Learning to Manipulate a Financial Benchmark

Megan Shearer
University of Michigan
Ann Arbor, MI, USA

Gabriel Rauterberg
University of Michigan
Ann Arbor, MI, USA

Michael P. Wellman
University of Michigan
Ann Arbor, MI, USA

ABSTRACT

Financial benchmarks estimate market values or reference rates used in a wide variety of contexts, but are often calculated from data generated by parties who have incentives to manipulate these benchmarks. Since the LIBOR scandal in 2011, market participants, scholars, and regulators have scrutinized financial benchmarks and the ability of traders to manipulate them. We study the impact on market welfare of manipulating transaction-based benchmarks in a simulated market environment. Our market consists of a single benchmark manipulator with external holdings dependent on the benchmark, and numerous background traders unaffected by the benchmark. Background traders use standard zero intelligence (ZI) strategies. We explore two types of manipulative trading strategies: manually adjusted ZI, and strategies generated by deep reinforcement learning. We find that manipulation decreases market surplus for the manipulator but increases it (to a lesser degree) for the background traders. It also decreases the quality of market information. Including the benchmark holdings, aggregate profits for the manipulator substantially increase. The negative impacts of manipulation, therefore, fall to the external counterparties to the manipulator's benchmark holdings, as well as anyone relying on benchmark information for decision making.

KEYWORDS

financial benchmark manipulation, algorithmic trading, agent-based modeling, deep reinforcement learning

ACM Reference Format:
Megan Shearer, Gabriel Rauterberg, and Michael P. Wellman. 2023. Learning to Manipulate a Financial Benchmark. 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/10.1145/3604237.3626847

1 INTRODUCTION

A ***financial market benchmark*** is a summary statistic over market variables, such as prices of specified securities at designated times. Benchmarks are employed by market participants for various purposes, including as reference measures for asset values (e.g., S&P 500), interest rates (LIBOR), and market volatility (VIX); to define derivative instruments; or as price terms in contracts [15]. Benchmarks in the form of reference measures can provide a concise reflection of market realities, thereby supporting decision making in the real economy. As such, accurate benchmark prices constitute a positive externality from functional financial markets [6]. Their use in financial instruments and contracts also serves a valuable function in commerce and risk management.

Given their role in market decisions and contracts, some entities may have stakes in benchmark values, and hence incentives to try to influence or *manipulate* them. For instance, the *London Interbank Offered Rate* (LIBOR) long served as a common benchmark, for upwards of $300 trillion worth of loans worldwide. Several major banks were implicated in schemes to manipulate LIBOR in the last decade, and criminal charges were brought against several individuals [24]. February 2018 saw accusations of manipulation in the CBOE *Volatility Index* (VIX), a measure of US stock market volatility based on the cost of buying certain options [3]. LIBOR was particularly vulnerable to manipulation because it was based on self-reported data provided by financially conflicted parties [12, 15]. In the wake of the LIBOR scandal, regulators, academics, and market participants lobbied for a transaction-based replacement, such as the Secured Overnight Finance Rate (SOFR) or the US Dollar Intercontinental Exchange (ICE) Bank Yield Index [12, 18]. Whereas it may be harder to manipulate transaction-based benchmarks, it is still possible, as in the alleged manipulation of the VIX in 2018 and the World Markets/Reuters Closing Spot Rates in 2014 [7].

Fig. 1 presents an example of how a transaction-based benchmark might be manipulated. In this case, the benchmark is calculated by the *volume-weighted price average* (VWAP) of the transactions. A manipulator shifts the VWAP benchmark downwards—generally at some cost in market profit—by submitting a series of marketable sell orders.

