Dynamic Calibration of Order Flow Models with Generative Adversarial Networks

Rama Cont
rama.cont@maths.ox.ac.uk
Mathematical Institute, University of Oxford
Oxford, UK

Mihai Cucuringu
mihai.cucuringu@stats.ox.ac.uk
Department of Statistics, University of Oxford
Oxford, UK
The Alan Turing Institute
London, UK

Jonathan A. Kochems[†]
jonathan.a.kochems@jpmorgan.com
JP Morgan
London, UK

Felix Prenzel[*][†]
felix.prenzel@maths.ox.ac.uk
Mathematical Institute, University of Oxford
Oxford, UK
JP Morgan
London, UK

ABSTRACT

Classical models for order flow dynamics based on point processes, such as Poisson or Hawkes processes, have been studied intensively. Often, several days of limit border book (LOB) data is used to calibrate such models, thereby averaging over different dynamics - such as intraday effects or different trading volumes.

This work uses generative adversarial networks (GANs) to learn the distribution of calibrations – obtained by many calibrations based on short time frames. The trained GAN can then be used to generate synthetic, realistic calibrations based on external conditions such as time of the day or volatility. Results show that GANs easily reproduce patterns of the order arrival intensities and can fit the distribution well without heavy parameter tuning.

The synthetic calibrations can then be used to simulate order streams which contain new dynamics such as temporary drifts, different volatility regimes, but also intra-day patterns such as the commonly observed U-shape that reflects stylized behaviour around open and close of market hours.

KEYWORDS

Limit Order Books, Synthetic Data, Generative Networks, Machine Learning, Finance, Market Microstructure

ACM Reference Format:
Rama Cont, Mihai Cucuringu, Jonathan A. Kochems, and Felix Prenzel. 2022. Dynamic Calibration of Order Flow Models with Generative Adversarial Networks. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3533271.3561777

1 INTRODUCTION

Limit order books (LOBs) provide the mechanism via which most stock trading is organized at most exchanges. For many applications, it is not only of interest to simulate the LOB state for a certain period of time, but also to model order-by-order data (the so-called MBO - market by order data). Such order flow models have been built under heavy use of point processes [3, 9, 14, 25, 30]. These point processes then produce every event affecting the LOB state. In most of the literature, order-by-order data of several days is used to obtain one single calibration of the model. The obtained parameters are then kept fixed which leads to stationary dynamics in the modeled order flow.

This study aims to improve this shortcoming of one single calibration by making use of existing approaches to simulate order flow and improving those with a hierarchical model on top of them. In particular, we build a probabilistic model based on generative adversarial networks (GANs), first introduced in [11], to learn to generate new, unseen calibrations of different order flow models. This allows to introduce new, realistic dynamics into existing models, while preserving their theoretical properties and, in some cases, interpretability.

We show in this study that GANs can learn high-dimensional distributions of calibrations of smaller time periods. Furthermore, these can be efficiently generated conditioned on other covariates, such as time-of-day or the current value of the volatility index (VIX). After training, the synthetic – but realistic – calibrations can be subsequently used to simulate order streams using the baseline model. This induces stylized behaviour like temporary drifts, different volatility regimes and intraday patterns one does not observe with one single calibration.

2 RELATED WORK

Ever since exchanges have started using LOBs as the mechanism for electronic trading, vast amounts of publicly available historical data have led to a wealth of studies analyzing their empirical properties

[*] Corresponding author
[†] Opinions expressed in this paper are those of the authors, and do not necessarily reflect the view of JP Morgan.

ICAIF '22, November 2–4, 2022, New York, NY, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561777

[6, 8], modeling their dynamics [2, 9, 17] or predicting price moves [29, 35]. Comprehensive introductions into the field are given in [1, 12].

One main line of research is to build models for the order flow of LOBs. This is to better understand their dynamics and to obtain analytical quantities for certain properties of the order book such as volatility or the expected time for queue depletion. [30] is among the first to use independent Poisson processes to model the inter-arrival times of different orders. [9] builds a stochastic model for limit order books also based on Poisson processes, allowing for different intensities depending on the depth of the book and derive several analytic quantities, such as the probability that the mid price increases (rather than decreases) or the probability of execution before mid price movement. [1] uses independent Poisson processes to simulate the order book assuming order sizes to follow a log-normal distribution. [14] proposes a queue reactive model in which intensities depend on the current state of the order book. They claim the model is able to reproduce realistic market impact.

To account for dependencies between the arrivals of limit orders, in particular self and cross-excitation of certain events, [2, 3] use Hawkes processes to model inter-arrival times of orders. Hawkes processes, incorporating an excitation kernel, are able to capture the salient clustering effect of orders, especially after the occurrence of a market order. [3] also studies the long-term behaviour of the LOB modeled with Hawkes process dynamics. The use of state-dependent Hawkes processes for LOBs is introduced in [25] to account for the different dynamics depending on the spread or the order imbalance. [19] uses non-linear Hawkes processes to model the order flow.

Also agent-based models have been used to model the dynamics of different agents in the market and then simulate the joint order flow in [7, 27]. [27] uses a decomposition of traders previously obtained by an analysis of the flash crash [15]. [7] provides a general frame work for this type of event stream modeling. However, in agent-based modeling it is often difficult to define the agents and obtain a calibration which leads to realistic effects of the synthetic data.

In recent years during the advent and fast development of many machine learning approaches, generative adversarial networks (GANs) [11] have become famous for probabilistic modeling. They are best known for their success in image [11, 28] and speech generation [10]. The well known training instability and mode collapses of GANs have led to plenty of research and new specifications to improve their convergence behaviour [4, 13, 22, 23]. The most famous specification probably being the Wasserstein GAN leveraging the Wasserstein distance to obtain a better training process. In order to condition generated samples on external states Conditional GAN is introduced in [24] in which the model obtains additional information on what kind of sample it shall produce.

