



Learning to simulate realistic limit order book markets from data as a World Agent

Andrea Coletta
andrea.coletta@jpmchase.com
J.P. Morgan AI Research
New York, USA

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

Aymeric Moulin*
amoulin@bamfunds.com
Balyasny Asset Management, L.P.
New York, USA

Tucker Balch
tucker.balch@jpmchase.com
J.P. Morgan AI Research
New York, USA

ABSTRACT

Multi-agent market simulators usually require careful calibration to emulate real markets, which includes the number and the type of agents. Poorly calibrated simulators can lead to misleading conclusions, potentially causing severe loss when employed by investment banks, hedge funds, and traders to study and evaluate trading strategies. In this paper, we propose a world model simulator that accurately emulates a limit order book market – it requires no agent calibration but rather learns the simulated market behavior directly from historical data. Traditional approaches fail short to learn and calibrate trader population, as historical labeled data with details on each individual trader strategy is not publicly available. Our approach proposes to learn a unique "world" agent from historical data. It is intended to emulate the overall trader population, without the need of making assumptions about individual market agent strategies. We implement our world agent simulator models as a Conditional Generative Adversarial Network (CGAN), as well as a mixture of parametric distributions, and we compare our models against previous work. Qualitatively and quantitatively, we show that the proposed approaches consistently outperform previous work, providing more realism and responsiveness.

KEYWORDS

GANs, synthetic data, time-series, financial markets

ACM Reference Format:

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 *3rd ACM International Conference on AI in Finance (ICAIF '22)*, November 2–4, 2022, New York, NY, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3533271.3561753>

*The research work was carried out when Aymeric Moulin was employed at J.P. Morgan AI Research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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/10.1145/3533271.3561753>

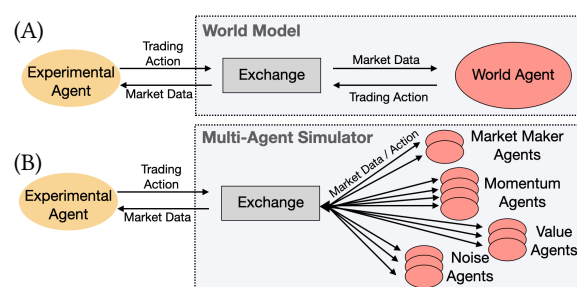


Figure 1: World Model (A) vs Multi-Agent (B) Simulator.

1 INTRODUCTION

Financial markets are among the most complex systems in existence. Naturally described as multi-agent systems, they comprise thousands of interacting heterogeneous participants. Nowadays, both researchers and traders heavily rely on artificial market models, to support the design of algorithms, as well as testing novel trading strategies. Artificial market models can help to isolate and study the impact of new algorithms to the price and volume of the stocks [19]; they can explain the nature of some rare financial market phenomena, such as bubbles and crashes [22]; or they can just be used to study and test trading strategies, before approaching the real market [6].

Previous work mostly focuses on multi-agent modeling, which is a natural bottom-up approach to emulate financial markets [8]. In these models, a number of decision-makers (agents or traders) and institutions, interact through prescribed rules to build the market. Several multi-agent simulators have been developed, by traders and researchers [3, 4, 23]. However, modeling a realistic market through a multi-agent simulation is still a major challenge [8, 18]. In fact, specifying how the agents should behave and interact in the simulation is not obvious. While some agents can be modeled following a common sense or historical analysis [9, 24], in general market participants adopt unknown proprietary trading strategies. Moreover, public available historical data does not include attribution to the various market participants, which makes the calibration of the agents challenging.

To overcome this challenge, learning to simulate from the data as a *world model* was introduced in [6]. This approach assumed a

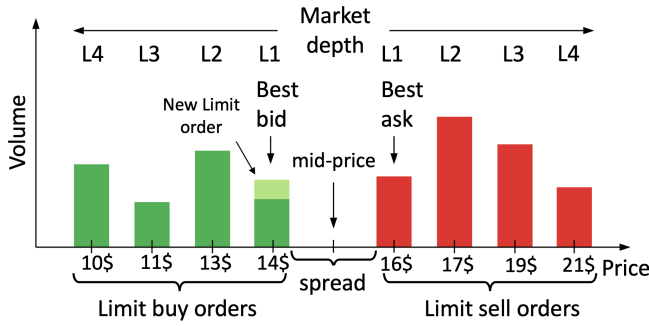


Figure 2: Order book definitions

unique agent trained as a conditional generative adversarial Network (CGAN) [17] from historical data, without the need of making assumptions about the individual market agent strategies.

The world agent was simplistic: it was only capable of placing limit orders which usually account for just 50% of all trading actions. Nevertheless, it was shown in [6] that this model could reproduce stylized facts as well as some form of market impact of trading - hence, this work provides a first attempt to a realistic and responsive *world model*. Figure 1 shows a classic multi-agent simulator compared to the *World Model*, in which all the background agents are represented with a unique *World Agent*.

In this paper, we improve the design of the CGAN-based world model by extending it to support all main market actions (i.e., market order, add limit order, cancel order, and replace order), and we describe it alongside another world model constructed explicitly as a mixture of parametric distributions. Moreover, we improve the CGAN robustness and stability by unrolling the model during the training.

We demonstrate that both approaches presented in this paper outperform previous work on world model construction by providing higher degree of simulation realism. We also emphasize and experimentally demonstrate that the GAN-based model applies to different heterogeneous stocks, as it does not make explicit assumptions about the data distributions.

2 BACKGROUND

In this section we introduce the readers to limit order book markets, with a brief introduction to market structure and mechanisms.