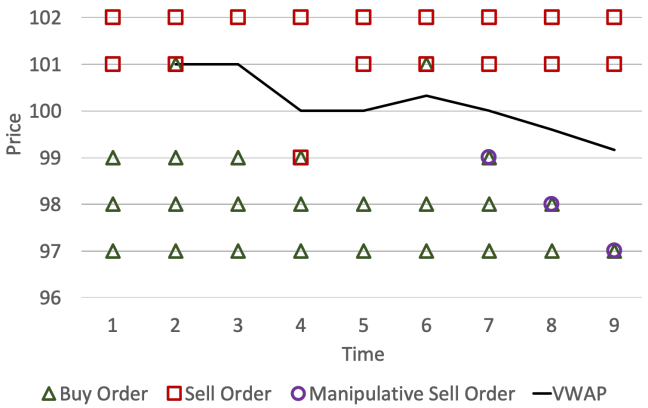


Figure 1: Hypothetical order book with unit orders over a brief trading period. A manipulator submits three marketable sell orders near the end, shifting the VWAP benchmark from 100.3 to 99.2.

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

ACM ISBN 979-8-4007-0240-2/23/11…$15.00
https://doi.org/10.1145/3604237.3626847

Prior work has employed theoretical models and historical data to study benchmark manipulation in financial markets [4, 11, 12, 14, 31]. Using a simulated market allows us to incorporate complex details of *market microstructure*, representing the actual mechanics of trade, interactions among market participants, and the structure of the market. By combining the agent-based model with game-theoretic reasoning, we can also consider the response of strategic agents to the presence of a benchmark manipulator, and consider a wide range of market settings, benchmark designs, and trading strategy options.

Our model employs a standard market mechanism organized around a limit order book for a single security. We assume a benchmark defined by transaction prices in this market. Trading agents may submit buy and sell orders, with orders executing immediately when matched, otherwise resting in the order book pending execution against a subsequent order. The market includes a manipulator agent, with external holdings of a contract tied to the benchmark. The rest of the market comprises background agents who have private reasons to trade, and a market maker who seeks profit by connecting these traders across time.

We consider three types of manipulation strategies. The first is a simple hand-crafted strategy, which extends the behavior of simple background traders by adjusting offers systematically in order to influence the benchmark in a certain direction. The other two types of benchmark manipulator generate their trading strategies through ***deep reinforcement learning*** (DRL). The two types correspond to qualitatively different RL algorithms, called ***deep Q-network*** (DQN) [26] and ***deep deterministic policy gradient*** (DDPG) [21]. In both cases, the agent is not explicitly instructed to manipulate, but rather learns a policy (mapping from market state to orders submitted) that effectively achieves manipulation; this is illustrated in Fig. 2. These policies are derived through simulated experience with the market model, given a reward function that credits the agent for its profits from the market combined with profits from its contract holdings tied to the benchmark.

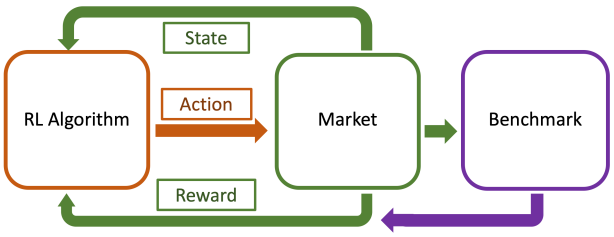


Figure 2: In training a trading strategy, the learning algorithm considers reward from market profit plus returns from benchmark holdings. This may result in a learned strategy that sacrifices market profits to influence the benchmark.

We determine the impact of benchmark manipulation by comparing market outcomes with and without manipulation. These comparisons reflect strategic responses of the background traders to the presence or absence of the manipulator. We find across a variety of settings that manipulation is profitable overall to the manipulators. The manipulation activity itself is costly, in that the manipulator must sacrifice trading profit to move the benchmark. The background traders actually benefit from the manipulation, as their aggregate gains from trading increase. The external parties dependent on the opposite side of the benchmark are the real losers from manipulation, with their losses captured in part by the manipulator and in part by the background agents whose trading is effectively subsidized.

This work represents the first automatic derivation of benchmark manipulators. Our key contributions:

- A model of financial benchmark manipulation, instantiated in an agent-based simulation environment.
- Trading strategies that effectively and profitably manipulate the benchmark in this model, including techniques for automatically generating manipulation strategies using deep reinforcement learning. We demonstrate successful learning to manipulate using two qualitatively different DRL algorithms, with and without the presence of market makers.
- Analysis of the impact of benchmark manipulation on market efficiency and on social welfare, accounting for some variation in market structure and strategic response.