The success of GANs has led to plenty of research studies concerning (financial) time series, applying and designing different architectures to problems from market predictions [36], fine tuning of trading strategies [16] and most importantly time series generation [18, 20, 26, 31–34]. Univariate and multivariate time series are generated in [20, 32, 34]. [33] uses GANs in order to generate volatility surfaces. A new conditional GAN based on signatures is presented in [26]. The application of GANs to limit order books is more scarce, event streams of LOBs are generated in [18].

In this study, we attempt to combine data-driven methods and commonly known LOB models – in particular Poisson order flow – in order to obtain the best of both worlds. This line of work is in the spirit of [21], which uses GANs to sample realistic correlation matrices.

3 LIMIT ORDER BOOKS AND EVENT STREAM MODELING

This section provides a brief introduction to limit order books and then presents the Poisson order flow as underlying model.

3.1 Limit order books (LOB)

We start by introducing the mathematical object typically modeled via point processes. First, a limit order has to be defined.

Definition 3.1. A limit order (LO) $x = (t, p, q, \alpha)$ is characterized by the following properties

- an arrival time $t \in \mathbb{R}^+$,
- a price $p \in \delta\mathbb{N}$ which is a multiple of the price tick $\delta > 0$,
- a quantity $q \in \mathbb{Z} \setminus \{0\}$ with $q > 0$ denoting buy orders and $q < 0$ denoting sell orders,
- the identity $\alpha \in A$ of the agent submitting the order.

A limit order is considered "outstanding" as long as it has neither been cancelled nor fully executed.

As public data from LOBSTER[1] is used, the assumption is made that A – the set of agents – only contains one agent. Hence the LOB only has one order source.

With $\mathcal{M}(\mathbb{R}_+ \times \delta\mathbb{N} \times A)$ as the (vector) space of signed measures on $\mathbb{R}_+ \times \delta\mathbb{N} \times A$, the LOB can be described as a signed measure

$$\begin{aligned} &\mu : \mathbb{R}_+ \times \delta\mathbb{N} \times A \longrightarrow \mathbb{Z}, \\ &([0,T], \{p\}, \{\alpha\}) \mapsto \mu([0,T], \{p\}, \{\alpha\}) \end{aligned} \tag{1}$$

where $\mu([0,T] \times \{p\} \times \{\alpha\})$ denotes the outstanding volume of orders submitted between time 0 and T at price p by α.

The queue size Q_p, is then the cumulative outstanding quantity of the orders submitted between $[0,T]$, $Q_p = \sum_{\alpha \in A} \mu([0,T], \{p\}, \alpha)$ which is the object modeled in this study. As per the single LOs, $Q_p < 0 (> 0)$ corresponds to buy (sell) queues. Furthermore

$$p_b = \sup_p \{p > 0, Q_p < 0\} \text{ and } p_a = \inf_p \{p > 0, Q_p > 0\}, \tag{2}$$

correspond to the best bid and ask price. The highest (lowest) price a market participant is willing to buy (sell) the corresponding stock. As [5] argues, the frequency of events occurring at a certain price level mainly depends on their relative price. That is why Q_p is often described in relative terms, i.e. i ticks away from the best opposite price

$$Q_i^a = Q_{p_b + i \cdot \delta} \text{ and } Q_i^b = Q_{p_a - i \cdot \delta}. \tag{3}$$

Note that, if there is a spread $s = p_a - p_b = 2\delta$, i.e. one empty level, then $p_b + \delta = p_a - \delta$, hence Q_1^a and Q_1^b refer to the same queue.

The LOB then evolves over time being affected by order events. We will hence consider the quantities at a certain point in time $t \in \mathbb{R}^+$. The order types which are widely modeled in the literature

[1] https://lobsterdata.com

are limit order, market order and cancellation order, altogether driving the main dynamics in a LOB.

(1) **Limit order submission:** An agent submits a limit order x at time t and price $0 < p < p_a(t)$ for bid orders and $p_b(t) < p < \infty$ for ask orders respectively. From time t on, order x is part of the limit order book. The change of the queue size is given by $\Delta Q_p(t) = q$.
(2) **Limit order cancellation:** An agent cancels their existing order x. Since the cancellation sets the order size to $q = 0$, x is not part of the LOB anymore. The change of the queue size at order price p reads $\Delta Q_p(t) = -q$.
(3) **Execution of a limit order via market order:** An agent submits a buy market order of quantity q at time t. This quantity is then immediately executed against sell limit orders on the ask side. There may be hidden limit orders which are not considered here as we are interested in modeling the lit market. In most exchanges, this is done via price-time priority. E.g. a market order of $q = 1$ would be executed against the oldest limit order (front of queue) at the best available price. If the market depth is smaller than the order size (e.g. $Q_{p_a}(t) < q$), the queue is depleted and the remaining quantity $q - Q_{p_a}(t)$ is executed against limit orders from the succeeding price levels (a phenomena often referred to as market sweep orders).

3.2 Poisson Order Flow

[1, 9, 30] present models assuming the arrival of Limit orders to follow independent Poisson processes. For each order type, different order arrival rates are assumed depending on the distance to the best opposite price:

(1) Limit orders arrive with intensity $\lambda_i^{L,\star}$ at side $\star \in \{a/b\}$ and $i \in \{1, \ldots k\}$ ticks away from the best opposite price.
(2) Cancellation orders arrive with intensity $\lambda_i^{C,\star}(t)$ at side $\star \in \{a/b\}$ and $i \in \{1, \ldots k\}$ ticks away from the best opposite price. Note that
$$\lambda_i^{C,\star}(t) = \lambda_i^{C,\star} \cdot Q_i^\star(t). \tag{4}$$
In essence, the cancellation rate $\lambda_i^{C,\star}(t)$ increases with increasing queue size which avoids the explosion of $Q_i^\star(t)$.
(3) Market orders arrive with intensity $\lambda^{M,\star}$ at side $\star \in \{a/b\}$.