2.1 Limit Order Book (LOB)

Financial markets offer a place for buyers and sellers to meet and trade on different assets. Modern electronic markets, such as NASDAQ, provide ad-hoc message protocols to facilitate trades and provides real-time information about the market order flow and state. In particular, the ITCH [13] protocol provides access to anonymized market data with highest granularity, including all the orders in the market. The main four fundamental orders are: *Market orders*, *Add limit orders*, *Cancel orders*, and *Replace orders*. They respectively indicate the intention of trading a given amount of shares at any price; the intention of trading shares at a fixed limit price; a cancellation of a previous limit order; and a modification to a previous limit order (e.g., a change in the price or quantity).

Most equity markets employ a *continuous-time double auction* mechanism to handle the stream of orders, and to execute a transaction whenever a buyer and seller agree on the price [2]. To store the supply and demand for each asset, the market exchange uses an electronic record called *limit order book (LOB)*. The LOB keeps record of all outstanding limit orders into different levels, organized by price, and it continuously updates them according to incoming orders. Figure 2 shows a snapshot of a LOB with the available supply (red bars) and demand (green bars). The first bars (L1) represent the first level, the second bars (L2) the second level, and so on. Each bar keeps the outstanding orders into a queue structure. An *add limit order* to buy will update the existing demand, increasing the queue size (see Figure 2 light green); while a *cancel order* will decrease the queue size, and consequently reduce supply or demand.

2.2 Artificial Market properties

Realism. Evaluating trading strategies against poorly calibrated market models can lead to poor and misleading conclusions, potentially causing severe loss when we employ these strategies on real markets. To assess the realism of artificial models, researchers commonly evaluate their ability to reproduce statistical properties of real markets called *stylized facts* [2, 7, 24]. For example, as asset daily returns usually have fat tail distribution and long-range dependence, we expected the same properties (or *stylized facts*) from artificial markets. In Section 5 we show that our approach outperforms existing work under a wide range of stylized facts. In particular, we consider *auto-correlations*, *heavy tails distribution*, and *long range dependence* to evaluate asset return properties. While we consider *order volumes*, *time to first fill*, *depth* and *market spread* distributions, to evaluate the volumes and order flow.¹

Responsiveness. Another desirable property of artificial market models is the *responsiveness* to exogenous trading orders: the model should emulate the market reaction to new orders, providing a tool to investigate strategies' impact on the market. For example, the arrival of several buy (sell) market orders commonly causes the rise (fall) of the price. This phenomenon is called *price impact*, and it is desirable that a responsive model exhibit this behavior.

In section 5 we evaluate the responsiveness of our model by simulating the arrival of a burst of buy/sell orders [1].

2.3 Generative Models and CGANs

In the last years generative models have been successfully employed in a wide range of scenarios, ranging from images to time-series. A generative model is any model able to learn a probability distribution p_{model} resembling the real data distribution p_{data} , from a set of real samples. Among generative models, we can identify two major approaches: a) models that *explicitly* estimate the probability density function; b) and models that *implicitly* learn to generate samples without the need of an explicit density function [10].

Generative Adversarial Networks (GANs). GANs are powerful generative models that consider two adversarial neural networks, which *implicitly* learn to generate data samples [11]. In particular, a generator G and a discriminator D are trained simultaneously to

¹We refer the reader to the work in [24] for a detailed introduction to stylized facts.

compete in the following min-max game:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

The generator $G(\mathbf{z})$ produces new realistic samples from a prior noise distribution $p_z(\mathbf{z})$, while the discriminator distinguishes between real and synthetic samples. Both networks aim at maximizing their own utility function: as the training advances, the discriminator D learns to reject synthetic samples generated by G , which in turn learns to generate more realistic data to fool the discriminator. This two-player game trains the model to generate realistic samples resembling the real data.

Our world model aims at generating realistic market actions in accordance with the current market, to provide realism and responsiveness to trading orders. Therefore, we consider a *conditional* GAN [17]. A CGAN generates realistic samples conditioned by some extra information \mathbf{y} , representing the market state in our case. It will include information about ongoing events, including outstanding trading orders. Both generator and discriminator incorporate this extra information resulting into the following game:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log(D(\mathbf{x}|\mathbf{y}))] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z}|\mathbf{y})))]$$

3 RELATED WORK

Artificial market models are increasingly adopted and developed to support researchers, practitioners, and traders [3, 5, 6, 8, 18, 25, 26].

Interactive Agent-Based Simulators (IABS) are popular models that represent the market by means of a set of trading agents that interact through prescribed rules. These simulators usually describe the agent population and their interaction by a set of common sense hand-crafted rules, made to mimic real markets [24].

In [9] Farmer et al. show how zero intelligence (ZI) agents are able to replicate some market dynamics, (i.e., spread and price) by placing random orders. ZI agents have no knowledge of the market, however they can emulate some market reactions to exogenous events, and Palit et al. [21] show that they also reproduce some stylized facts of real markets, like the fat tails and long-range dependencies. Recently, M.P. Wellman and X. Wang [26] use ZI agents along heuristic belief learning (HBL) agents, which actually exploit order book information, to simulate and study spoofing attacks.

However, the recent work of S. Vyetenko et al. [24] discusses how the calibration of a realistic population of agents is a challenging task, as historical labeled data, with details about traders and their individual strategies, are not publicly available. The authors study different hand-crafted agent configurations, and they propose related stylized facts to evaluate the realism of simulation. In Section 5 we evaluate our world model against classic multi-agent simulators, using an agent configuration proposed in [24].

Other work on market modeling leverages learning approaches, either by using adaptive agents, which learn from experience and evolve towards a more realistic behavior [14, 16], or by directly generating synthetic LOB data [6, 15].

