have already been tested in a toy LOB event stream dataset. In this study, we extend the research on combining neural point processes with LOB modelling on real event stream data, thus providing the research community a new perspective on modelling the LOB.

3 MODEL FORMULATION

3.1 Problem Description and Notation

Samples are arranged as a set of event sequences $C = \{S^1, \ldots, S^J\}$. $S^i = (e_1^i, \ldots, e_J^i) = ((t_1^i, y_1^i), \ldots, (t_J^i, y_J^i))$ in which $t_j^i \in \mathbb{R}^+$ and $y_j^i \in \{0, 1, 2, 3\}$ respectively represent the time and event type of the j-th arrival of the i-th event sequence. Event arrivals are categorized into four classes: submission of order at the bid side ($y = 0$), cancellation of order at the bid side ($y = 1$), submission of order at the ask side ($y = 2$), and cancellation of order at the ask side ($y = 3$). The state sequence coupled with S_i is denoted as $X^i = (x_1^i, ..., x_J^i)$. The event stream for a specific financial asset during each day is manually sliced into sub-streams of equal length J, using the rolling window approach of step size one. Given sequences S and X of length J, the model receives the first $J-1$ events as input and makes a prediction on the j-th event type y_J^i.

3.2 State Dependent Neural Hawkes Process

One distinct characteristic of event streams in a LOB market is that the arrival of events not only depends on the event history, but is also influenced by the current market state [11, 23]. For example, at any given time, the next event type is influenced by whether the current bid-ask spread is large or small, and whether or not the aggregate volume of orders is greater on the ask side or bid side.

In a state-dependent neural Hawkes process, the influence of past event history laid on current event intensity rate during time interval $(t_j, t_{j+1}]$ is no longer additive. First, a neural network is used to iteratively encode history information into a hidden state, and the intensity rate is decoded from such a latent state when making predictions. Second, in order to incorporate time information, the exponential decaying mechanism is laid upon the latent state instead of the intensity rate directly. Considering the aforementioned event-state interaction mechanism, the parameters of the CT-LSTM at j-th discrete inputs updates in the following manner:

$$\tilde{h}_j^- = \text{concat}((\tilde{y_j} + \tilde{x_j}), h_j^-) \tag{3}$$
$$i_j = \text{sigmoid}(\text{LC}_i(\tilde{h}_j^-)) \tag{4}$$
$$\bar{i}_j = \text{sigmoid}(\text{LC}_{\bar{i}}(\tilde{h}_j^-)) \tag{5}$$
$$f_j = \text{sigmoid}(\text{LC}_f(\tilde{h}_j^-)) \tag{6}$$
$$\bar{f}_j = \text{sigmoid}(\text{LC}_{\bar{f}}(\tilde{h}_j^-)) \tag{7}$$
$$o_j = \text{sigmoid}(\text{LC}_o(\tilde{h}_j^-)) \tag{8}$$
$$z_j = \tanh(\text{sigmoid}(\text{LC}_z(\tilde{h}_j^-)) \tag{9}$$
$$c_j = f_j \odot c_j^- + i_j \odot z_j \tag{10}$$
$$\bar{c}_j = \bar{f}_j \odot \bar{c}_{j-1} + \bar{i}_j \odot z_j \tag{11}$$
$$\delta_j = \text{softplus}(\text{LC}_d(\text{concat}((\tilde{y_j} + \tilde{x_j}), h_j^-)) \tag{12}$$

in which i, f, o are gating controls in a normal LSTM unit; $\tilde{y}$ and $\tilde{x}$ are the representations of event and state after embedding; c and $\bar{c}$ are the initial cell state and the cell state decaying target; h_j^- is the hidden state just before receiving input at time t_j; δ is the parameter to control decaying pace; LC represents linear connection layer;

$\odot$ is the operation of Hadamard product. During $(t_j, t_{j+1}]$, the cell state, hidden state and intensity rate at time t can be derived as:

$$c(t) = \bar{c}_j + (c_j - \bar{c}_j) \exp\left(-\delta_j \left(t - t_j\right)\right) \tag{13}$$

$$h(t) = o_j \odot \tanh(c(t)) \tag{14}$$

$$\lambda_k(t) = \text{softplus}(\text{LC}_\lambda(h(t))[k]) \tag{15}$$

By considering the event-state interaction mechanism for a stochastic point process [22], after event e_j happens, the state transition follows the distribution:

$$P(x_j \mid y_j, F_{j-}^{S,X}) \sim \phi_{y_j, x_j^-} \tag{16}$$

in which $F_{j-}^{S,X}$ indicates event stream and state history just before time t_j. $\phi_{y_j,x_j^-}$is a discrete distribution when the event type at time t_j is y_j and the state just before time t_j is x_j^-. Under such a formation, the intensity rates of events are influenced by the state, and the state transition thereafter is in turn influenced by which type of event happens at the moment. We use a market condition indicator similar to [5], which is calculated as $I = (v^{b(1)} - v^{a(1)})/(v^{b(1)} + v^{a(1)})$ and the state is denoted as:

$$x = \begin{cases} 0 & I \in [-1, -\theta) \\ 1 & I \in [-\theta, \theta] \\ 2 & I \in (\theta, 1] \end{cases} \tag{17}$$

Such an 'imbalance' indicator describes whether the buying or the selling side for an asset takes dominance. If the order volume on the ask side is much more than the bid side, the market is an 'ask market' and the price tends to go downwards; Likewise if the volume on the bid side is much more than the ask side, the market is a 'bid market' and the price tends to go upwards; If the volume on both sides are roughly balanced, we call it a 'balanced market' and fluctuation in price tends to have neutral direction.