For all intensities, $\lambda \in \mathbb{R}_{\geq 0}$ must hold. An exemplary visualization of the model, along with the locations in the book to which the different $\lambda_i^{L,\star}$ rates pertain to, is depicted in Figure 1.

The whole set of order arrival intensities and thus parameterization of the model is given by

$$\Lambda = \{\lambda^{M,a}, \lambda^{M,b}\} \cup \{\lambda_i^{L,\star}, \lambda_i^{C,\star} | \star \in \{a, b\}, i \in \{1, \ldots, k\}\} \in {\mathbb{R}_{\geq 0}}^{4k+2} \tag{5}$$

$$= (\lambda_1, \ldots, \lambda_i, \ldots, \lambda_{4k+2}) \in {\mathbb{R}_{\geq 0}}^{4k+2}. \tag{6}$$

The second part of (5) is a reindexing of the first part for convenience later on. For any variable $x_i \overset{iid}{\sim} Po(\lambda_i)$ it holds that $\sum_i x_i \sim Po(\sum_i \lambda_i)$. Hence, the counting process of all orders is

$$Po\left(\sum_{\lambda \in \Lambda} \lambda\right) \tag{7}$$

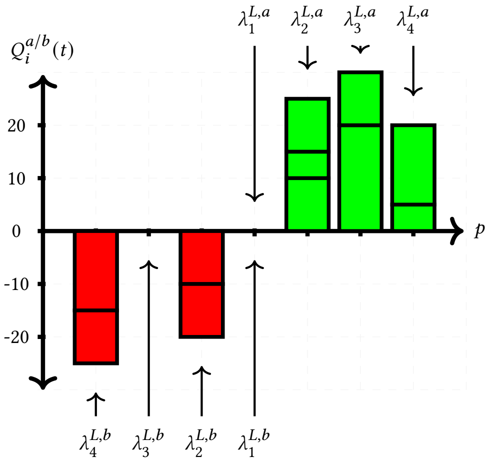

Figure 1: Exemplary order book state with indications where arrival rates $\lambda_i^{L,\star}$ currently "sit".

distributed and inter-arrival times between two consecutive orders follow an exponential distribution with rate $\frac{1}{\sum_{\lambda \in \Lambda} \lambda}$. The single events depend on the relative size of arrival rates. For example, the probability of the next event drawn of the process $Po\left(\sum_{\lambda \in \Lambda} \lambda\right)$ to be a buy market order is equal to $\frac{\lambda^{M,a}}{\sum_{\lambda \in \Lambda} \lambda}$.

While in [9, 30] a constant order size (e.g. the mean of all limit orders) is assumed, [2] uses a log-normal distribution for the numerical simulation of the LOB. Alternatively, an empirical distribution may be used. Using a unit order size as in [9] allows to derive various interesting micro scale quantities using conditional Laplace transforms. [9] in particular derives the probability for a price increase, an order to be executed before a price move, and the probability of two orders at bid/ask being executed before a price moves. [2] shows on a more macro level that the resulting price paths created with Poisson order flow converge to a diffusion process with Gaussian noise, e.g. normal distributed returns.

It is important to keep in mind that level i indicates the distance in ticks from the **best opposite price**, e.g. $p_a(t) - i \cdot \delta$ or $p_b(t) + i \cdot \delta$ respectively. In case the spread, $p_a(t) - p_b(t)$, exceeds δ (e.g., $p_a(t) - \delta = p_b(t) + \delta$), $\lambda_1^{L,a}$ and $\lambda_1^{L,b}$ "sit" at the same price level. The case when $s(t) = 2\delta$ is visualised in Figure 1. The overlap ensures that both bid and ask orders may be posted in the empty levels between $p_b(t)$ and $p_a(t)$.

Regarding the esitmation of the Poisson order flow, estimator for limit and market order submissions is given by

$$\hat{\lambda}_i^{L,\star} = \frac{N_{i,T}^{L,\star}}{T} \text{ and } \hat{\lambda}^{M,\star} = \frac{N_T^\star}{T}. \tag{8}$$

$N_{i,T}^{L,\star}$ and $N_T^{M,\star}$ denote the limit (market) number of orders in the given time period of length T. The intensity of cancellation orders is proportional to the queue size as each order as is assumed to have equal cancellation probability. The estimate is thus scaled by the average queue size

$$\hat{\lambda}_i^{C,\star} = \frac{N_{i,T}^{L,\star}}{T \cdot \overline{Q^{\star}}_i}. \tag{9}$$

We refer to [1] chapter 9 for the detailed description of the simulation and estimation of the parameters.

4 GENERATING REALISTIC CALIBRATIONS

Typical order flow models such as the Poisson order flow are often estimated using several days of data, see e.g. [1, 9]. Once estimated, Λ generally remains fixed. This leads to stationary modeling of order flow, regardless of, for instance, the time of the day or the volatility of the given day. The flow purely depends on the calibration in Equation (5).

Our present work suggests to generate "synthetic" calibrations in order to induce different, unseen but realistic dynamics, thereby rendering the order flow non-stationary. Instead of estimating order intensities for an entire time horizon (e.g. one day or month as done in [1]), one may estimate many calibrations over smaller time periods in order to obtain an entire data set of calibrations with distribution $\mathbb{P}_r(\Lambda)$. This consequently entails the assumption that while markets are not stationary over an entire day, they may be more stationary for a short time interval.