The work of J. Li et al. [15] shows a first attempt to generate realistic LOB data. However, they focus on synthetic data generation, not considering market simulation and properties like responsiveness. The work of Coletta et al. [6] introduces the *world model* to learn how simulate a realistic market from data. This approach assumed

a unique agent trained as a CGAN from historical data, without the need of individual market agent strategies or data. However, the model considers only add limit orders, which account for just 50% of all trading actions, resulting in a partial market representation. We extend this work to all the main trading actions, and we evaluate it against our proposal in Section 5.

4 THE WORLD MODEL

The *world model* provides a novel approach to market simulation: it considers a unique *world agent* trained on historical data to emulate the whole traders' population, without the need of individual market agent strategies. The *world agent* observes the market state and generates the next trading action emulating the real traders' behavior. Figure 1 shows the *world agent* representing the whole ensemble of trading agents.

Formally, the world agent can be described as a conditional probability distribution $\mathcal{F}(\mathbf{x}|\mathbf{y})$ that generates the next market action \mathbf{x} given some information \mathbf{y} about the market.

The action \mathbf{x} represents a trading order to the exchange, which advances the simulation into a new market state. Thus, by iteratively generating new orders the simulation advances in time, exploiting the world agent to generate the market.

Actions. We consider 4 possible actions representing the main trading orders, which are defined as follows:

- **Add Limit Order** is a 3-tuple composed by $\langle \text{depth}, \text{side}, \text{quantity} \rangle$
- **Market Order** is a 2-tuple composed by $\langle \text{side}, \text{quantity} \rangle$
- **Cancel Order** is a 3-tuple composed by $\langle \text{cancel depth}, \text{side}, \text{queue position} \rangle$
- **Replace Order** is a 5-tuple composed by $\langle \text{cancel depth}, \text{side}, \text{queue position}, \text{new depth}, \text{new quantity} \rangle$

We introduce $p_a^i(t)$, $v_a^i(t)$, $p_b^i(t)$, $v_b^i(t)$ as the price and volume size at i -th level of the LOB, at time t , for ask and bid respectively. The *depth* $d(t)$ of a limit order describes the order price $p(t)$ with respect to the best-bid and best-ask as follows:

$$p(t) = \begin{cases} p_b^1(t) - d(t), & \text{If side = BID} \\ p_a^1(t) + d(t), & \text{Else} \end{cases} \quad (1)$$

we consider *depths* rather than prices to improve model stability: depths are almost stationary, conversely to prices that change over time.

The *side* and *quantity* describe the amount of shares and the side of an order (i.e., bid or ask), respectively.

Cancel depth and *queue position* are used to cancel and replace orders, as they accurately describe cancellation and replacement dynamics of real markets [2]. In particular, the *cancel depth* identifies the order book level, while the *queue position* identifies the specific order at that level.

Market state. Modeling the market state plays the fundamental role of conditioning the generative model, to produce accurate and responsive actions. We introduce a set of features that best describe the current market \mathbf{s}_t at time t , and the ongoing events.

We first define the *volume imbalance* $I^i(t)$ as the demand and supply inequality within the first i levels:

$$I^i(t) = \frac{\sum_{j=1}^i v_b^j(t)}{\sum_{j=1}^i v_b^j(t) + v_a^j(t)} \quad (2)$$

The volume inequality is a strong predictor of the future price change [2], and provides a view of the current market state.

Along the imbalance, we also define the *absolute volume* $V^i(t)$ within the first i levels:

$$V^i(t) = \sum_{j=1}^i v_b^j(t) + v_a^j(t) \quad (3)$$

This feature helps the model balancing between cancel and limit orders, and generating accurate *quantities* and *depths*, to keep consistent volumes over the day.

We define the *order-sign imbalance* $O_N(t)$ for a history window of N events as follows:

$$O_N(t) = \frac{1}{N} \sum_{j=t-N}^t \epsilon(j) \quad (4)$$

where $\epsilon(j)$ is the sign of a market order at event-time j , if any. We consider $\epsilon(j) = 1$ for a sell market order, and $\epsilon(j) = -1$ for a buy market order. This feature provides knowledge about the price trend in the recent history, and about price impact phenomena.

We consider the market *spread* $\delta(t)$, defined as:

$$\delta(t) = p_a^1(t) - p_b^1(t) \quad (5)$$

It helps the model balancing between liquidity provider and liquidity taker behavior, and to shape order *depths*.

Finally, we consider the *price return* $r_N(t)$ for a history window of N events, defined as follows:

$$r_N(t) = \frac{m(t)}{m(t-N)} - 1 \quad (6)$$

The returns describe the current market trends.

4.1 Explicit model: mixture of parametric distributions

We now introduce a simple and understandable world agent model based on classic parametric distributions. We observe that the considered trading actions are composed of ordinal features with an unbounded range (e.g. quantity) but also relatively well-balanced categorical features (i.e., side and order type). Therefore, we consider a world agent expressed as a product of successive conditional distributions. We first condition the generation process with categorical features, which allows us to break down the order distributions into smaller conditional pieces, which are modeled with simple distributions. Figure 3 shows the proposed approach in which the *order type* and *side* break down the complexity of the generation process, and condition the ordinal features (e.g., depth and quantity). Notice that, we use the following abbreviations: *LO*, *MO*, *REP*, and *CAN*, to identify add limit orders, market orders, replace orders, and cancel orders, respectively.