3.3 Stacking of CT-LSTM Units

In the original neural Hawkes process, a single CT-LSTM unit is used to encode input and time information into the latent state with decaying kernels. The intensity rates for all types of events are decoded from the single latent state. Parameters related to encoding time information (i.e., c, $\bar{c}$, δ) are shared across all event types, and the expression of the decaying mechanism for different event type intensity rates is bounded.

With the aim of enriching the expression of latent state evolution for different types of events, we stack K CT-LSTM in parallel to form a PCT-LSTM module for encoding input and time information. Each unit governs the evolution of the intensity rate for different event types. At the same time, to enable parameter sharing across different units, we modify the mechanism used to update hidden states at discrete input timesteps (shown in Eq. 3). The decayed hidden states of different units together with inputs are first passed through a common linear fully connected layer, and the outputs are then used as hidden states in respective units:

$$\tilde{h}_{j,1}^-, ..., \tilde{h}_{j,K}^- = \text{LC}_h(\text{concat}((\tilde{y_j} + \tilde{x_j}), h_{j,1}^-, ..., h_{j,K}^-) \tag{18}$$

We find in experiments that this shared linear layer at discrete input timesteps across stacked units can effectively improve prediction

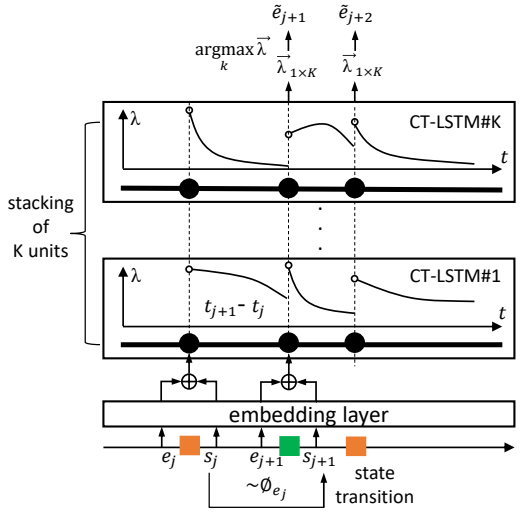


Figure 2: Model framework.

accuracy, with an improvement of approximately 2% in terms of type prediction accuracy compared with the model in which this parameter sharing layer is absent. After the hidden state is updated, the intensity rate for event type k in a single CT-LSTM unit module and a stacked PCT-LSTM module are respectively indicated as:

$$\lambda_k(t) = \text{D}(h(t))[k] = \text{D}(\text{CTLSTM}(S, X, t))[k] \tag{19}$$

$$\lambda_k(t) = \text{D}_k(h_k(t)) = \text{D}_k(\text{CTLSTM}_k(S, X, t)) \tag{20}$$

The general framework of the proposed state-dependent parallel neural Hawkes process model is illustrated in Fig. 2

3.4 Loss Function

Given an intensity function $\lambda(t)$, the conditional density function that the next event happens at time t conditioned on history $F_{j-}^{S,X}$ can be derived as [8]:

$$p_j(t) = P(t_j = t \mid F_{j-}^{S,X}) = \lambda(t) \text{exp}\left(-\int_{t_{j-1}}^{t} \lambda(s) ds\right) \tag{21}$$

For the ease of gradient computation, the sum of logarithm probability of $p(t)$ is used instead in deriving the loss function. The logarithm probability for an event sequence C can be computed as:

$$L_1 = \sum_{j=1}^{J-1} \sum_{k=0}^{K} \log \lambda_{x_{j+1}}(t_{j+1}) - \int_{t_j}^{t_{j+1}} \lambda_k(s) ds \tag{22}$$

We call L_1 the Hawkes loss term. By maximizing Eq. 22, the probability that event y_j^i happens at time t_j^i in sequence S^i is maximized. Eq. 22 maximizes the intensity function at event arrival times and minimizes the function at non-event time. We notice that even though the Hawkes loss term maximizes the probability of arrival times for each event type, it does not consider the 'competition' for arrivals across different event types. Based on [8], if the arrival time t_j is known, the event type with highest probability of

arrival is $\arg\max_k \lambda_k(t_j)$. By minimizing the cross-entropy function between the distribution of $\{\lambda_1(t_j), ..., \lambda_K(t_j)\}$ and the real event type y_j, the event type prediction at each timestep can be improved. A similar terminology is also considered in [13]. In this sense, the second loss term can be derived as:

$$L_2 = \sum_{j=1}^{J-1} \sum_{k=0}^{K} -\rho_k(t_{j+1}) \log \lambda_k(t_{j+1}) \tag{23}$$

in which $\rho_k(t_j) = 1$ when $y_j = k$ otherwise $\rho_k(t_j) = 0$.