The aim then is to approximate this distribution and draw samples from it. To this end, one may suggest a parametric distribution with a certain correlation structure. This study, instead, leverages generative adversarial networks (GANs) to learn generating calibrations without any assumptions on the form of distribution.

As introduced in [11], a GAN consists out of two models (generally neural networks):

(1) Generator (g): uses a noise seed from e.g. $Z \in \mathbb{R}^z, Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_z)$ to map it into the sample space of Λ
(2) Discriminator (d): gets real samples and "fake" samples – generated by g – with the aim to distinguish between those.

One design for the g is a simple unconditioned case

$$\Theta^{(g)} \times \mathbb{R}^z \rightarrow \mathbb{R}_{\geq 0}^{4k+2} \tag{10}$$

$$(\theta^{(g)}, Z) \mapsto g_{\theta^{(g)}}(Z) = \Lambda. \tag{11}$$

$\Theta^{(g)}$ is the parameter space of a neural network and $\mathbb{R}^Z$ the space of the noise input.

A generator as in Equation (10) is unconditional. Yet, the calibrations potentially differ substantially depending on variables such as time of day, momentum or volatility in the market. Conditional GAN [24] can be applied to condition the generated distribution on external (market) conditions, e.g.

$$\Theta^{(g)} \times \mathbb{R}^z \times \mathbb{R}^s \rightarrow \mathbb{R}_{\geq 0}^{4k+2}, \tag{12}$$

$$(\theta^{(g)}, Z, S) \mapsto g_{\theta^{(g)}}(Z, S) = \Lambda. \tag{13}$$

The vector $S \in \mathbb{R}^s$ is used by the generator to condition the generated calibrations of the order flow. The time of day and the state of the volatility index VIX of the trading day are used in this example, i.e. $S = (time, vix), S \in \mathbb{R}^2$.

The conditional discriminator is given by

$$\begin{aligned} d : \Theta^{(d)} \times \mathbb{R}_{\geq 0}^{4k+2} \times \mathbb{R}^s &\rightarrow [0, 1] \\ (\theta^{(d)}, \Lambda, S) &\mapsto d_{\theta^{(d)}}(\Lambda, S). \end{aligned} \tag{14}$$

In its first version from [11], the GAN is then trained in a minmax game with loss function

$$\begin{aligned} \min_{\theta^{(g)}} \max_{\theta^{(d)}} \, & \mathbb{E}_{x \sim \mathbb{P}_r(x)} [\log d_{\theta^{(d)}}(x)] \\ & + \mathbb{E}_{z \sim \mathbb{P}_z(z)} [\log 1 - d_{\theta^{(d)}}(g_{\theta^{(g)}}(z))]. \end{aligned} \tag{15}$$

In words, the discriminator tries to correctly classify the calibrations as real or fake while the generator tries to fool the discriminator. In theory, the two players reach a Nash equilibrium in which the discriminator is unsure about every sample and outputs 0.5 for every sample [11]. In practical applications, however, GANs are difficult to train and to tune which has led to a lot of research trying to improve their training stability.

This setting and the condition S can further be extended with other conditions (e.g. previous calibrations, average traded volume during the time period etc.), and will constitute future work.

5 RESULTS

5.1 Evaluation metrics

To track the similarity of the correlation, the Frobenius norm of the two matrices' differences is used. Let $\Sigma \in \mathbb{R}^{p \times p}$ be a correlation matrix, where $\sigma_{i,j} \forall i, j \in \{1, \ldots p\}$ is the correlation between the i-th and j-th variable. The total number of variables is p. The Frobenius norm reads

$$\|\Sigma\|_F = \sqrt{\sum_i^p \sum_j^p |\sigma_{i,j}|^2}. \tag{16}$$

In the present case, $\sigma_{i,j} = \mathbb{E}(\lambda_i \lambda_j) - \mathbb{E}(\lambda_i)\mathbb{E}(\lambda_j)$ and $p = 4k + 2$. During training, $\|\Sigma_{fake} - \Sigma_{real}\|_F$ is computed every 5 epochs.

To observe the convergence of the marginal distributions, the W1 distance (17) is used

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|], \tag{17}$$

where $\mathbb{P}_r$ is the real and $\mathbb{P}_g$ is the generated distribution. Both the maximum as well as the mean of the W1 distances of each variable are tracked, i.e.

$$\frac{1}{4k+2} \sum_{i=1}^{4k+2} W(\mathbb{P}_{r_i}, \mathbb{P}_{g_i}) \quad \text{and} \quad \max_{i \in \{1, \ldots, 4k+2\}} W(\mathbb{P}_{r_i}, \mathbb{P}_{g_i})$$

where $\mathbb{P}_{r_i}$ ($\mathbb{P}_{g_i}$) is the real (generated) marginal distribution of λ_i.

To track the effect of the state conditions S_t, samples are generated conditioned on the range of the data. E.g. for daytime, the trading time of the NASDAQ stock (9:30 - 16:00) is split into even intervals with boundaries $(t_0, \ldots, t_j, \ldots, t_n)$ with

$$t_j - t_i = (j - i) \cdot c \; \forall \; j, i \in \{0, \ldots, n\}, j > i$$

where $t_0 = 9.5$ (market open) and $t_n = 16$ (market close). For each interval (t_{j-1}, t_j), times are uniformly drawn from $U(t_{j-1}, t_j)$. All other conditions are randomly sampled from the range of the training data. It is of particular interest how the expected value of a certain variable λ_i deviates from the observed data. So in particular

