Conditional Generators for Limit Order Book Environments: Explainability, Challenges, and Robustness

Andrea Coletta
J.P.Morgan AI Research
London, United Kingdom
andrea.coletta@jpmchase.com

Joseph Jerome
Dept. of Computer Science, University of Liverpool
Liverpool, United Kingdom
j.jerome@liverpool.ac.uk

Rahul Savani
The Alan Turing Institute; and Dept. of Computer Science, University of Liverpool
Liverpool, United Kingdom
rahul.savani@liverpool.ac.uk

Svitlana Vyetrenko
J.P.Morgan AI Research
New York, USA
svitlana.s.vyetrenko@jpmchase.com

ABSTRACT

Limit order books are a fundamental and widespread market mechanism. This paper investigates the use of conditional generative models for order book simulation. For developing a trading agent, this approach has drawn recent attention as an alternative to traditional backtesting, due to its ability to react to the presence of the trading agent. We explore the dependence of a state-of-the-art conditional generative adversarial network (CGAN) upon its input features, highlighting both strengths and weaknesses. To do this, we use "adversarial attacks" on the model's features and its mechanism. We then show how these insights can be used to improve the CGAN, both in terms of its realism and robustness. We finish by laying out a roadmap for future work.

KEYWORDS

GANs, synthetic data, time-series, financial markets

ACM Reference Format:
Andrea Coletta, Joseph Jerome, Rahul Savani, and Svitlana Vyetrenko. 2023. Conditional Generators for Limit Order Book Environments: Explainability, Challenges, and Robustness. In *4th ACM International Conference on AI in Finance (ICAIF '23), November 27–29, 2023, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3604237.3626854

1 INTRODUCTION

This paper deals with the construction of robust and realistic limit order book (LOB) environments for the training and evaluation of trading strategies. LOBs are a fundamental market mechanism [20], which are used across a significant proportion of financial markets, including all major stock and derivative exchanges. The benefits of having robust and realistic simulators for these markets are numerous. While the ultimate test of the profitability of a trading strategy is to trade it live in a real market [36], this testing approach is rarely feasible for academic researchers, and even for industry practitioners it is potentially expensive. Thus effective simulation-based evaluation frameworks are highly desirable. They would allow the study of markets under different assumptions and the further investigation of AI techniques for training trading strategies, including for *market making* [17, 26, 46] and *optimal execution* [33, 35].

The most prevalent choice for simulation is market replay ("backtesting"), but simulators based on this approach have no reactivity [3]. In this paper we focus on conditional generative models, which provide a solution to the issue of reactivity [10, 11, 27, 29, 32, 44].

In this paper, we motivate and study the problem of developing realistic and robust conditional generative models for simulating LOBs. Throughout the paper we use a specific conditional generator introduced in [10], which we refer to as `LOBGAN`, along with high-fidelity historical order book data from LOBSTER [24]. We here summarize our contributions:

- We extend prior work demonstrating the benefits of `LOBGAN` by doing a *price impact analysis for market and limit orders separately*, which reveal a new strength and weakness of `LOBGAN`.
- We provide a *new technique for analysing the conditioning of generative LOB models*. This technique helps both with explainability and the design of better models.
- We develop and test trading strategies to *study and stress test the robustness* of `LOBGAN`.
- Using our insights, we develop *new* `LOBGAN` *models*, which we demonstrate are better in terms of *both* realism and robustness.
- We provide *recommendations* for how to use `LOBGAN` (or similar generative models) *in practice*, and future research directions.

While our exploration uses this specific model family, our contributions can be applied more generally.

Remark: our adversarial attacks are not *"market manipulation"*. Our goal in this paper is to understand and develop the methodology for designing realistic and robust conditional generators. As such, we design adversarial attacks to exploit and show weaknesses of such models by manipulating the features and mechanism of the models. The term "market manipulation" has a specific meaning and refers to behaviour such as spoofing and quote stuffing, whereby orders are placed, with no intention of them being executed, and with the goal of deceiving and manipulating other market participants. In the interests of clarity, we note that none

ICAIF '23, November 27–29, 2023, Brooklyn, NY, USA

ACM ISBN 979-8-4007-0240-2/23/11.
https://doi.org/10.1145/3604237.3626854

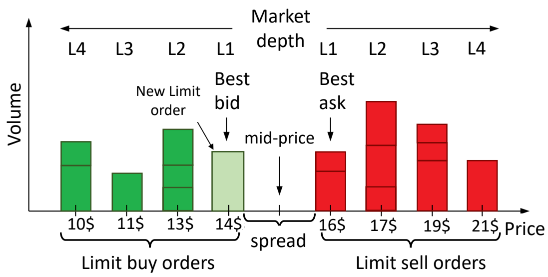

Figure 1: Limit Order Book (LOB)

of the strategies we present in this paper would be considered as market manipulation, but we rather focus on adversarial attacks on the deep neural network generative model [4, 34, 48].

2 PRELIMINARIES

Generative Adversarial Networks (GANs). GANs are deep generative models that generate samples by *implicitly* learning to generate data without the need for an explicit density function [18]. GANs employ two neural networks, G and D, and an adversarial training procedure: a generator G takes as input a vector $\mathbf{z}$ from a distribution p_z (typically a multi-dimensional Gaussian) and outputs a sample x with the goal of fooling the discriminator D, which in turn tries to estimate whether x is real (i.e., x belongs to the ground truth training set) or fake (i.e., x was generated by G). LOBGAN is a Wasserstein Conditional GAN [21, 31] that conditions on recent market data to generate the sample x (i.e., the next order).

Limit Order Books (LOB). Figure 1 gives an example of the LOB structure, which stores all the outstanding limit orders in the *order book*. Unlike limit orders, market orders are not stored in the book if they are not immediately matched on arrival. Buy limit orders (bids) are shown as green bars and sell orders (asks) as red bars, with the bid book (at lower price levels) on the left, and the ask book (at higher price levels) on the right. New buy (sell) orders (market or limit) are matched, if possible, against existing orders according to a price-time priority – first matching against the lowest (highest) price and then matching orders at the best price according to time priority (i.e., transacting first against limit orders that arrived earlier). The light green bar represents a new limit order, which has become the new best bid (with highest price). In a real market, the key question would be what happens next? For a market replay environment, which lacks of reactivity, the answer is simply *whatever happened next in the historical data*, even if the new limit limit order came from an exogenous trading agent rather than the historical data. I.e., market replay assumes that a trading agent can place orders without affecting subsequent order flow, which is unrealistic. In a LOBGAN-simulated environment, subsequent order flow depends on the current state of the market, so that when an exogenous agent interacts with the book and changes it, the LOBGAN generates orders *conditioned on* the updated order book. This *reactivity* is a primary advantage of the CGAN approach.