The third loss term deals with the state transition process by attempting to maximize the likelihood of the coupled state sequence that is sampled from the state transition matrix $\phi_{K\times W\times W}$, as:

$$L_3 = \sum_{j=1}^{J-1} \log \phi(y_{j+1}, x_j, x_{j+1}) \tag{24}$$

The overall loss function combines the aforementioned three losses together: L_1 that maximizes the sum of log probability of $p(t)$ in terms of event arrival times; L_2 that maximizes the probability that the correct type of event happens at each arrival time; and L_3 that maximizes the likelihood of the sampled state sequence. The overall loss is calculated as:

$$L = L_1 - \eta_1 L_2 + \eta_2 L_3 \tag{25}$$

in which η_1 and η_2 are weighting coefficients. The training of parameters can be achieved by applying the gradient ascending algorithm. In practice, the loss term L_3 can be separated from the total loss and be trained separately as the parameters involved are isolated from the Hawkes loss term L_1 and cross-entropy term L_2.

Once the neural point process model is trained, the model is able to predict the time and type of next event arrival. Given the density function for next event time, shown in Eq. 21, the expected time with minimum L2 loss can be derived as $\hat{t}_j = \int_{j-1}^{\infty} t * p_j(t) dt$; and given time t_j, the most likely event type would be $\arg\max_k \lambda_k(t_j)/\lambda(t_j)$; or without knowledge of time, the most likely event type would be $\arg\max_k \int_{j-1}^{\infty} \lambda_k(t)/\lambda(t) * p_j(t) dt$.

3.5 LOB Event Stream Simulation

Given a well-trained model, an event-state sequence can be sampled using Ogata's thinning algorithm [24], shown in Algorithm 1. By iteratively incorporating simulated event and state into the input and removing the first event and state in the input to form new inputs, event and state pairs can be constantly generated. This approach contrasts with the LOBRM simulation presented in [27, 28] in the sense that the simulation generated by LOBRM is composed of separate predictions on LOB states, while the intended simulation here is generated as a result of events' interaction and accumulation.

With a model being able to stochastically sample event streams based on its knowledge learned from historical event stream data, a realistic simulation of the LOB data can be achieved. It is worth noting that such a simulation is remarkably different from the mainstream agent-based LOB simulation. The simulation is 'realistic' judging from the fact that the generation of events is from knowledge learned on real data instead of from agents with subjectively

Algorithm 1 Ogata's thinning algorithm for iterative sampling of event-state stream

Input: $(e_1, ..., e_{j-1})$, $(x_1, ..., x_{j-1})$
set the upper bound λ^* for all $\lambda(t)$.
for $k = 0$ to K **do**
  let $t^k \leftarrow t_{j-1}$.
  repeat
    sample $\delta t \sim \text{Exp}(\lambda^*)$, $u \sim \text{Unif}(0, 1)$.
    $t^k \leftarrow t^k + \delta t$.
  until $\lambda_k(t^k)/\lambda^* \geq u$
end for
$t_j \leftarrow \min_k t^k$, $y_j \leftarrow \arg\min_k t^k$, $e_j \leftarrow (t_j, y_j)$
sample $x_j \sim \phi(y_j, x_{j-1})$
return e_j, x_j

set strategies and parameters. Such a realistic simulation of the LOB data allows for an alternative way to validate some empirical findings on the real LOB market. To complete the design of the simulator, apart from the type and time of each event, price and volume information affiliated with the event is also needed. Based on empirical studies, the relative price ($p^a - p^{a(1)} \geq 0$ or $p^{b(1)} - p^b \geq 0$) and volume of submitted orders follow a power law distribution [2, 34]. Thus, a truncated discrete power law distribution with parameter a and boundary b in the form of

$$p(x) = 1/(x+1)^a, \quad x = 0, \ldots, b \tag{26}$$

is used to fit those distributions from real data. Some special settings concerning order type and price include: the order cancellation on a specific price level follows first-in-first-out terminology; market orders appear as a given percentage of submitted orders; orders with price that cross the spread ($p^a - p^{a(1)} < 0$ or $p^{b(1)} - p^b < 0$) appear as a given percentage of submitted limit orders. With these settings, a minimal and realistic simulation of the LOB can be achieved. The quality of the generated LOB data can be judged by exploring whether stylized facts found in real data exist in the simulated data.

4 EXPERIMENTS

4.1 Prediction

4.1.1 Dataset. Experiments are conducted on the real world LOB event stream data of five days' length for three stocks, ticker symbol INTC (Intel), MSFT (Microsoft), and JPM (JP Morgan), provided by the financial data provider LOBSTER.[1] On average there are 0.5 million event updates per trading day per stock. By using a rolling window approach to form times series samples with size $J = 50$, approximately 4 million samples are involved in experiments.