The decomposition makes the world agent easier and more understandable: we can fit each distribution directly on the data, and

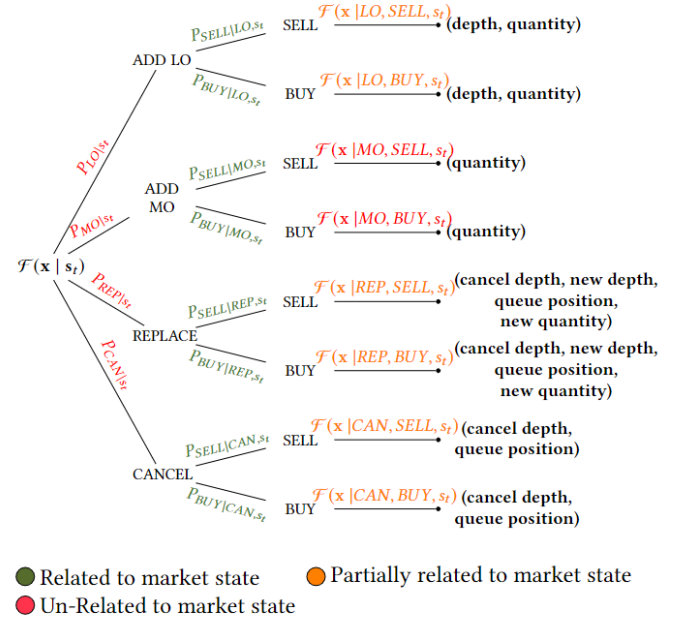


Figure 3: Explicit model : mixture of parametric distributions.

independently from other distributions. We use classic and well-studied distributions, which have been carefully chosen after an accurate analysis of data, and according to existing literature [2, 24]. We use closed-form maximum likelihood or moment matching estimators to fit the distributions' parameters. For example, $P_{LO|s_t}$ is fitted as a fixed probability estimated using empirical proportions of actions in the historical data. At the second level of conditioning, $P_{SELL|LO,s_t}$ is fitted using only historical data in which the trading actions are limit order placements. To easily decompose the problem, and make it tractable and understandable, we make the following assumptions: - for limit order placement and replacement, we assume that *depth* and *quantity* are independent; - for order replacement, we assume that *new depth* and *new quantity* are independent from the former ones.

We now introduce the distributions used for the categorical features. For the *order type* we use a multinomial distribution fitted on historical data, while the *side* consists of a binomial distribution conditioned on the order type and the volume imbalance $I^5(t)$, using a logistic model per each type. The logistic models show high probability of generating a BID limit order when $I^5(t) \approx 0$, which indicates that most of the volume is on the ASK side, and vice versa. Once we identify the order type and side, we use the following distributions to capture each specific order feature:

- We describe the **depth** of a limit or replace order by a mixture of a beta-binomial distribution and an empirical multinomial distribution. The first distribution models negative depths while the latter accounts for positive depths. The probability of having a negative depth is modeled with a logistic regression, dependent on the market spread $\delta(t)$ and volume imbalance $I^5(t)$.

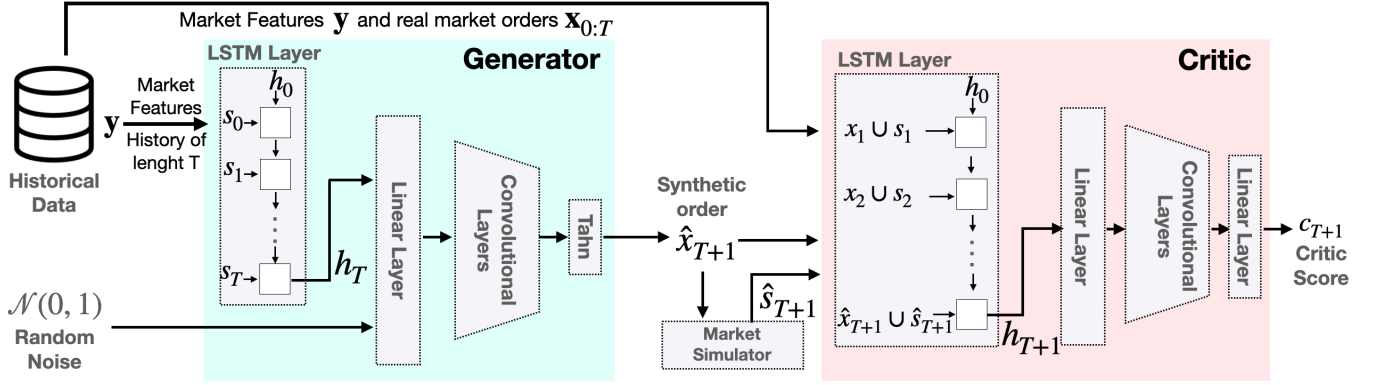


Figure 4: CGAN Architecture.

- We represent the order **quantity** as a mixture of two negative-binomial distributions. We observe that most of the investors trade quantities that are multiples of 100 shares, therefore one distribution describes quantities that are multiples of 100, and the other distribution is used for quantities that are not multiples of 100.
- We represent the **cancel depth** using a negative-binomial distribution, considering replace and cancel orders separately.
- We model the **cancel queue position** using a beta-binomial distribution, considering replace and cancel orders separately.

Finally, we model the **inter-arrival time** of orders by fitting a gamma distribution on historical data. The inter-arrival time is the time between two consecutive orders, and it determines when the world agent is triggered to generate another trading action. Both our world models use this gamma distribution to model the inter-arrival times. In the experimental section, Figure 6 shows the *time to fill* for limit orders, which is strongly related to the order **inter-arrival times**. As both world models show a realistic time to fill, we can conclude that the gamma distribution generates **inter-arrival times** properly.

4.2 CGAN model

We now introduce a CGAN-based *world agent* implemented through a conditional Wasserstein GAN with gradient penalty (WGAN-GP) [12]. We consider a WGAN-GP as it provides a more stable training and allows to deal with discrete data: it minimizes the Wasserstein-1 distance between real and synthetic data distributions, which is continuous and differentiable almost everywhere. In a WGAN-GP the discriminator does not classify samples, but it rather outputs a real value evaluating their realism, thus we refer to it as a *Critic*. Note that in contrast with explicit model learning described in the previous section, the CGAN architecture does not take any parametric assumptions, and hence can be more easily extended to training on data that represents stocks with different dynamics.