$\mathbb{E}_{\mathbb{P}_{g_i}}(\lambda_i | t \in [t_{j-1}, t_j)) - \mathbb{E}_{\mathbb{P}_{r_i}}(\lambda_i | t \in [t_{j-1}, t_j))$, the difference between the conditional expectations for λ_i drawn from the generated, marginal distribution $\mathbb{P}_{g_i}$ and the data distribution $\mathbb{P}_{r_i}$. Similar to the W1 distance, we compute the (squared) sum of all deviations and the maximum in order to keep track of the general convergence, as well as the "worst" case

$$\sum_{i=1}^{4k+2} \sum_{j=1}^{n} \left[\mathbb{E}_{\mathbb{P}_{g_i}}(\lambda_i | t \in [t_{j-1}, t_j)) - \mathbb{E}_{\mathbb{P}_{r_i}}(\lambda_i | t \in [t_{j-1}, t_j)) \right]^2 \tag{18}$$

and

$$\max_{i \in \{1,\dots,4L+2\}} \sum_{j=1}^{n} \left[\mathbb{E}_{\mathbb{P}_{g_i}}(\lambda_i | t \in [t_{j-1}, t_j)) - \mathbb{E}_{\mathbb{P}_{r_i}}(\lambda_i | t \in [t_{j-1}, t_j)) \right]^2 . \tag{19}$$

For the condition effect using the volatility index VIX, we assume an equidistant grid similar to the time conditioning above. The deviation is then computed equivalently to (18).

5.2 Parameter Simulation

To create a data set containing many different calibrations, intensities up to 5 ticks away (i.e. $k = 5$) from the best opposite price were estimated for 6 minute buckets using LOBSTER data for one year of the INTC ticker. The result is a data set of $\sim$ 16000 calibrations. Setting the intervals to 6 minutes implies the assumption of arrival rates to be stationary within this time period. The length of the time interval is subject to discussion and potentially even deserves an independent study. Due to the non-negativity of Λ, we model the standardized log intensity, i.e.

$$\tilde{\lambda}_i = \frac{\log(\lambda_i) - \mu_{\log(\lambda_i)}}{\sigma_{\log(\lambda_i)}}. \tag{20}$$

This standardized data set is used in a first step to train a GAN to learn (12). The results shown below use the loss setting from [11]. Both networks are fully connected neural networks with three hidden layers and either 32 or 64 hidden neurons. We train for up to 2000 iterations through the training data set with a learning rate of 0.00002 which takes about 30 minutes. Batch size is set to 32 and the noise dimension is 10. The ADAM optimizer is used for training.

Figure 2 visualizes all metrics during the training process on a logarithmic scale. Several things can be observed.

The first glimpse of the data is learnt very fast by the GAN. Longer training tends then to improve results up to a certain level. Looking at any of the metrics, there is a global decreasing trend with increasing number of epochs up until epoch 1000. Then the metrics start to fluctuate around their current level.

GANs are known to be unstable during optimization (see e.g. [4, 13, 23]). Also, training this model to generate the calibrations shows some fluctuations in the tracked metrics. These fluctuations appear to be aligned across different metrics, in particular to be observed in the later training stage.

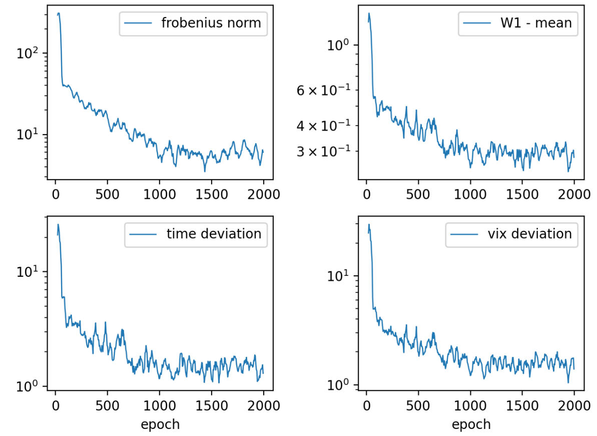

Figure 2: Rolling mean (5 epochs) of observed metrics on a log scale during training.

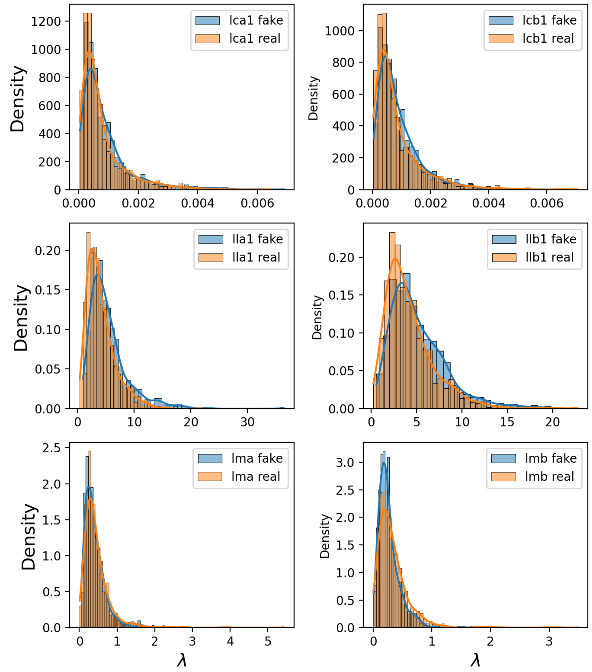

Figure 3: Intensities for cancellation, submission and market order intensities (one bid and one ask).

Figure 3 shows the generated unconditional distributions for six of the 22 intensities (corresponding to $k = 5$), namely $\lambda_1^{C,a}$, $\lambda_1^{C,b}$, $\lambda_1^{L,a}$, $\lambda_1^{L,b}$, $\lambda^{M,a}$ and $\lambda^{M,b}$; • *lca1* corresponds to the "lambda cancellation ask" 1 tick away from the best bid price, $Q_1^a(t)$; • *lla1* corresponds to the "lambda limit order" submissions at $Q_1^a(t)$; • *lma* denoted the intensity "lambda market order" on the ask side $\lambda^{M,a}$. In line with the mean W1 distance shown in the upper right plot in Figure 2, the distributions are mostly well fit, apart from slight deviations.