4.1.2 Model Comparison. This section compares testing results for the proposed model against different state-of-the-art models that predict event arrivals. The performance of five models are compared: (1) Hawkes: state-dependent stochastic Hawkes point process model with exponential decaying kernel [22]; (2) LSTM: naïve LSTM model, with no modelling of event intensity rates [16]; (3) SAHP: self-attentive neural Hawkes process that uses the attention mechanism instead of recurrent input structure [13, 32];

[1] A sample dataset can be found at https://lobsterdata.com/info/DataSamples.php

Table 1: Model comparison results. Best performance values underlined.

Model	MSFT			INTC			JPM		
	Neg log prob	Time loss	Acc. type	Neg log prob	Time loss	Acc. type	Neg log prob	Time loss	Acc. type
Hawkes	0.67	1.25	42.67 / 43.12	0.80	1.24	40.10 / 39.48	-0.41	1.04	48.20 / 48.14
LSTM	-	-	47.64	-	-	46.04	-	-	58.05
SAHP	-0.60	1.14	47.73 / 46.92	-0.36	1.08	46.87 / 45.72	-1.72	1.06	59.15 / 57.24
CT-LSTM	-0.81	1.03	46.81 / 46.64	-0.52	1.00	44.22 / 42.99	-2.03	0.91	59.62 / 58.38
PCT-LSTM	<u>-0.87</u>	<u>1.03</u>	<u>49.66</u> / <u>48.98</u>	<u>-0.61</u>	<u>0.97</u>	<u>48.38</u> / <u>46.88</u>	<u>-2.09</u>	<u>0.87</u>	<u>60.97</u> / <u>59.47</u>
PCT-LSTM (non-sd)	-0.81	1.03	48.79 / 48.12	-0.60	0.98	48.16 / 46.43	-2.07	0.88	60.43 / 58.89
PCT-LSTM ($\eta_1 = 1$)	-0.53	1.04	49.11 / 47.90	-0.31	1.04	48.31 / 46.48	-1.57	0.95	60.33 / 57.29

(4) CT-LSTM: neural Hawkes process that uses a single CT-LSTM unit for the modelling of the intensity rates for all types of events [21]; (5) PCT-LSTM: the proposed state-dependent neural Hawkes process with stacked CT-LSTM units to model the intensity rates for different types of events separately; (6) PCT-LSTM (non-sd): a variant of the PCT-LSTM model that does not contain an event-state interaction mechanism; (7) PCT-LSTM ($\eta_1 = 1$): a full PCT-LSTM model trained with a mixed loss function that contains a cross-entropy term. The same metrics used in [21] are adopted to measure performance: (1) negative logarithm likelihood $p(t)$ per event; (2) next event time prediction accuracy, measured as the absolute distance between real and predicted time after common logarithm operation, which is necessary as the ground truth for time spans from several microseconds to seconds; (3) next event type prediction accuracy in percentage, when the next even time is known and not known. To enable parallel comparison, both event and state are included as inputs for all models.

In terms of parameter settings for all models, all linear layers in LSTM units are set with two layers, 16 units and Tanh activation, except for that the Softplus activation is used for the linear layers for deriving decay coefficient (>0). The attention module in SAHP is of 4 heads. The embedding layers for event type and state indicator are also of two layers, 16 units and Tanh activation. The decoding layer for intensity rates is of one layer, 16 units and Softplus activation that forces outputs to be positive. Optimizer is RMSprop, learning rate is 2e-3 and all models are trained 200 iterations.

Testing results are shown in Table 1. We find that the performance of neural network family models is better than the pure stochastic Hawkes model in terms of all criteria, as a stochastic Hawkes model is controlled by only a few parameters. This finding is in accordance with existing literature [21, 32]. The LSTM model does not contain a point process terminology to derive the intensity rate, thus only an event type prediction is illustrated here. The performance improvements brought by SAHP or CT-LSTM are mixed, which differs from the conclusion drawn in [32] that SAHP is better than CT-LSTM. The main reason for this contradiction might be the time series input in this research is comparatively short with a length of 50, whilst the time series studied in [32] has a maximum length of over 200, and the self-attention module is more effective for long sequences. PCT-LSTM achieves the overall best performance compared with all alternatives, with the lowest negative log likelihood and time loss, and the highest event type prediction accuracy. This improvement validates that the separate modelling for intensity rates for different event types through the PCT-LSTM unit is effective in multi-variate point process modelling.