Realism in isolation and interactive realism. While reactivity in response to incoming orders from exogenous agents is a strength of the CGAN approach, so far evaluation of CGAN LOB models has focused primarily on the realism of the CGAN outputs when the only inputs come from model itself. In this paper, we extend the focus to include realism of outputs when at least one additional trading agent is also in the system, introducing the following terminology: **Realism in isolation** is when no agents are added to the simulation, and the desideratum is that the simulation outputs should look realistic in terms of their statistical properties. This is discussed in detail for LOBs in [50], and studied in existing CGAN LOB models [10, 11, 29]; **Interactive realism (i.e., realistic reactivity)** requires that the simulator reacts realistically when, an external agent also places orders. While, market replay cannot provide interactive realism by definition, a CGAN model can, by conditioning on recent market action, which includes the actions of external agents.

Realism metrics. In general, a single metric to measure the realism of a synthetic LOB market does not exist; thus we evaluate how well a range of statistical properties align with those of real markets [6]. These statistical properties as often referred as *stylized facts* [50]. These stylized facts are used to answer two fundamental questions during the CGAN training process: a) *at which training epoch has the model stabilized?* b) *given two trained CGAN models, which one is more realistic?*. In fact, due to the adversarial training procedure, the CGAN loss is by no means a perfect indicator of the generator quality [19]. Instead, to create LOBGAN [10], at the end of each epoch the CGAN was unrolled in closed-loop simulation to generate multiple days of synthetic market data. Then a human-in-the-loop (HITL) approach [5] was applied using visual comparisons of the synthetic data and the real market data, in particular, for the *price series*, *volume at best bid and ask*, and the *spread*. This approach is based on the fact that humans can easily distinguish between real stock price series and synthetic price series generated by simple popular stock price models [30]. Finally, once a "best" LOBGAN model was selected, it was tested against a wider range of all the stylized facts [10].

Price Impact. As mentioned in the previous paragraphs, the CGAN simulator should ideally react realistically to exogenous agent orders. One way to evaluate this realism is to measure its response to exogenous agent orders through the *price impact* [6], which is the effect that the order has on the price. In particular buy (sell) orders tend to push the price of the asset up (down). We consider the price impact defined as the (reaction) *impact path* [6], i.e., the average price dislocation between the beginning and the end of a metaorder execution (a collection of smaller orders in one direction): $\mathcal{I}^{\text{react.}}_{t+l}(\text{exec}_t \mid \mathcal{F}_t) = \mathbb{E}\left[P^{\text{mid}}_{t+l} \,\middle|\, \text{exec}_t, \mathcal{F}_t\right] - \mathbb{E}\left[P^{\text{mid}}_{t+l} \,\middle|\, \text{no exec}_t, \mathcal{F}_t\right]$. In particular, we measure the price difference between a simulation with and without the metaorder, and compare the resulting impact paths against the form of these paths that have been found in the empirical price impact literature (Figure 3). Note: this approach cannot be implemented with market replay or in a real market as the two situations (the metaorder arriving or not) are mutually exclusive.

3 THE BENEFITS OF LOBGAN

In this section, we discuss and show the benefits of conditional generators over traditional market replay simulators. Going beyond what one typically finds in the literature, we investigate the price impact of *market* and *limit* orders on LOBGAN separately, which extends the analysis in [10]. This allows us to better evaluate and understand the benefits of LOBGAN over market-replay.

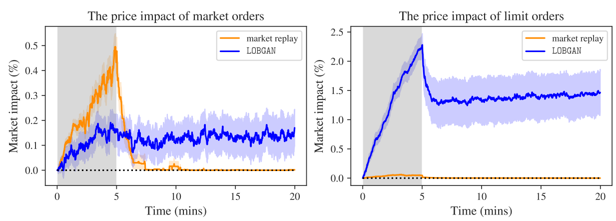

Figure 2: The average price impact path of a TWAP market and limit metaorder in the CGAN environment and in the market replay environment. This TWAP execution constitutes an average of 70% and 60% of the traded Percent of Volume (POV) over its 5 minute execution window (shaded in grey), for market and limit orders respectively. Error bars represent 5th-95th% confidence intervals. Trajectories start at 30 equally-spaced times across two days for a total of 60 trajectories.

Impact path of market orders. We first evaluate LOBGAN and market-replay environments with a Time-Weighted Average Price (TWAP) agent that splits a large order into small market orders evenly executed over time (e.g., 5 minutes). Figure 2 shows the average impact path when the TWAP agent is included in the simulation environment. Notice that, although in the market-replay environment the subsequent order flow does not *react* to the orders of the the TWAP agent, it still exhibits a price impact for sufficiently larger incoming market orders: a large buy (sell) order could take out a large number of ask (bid) price levels in the LOB, and create an arbitrarily large increase in the midprice. However, this increase is an instantaneous spike, with the midprice reverting to its old level when new historical asks arrive. By contrast, for LOBGAN the new order flow does react to the TWAP agent, and the price impact is larger, persistent, exhibits a slightly mean-reverting trend, and is well-aligned with the expected impact from the literature (see Figure 3).

Impact path of limit orders. Whilst the impact of market orders has been much studied, the impact of limit orders has received less attention. On average, relatively large limit orders do have a significant market impact, pushing the price up for large bids and down for large asks [22]. In Figure 2, we compare the price impact of limit orders in a LOBGAN and market replay environment. The market-replay simulation produces minimal market impact, since non-aggressive limit orders (limit orders that do not enter the spread) only affect the price by being filled instead of other limit orders deeper in the book. And the future evolution of order flow is not affected by their presence leading to the tiny market impact. In contrast, these incoming limit orders can impact subsequent order flow in LOBGAN, and this impact, as shown in Figure 2, has the same characteristic shape in response to a meta-order as found in the literature and shown in Figure 3. It is worth mentioning that LOBGAN exhibits a greater price impact with limit orders than with market orders. This phenomenon is partially related to the $\texttt{BookImbalance}_1$ feature which is generally a strong predictor of the sign of future price changes [6]. We show later that LOBGAN is over-reliant on this feature and large limit buy orders alter the $\texttt{BookImbalance}_1$ and drive the price up. We investigate this effect in Section 4.2.