Following a discussion of related work in the next section, we describe the market environment in §3. §4 introduces the benchmark manipulator and methods for learning to manipulate. §5 describes our experimental design, and §6 presents the results and our analysis of the effect of benchmark manipulation. As the ability to automatically generate manipulation strategies presents significant new challenges for financial regulation, §7 provides commentary on how this study can inform policy.

2 RELATED WORK

Martínez-Miranda et al. [23] studied market manipulation using a Markov decision model, identifying conditions that are relatively favorable for manipulative strategies. Wang et al. [41] developed an agent-based model of market manipulation, demonstrating settings where a spoofer can effectively influence market prices despite the presence of rationally responding traders. The current work builds on this agent-based approach, employing a similar market model extended to include a financial benchmark.

Mizuta [25] showed that a genetic algorithm combined with agent-based simulation can learn a sequence of actions that profits in a specified simulation scenario by influencing the prices offered by other trading agents following a fixed market-sensitive strategy. Wang and Wellman [42] studied methods to *adapt* a spoofing strategy to evade detection, within an adversarial learning framework. Byrd [9] considered the problem of learning *not* to spoof: that is, ensuring that an RL algorithm does not inadvertently manipulate a market in particular ways.

Significant attention has been paid to the potential of automating market manipulation through misinformation campaigns, in social media and other forums. Yagemann et al. [47] study the potential for conducting *market-based* manipulation at scale, through botnet hijacking of brokerage accounts. On the basis of SEC data and agent-based simulation, they find that such attacks appear to be quite feasible.

The majority of prior research on ***benchmark manipulation*** is either theoretical or based on analysis of historical market data.

Duffie and Dworczak [12] introduce a theoretical model to analyze the robustness and bias of alternative benchmark constructions, and find that VWAP is optimal among linear benchmarks. Duffie [11] also considers robustness to manipulation in design of an auction mechanism to convert LIBOR-based contracts to employ the replacement SOFR benchmark.

Bariviera et al. [4] and Eisl et al. [14] use historical data to find instances of manipulation of interest-rate benchmarks and provide suggestions for more robust benchmarks and regulation. Rauch et al. [31] also use historical data to find instances of benchmark manipulation in LIBOR and investigate which banks were potentially involved in the 2011 scandal. Griffin and Shams [17] examine spikes at time of settlement as evidence for possible manipulation of the VIX benchmark. Such findings have underscored concerns and contributed to policy discussions around reforms of financial benchmarks [13, 15, 19, 38].

There exists a significant amount of prior work that focuses on the goal of developing trading strategies using reinforcement learning (RL). Previous studies address this in agent-based simulation and with historical data. Numerous simulation-based studies demonstrate the learning of profitable trading studies from a discrete or continuous observation space and an action space [1, 33, 35, 44]. Likewise many have demonstrated successful RL of trading strategies using historical data. Most employ DRL with a discrete or continuous observation space and discrete action space [10, 20, 27–29, 36, 45, 49], but some recent work considers continuous observation and action spaces [22, 30, 45, 46, 48]. Not surprisingly, given the profit potential of any advantage in trading strategy, advances in RL and DRL are quickly implemented in this domain. What is reported in public research is undoubtedly just the tip of an iceberg.

3 MARKET ENVIRONMENT

Our model comprises a single security traded through a limit order book, with a transaction-based benchmark calculated at the end of the trading period. The common value of the security at time t is given by the ***fundamental*** r_t, which is generated by a stochastic process. This model is implemented in `market-sim`, a market simulation platform originally developed by Wah [39] and employed in many agent-based finance studies [40, 41, 44].