The learned state transition matrix is partially shown in Eq. 27:

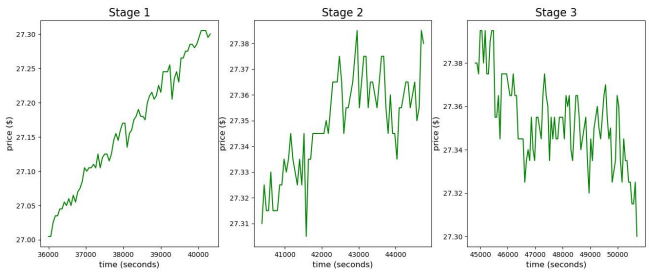
$$\left(\begin{pmatrix} 0.93 & 0.07 & <0.01 \\ \approx 0.01 & 0.99 & \approx 0.01 \\ <0.01 & 0.02 & 0.97 \end{pmatrix}, \quad \begin{pmatrix} 0.97 & 0.03 & <0.01 \\ \approx 0.01 & 0.99 & \approx 0.01 \\ <0.01 & 0.07 & 0.93 \end{pmatrix} \quad ... \right) \tag{27}$$

We see that inertia exists in market condition transition, with more than 90% possibility that the market condition remains the same as the condition at last timestep. Distributions of state transition shift from $\phi[0]$ to $\phi[1]$ when the arrived event type changes from a bid submission to an ask submission. This indicates the existence of a feedback loop, in which the current state influences the arrival of next event type and conversely the arrived event type influences the distribution of state transitions. Also, a certain degree of asymmetry exists in state transitions: when arrived event type is a bid submission, the probability of state transforming from an ask market to a neutral market (0.07) is higher than transforming from a bid market to neutral market (0.02); yet, when arrived event type is an ask submission, the probability of transforming from an ask market to neutral market (0.02) is lower than transforming from a bid market to a neutral market (0.07). This asymmetry is intuitive.

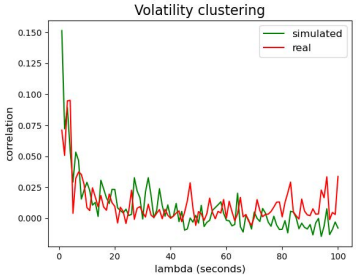
4.1.3 Discussions on the Model Settings. The superiority of stacking CT-LSTM units has been verified in the last section. In this section we consider the performance improvement brought by the inclusion of a state indicator, and the impact of the modified cross-entropy loss function. Results are presented in the bottom two rows of Table 1. Comparing the results from PCT-LSTM (non-sd) with the results from PCT-LSTM, we find all criteria are inferior when state information is removed from input, indicating that the market condition indeed helps the modelling of intensity rate evolution and therefore improves prediction. We then compare the performance of PCT-LSTM ($\eta_1 = 1$) with PCT-LSTM. We find that the inclusion of the cross-entropy loss term does not help improve prediction performance of the PCT-LSTM. However, we find this cross-entropy term indeed facilitates prediction for SAHP across all criteria, as indicated in [13], and the testing results of SAHP in Table 1 are obtained by including this term. In this sense, we can flexibly choose whether to include the cross entropy loss term according to which type of model we are implementing.

4.2 Simulation

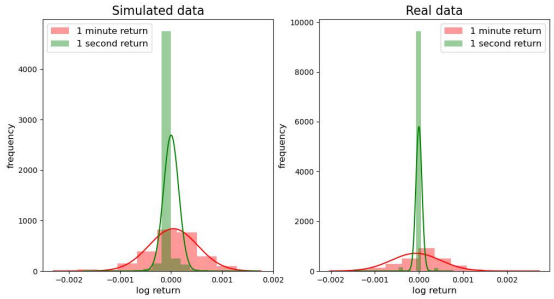
4.2.1 Parameter Setting. This section presents how to generate a simulated LOB of five price levels for a small-tick stock using the



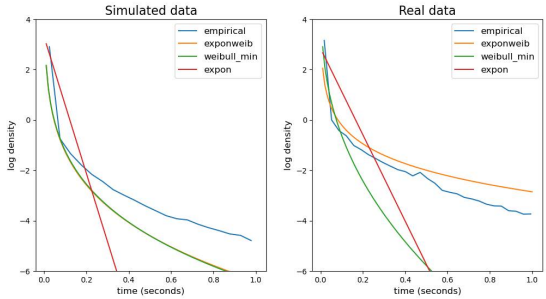
(a) Mid-price evolution of simulation. Percentage of market orders is controlled to evolve market from a buy to a neural to a sell market.



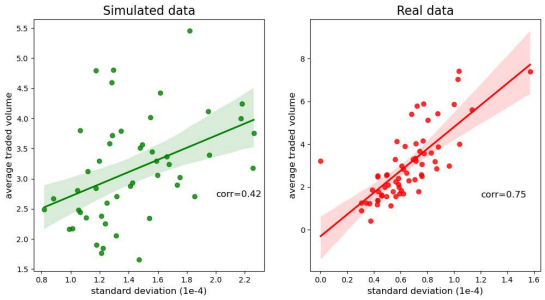
(b) Volatility clustering: sim (green); real (red). Positive correlation over short time lags, correlation decays to zero as lag times increase.



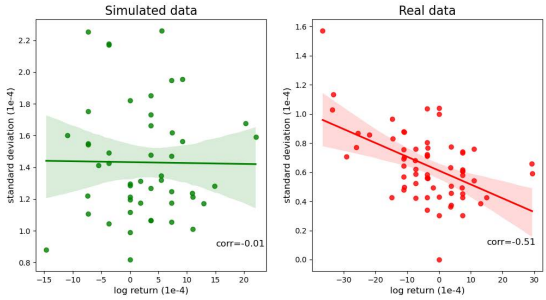
(c) Log returns: Log returns are normally distributed for sim (left) and real (right) data. As sampling frequency increases, kurtosis increases.