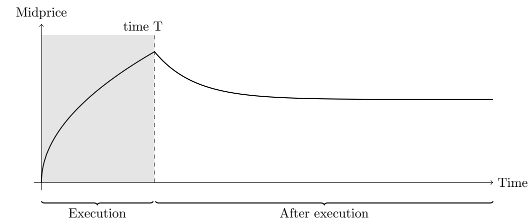

Figure 3: The impact of a buy metaorder from the literature (e.g., [6]) showing temporary price impact (until time T), permanent price impact (the long-run level) and transient price impact (the difference between price impact at the peak and the long-run level).

Additional benefits. Two important further benefits of the CGAN approach are: **Data shareability** – generative models offer a way for realistic data to be shared with academia; **Data Variability** – generative models can generate a countless number of different market scenarios, while backtesting is prone to "time-period bias".

4 THE CONDITIONING OF LOBGAN

In this section we explore how different market state features impact the order flow generated by LOBGAN.

4.1 Feature definitions

Since extracting features from raw data is difficult [45], LOBGAN uses hand-crafted features that have been successfully used in other parts of the market microstructure literature [6, 10].

Order book features. Denote the i^{th} best bid and ask prices by $P_i^b(t)$ and $P_i^a(t)$ respectively, and the corresponding volumes as $V_i^b(t)$ and $V_i^a(t)$. We define the following features at time t:

- *Total volume*, top n levels: $\texttt{TotalVol}_n(t) = \sum_{i=1}^n V_i^b(t) + V_i^a(t)$.
- *Book imbalance*, top n levels:

$$\texttt{BookImbalance}_n(t) = \frac{\sum_{i=1}^n V_i^b(t)}{\sum_{i=1}^n (V_i^b(t) + V_i^a(t))} = \frac{\sum_{i=1}^n V_i^b(t)}{\texttt{TotalVol}_n(t)}.$$

- *Spread*: $\texttt{Spread}(t) = P_1^a(t) - P_1^b(t)$.
- *Midprice*: $P_t^{\text{mid}} = \frac{1}{2}(P_1^b(t) + P_1^a(t))$.
- *Return* since time $t - \Delta$: $\texttt{PctReturn}_\Delta(t) = (P_t^{\text{mid}} - P_{t-\Delta}^{\text{mid}})/P_{t-\Delta}^{\text{mid}}$.
- Suppose that the order book events occur at times t_i for $i \in \mathbb{N}$.

Then, the midprice at time t is equal to the midprice at time $t_{j^*(t)}$ for $j^*(t) = \sup\{j \in \mathbb{N} : t_j \le t\}$. After the n^{th} order book event, we may then also define the *n-event percentage return* at time t:

$$\texttt{EventPctReturn}_n(t) = \frac{P_{t_{j^*(t)}}^{\text{mid}} - P_{t_{j^*(t)-n}}^{\text{mid}}}{P_{t_{j^*(t)-n}}^{\text{mid}}}.$$

We use *the touch* to refer to the best bid and best ask price levels. When we refer to quoting at a distance from the touch, this means that we post a limit order at that distance away from the best price on the corresponding side of the order book. This is quoted in *ticks* (the minimum price difference between two price levels in a LOB).

Trade features. Suppose that trades occur at times s_i for $i \in \mathbb{N}$ and let $V_{s_i}^{\text{trade}}$ be the signed volume of the trade; if the trade is seller-initiated then it takes a positive sign, if it is buyer-initiated, it takes a negative sign.[1] Then, for $t \geq \Delta$ one can define the *trade volume imbalance* of window size Δ at time t by

$$\texttt{TradeImbalance}_{\Delta}(t) = \frac{\sum_{t-\Delta \leq s_i \leq t} \mathbb{1}_{V_{s_i}^{\text{trade}} \geq 0} V_{s_i}^{\text{trade}}}{\sum_{t-\Delta \leq s_i \leq t} \left| V_{s_i}^{\text{trade}} \right|}.$$

`LOBGAN` features. `LOBGAN` conditions on: 1) *order book imbalance* for $n = 1$ and $n = 5$ levels; 2) *total volume* at the first level and at the top five levels of the book; 3) spread; 4) *n-event midprice percentage return* for $n = 1$ and $n = 50$ order book events; 5) *trade volume imbalance* over the last minute and over the last five minutes. `LOBGAN` concatenates the feature values over the last 30 seconds.

4.2 `LOBGAN` features dependence

The conditional nature of `LOBGAN` is crucial for the stability of CGAN training, but it also ensures reactivity when an exogenous agent interacts with the order book. In this section, we investigate how the order book dynamics depend upon the input feature vectors and which features produce the largest changes in the trajectories of the trained CGAN.

We note that analysing the effect of individual features is extremely difficult for a number of reasons: firstly, there are cross-correlations between all of the input features (see Figure 4); second, we do not only want to investigate the distributional properties of a *single* output of the CGAN – but rather the order flow over time when the CGAN *repeatedly* processes its previous generated orders, which compounds any "errors" in the order flow; finally, there is also a "mechanism effect" that comes directly from the order book mechanism, which should ideally be separated from the effect of the features.

To investigate the effect of various features on the generated order flow, we first roll out 60 trajectories each lasting 20 minutes, starting on a lattice of evenly spaced points across two trading days in January 2021. These act as baseline trajectories. We then repeat the process twice for each input feature of the CGAN, fixing its values to be equal to the 5^{th} and 95^{th} percentiles of the empirical training distribution. We allow all of the other features to update as the order book updates. Notice that, here that we do not actually change the *past* orderbook or trades that occur, but rather `LOBGAN`'s *perception* of them. We then investigate the properties of the time series for each of these trajectories, with the goal of measuring the effect of fixing these features on certain key properties of the time series data [50]. In particular, if the trajectory statistics for these rollouts with extreme values for the conditioning appear much the same as the baseline trajectories, then these features are candidates for ablation; on the other hand, if there is a clear dependence of any of the output "stylized facts" on the feature being perturbed, then this knowledge provides partial explainability of the CGAN. We will see later that it can also be used to construct "adversarial" strategies for the CGAN model. This approach requires that the correlations between the input features not be too large, which is indeed a reasonable assumption based on Figure 4.

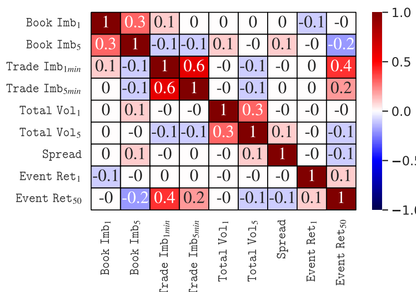

Figure 4: The hand-crafted features correlation matrix.