Model input. Our CGAN generator $G(z|y)$ takes as input a vector of Gaussian random noise $z \sim \mathcal{N}(0, 1)$ and a vector y containing market information. We represent the market state at time t as an

ordered vector s_t defined as follows:

$$s_t = \{I^1(t), I^5(t), O_{128}(t), O_{256}(t), V^1(t), V^5(t), \delta(t), r_1(t), r_{50}(t)\}$$

To capture the market evolution over time, we define y as the concatenation of the last T historical market states: $y_t = \{s_{t-T}, \dots, s_t\}$. Notice that, while most of the features take values in $[-1, 1]$, we normalize $V^1(t)$, $V^5(t)$ and $\delta(t)$ between -1 and 1, using a mix-max scaler.

Model Output. The generator outputs a synthetic trading action, which may have different attributes upon its type: a *market order* has a *side* and a *quantity*, while an *add limit order* requires also a specific *depth*.

To have an universal representation for all the orders, we consider an output vector \hat{x} including all the possible attributes:

$$\hat{x} = (\text{depth}, \text{cancel depth}, \text{qty}_x, \text{qty}_{100x}, \text{qty type}, \text{order type}, \text{side})$$

The *order type* assumes values in $\{-1, 0, 1\}$ and it discriminates between *market orders*, *add limit orders* and *cancel orders*. In our CGAN architecture the *replace orders* are represented and learned by a sequence of a cancel and an add limit order. The order *quantity* is represented by 3-attributes, namely qty_x , qty_{100x} and *qty type*, and it is defined as follows:

$$\text{quantity} = \begin{cases} \text{qty}_x, & \text{If qty type} = 1 \\ 100 \cdot \text{qty}_{100x}, & \text{Else} \end{cases} \quad (7)$$

As described in the previous section, most of the investors trade multiples of 100 shares, thus we use *qty type* to discriminate their orders, and qty_{100x} to learn the hundreds digits of the quantity. The *side* assumes values in $\{-1, 1\}$ and it distinguishes SELL and BUY orders. Finally, the *depth* and *cancel depth* assume discrete values in \mathbb{Z} , and they represent the price of the order to add or modify, respectively. To reduce the action space, the *queue position* is predicted through a beta-binomial distribution considering its stable and regular distribution.

Notice that, all the non categorical attributes are normalized between -1 and 1. Moreover, depending on the *order type*, only some attributes are meaningful and used: for an *add limit order* we consider both *depth*, *side* and *quantity* attributes, but for a *cancel order* we consider just the *cancel depth* and the *side*.

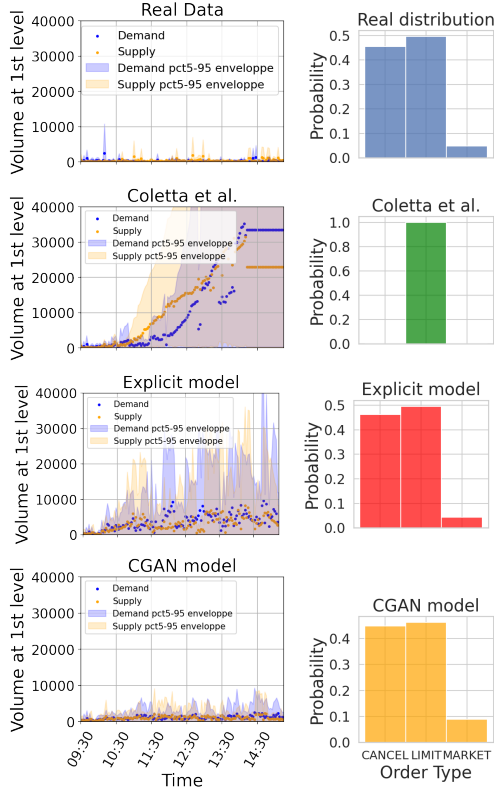


Figure 5: Real vs Synthetic order distributions (AVXL). Proposed world models generate more realistic markets compared to previous work: they generate all main market actions which leads to more realistic volumes.

Model training and architecture. Figure 4 shows the proposed CGAN architecture, which extends the previous model presented in [6]. In particular, our CGAN improves stability and responsiveness by unrolling the model during the training. In [6] the model is trained using only ground truth market states: at each training iteration the CGAN receives a real market state s_t to generate the next order x_{t+1} . At test time, when the model is employed, and unrolled in a closed-loop simulation, the CGAN may encounter unseen states induced by a previous sequence of sub-optimal actions. Unseen states can lead to poor and misleading simulation (e.g., exponential market growth). To mitigate unseen and unrealistic market states, we feed the generated orders into a simulator that advances the market state, and we let the *critic* evaluates both generated orders and states during the training. Most important, we unroll the model during training: we generate k steps ahead, feeding the model with the previous generated market states. This approach enforces the model to deal with synthetic market states, improving stability and realism: actions that led to unrealistic states will be penalized, while we also minimize unseen states. Notice that, we increase the value of k during the training epochs, as the generator learns to generate more realistic orders.

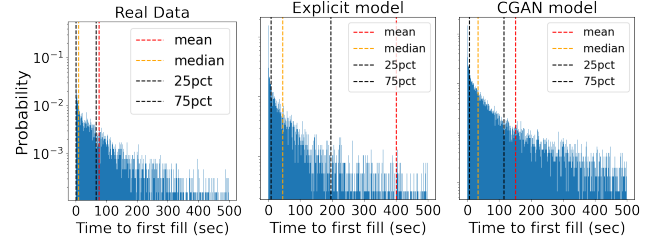


Figure 6: Time to first fill (seconds): CGAN model times are closer to historical ones.