The order intensities, i.e. any λ, are non-stationary and in particular dependent on the time of the day. This impacts strongly the dynamics in the LOB. Figure 4 visualizes the expected value of the generated and the real order intensities condition on some time bucket. The solid orange line shows $\mathbb{E}_{\mathbb{P}_{r_i}}(\lambda_i | t \in [t_{j-1}, t_j))$, the expected value of the estimates for the bucket of the data, and the solid blue line shows $\mathbb{E}_{\mathbb{P}_{g_i}}(\lambda_i|t \in [t_{j-1}, t_j))$, the mean generated value of the GAN for the particular point in time. The dashed lines indicate

$$\mathbb{E}_{\mathbb{P}_{g_i}}(\lambda_i|t \in [t_{j-1}, t_j)) \pm \sqrt{VAR_{\mathbb{P}_{g_i}}(\lambda_i|t \in [t_{j-1}, t_j))},$$

i.e. mean ± standard deviation of the value from the conditional order intensity.

In particular, the cancellations, *lca1* and *lcb1* are well fit, capturing the decreasing pattern throughout the day. Also the lambdas of limit order submissions, i.e. *lla1* and *llb1* show the U-shaped trend throughout the day, though with a slight over estimation. The most notable deviation is during the morning, which shows a significantly larger expected value for market orders on both bid as well as ask side.

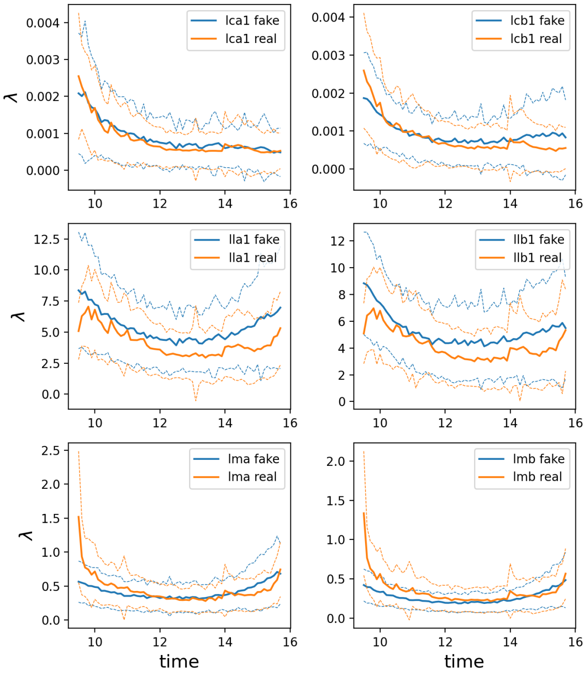

Figure 4: Mean intensities and standard deviations for six different intensities conditioned on the time of the day.

Figure 5 shows the analogue plot to Figure 4 but when conditioning on different values of the volatility index VIX. As before, the orange (blue) line shows the mean value for the real (fake) calibrations conditioned on a given value of the volatility index. Again, particularly for *lca1* and *lcb1*, the cancellation intensities one tick away from the best opposite prices are very well fit as the upper two plots indicate. Generally, for all intensities, the increasing pattern is clearly recognizable indicating the GAN easily picks up the pattern to condition effectively on time and volatility.

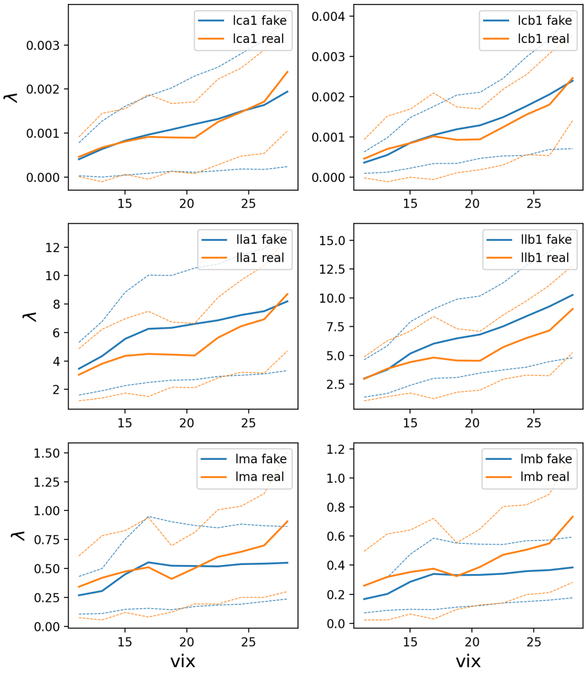

Figure 5: Mean intensities and standard deviations for six different intensities conditioned on the volatility index VIX.

5.3 Event Stream Simulation with Dynamic Parameters

Results show that GANs automatically learn much of the structure of the calibrations with promising fits and without tuning of hyper parameters, as outlined above. Lastly, we want to use the generated calibrations Λ to simulate the order stream and compare it to both real data as well as the Poisson order flow with fixed parameters.

To this end, we will simulate entire trading days of LOB data and compare the statistics of the behaviour of different paths of the LOB. This is done in two steps, which are then repeated in an iterative fashion.

(1) The condition S for the sample/calibration is set and a calibration is generated.
(2) The generated calibration is used to simulate the next time interval[2]. For this time interval, the calibration does not change.

The time frame for which the calibration is kept constant in this example at the same value as for the estimation for the data set – 6 minutes. This implies the parameters change abruptly which is not necessarily optimal or correct and subject for future research to investigate different (smoother) transitions between two consecutive calibrations or other time horizons.