`BookImbalance`$_1$. When the book imbalance at the touch is larger – due to a larger volume of buy limit orders than sell limit orders – the trajectories generated by `LOBGAN` noticeably trend upwards more on average (see left chart in Figure 5). This effect has been frequently observed in analyses of historical data, where book imbalance is often a predictor of midprice moves [6, 55]. By investigating the properties of the individual orders outputted by the CGAN during these trajectories, we can answer the question of *how* this trend appears as follows: 1) the `BookImbalance`$_1$ features affects the direction of outputted market orders: when the imbalance is lower, `LOBGAN` outputs a larger proportion of *sell* market orders with a consequent drop in the price, and vice-versa; 2) if the input `BookImbalance`$_1$ is lower, there is an increase in the quantity (*cancellation volume* − *limit volume*) on the sell side when compared with the baseline trajectory, and vice-versa. Interestingly, in both cases the difference on the bid side of the book is unaltered, likely due to a simple bias in the training data.

`Spread`. The spread plays a key role in the dynamics of the trajectories generated by `LOBGAN`. In particular, it is highly mean-reverting. This is essential for the stability of the outputted market dynamics. This can be seen in the right chart of Figure 5 – when `LOBGAN` perceives the spread to be small, it outputs orders that increase the spread; when the spread is large, `LOBGAN` tries to close it. After a careful investigation, we notice that `LOBGAN` does this primarily via the distribution of *order types*: when the spread is large (i.e., 95^{th} percentile), `LOBGAN` places more limit orders so that more liquidity is provided to the market and the spread tightens. When the spread is small (i.e., 5^{th} percentile), the order type distribution shifts in favour of deletions and the spread increases significantly.

Other features. `TradeImbalance`$_1$ and `TradeImbalance`$_5$ both have a large effect on midprice dynamics, with `TradeImbalance`$_1$ playing a particularly big role. The main cause of the resulting upward (downward) prices movement are the larger proportion of buy (sell) market orders. The rollouts when individually fixing the features `TotalVolume`$_1$, `TotalVolume`$_5$, `PctReturn`$_{1\text{min}}$ and `PctReturn`$_{5\text{min}}$ in turn, clearly show that none of them has a meaningful effect. Therefore, they are candidates for ablation: we remove these features and investigate how time to model convergence and model realism (see Section 2) are affected. Without `TotalVolume`$_1$ and `TotalVolume`$_5$ the

[1] A trade is seller-initiated if the sell order arrives at the market after the buy order, and is buyer-initiated if the buy order arrives after the sell order.

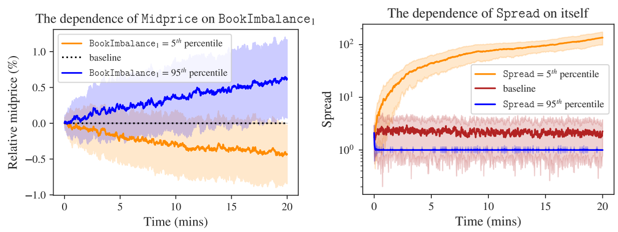

Figure 5: `LOBGAN` feature dependance for `BookImbalance`$_1$ and spread. Left: effect on price. Right: effect on spread.

model achieves comparable realism, meaning that it is able to unconditionally learn the average volume of the orders (we recall that `LOBGAN` is trained against a discriminator that rejects unrealistic volumes.) Without `PctReturn`$_{1\mathrm{min}}$ and `PctReturn`$_{5\mathrm{min}}$ we observe substantially more training time, i.e., more unrealistic markets in the early phases, yet comparable performance at the end of the training procedure. As the returns are used also by the discriminator, we can conclude that they help more with rejecting unrealistic markets during training than with conditioning the generation.

5 LOBGAN ADVERSARIAL ATTACKS

In this section, we show that certain simple trading strategies are able to exploit the `LOBGAN` model, completing profitable round trip meta-orders. This motivates the need for more robust simulation. All of these strategies start with zero inventory and liquidate any terminal inventory and so – whilst the profit and loss curves are marked to market (MtM) – the terminal profit is a pure increase in cash holdings of the agent between the start and end of the episode.

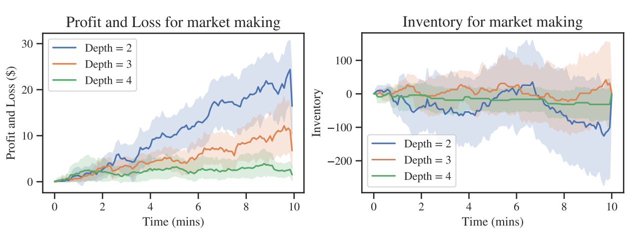

Figure 6: Left: profit and loss for a family of market-making strategies that symmetrically posts a (relatively) large volume at a fixed number of levels (its depth) from a target. Right: the associated inventories.

Market-making. Market makers (i.e., liquidity providers) are a ubiquitous and crucial type of participant in high-frequency financial markets. They usually post limit orders on both sides of the book, offering to buy or sell, with the aim of earning the difference between the bid and the ask whenever they complete a round trip trade. In this section, we introduce a naive symmetric market-making strategy that posts a symmetric volume around the midprice of the asset. These strategies update every 5 seconds – maintaining a fixed volume on each side the book at a fixed depth. They do so by cancelling existing orders that, after an orderbook update, are no longer at the desired depth, replacing them with orders at the desired depth. This simple liquidity provision strategy ends up being consistently profitable in the trained `LOBGAN`-based simulator (see left hand panel of Figure 6), and is robust across a range of depths. While market making can be a highly profitable activity in real markets, it is not realistic for such a simplistic strategy, which uses a fixed distance from the touch on both sides at all times (i.e., it never skews its spread), to be so consistently profitable. It is worth noting that this strategy is not profitable if it posts exactly at the touch, as the frequency at which the agent gets filled can cause the agent to accumulate a overly large signed inventory that needs to be liquidated at the terminal time (at a cost, through market impact) to satisfy the condition of being a round-trip meta-order.[2] The right panel of Figure 6 shows the mean inventory accumulated by the strategy. The inventory mean-reverts, which maintains the inventory risk of the strategy within reasonable bounds. In the extended version of this work, we also train and study a very simple but profitable market making agent using reinforcement learning [9].