3.1 Benchmark

The benchmark we employ is ***volume-weighted average price*** (VWAP), which Duffie and Dworczak [12] showed should be hardest to manipulate among a class of transaction-based benchmarks. As its name suggests, VWAP sums the prices weighted by quantity of transactions. Suppose there are N transactions at quantity and price (q_i, p_i) over the trading horizon T. Then VWAP is given by:

$$\beta_T = \frac{\sum_{i=1}^{N} q_i p_i}{\sum_{i=1}^{N} q_i}.$$

In our market scenario, agents submit only single-unit orders, thus the VWAP benchmark becomes:

$$\beta_T = \frac{\sum_{i=1}^{N} p_i}{N}.$$

3.2 Agents in the Market

The benchmark manipulator operates in a market populated by background agents employing the ***zero intelligence*** (ZI) trading strategy [16] in a version described by Brinkman [8]. ZI background traders arrive according to a Poisson process, and on each arrival perceive the market state (current price quote, recent transaction prices, plus a noisy observation of the fundamental).[1] From this they construct an estimate $\hat{r}_t$ of the final fundamental based on information observed up to t. Finally, they submit a buy or sell order (decided by coin-flip) for a single unit. The new order replaces its previous offer, if any, on the order book. The price of the limit order at time t is set at the agent's estimated valuation for the good, $v(t)$, offset by a requested surplus ζ_t. Valuation $v(t)$ is the sum of the estimated fundamental value $\hat{r}_t$, and an agent-specific private value. Private values are vectors expressing diminishing marginal value for units of the security, drawn i.i.d. from a specified distribution for each agent at the start of the market. The requested surplus ζ_t is chosen for each order uniformly at random, from an interval whose endpoints are parameters of the ZI strategy. The ZI agent employs one additional strategic parameter, $\eta \in [0, 1]$, in deciding to submit an executable order instead if it would be able to obtain at least fraction η of its requested surplus from the current order book.

Some market instances also include a ***market maker*** (MM), which follows the MM strategy described by Wah et al. [40].

4 BENCHMARK MANIPULATION STRATEGIES

Like the background traders, the benchmark manipulator operates in the market by submitting single-unit limit orders to buy and sell the market security. Also like these traders, the manipulator accrues profit from the market as the sum of trading cash flow and value of terminal holdings, where this value in turn is the sum of common and private value elements. What distinguishes the manipulator is that it also obtains payoff based on holdings of a contract tied to the benchmark. An example of a manipulator's ***contract holdings*** may be the stake they have in a publicly traded acquisition target. If the acquisition contract is tied to the price of the stock, then the manipulator may benefit through market actions that increase the stock price. The benchmark in this example is a function of trading price history. In our model, an agent with ψ units of contract holdings receives a payment of $\psi\beta_T$ when the market ends with final benchmark value β_T.

Let $V(t)$ denote the value of the agent's market position at time t, defined as valuation of current market holdings plus cash flow from transactions to that time. The total profit of a benchmark manipulator $B(t)$ is:

$$B(t) = V(t) + \psi\beta_t. \tag{1}$$

If $sign(\psi)$ is positive (negative), then the manipulator benefits from higher (lower) benchmark levels. By choosing higher or lower order prices than it would otherwise, it may be able to influence the benchmark. Doing so entails some loss of profit in the securities

[1] Trading based on a combination of market information and noisy fundamental information is a common feature in agent-based finance studies [5, 34]. Trader attention to market information is necessary for the possibility of spoofing [41], and may also provide a channel facilitating benchmark manipulation.

market, but may be worthwhile if the gain in payment from contract holdings is sufficient.

4.1 Zero Intelligence Manipulation

The first manipulation strategy we consider is *ZIM*, an adjusted version of the ZI strategy that attempts to influence the benchmark. A standard ZI agent arriving at time t submits orders priced at $p^{\mathbf{ZI}}(t) = v(t) \pm \zeta_t$, where ζ_t is the requested surplus. A ZIM agent offsets $p^{\mathbf{ZI}}(t)$ by $sign(\psi)\chi$, where χ is a strategic parameter:

$$p^{\mathbf{ZIM}}(t) = p^{\mathbf{ZI}}(t) + sign(\psi)\chi. \tag{2}$$

This manipulator also employs the strategic parameter $\eta \in [0, 1]$ to submit a marketable order if the current quote is sufficiently favorable. There is a subtle difference in how η applies for ZIM compared to ZI. For a ZI agent, it is always the case that requested surplus $\zeta_t \geq 0$. With offset, however, a ZIM agent's total requested surplus may be negative. If in fact $\zeta_t \pm sign(\psi)\chi < 0$, then the manipulator is willing to accept *any* portion of its requested surplus. Specifically, when buying, the manipulator prices its order at ASK_t rather than $p^{\mathbf{ZIM}}(t)$ if:

$$\text{ASK}_t \leq v(t) + \max\left\{\eta\big(sign(\psi)\chi - \zeta_t\big), \big(sign(\psi)\chi - \zeta_t\big)\right\}.$$