Figure 6 shows the average of 100 exemplary simulations in comparison to the statistics of one exemplary day. The upper left plot shows the number of orders throughout the day. The simulations clearly exhibit the U-shaped pattern which can also be seen in the

[2] For detailed information of the algorithm see chapter 9 in [1]

data of the real day. Furthermore, the high VIX conditioned simulations exhibit a higher level of orders that the lower VIX conditioned simulation. This is in line with Figure 5 which shows larger order intensities for higher VIX values and hence also more orders. For the Poisson order with a constant calibration, the number of orders is flat.

In the upper right, Figure 6 shows exemplary price paths from the simulation. It can be seen that the dynamic calibration clearly induces drifts and non-stationary dynamics in particular for the high VIX conditioned price path. It can also be noted that the real price path is more volatile despite a lower number of events. Part of the reason is the mean reverting behaviour of the Poisson order flow which leads to decreasing cancellation intensities for smaller queue sizes. This lets the queue rather increase than decrease and the consequence are fewer price changes.

The lower left plot in Figure 6 indicates the volatility of the price changes throughout the day. In the present case, the price change is the mid price change during one second. Hence, the volatility for calibration bucket i reads $\sigma_i = \frac{1}{360}\sum_{j=1}^{360}(r_{i,j} - \bar{r}_i)^2$ where $r_{i,j} = \frac{p_{i,j} - p_{i,j-j}}{\delta}$ the j-th price change in the bucket i and $\bar{r}_i$ its mean. For comparison, the rolling volatility from the real exemplary day using 360 price changes is shown, which exhibits a decreasing pattern throughout the day. As mentioned, the calibrations conditioned on a high VIX value lead to higher volatility of the one-second price changes. Both simulations show a similar decreasing pattern as the real data (with a slight increase towards the end of the day), however, they are at a lower level than the sample real day. Again, it can be noted that the Poisson order flow tends to exhibit less volatility than the real data (despite more events). For constant calibration, the volatility throughout the day is again constant.

The lower right plot in Figure 6 shows the distribution of the spread for the simulations and reference data. For all settings, the spreads of the model tend to be tighter. The reason for this is the baseline model. Assuming a spread of two ticks 2δ, we have the situation in Figure 1. Both $\lambda_1^{L,a}$ and $\lambda_1^{L,a}$ now sit at the empty level. The distribution shows that these intensities are much larger than in deeper levels. Thus, the probability of a limit order being posted in the empty level,

$$\frac{\lambda_1^{L,a} + \lambda_1^{L,b}}{\sum_{\lambda \in \Lambda} \lambda}, \tag{21}$$

is rather large. This leads to quickly disappearing spreads in the Poisson order flow. In the future one could add some state dependency to improve the spread behaviour of the model. Table 1 shows the average intensities from the data set scaled by their sum, e.g.

$$\frac{\lambda_i^{L,\star}}{\sum_{i=1}^{k}(\lambda_i^{L,a} + \lambda_i^{L,b})} \tag{22}$$

which is equivalent to the probability where the next limit order submission will occur. It can be seen that in case of a spread equal to 2δ (as in Figure 1), the next limit order will be placed in the empty level between $p_b(t)$ and $p_a(t)$ with a probability of more than 76%. As consequence, any spread $s(t) > \delta$ then tends to quickly disappear.

i	1	2	3	4	5
$\lambda_i^{L,a}$	0.379	0.058	0.026	0.016	0.015
$\lambda_i^{L,b}$	0.384	0.059	0.026	0.016	0.015

Table 1: Table indicating limit order submission intensities for different levels scaled by their total sum.

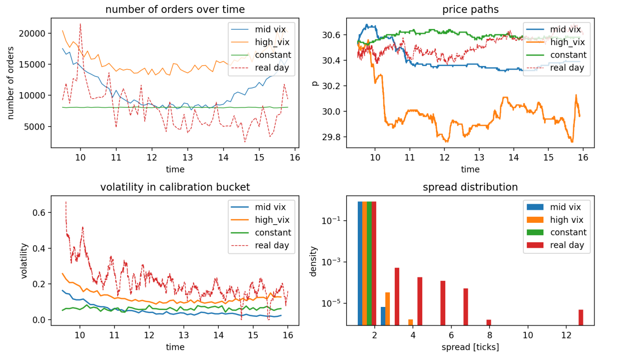

Figure 6: Order book simulation over the entire trading day for both a high and a low vix trading day.

6 CONCLUSION

Many models have been built to model the arrivals of orders in LOBs, the order flow. Amongst others, point processes have extensively been used as outlined in Section 2. Most models, however, are stationary once they are estimated, and several days of data are used for parameter estimation.

This work proposes to build a probabilistic model on top of existing LOB models to learn the distribution of calibrations of smaller time periods conditioned on external variables. The aim of this hierarchical model is to introduce new dynamics and non-stationarity to the baseline model, while preserving its (theoretical) properties. In particular, we train a conditional GAN similar to the design presented in [24] to learn the conditional distribution of the calibrations for the Poisson order flow, as described and simulated in [1]. The results show GANs can capture much of the structure in the data. Furthermore, the approach also learns the conditions on external variables, such as time of day and volatility index VIX, without extensive parameter tuning.

We then use the model to generate entire days of LOB data by iterative generation of calibrations and simulation of the order stream with the synthetic calibrations. Most importantly, the resulting simulations exhibit macro properties which are not observed with using only one global calibration, such as U-shaped intraday pattern, decreasing volatility over the day, or temporary drifts caused by imbalances in the order intensities.