(d) Inter-arrival times: Exponentiated Weibull distribution is found to be the best fit for both simulated (left) and real (right) data.



(e) Volatility-volume: Positive correlation of returns standard deviation and average traded volume in sim (left) and real (right) data.



(f) Volatility-returns: Returns standard deviation and log returns are negatively correlated in real (right) data but not in sim (left) data.

Figure 3: Stylized facts verification for both simulated and real LOB data.

Table 2: Parameter settings for simulated LOB. Samples for learning price and volume distribution are truncated at 95% to remove outliers. Power law coefficients a are learned from the ask side one-day INTC data; sp indicates bid-ask spread.

	Price ≥ 0	Volume	Market pct.	Price $= -1$ pct.
$sp = 1$	$a = 1.5$	$a \in [1.2, 1.5]$	0.025	-
$sp > 1$	$a = 4.3$	$a \in [0.4, 1.9]$	0.0025	0.1

methodology described in Sec. 3.5. Event type and time are sampled from a well-trained neural Hawkes model that models event intensity rates, and the price and volume information of each order are sampled from empirical discrete power law distributions learned on real data using maximum likelihood estimation (MLE). Parameters involved in generation are illustrated in Table 2. Other empirical characteristics of the LOB of small-tick stocks, together with several empirical observations on the three LOB datasets studied in this paper, are also considered in simulation, including (1) orders tend to densely arrive around top price levels and price intervals between the top five price levels at both bid and ask side are always one tick (>99% of LOB snapshots), and (2) bid-ask spread are dominantly one tick (>50% of LOB snapshots).

4.2.2 Stylized Facts Verification. A LOB containing 400k updates, spanning approximately 4 hours, is simulated and the existence of

several stylized facts is investigated and compared with the real LOB data.[2] For the selection of stylized facts, we consider the research conducted in [18, 30]. Stylized facts regarding the order stream, together with the resulting intraday LOB time series, are selected for investigation. These include: the evolution of mid-price; the absence of autocorrelation in returns; the aggregational normality of returns; volatility clustering; the positive relation between trade volume and volatility; the negative relation between returns and volatility; and the exponential distribution of inter-arrival times.

Some other stylized facts, including the power law distribution of order price and volume, are embedded in the methodology to generate simulated data so we do not investigate their existence. We also disregard several stylized facts regarding interday and seasonal patterns, as the simulator is specifically designed for generating intraday LOB data. Finally, several facts regarding the pattern of market trades are not considered, as market orders are designed to appear randomly as a percentage of limit orders. More stylized facts can be investigated in future extensions of the simulator when more advanced settings are introduced.

First, we illustrate some notations involved in calculation. At time t, the mid-price is $m_t = \frac{(p_t^a + p_t^b)}{2}$. Define a time scale Δt which can range from milliseconds to hours, the return is usually calculated in logarithmic form as $r_{t,\Delta t} = \ln m_{t+\Delta t} - \ln m_t$. Slice LOB time series into small non-overlapping time intervals with length τ, then V_τ is the traded volume within τ and $\sigma_{\tau,\Delta t}$ is the standard deviation of log returns in scale Δt over interval τ.

Mid-price evolution. This fact indicates that the evolution of the mid-price for liquid financial assets essentially follows a random walk and the mid-price remains volatile during any trading period [18]. As shown in Fig. 3a, the mid-price fluctuates across the simulated time period instead of being still. Specifically, the simulation was manually conducted through three periods: buy market, neutral market, and sell market. In a buy market the percentage of market buy as a fraction of limit buy is doubled, and in a sell market the percentage of market sell as a fraction of limit sell is doubled. We can see clearly from the figure that the market sequentially exhibits sharp upwards (stage 1), mild upwards (stage 2; note that y-axis has much smaller range) and downwards (stage 3) trends. The market retains a slight upwards trend during the neutral market (stage 2) due to the fact that the bid submissions remain approximately 1.2 times more active than ask submissions during the simulation period. During each specific period, the resulting price also shows high volatility.

Volatility clustering. This fact indicates that high volatility events tend to cluster in time. The function that used to quantify this feature is $f(\tau) = \text{corr}(r^2_{t+\tau,\Delta t}, r^2_{t,\Delta t})$ [6]. This function remains positive and shows a downward tendency when delay in time τ increases. Fig. 3b confirms this property is exhibited in both simulated and real data ($\Delta t = 1$ sec), and is similar to that shown in [6].

Normality of log returns. This fact indicates that the distribution of asset log returns follows a normal distribution; at the same time, when sampling frequency changes from low to high, the tail of the distribution becomes lower [30]. One way to verify this fact is to observe an increasing kurtosis as sampling frequency increases. Fig. 3c shows the distribution of $r_{t,\Delta t}$ when the sampling frequency is high ($\Delta t = 1$ sec) and low ($\Delta t = 1$ min) can both be fitted with a Gaussian distribution. The distribution of high frequency return shows higher kurtosis and low tails (kurtosis > 10), while the low frequency return shows lower kurtosis and high tails (kurtosis ≈ 1). This phenomenon is found in both simulated and real data.