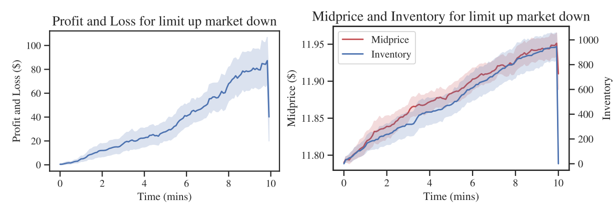

Figure 7: Left panel: profit and loss for a simple strategy that places buy limit orders over 10 minutes and then liquidates with a market order. Right panel: corresponding midprice and inventory.

Accumulating positive inventory using limit orders and then liquidating. As shown in Section 4.2, `LOBGAN` is quite sensitive to the order book imbalance. In this section, we show that a trading strategy can target this feature by continually placing a large volume of limit orders on one side of the book. The strategy also places a small volume of orders on the other side of the book. It accumulates inventory from the larger orders, and after liquidation at the end of the episode makes a profit from the overall round-trip meta-order. A specific example is an agent that maintains a limit order of size 200 at one tick away from the touch on the bid side of the book and an order of size 20 one tick away from the touch on the ask.[3] As described in Section 4.2, by making `BookImbalance`$_1$ larger, this strategy pushes the price up whilst accumulating inventory as the price goes up. The strategy is then able to liquidate at the terminal time for a cost that is less than the value it gained by pushing the price up, completing a profitable round-trip meta-order. Figure 7 shows the profit and loss of this strategy, along with the effect that this strategy has on the midprice dynamics as well as the strategy's accumulated inventory. We recall that this is an adversarial attack, like all of our strategies, created to highlight the `LOBGAN` model weaknesses. Moreover, our strategy is not a form of (illegal) market manipulation, like spoofing, since every limit order we place has a good chance to be, and often is, executed.

[2]It seems highly likely that a more adaptive market-making strategy (e.g., [26]) could avoid this problem by managing inventory risk better.
[3]The strategy works much less well without the ask side, as the agent accumulates a very large inventory, which is expensive to liquidate at end.

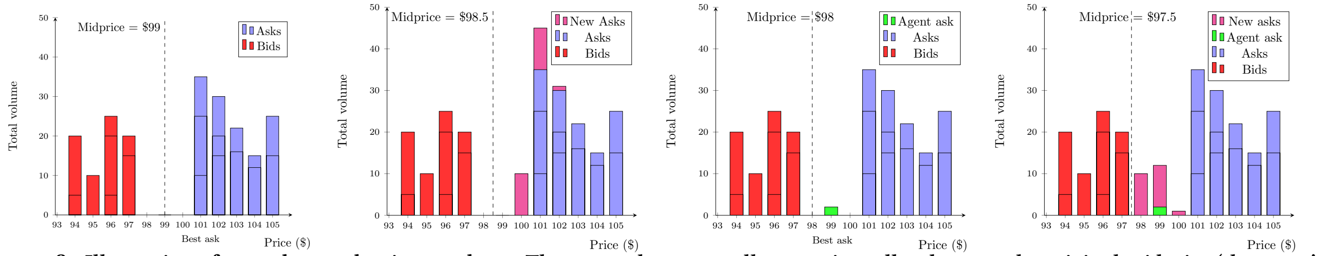

Figure 8: Illustration of a *market-mechanism weakness*. The agent places a small aggressive sell order near the original midprice (the agent's sell order then becomes the best ask) which puts downwards pressure on the midprice. This is because ask orders generated by LOBGAN – which are assigned a price relative to the best ask – have a lower price than if the agent's order was not present.

The weakness of the LOBGAN placing mechanism. The trained CGAN outputs orders with relative prices called *depths*. For example, if the CGAN outputs a bid limit order with depth 1, then it is placed one tick away from the current touch on the bid side of the book. This ensures that the order price distribution is stationary, improving the performance and convergence of the CGAN compared to training the CGAN using absolute prices. However, as we will show in this section, this rule for assigning prices to orders can be exploited by an agent. A limit order that is placed at a better price than the touch is called an *aggressive* limit order. The arrival of aggressive limit orders, and the execution or cancellation of orders at the touch, are the ways in which the midprice changes. In particular, restricting to the bid side of the book for clarity, the midprice decreases when sell market orders arrive and are executed, or when the bid touch is fully cancelled; the midprice increases when aggressive bid limit orders are placed. These same events cause the spread to increase and decrease.

The adversarial strategy in this section is the following: maintain a single limit order at the touch on the side of the book that the agent wishes to move inwards. For example, we describe the strategy for a case in which the agent wishes to move the price down. This strategy is shown in Figure 8, where the bottom and top rows show the market evolution with and without the strategy, respectively. In this market, the best ask price is 101$ (first chart), thus the agent can place a new ask limit order at 99$ being at the touch of the side he wishes to move inwards (third left chart). This means that the prices of all new incoming sell limit orders from LOBGAN (which are priced relative to the best ask) are now relatively lower (fourth chart) than they would have been had the aggressive order of the agent not been added to the order book (second chart). In short, the agent's aggressive ask order means that the best ask is lower than it would have been without and this modifies subsequent order placement by LOBGAN. Then, the agent places an order of a larger size (we chose 300) at a fixed depth away from the ask touch (to get a better sale price) to accumulate a negative inventory and profit as the price goes down further.

We recall that also in this case, our adversarial strategy is demonstrating a weakness of the LOBGAN placing mechanism, which is not representative of a real market. The resulting strategy is consistently profitable across a variety of larger-order depths (see Figure 9).

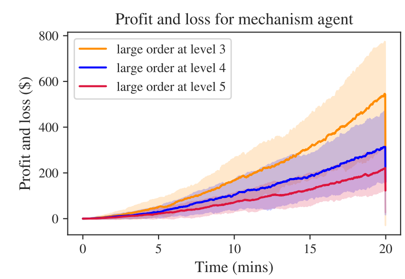

Figure 9: Profit and loss for market mechanism adversarial attack (Section 5) for different depths and (main) volume 300.

6 IMPROVED LOBGAN MODELS

This section presents various solutions aimed at improving the robustness of LOBGAN and the simulators that are built upon it. Based on the the strategies from the previous section, we identify four possible limitations of the current LOBGAN-based simulator:

• **Representativeness** - by using a restricted set of hand-crafted features the simulator has a limited view of the market which could lead to unrealistic behaviour when it conditions on certain market regimes. In particular, any predictive feature of the market that is uncorrelated to the input features of LOBGAN will be invisible to it.

• **Overimportance of certain features** - by only including a limited number of human-interpretable market features features there is a high risk of the model becoming over-reliant upon them, thereby facilitating unrealisticly profitable strategies.