Despite the promising results, there are several directions for future research. First, the fit of the GAN and its training stability can be further improved as this study did not focus on extensive tuning. Second, one can study how long markets may be assumed to be stationary, and what is a suitable time interval that renders these intervals stochastic. Third, our approach adds non-stationary features in the macro perspective. It does not improve the model basis and certain weaknesses remain. E.g. the Poisson order flow with dynamic calibrations still does not show excitation effects. Furthermore, the Poisson order flow seems to produce less price changes than the real data shows. One reason for this is the mean reverting behaviour of queue dynamics. Also, the estimation of the Poisson order flow is fairly simple. The calibration of Hawkes processes with a GAN may circumvent the computationally expensive estimation of its parameters. A possible extension would thus be to use other baseline LOB models, in particular Hawkes processes. Lastly, it would be interesting to investigate to which extend such dynamic calibrations can help to design and train more robust trading algorithms.

ACKNOWLEDGMENTS

Felix Prenzel has been supported by the EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation (EP/S023925/1).

REFERENCES

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

[2] Frédéric Abergel and Aymen Jedidi. 2013. A mathematical approach to order book modeling. *International Journal of Theoretical and Applied Finance* 16, 05 (2013), 1350025.

[3] Frédéric Abergel and Aymen Jedidi. 2015. Long-time behavior of a Hawkes process–based limit order book. *SIAM Journal on Financial Mathematics* 6, 1 (2015), 1026–1043.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875* (2017).

[5] Jean-Philippe Bouchaud, J Doyne Farmer, and Fabrizio Lillo. 2009. How markets slowly digest changes in supply and demand. In *Handbook of financial markets: dynamics and evolution.* Elsevier, 57–160.

[6] Jean-Philippe Bouchaud, Marc Mézard, Marc Potters, et al. 2002. Statistical properties of stock order books: empirical results and models. *Quantitative finance* 2, 4 (2002), 251–256.

[7] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2019. Abides: Towards high-fidelity market simulation for ai research. *arXiv preprint arXiv:1904.12066* (2019).

[8] Rama Cont. 2011. Statistical modeling of high-frequency financial data. *IEEE Signal Processing Magazine* 28, 5 (2011), 16–25.

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

[10] William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736* (2018).

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems.* 2672–2680.

[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] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems.* 5767–5777.

[14] Weibing Huang, Charles-Albert Lehalle, and Mathieu Rosenbaum. 2015. Simulating and analyzing order book data: The queue-reactive model. *J. Amer. Statist. Assoc.* 110, 509 (2015), 107–122.

[15] Andrei Kirilenko, Albert S Kyle, Mehrdad Samadi, and Tugkan Tuzun. 2017. The flash crash: High-frequency trading in an electronic market. *The Journal of Finance* 72, 3 (2017), 967–998.

[16] Adriano Koshiyama, Nick Firoozye, and Philip Treleaven. 2021. Generative adversarial networks for financial trading strategies fine-tuning and combination. *Quantitative Finance* 21, 5 (2021), 797–813.

[17] Charles-Albert Lehalle, Olivier Guéant, and Julien Razafinimanana. 2011. High-frequency simulations of an order book: a two-scale approach. In *Econophysics of Order-driven Markets.* Springer, 73–92.

[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. 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] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. 2018. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems.* 1596–1607.

[21] Gautier Marti. 2020. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 8459–8463.

[22] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406* (2018).

[23] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. 2017. The numerics of gans. In *Advances in Neural Information Processing Systems.* 1825–1835.

[24] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[25] Maxime Morariu-Patrichi and Mikko S Pakkanen. 2018. State-dependent Hawkes processes and their application to limit order book modelling. *arXiv preprint arXiv:1809.08060* (2018).

[26] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. 2020. Conditional Sig-Wasserstein GANs for Time Series Generation. *arXiv preprint arXiv:2006.05421* (2020).

[27] Mark Paddrik, Roy Hayes, Andrew Todd, Steve Yang, Peter Beling, and William Scherer. 2012. An agent based model of the E-Mini S&P 500 applied to Flash Crash analysis. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr).* IEEE, 1–8.

[28] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[29] Justin Sirignano and Rama Cont. 2019. Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance* 19, 9 (2019), 1449–1459.

[30] Eric Smith, J Doyne Farmer, L szl Gillemot, Supriya Krishnamurthy, et al. 2003. Statistical theory of the continuous double auction. *Quantitative finance* 3, 6 (2003), 481–514.

[31] Kaleb E Smith and Anthony O Smith. 2020. Conditional GAN for timeseries generation. *arXiv preprint arXiv:2006.16477* (2020).

[32] Shuntaro Takahashi, Yu Chen, and Kumiko Tanaka-Ishii. 2019. Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications* 527 (2019), 121261.

[33] Magnus Wiese, Lianjun Bai, Ben Wood, and Hans Buehler. 2019. Deep hedging: learning to simulate equity option markets. *Available at SSRN 3470756* (2019).

[34] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. 2019. Quant GANs: deep generation of financial time series. *arXiv preprint arXiv:1907.06673* (2019).

[35] Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2019. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing* 67, 11 (2019), 3001–3012.

[36] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. 2018. Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering* (2018).

A DISCLAIMER

Opinions and estimates constitute our judgement as of the date of this Material, are for informational purposes only and are subject to change without notice. This Material is not the product of J.P. Morgan's Research Department and therefore, has not been prepared in accordance with legal requirements to promote the independence of research, including but not limited to, the prohibition on the dealing ahead of the dissemination of investment research. This Material is not intended as research, a recommendation, advice, offer or solicitation for the purchase or sale of any financial product or service, or to be used in any way for evaluating the merits of participating in any transaction. It is not a research report and is not intended as such. Past performance is not indicative of future results. Please consult your own advisors regarding legal, tax, accounting or any other aspects including suitability implications for your particular circumstances. J.P. Morgan disclaims any responsibility or liability whatsoever for the quality, accuracy or completeness of the information herein, and for any reliance on, or use of this material in any way.