Event inter-arrival time. This fact indicates that the distribution of event inter-arrival times follows an exponential or a Weibull distribution [1]. Fig. 3d presents the empirical density curve of time distribution (blue line) fitted with exponential (red), Weibull (green), and exponentiated Weibull (orange) distributions using MLE. Both simulated and real data have the best goodness of fit with the exponentiated Weibull distribution (for simulated data, Weibull and exponentiated Weibull curves are almost identical).

Volatility/volume +ve correlation. This facts indicates that the standard deviation of log return $\sigma_{\tau,\Delta t}$ and traded volume V_τ have positive correlation, in the sense that trading activities of large volumes are likely to introduce higher volatility [4]. In [30], the mean value for correlation coefficient for historical data lies around 0.4. Fig. 3e plots the standard deviation against average trade volume every 5 minutes. The correlation coefficient for simulated and real data are 0.48 and 0.75 respectively, which confirm the existence of this fact.

Volatility/returns -ve correlation. This fact, also called the 'leverage effect' [3], indicates that volatility and asset returns are negatively correlated. The volatility commonly used here is the implied volatility derived from a particular volatility index (e.g., the VIX) or the price of financial derivatives. Here, as we do not have access to the implied volatility of assets, we use the standard deviation of log return $\sigma_{\tau,\Delta t}$ as an alternative. Following [14], we investigate the linear correlation between volatility and log return. Fig. 3f shows that real data exhibits significant negative correlation between log return and standard deviation of return, with a correlation coefficient of 0.51. As for the simulated data, the correlation is not significantly different from zero. The simulated data fails to synthesise this stylized fact.

Absence of autocorrelation. This fact, a reflection of the 'efficient markets' hypothesis [10], indicates that price movements for liquid assets do not exhibit significant autocorrelation. The absolute value for autocorrelation of log return time series $f(\tau) = \text{corr}(r_{t+\tau,\Delta t}, r_{t,\Delta t})$ was found to be lower than 0.2, and as time lag increases the coefficient converges to zero [6]. This phenomenon is confirmed in both simulated and real data and the autocorrelation coefficient fast decays to near zero after around 10 seconds lag for 1 second return data (no figure shown).

[2] Simulated data can be found at https://github.com/ashimoo/SDPNHP

5 CONCLUSION AND FUTURE WORKS

We have presented the state-dependent parallel neural Hawkes process for limit order book event stream prediction and simulation.

This research is the first to systematically investigate the utilization of neural Hawkes models in the field of LOB modelling. The main contributions are: (1) introducing an event-state interaction mechanism into a neural Hawkes process that enables improved prediction performance and efficient sampling; (2) improving the structure of the CT-LSTM module used in the original neural Hawkes process to a PCT-LSTM module with stacked units, in which the evolution of intensity rates for different types of events are governed by separate kernels; (3) extending the task from LOB event prediction to realistic LOB data simulation by combining a well-trained neural Hawkes model for event type and time sampling, and a rule-based generator for order price and volume information generation. Through experiments, we find the proposed model outperforms other mainstream point process models in the specific task of LOB event and time prediction. Meanwhile, the simulated LOB data generated from a well-trained neural Hawkes model using the proposed methodology exhibits several stylized facts shown to also exist in real LOB data. This way of generating simulated LOB data provides a new alternative to existing methods. As a whole, this research has the potential to benefit both financial microstructure researchers and high frequency trading practitioners.

Limitations and Future Works. The performance of the state-dependent parallel neural Hawkes process is only validated on the task of LOB event prediction. Future studies can be carried out to test whether the proposed model can also benefit other types of tasks. On the other hand, several settings of the simulator are simplistic, for instance the occurrence of market orders and the relative price distribution of arriving orders. For enhanced realism, More advanced and realistic features will be introduced into the simulator. The proposed neural Hawkes-based simulator can also be combined with agent-based simulators to develop a hybrid simulator that enjoys unique characteristics from both methodologies.

ACKNOWLEDGMENTS

ZS's PhD is funded by a China Scholarship Council / University of Bristol joint-scholarship. JC is sponsored by Refinitiv.

REFERENCES

[1] Frédéric Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. 2016. *Limit order books.* Cambridge University Press, Cambridge, UK.

[2] Jean-Philippe Bouchaud, Marc Mézard, and Marc Potters. 2002. Statistical properties of stock order books: empirical results and models. *Quantitative finance* 2, 4 (2002), 251.

[3] Jean-Philippe Bouchaud and Marc Potters. 2001. More stylized facts of financial markets: leverage effect and downside correlations. *Physica A: Statistical Mechanics and its Applications* 299, 1-2 (2001), 60–70.

[4] Olivier Brandouy, Angelo Corelli, Iryna Veryzhenko, and Roger Waldeck. 2012. A re-examination of the "zero is enough" hypothesis in the emergence of financial stylized facts. *Journal of Economic Interaction and Coordination* 7, 2 (2012), 223–248.