• **Interactiveness** - independently from the chosen features, LOBGAN is trained in a closed-loop: during the training the model learns to generate orders from ground truth past states and novel states induced by its previous orders. While this training alleviates compounding errors (i.e., it reduces the possibility that previous suboptimal decisions induce unseen states and failures), the inclusion of an interactive trading agent during training will almost certainly be able to create novel or adversarial states that are not seen in the current training of LOBGAN.

• **LOBGAN order placement** - by placing orders relative to dynamic market features (i.e., best bid/ask) LOBGAN enables a trading agent to manipulate the next placed LOBGAN orders by altering these features, as seen in Section 5. Thus, using the touch for relative order placement is a limitation of LOBGAN.

We now propose three improved LOBGAN models by addressing the *representativeness* and *overimportance of features* limitations. We discuss a solution to *interactiveness* and *order placement* in Section 8. The LOBGAN model uses a set of hand-crafted features introduced in Section 4.1 to create its own representation of the financial market. However, this representation can be limited and cause misleading behaviour under certain market regimes. Learning the market dynamics from raw orderbook observations would be ideal, but it is difficult and computationally expensive in general [52]. Instead, here we show how to improve *representativeness* by introducing a new LOBGAN model that augments the features detailed in Section 4.1 with new features that allows the CGAN to have a more detailed view of the current market state.

LOBGAN v1. The features that are added to this new version of LOBGAN, which we refer to as LOBGAN v1 (with the original LOBGAN being denoted by LOBGAN v0), are: 1) the *order book imbalance* for $n = 10$ (i.e., for the top 10 levels of the book); 2) the *total volume* at the top 10 levels of the book; 3) the *midprice percentage return* over the last $\Delta \in \{1, 5\}$ minutes; 4) the *trade volume imbalance* over the last 10 minutes; 5) the *total execution volume* over the last $\Delta \in \{1, 5, 10\}$ minutes. The *total execution volume* of window size Δ minutes at time t is defined by $\texttt{TradeVolume}_\Delta(t) = \sum_{t-\Delta \leq s_i \leq t} \left| V^{\text{trade}}_{s_i} \right|$, where s_i is the time at which the trade occurs, and $V^{\text{trade}}_{s_i}$ is the signed volume of the trade. We also remove the two *n-event midprice percentage return* features to prevent the overlap with the new time-based midprice returns.

LOBGAN v1 achieves similar realism for *volume* and *spread* time-series, with a clear resemblance between the simulated and real time series. Interestingly, the new model has slightly better *price series*: they exhibit significant diversity and better symmetry. We believe the new features, especially the time-based *midprice percentage return*, are responsible for the increased realism (in isolation) of prices. As well as improving realism in isolation, LOBGAN v1 is also more robust to the adversarial strategies from Section 5. In Figure 10 we see that LOBGAN v1 makes improvements in terms of robustness to both the $\texttt{BookImbalance}_1$ strategy and market making strategy. Both strategies are now less profitable: the first just about breaks even on average; the second loses money.

The introduction of new hand-crafted features in the last section partially solved the over-reliance on some features discussed in Section 4.2. However, in general it is by no means guaranteed that adding new features will preclude the model overly relying on just few of them. We next further investigate strategies to mitigate over-reliance on certain features. For simplicity, we focus on $\texttt{BookImbalance}_1$ and the related adversarial strategy.

LOBGAN v2. We next investigate the effect of simply removing $\texttt{BookImbalance}_1$ from the feature set for LOBGAN v1. This model is a first *naïve* attempt to show the advantages and disadvantages of just removing relevant features. We start by considering the realism of the resulting LOBGAN v2. The stylized facts show less realism for both *volume* and *spread* w.r.t. the real data: volume accumulates over time while spread has higher variance. Most importantly, the proposed model shows *price series* with strong trends, moving more than 20% relative to the market open. $\texttt{BookImbalance}_1$ is a strong indicator of market direction [6], and when it is included as a feature, the CGAN uses it to generate more realistic time-series than it does without it. However, the goal of LOBGAN v2 was primarily to improve the robustness of the CGAN to the adversarial attacks on $\texttt{BookImbalance}_1$. As we can see in Figure 10, this goal was clearly achieved. In particular, the adversarial strategy loses its control over the midprice dynamics. And furthermore, LOBGAN v2 is also robust to our simplistic market-making strategies. They make a slight loss and have a high variance, both undesirable from a risk and reward perspective.

LOBGAN v3. We now introduce a more sophisticated attempt, namely LOBGAN v3, to reduce the model dependency on $\texttt{BookImbalance}_1$ via a randomized version of this feature. We remove the order book imbalance for levels 1, 5, and 10, and we add the *χ-level order book imbalance* for a random variable χ. Here, we simply take χ to be uniformly distributed on the set $\mathcal{S} = \{1, 2, 3, 4, 5\}$ so that at each time period one of these levels is chosen uniformly at random to define this feature. LOBGAN v3 shows an improvement in realism for both *volume* and *spread*. LOBGAN v3 better captures the stylized facts of real data, even though the generated *price series* still have stronger trends, moving around 10% compared to the market open. Moreover, as shown in Figures 10, we can see that LOBGAN v3 improves upon all of the previous versions of LOBGAN in terms of robustness to both the $\texttt{BookImbalance}_1$ adversarial strategy and the market-making strategy. Both of these naïve trading strategies lose a substantial amount of money on average and have high variance. With improvements in realism and robustness over LOBGAN v2, this new version is clearly better.

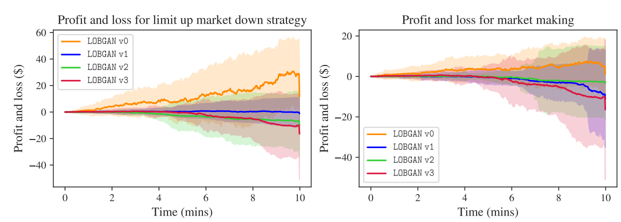

Figure 10: Profit and loss trajectories for strategies from Section 5: $\texttt{BookImbalance}_1$ strategy (left); market making strategy (right).

To conclude this section, we highlight three practical recommendations for how to use LOBGAN or similar models given our insights:

Training LOBGAN with diverse feature sets. We recommend to both explore further predictive features that are not highly correlated with the features already used in versions v0-v3 of LOBGAN and to train a variety of models using different subsets of features. The overarching goal is to choose as many uncorrelated and predictive features as possible. This will improve realism, as the model is able to form a more nuanced view of the market, and robustness, as the model should then not be over-reliant on just one or two features.