5 EXPERIMENTS

In this section we evaluate our world model by comparing the two proposed approaches in terms of realism and responsiveness. We train our models using NASDAQ TotalView data [20] sent via ITCH protocol [13] replayed at a simulated exchange at the trading action level. We consider four small-tick stocks, i.e., AVXL, AINV, CNR and AMZN, we use 3 to 4 days of data to train the models, and 9 days for testing. The results are averaged for each stock, over the 9 days period. We implement our models extending ABIDES simulator [3], and we feed real data market from 09:30 to 10:00 to initialize the simulated market, and condition the models. For simplicity, we refer to the model implemented through a mixture of parametric distributions as the *explicit model* (see Section 4.1), while we refer to the CGAN-based model as the *CGAN model* (see Section 4.2).

Previous work comparison. We first compare our models against the previous work of Coletta et al. [6]. Figure 5 *right column* charts show the order types in the real and synthetic markets. While both our models closely resemble the market structure, the previous work of Coletta et al. [6] (*green chart*) only represents limit orders, accounting for just 50% of all orders and lacking of realism.² The *left column* charts show the demand and supply (i.e., outstanding limit orders) at the first level of the order book, for real and simulated markets. The orange dots represent the average ASK volumes (supply), while the blue dots represent the average BID volumes (demand). The filled area represents the values between the 5th and 95th percentile. The charts clearly show unrealistic volumes that exponential increase for the work in [6] (*second chart*), mainly caused by the absence of cancel and market orders. Instead, our CGAN model faithfully reproduces real market volumes (*fourth chart*), while the explicit model overestimates the volumes (*third chart*) but it keeps reasonable average values (dots) and does not show exponential growth.

Volumes and order flow stylized facts. We now evaluate the ability of the explicit and CGAN model to reproduce a set of market stylized facts for a given stock, namely AVXL. We first investigate the *time to first fill*, defined as the time elapsed between the placing of a limit order and its actual execution. This stylized fact describes how closely the synthetic market captures the real market dynamics and liquidity. Liquid markets usually have low *time to first fill*, compared to less liquid ones.

Figure 6 shows that both proposed models generate a realistic *time to first fill*, with a median time less than 50 seconds (yellow vertical

²We represent replace orders as a sequence of a cancel and a limit order to keep the representation consistent with CGAN model.

line). The CGAN model produces a slightly more accurate market compared to the explicit model: the average (red line) and 75th percentile (black line) values are closer to the real ones.

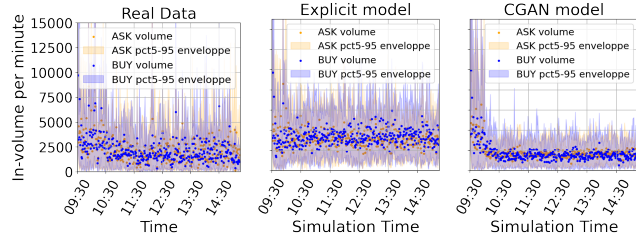


Figure 7: Aggregated volume of limit orders: both world models generate realistic volumes.

Then, we analyse the incoming volume in real and simulated markets. Figure 7 shows the volumes of add limit orders, aggregated in a minute time window: the dots represent the average incoming volumes per minute, while the filled area represents the 5th and 95th percentile values. This stylized fact represents the liquidity provided by market participants during the day. While both models closely resemble the real market, the explicit model slightly overestimates the volumes (as shown also in Figure 5).

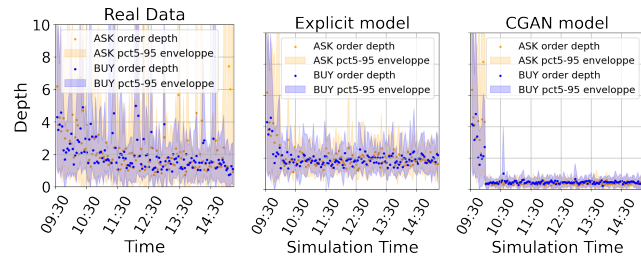


Figure 8: Depth of limit orders: explicit model reproduces realistic order depths.

Figure 8 shows the average depth of limit orders, for BID orders (blue dots) and ASK orders (orange dots). The chart shows how the explicit model closely resembles the real market data, even if the data have slightly less variance. Instead, the CGAN model only partially matches the real data: it shows a narrow data distribution, with most of the depths close to zero.

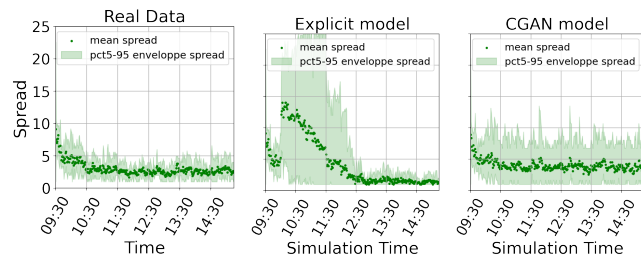


Figure 9: Market spread: CGAN model closely reproduces real market spread.

Figure 9 shows the market spread (see Eq. 5) over the day. The green dots represent the average spread in the real and simulated markets, and the filled area shows the 5th and 95th percentile values. The CGAN model has the best performance, it closely replicates the real market behavior, while the explicit model shows a slight adaptation problem to real market data. When first employed in the market at 10:00 (after the market is initialized with real data), the explicit model doubles the market spread, which decreases and stabilizes only after 12:00.