[5] Alvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal. 2018. Enhancing trading strategies with order book signals. *Applied Mathematical Finance* 25, 1 (2018), 1–35.

[6] Rama Cont. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance* 1, 2 (2001), 223.

[7] Rama Cont, Sasha Stoikov, and Rishi Talreja. 2010. A stochastic model for order book dynamics. *Operations research* 58, 3 (2010), 549–563.

[8] Daryl J Daley and David Vere-Jones. 2003. *An introduction to the theory of point processes. Volume I: elementary theory and methods.* Springer, New York, NY.

[9] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, New York, NY, 1555–1564.

[10] Eugene F Fama. 1991. Efficient capital markets: II. *The journal of finance* 46, 5 (1991), 1575–1617.

[11] Federico Gonzalez and Mark Schervish. 2017. Instantaneous order impact and high-frequency strategy optimization in limit order books. *Market Microstructure and Liquidity* 3, 02 (2017), 1850001.

[12] Martin D. Gould, Mason A. Porter, Stacy Williams, Mark McDonald, Daniel J. Fenn, and Sam D. Howison. 2013. Limit order books. *Quantitative Finance* 13, 11 (2013), 1709–1742.

[13] Yulong Gu. 2021. Attentive Neural Point Processes for Event Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35(9). AAAI Press, Palo Alto, CA, 7592–7600.

[14] Qian Han, Biao Guo, Doojin Ryu, and Robert I Webb. 2012. Asymmetric and negative return-volatility relationship: The case of the VKOSPI. *Investment Analysts Journal* 41, 76 (2012), 69–78.

[15] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Andreas GF Hoepner, David McMillan, Andrew Vivian, and Chardin Wese Simen. 2021. Significance, relevance and explainability in the machine learning age: an econometrics and financial data science perspective. *The European Journal of Finance* 27, 1-2 (2021), 1–7.

[18] Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael Wellman. 2020. Generating realistic stock market order streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34(01). AAAI Press, Palo Alto, CA, 727–734.

[19] Xiaofei Lu and Frédéric Abergel. 2018. High-dimensional Hawkes processes for limit order books: modelling, empirical analysis and numerical calibration. *Quantitative Finance* 18, 2 (2018), 249–264.

[20] Frank McGroarty, Ash Booth, Enrico Gerding, and VL Raju Chinthalapati. 2019. High frequency trading strategies, market fragility and price spikes: an agent based model perspective. *Annals of Operations Research* 282, 1 (2019), 217–244.

[21] Hongyuan Mei and Jason M Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. *Advances in Neural Information Processing Systems* 30 (2017), 6757–6767.

[22] Maxime Morariu-Patrichi and Mikko S Pakkanen. 2021. State-dependent Hawkes processes and their application to limit order book modelling. *Quantitative Finance* 0, 0 (2021), 1–21. https://doi.org/10.1080/14697688.2021.1983199.

[23] Ioane Muni Toke and Nakahiro Yoshida. 2017. Modelling intensities of order flows in a limit order book. *Quantitative Finance* 17, 5 (2017), 683–701.

[24] Yosihiko Ogata. 1981. On Lewis' simulation method for point processes. *IEEE transactions on information theory* 27, 1 (1981), 23–31.

[25] Christine A Parlour. 1998. Price dynamics in limit order markets. *The Review of Financial Studies* 11, 4 (1998), 789–816.

[26] Ioanid Roşu. 2009. A dynamic model of the limit order book. *The Review of Financial Studies* 22, 11 (2009), 4601–4641.

[27] Zijian Shi and John Cartlidge. 2021. The Limit Order Book Recreation Model (LOBRM): An Extended Analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Cham, Switzerland, 204–220.

[28] Zijian Shi, Yu Chen, and John Cartlidge. 2021. The LOB Recreation Model: Predicting the Limit Order Book from TAQ History Using an Ordinary Differential Equation Recurrent Neural Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35(1). AAAI Press, Palo Alto, CA, 548–556.

[29] Eric Smith, J Doyne Farmer, László Gillemot, and Supriya Krishnamurthy. 2003. Statistical theory of the continuous double auction. *Quantitative finance* 3, 6 (2003), 481.

[30] Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Balch. 2020. Get real: Realism metrics for robust limit order book market simulations. In *Proceedings of the First ACM International Conference on AI in Finance*. ACM, New York, NY, 1–8.

[31] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31(1). AAAI Press, Palo Alto, CA, 1597–1603.

[32] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-Attentive Hawkes Process. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). MLResearchPress, 11183–11193.

[33] Yizhou Zhang, Karishma Sharma, and Yan Liu. 2021. VigDet: Knowledge Informed Neural Temporal Point Process for Coordination Detection on Social Media. *Advances in Neural Information Processing Systems* 34 (2021), 14 pages.

[34] Ilija Zovko and J Doyne Farmer. 2002. The power of patience: a behavioural regularity in limit-order placement. *Quantitative finance* 2, 5 (2002), 387.