Using multiple models. By using multiple models, one can improve the robustness of trading strategy evaluation, e.g., by using each model independently, and then evaluating the distribution of performance across the models. The evaluation could use a variety of statistics of this distribution; e.g., a worst-case approach would use the worst performance across models.

Use existing trading agents for calibration/model selection. If the user has access to historical true trade data, e.g., from execution and market making agents, this data could be used to calibrate LOBGAN to ensure that it gives similar profit as the test agents did in historical live trading. While this will almost never by feasible for an academic project, it would be very natural for a commercial trading entity.

7 RELATED WORK

Conditional generative models. CGANs were introduced in [31], where they were demonstrated and evaluated using the MNIST dataset of images of handwritten digits. While they have most prominently been used in the context of image generation tasks, they have also been explored for generating other types of data, including tabular data [53] and time series data [15]. Important early works on GANs include [1] (who introduced the Wasserstein GAN used by Stock-GAN [29] and LOBGAN [10]) and [21], who made fundamental contributions to improving Wasserstain-GAN training such as gradient penalties (used by LOBGAN). Other generative model approaches that have been applied to time series include Conditional Variational Autoencoders [7], normalizing flows [42], state-space layers and autoregressive models [56], and denoising diffusion models [8, 23, 41]. It is an interesting direction to explore what these other approaches can offer for LOB simulation, analysing, for example, recent work using auto-regressive models [25, 32]. Although we believe the feature exploitation demonstrated in this paper could apply to any conditional model if explicit care to avoid it is not taken.

Reinforcement Learning. In adversarial reinforcement learning, training incorporates an adversarial agent, who is given (limited) control over (e.g., the transitions of) the environment [37, 47]. This is closely related to what we propose in Section 8, namely introducing trading agents during CGAN training. There are also existing works that use RL in order to learn a simulator [43, 51]. However, these works tend to look at settings where a downstream task that the simulator will be used for, such as image classification, provides a natural reward function, such as accuracy. The desiderata in our setting are complex and multi-faceted, and to apply RL a key challenge would be design of the reward function for this multi-objective problem.

Agent-based models (ABMs). For a long time, ABMs have offered the promise of reactive financial market simulation. ABMs have been useful for investigating structural properties of financial markets [14]. However, due to both the lack of agent-level historical data, and the difficulty of calibrating ABMs, they are not currently a viable option for realistic strategy evaluation (see, e.g., [2, 12, 28, 38, 39]). Future work may focus on adapting the recent tensorized and differentiable ABMs [40] to financial market simulation, enabling a fast calibration of a population of traders by using gradient-based approaches; or on using novel GPU-Accelerated limit order book simulators [16] to speed-up the calibration of existing models.

8 CONCLUSION AND FUTURE WORK

We explored the benefits and challenges of building conditional generative models for LOBs. We distinguished between realism in isolation and interactive realism, and explored the latter in depth with the LOBGAN family of models. After analysing LOBGAN's dependence on features, and demonstrating weaknesses of it features and mechanism via adversarial attacks, we designed better, more realistic models. We finish by outlining directions for further work.

Using raw market features. Existing generative models for LOBs, including LOBGAN, use hand-crafted features. It is an open challenge to effectively train a conditional model using raw market features. One piece of work in this direction is [54] which uses convolutional layers to extract features and create a representation using raw order book snapshots. In preliminary experiments, we found that the training approach used by LOBGAN does not work immediately with raw features – new innovations may be needed. For instance, (denoising) diffusion models [23, 41] could be a good alternative generative model that may be good at processing raw orderbook data, however, they are more computationaly expensive than CGANs, which may be problematic [13]. Transformers are also worth exploring [49].

Incorporating RL-based adversarial attacks in training. We demonstrated how interactive realism and corresponding adversarial attacks on the features and mechanism of LOBGAN could be used to build better generative models. It is appealing to try and systematize this process, for example, by using reinforcement learning to create adversarial agents *during* the CGAN training process. A challenge here is to effectively balance the CGAN's training objective between interactive realism and realism in isolation.

Metrics for model selection. Human assessment of realism, as described in Section 2, currently brings an important sanity check to the process. Moreover, there are so many aspects to realism that aggregating them into a single realism score requires much more research. Still, developing such metrics is an important direction to pursue since it would open up many possibilities. For example, with metrics for realism in isolation and interactive realism one could explore whether there is a trade-off between these two types of realism, and with a combined realism metric for model quality one could do automated model selection within the training process.

Order placement mechanism. Our simple adversarial strategy exploits the fact that the LOBGAN-based simulator creates orders relative to touch. A more robust notion of price could fix this: new orders would instead be placed relative to this robust price, which should be designed to be more resilient to temporary and exogenous trading orders than the touch or midprice.

ACKNOWLEDGMENTS

Jerome and Savani were supported by a J.P.Morgan AI Research Faculty Research Award. Jerome was also supported by a G-Research Early Career Grant.

DISCLAIMER

product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proc. of ICML*.

[2] Yuanlu Bai, Henry Lam, Tucker Balch, and Svitlana Vyetrenko. 2022. Efficient Calibration of Multi-Agent Simulation Models from Output Series with Bayesian Optimization. In *Proc. of ICAIF*.

[3] Tucker Hybinette Balch, Mahmoud Mahfouz, Joshua Lockhart, Maria Hybinette, and David Byrd. 2019. How to Evaluate Trading Strategies: Single Agent Market Replay or Multiple Agent Interactive Simulation? *arXiv:1906.12010* (2019).

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Proc. of ECML PKDD*.

[5] Ali Borji. 2019. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65.

[6] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould. 2018. *Trades, quotes and prices: financial markets under the microscope.* Cambridge University Press.

[7] Hans Bühler, Blanka Horvath, Terry J. Lyons, Imanol Perez Arribas, and Ben Wood. 2020. A Data-driven Market Simulator for Small Data Environments. *arXiv:2006.14498* (2020).

[8] Andrea Coletta, Sriram Gopalakrishan, Daniel Borrajo, and Svitlana Vyetrenko. 2023. On the Constrained Time-Series Generation Problem. *arXiv preprint arXiv:2307.01717* (2023).

[9] Andrea Coletta, Joseph Jerome, Rahul Savani, and Svitlana Vyetrenko. 2023. Conditional Generators for Limit Order Book Environments: Explainability, Challenges, and Robustness. *arXiv preprint arXiv:2306.12806* (2023).