Finally, we show an example of the generated time-series in Figure 10. The charts show the normalized mid-price: the left chart shows the real samples, the middle chart shows the explicit model mid-prices, and the right chart shows the CGAN model ones. Both models shows promising diverse and realistic time-series data.

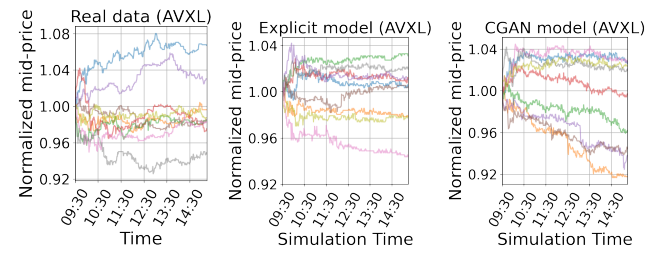


Figure 10: Generated time-series (AVXL) show diversity and realism.

Training on different stocks. We now discuss how our models apply to different stocks. Figure 11 shows the orders generated by the explicit and CGAN model for three different stocks, namely CNR, AINV and AMZN. The blue bars show the real data distributions, the orange bars show the CGAN synthetic data, and the bars with red lines outline the explicit model synthetic data (we use empty bars to improve readability). The first two charts show the *cancel depth* and *depth* of CNR and AINV orders, respectively. The charts show that both proposed models are able to generate realistic data (i.e., the bars mostly overlap). The last chart shows the *depth* distribution for AMZN orders. While the CGAN model is able to generate realistic data, the explicit model fails to reproduce the *depth* of the orders (i.e., it generates only values close to 0). This chart shows the main advantage of the CGAN model: with fewer assumptions on underlying data structure, the model is able to represent a wider range of stocks, with different behaviors and distributions.

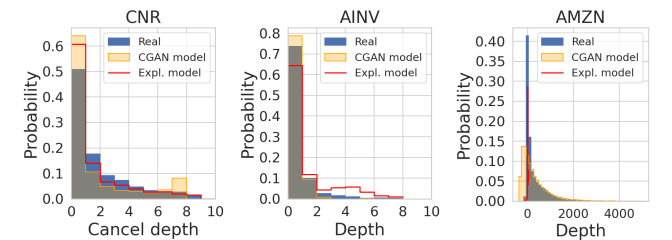


Figure 11: Training on different stocks (AINV, CNR, AMZN): the CGAN model adapts better compared to the explicit model.

Responsiveness. Responsiveness to exogenous orders is a desirable property of financial market simulators, and it allows to investigate the impact of a strategy on the market. To evaluate the responsiveness of our models, we study the price impact caused by an experimental Percent-Of-Volume (POV) agent [1, 6]. This agent submits a burst of buy/sell orders within a limited time window (e.g., 30 minutes) to buy/sell a target amount of shares. This target amount is a percentage $\lambda \in (0, 1]$ of the total transacted volume in the history, for the same time window.

Figure 12 shows the simulated market with the λ -POV agent, with $\lambda \in [0.1, 0.2, 0.5]$. The agent acts only in a time window of 30 minutes, between 10:30 and 11:00 (gray area in the charts). The charts show the *market impact* as the normalized mid-price difference between the simulation with and without the experimental agent. The results average 25 different runs: the black line shows the average mid-price difference; the gray shaded region represents one standard deviation; and the red lines represent the 5-th and 95-th percentile. The left charts show the explicit model, while the right charts show the CGAN-based model.

Both models exhibit the price impact, i.e., the burst of buy trades at 10:30 causes prices to rise, showing a substantial deviation w.r.t. the mid-price in the simulation without the experimental agent. The greater the value of λ , the higher the impact on the price. In particular, the average mid-price difference in the CGAN model with $\lambda = 0.1$ (top right chart) reaches the 0.5%, while with $\lambda = 0.5$ (bottom right chart) it increases over the 1%. With $\lambda \in [0.1, 0.2]$ the CGAN model also shows a mean-reversion to the average price, after the price impact. The mean-reversion effects are weaker for the explicit model, and the price does not return to its average value. With $\lambda = 0.5$ (bottom charts) the experimental agent alters the price trend permanently, i.e., the prices do not return to their average levels. In summary, the observed market impact and price reversion phenomena that arise in simulation, using our world agent approaches, are consistent with observations of the real market [2].

Asset returns stylized facts. Finally we evaluate the realism of generated time-series against a Multi-Agent Configuration (MA-Config), calibrated with 5000 noise, 100 value, 1 market maker, and 25 momentum agents, according to the configurations used in [24]. The first two charts show the *Minutely Log Returns* and the *Autocorrelation*, respectively, which demonstrate that our models are closest to historical data compared to the multi-agent configuration (i.e., real and synthetic distributions overlap). The third chart shows the average autocorrelation of square returns as a function of time lag. It decays for both historical and our synthetic data, as time lag increases, while the multi-agent simulator shows an increasing trend. We conclude that our models provide a more realistic simulation, compared to the hand-crafted multi-agent configuration.

6 CONCLUSIONS

In this paper we introduced a world model simulator to ease financial simulation, and improve realism and responsiveness. We proposed two approaches to the world model based on a Conditional Generative Adversarial Network (CGAN), and a mixture of parametric distributions. We proved that our world models can learn to simulate realistic markets once trained on historical data, without the need of access to individual and proprietary strategies.

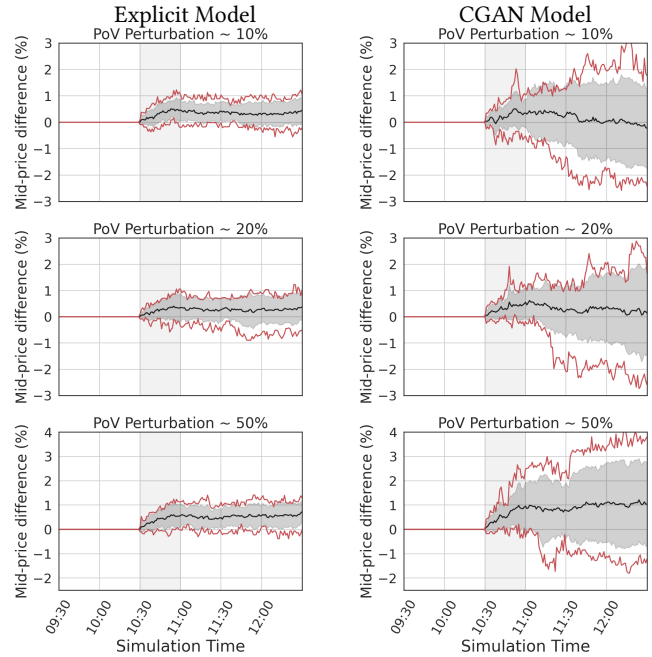


Figure 12: POV Agent Experiment: explicit model (left) and CGAN-based model (right) exhibit price impact and mean reversion trend.

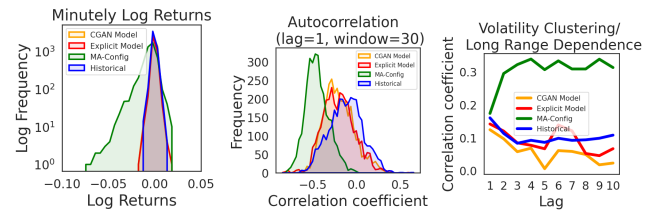


Figure 13: Asset returns stylized fact (AVXL): the world models described in the paper produce stylized facts closer to the historical ones when compared to the multi-agent simulation.

We also demonstrated that our models improve previous state-of-art solutions, providing more realistic simulation. We discussed the main advantages of the CGAN model, and we demonstrated that it is able to represent a wider range of small-tick stocks. For the future work, we would like to explore and improve the CGAN performance on large-tick stocks, which require a more sophisticated training procedure due to their quasi-degenerate data distributions.

DISCLAIMER

This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates (“J.P. Morgan”), and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial 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] 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 preprint arXiv:1906.12010* (2019).
- [2] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould. 2018. *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press.
- [3] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2020. ABIDES: Towards High-Fidelity Multi-Agent Market Simulation. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 11–22.
- [4] Carl Chiarella and Giulia Iori. 2002. A simulation analysis of the microstructure of double auction markets. *Quantitative finance* 2, 5 (2002), 346.
- [5] Christopher J Cho and Timothy J Norman. 2021. Bit by bit: how to realistically simulate a crypto-exchange. In *Proceedings of the Second ACM International Conference on AI in Finance*. 1–9.
- [6] Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetenko, and Tucker Balch. 2021. Towards Realistic Market Simulations: A Generative Adversarial Networks Approach. In *Proceedings of the Second ACM International Conference on AI in Finance (ICAIF)*.
- [7] Rama Cont. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance* 1, 2 (2001), 223.
- [8] J Doyne Farmer and Duncan Foley. 2009. The economy needs agent-based modelling. *Nature* 460, 7256 (2009), 685–686.
- [9] J Doyne Farmer, Paolo Patelli, and Ilija I Zovko. 2005. The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences* 102, 6 (2005), 2254–2259.
- [10] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016).
- [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*, Vol. 27.
- [12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028* (2017).
- [13] Nasdaq Inc. 2020. NASDAQ TotalView-ITCH 5.0. <https://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/NQTVITCHSpecification.pdf>
- [14] Blake LeBaron and Ryuichi Yamamoto. 2007. Long-memory in an order-driven market. *Physica A: Statistical mechanics and its Applications* 383, 1 (2007), 85–89.
- [15] 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.
- [16] Andrew W Lo. 2005. Reconciling efficient markets with behavioral finance: the adaptive markets hypothesis. *Journal of investment consulting* 7, 2 (2005), 21–44.
- [17] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv:1411.1784* [cs.LG]
- [18] Takanobu Mizuta. 2016. A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. *Available at SSRN 2710495* (2016).
- [19] Takanobu Mizuta. 2020. An agent-based model for designing a financial market that works well. In *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 400–406.
- [20] NASDAQ. [n.d.]. Nasdaq Total View. <https://www.nasdaq.com/solutions/nasdaq-totalview>
- [21] Imon Palit, Steve Phelps, and Wing Lon Ng. 2012. Can a zero-intelligence plus model explain the stylized facts of financial time series data?. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 653–660.
- [22] James Paulin, Anisoara Calinescu, and Michael Wooldridge. 2018. Agent-based modeling for complex financial systems. *IEEE Intelligent Systems* 33, 2 (2018), 74–82.
- [23] Marco Raberto, Silvano Cincotti, Sergio M Focardi, and Michele Marchesi. 2001. Agent-based simulation of a financial market. *Physica A: Statistical Mechanics and its Applications* 299, 1-2 (2001), 319–327.
- [24] Svitlana Vyetenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Derovic, Manuela Veloso, and Tucker Hybinette Balch. 2020. Get Real: Realism Metrics for Robust Limit Order Book Market Simulations. In *ACM International Conference on AI in Finance (ICAIF)*.
- [25] Jianling Wang, Vivek George, Tucker Balch, and Maria Hybinette. 2017. Stockyard: A discrete event-based stock market exchange simulator. In *2017 Winter Simulation Conference (WSC)*. IEEE, 1193–1203.
- [26] Xintong Wang and Michael P Wellman. 2017. Spoofing the Limit Order Book: An Agent-Based Model. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 651–659.