[10] Andrea Coletta, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. 2022. Learning to simulate realistic limit order book markets from data as a World Agent. In *Proc. of ICAIF*.

[11] Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. 2021. Towards realistic market simulations: a generative adversarial networks approach. In *Proc. of ICAIF*.

[12] Andrea Coletta, Svitlana Vyetrenko, and Tucker Balch. 2023. K-SHAP: Policy Clustering Algorithm for Anonymous Multi-Agent State-Action Pairs. In *Proc. of ICML*.

[13] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2022. Diffusion models in vision: A survey. *arXiv:2209.04747* (2022).

[14] Vincent Darley and Alexander Outkin. 2007. *A NASDAQ Market Simulation: Insights on a Major Market from the Science of Complex Adaptive Systems.* World Scientific.

[15] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. 2017. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *CoRR* abs/1706.02633 (2017).

[16] Sascha Frey, Kang Li, Peer Nagy, Silvia Sapora, Chris Lu, Stefan Zohren, Jakob Foerster, and Anisoara Calinescu. 2023. JAX-LOB: A GPU-Accelerated limit order book simulator to unlock large scale reinforcement learning for trading. *arXiv preprint arXiv:2308.13289* (2023).

[17] Bruno Gasperov and Zvonko Kostanjcar. 2021. Market Making With Signals Through Deep Reinforcement Learning. *IEEE Access* 9 (2021), 61611–61622.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proc. of NIPS*.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[20] 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.

[21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of Wasserstein GANs. In *Proc. of NIPS*.

[22] Nikolaus Hautsch and Ruihong Huang. 2012. The market impact of a limit order. *Jour. of Econ. Dyn. and Cont.* 36, 4 (2012), 501–522.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proc. of NeurIPS* (2020).

[24] Ruihong Huang and Tomas Polak. 2011. LOBSTER: Limit order book reconstruction system. *Available at SSRN 1977207* (2011).

[25] Hanna Hultin, Henrik Hult, Alexandre Proutiere, Samuel Samama, and Ala Tarighati. 2023. A generative model of a limit order book using recurrent neural networks. *Quantitative Finance* (2023), 1–28.

[26] Joseph Jerome, Gregory Palmer, and Rahul Savani. 2022. Market Making with Scaled Beta Policies. In *Proc. of ICAIF*.

[27] Chia-Hsuan Kuo, Chiao-Ting Chen, Sin-Jing Lin, and Szu-Hao Huang. 2021. Improving Generalization in Reinforcement Learning-Based Trading by Using a Generative Adversarial Market Model. *IEEE Access* 9 (2021), 50738–50754.

[28] Francesco Lamperti, Andrea Roventini, and Amir Sani. 2018. Agent-based model calibration using machine learning surrogates. *Jour. of Econ. Dyn. and Cont.* 90 (2018), 366–389.

[29] Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael P. Wellman. 2020. Generating Realistic Stock Market Order Streams. In *Proc. of AAAI*.

[30] Benoit B Mandelbrot and Richard L Hudson. 2010. *The (mis) behaviour of markets: a fractal view of risk, ruin and reward.* Profile books.

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

[32] Peer Nagy, Sascha Frey, Silvia Sapora, Kang Li, Anisoara Calinescu, Stefan Zohren, and Jakob Foerster. 2023. Generative AI for End-to-End Limit Order Book Modelling: A Token-Level Autoregressive Generative Model of Message Flow Using a Deep State Space Network. *arXiv preprint arXiv:2309.00638* (2023).

[33] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. 2006. Reinforcement learning for optimized trade execution. In *Proc. of ICML*.

[34] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. of CVPR*.

[35] Brian Ning, Franco Ho Ting Lin, and Sebastian Jaimungal. 2021. Double Deep Q-Learning for Optimal Execution. *Applied Mathematical Finance* 28, 4 (2021), 361–380.

[36] Robert Pardo. 2008. *The Evaluation and Optimization of Trading Strategies, 2nd Ed.* Wiley.

[37] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust Adversarial Reinforcement Learning. In *Proc. of ICML*.

[38] Donovan Platt. 2020. A comparison of economic agent-based model calibration methods. *Jour. of Econ. Dyn. and Cont.* 113 (2020), 32 pages.

[39] Donovan Platt and Tim Gebbie. 2017. The Problem of Calibrating an Agent-Based Model of High-Frequency Trading. *arXiv:1606.01495* (2017).

[40] Arnau Quera-Bofarull, Ayush Chopra, Joseph Aylett-Bullock, Carolina Cuesta-Lazaro, Anisoara Calinescu, Ramesh Raskar, and Michael Wooldridge. 2023. Don't Simulate Twice: One-Shot Sensitivity Analyses via Automatic Differentiation. In *Proc. of AAMAS*.

[41] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. 2021. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *Proc. of ICML*.

[42] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. 2021. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *Proc. of ICLR*.

[43] Nataniel Ruiz, Samuel Schulter, and Manmohan Chandraker. 2019. Learning To Simulate. In *Proc. of ICLR*.

[44] Zijian Shi and John Cartlidge. 2023. Neural Stochastic Agent-Based Limit Order Book Simulation: A Hybrid Methodology. In *Proc. of AAMAS*.

[45] Justin A Sirignano. 2019. Deep learning for limit order books. *Quantitative Finance* 19, 4 (2019), 549–570.

[46] Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. Market Making via Reinforcement Learning. In *Proc. of AAMAS*.

[47] Thomas Spooner and Rahul Savani. 2020. Robust Market Making via Adversarial Reinforcement Learning. In *Proc. of IJCAI*.

[48] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv:1312.6199* (2013).

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proc. of NIPS*.

[50] 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 *Proc. of ICAIF*.

[51] Song Wei, Andrea Coletta, Svitlana Vyetrenko, and Tucker Balch. 2023. ATMS: Algorithmic Trading-Guided Market Simulation. *arXiv preprint arXiv:2309.01784* (2023).

[52] Yufei Wu, Mahmoud Mahfouz, Daniele Magazzeni, and Manuela Veloso. 2022. Towards Robust Representation of Limit Orders Books for Deep Learning Models. *arXiv:2110.05479* (2022).

[53] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular data using Conditional GAN. In *Proc. of NeurIPS*.

[54] 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.

[55] Ban Zheng, Eric Moulines, and Frédéric Abergel. 2012. Price jump prediction in limit order book. *arXiv:1204.1381* (2012).

[56] Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. 2023. Deep latent state space models for time-series generation. In *Proc. of ICML*.