When selling, it prices its order at BID_t rather than $p^{\mathbf{ZIM}}(t)$ if:

$$\text{BID}_t \geq v(t) + \min\left\{\eta\big(sign(\psi)\chi + \zeta_t\big), \big(sign(\psi)\chi + \zeta_t\big)\right\}.$$

4.2 Manipulation with Deep RL

We also develop manipulative strategies using two qualitatively distinct DRL algorithms: deep Q-network (DQN) [26] and deep deterministic policy gradient (DDPG) [21].

4.2.1 Deep Q-Network. DQN is a model-free, off-policy value learning algorithm. ***Value learning*** is the task of inducing a function representing the value of relevant situations. DQN is ***model-free*** as it does not incorporate explicit representations of the environment dynamics in value learning. A ***policy*** defines the agent's behavior in terms of a mapping from states to actions. In the value-based approach, the learned policy is implicit in the learned value function. DQN is ***off-policy*** as the learned policy may be unconnected from the policy used to generate experiences. Off-policy learning is imperative in our context, as the interface to `market-sim` does not permit updating the policy while the market is active. Thus all training occurs between market runs.

DQN combines ***Q-learning*** and ***deep neural networks*** (DNNs) to learn Q-values in environments with rich sensory data. A ***Q-value*** is the estimated value of total discounted reward for the remainder of an episode, for a given state-action pair (s, a). Suppose the agent arrives to the market in state s and takes action a, leading to state s' and producing immediate reward ρ. We record the experience tuple (s, a, s', ρ) to learn from and update Q-values once the episode is complete. DQN uses a DNN to learn a hierarchical abstract representation of a complex state space. This DNN estimates Q-values over a discrete action space. DQN updates the DNN parameters θ using the stochastic gradient descent updating rule:

$$\Delta\theta = \alpha\left[(\rho + \gamma \max_{a'} Q_\theta(s', a')) - Q_\theta(s, a)\right] \nabla_\theta Q_\theta(s, a),$$

where $Q_\theta(s, a)$ is the estimated Q-value given the current DNN parameters and state-action pair, α is the learning rate, and γ is the discount factor.

4.2.2 Deep Deterministic Policy Gradient. DDPG is a model-free, off-policy actor-critic algorithm. An ***actor-critic*** algorithm combines policy learning and value learning. ***Policy learning*** tries to directly learn a policy function that maximizes the agent's reward. The actor maintains a parametrized policy function and the critic a value function, represented as a DNN (like DQN). The actor is updated given the learned parameters from the critic θ^Q, and by applying the chain rule to the expected return from the distribution J with respect to the parameters of the actor θ^μ:

$$\begin{aligned}\Delta_{\theta^\mu} J &\approx \mathrm{E}_{s_t \sim \nu^\pi}\left[\Delta_{\theta^\mu} Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t|\theta^\mu)}\right] \\ &= \mathrm{E}_{s_t \sim \nu^\pi}\left[\Delta_a Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t)} \Delta_{\theta^\mu} \mu(s \mid \theta^\mu) \mid_{s=s_t}\right],\end{aligned}$$

where ν^π is the discounted state visitation distribution for a stochastic behavior policy π. The actor learns a distribution over the action space, which is mapped to a continuous action space. Noise $\mathcal{N}$ is added to the actor's policy for exploration:

$$\mu'(s_t) = \mu(s_t \mid \theta_t^\mu) + \mathcal{N}.$$

4.2.3 State Space. The benchmark manipulator's state space includes all the agent's private information. This includes its private valuation of the traded security, contract holdings, and current holdings of the security. We also include the *side* of the current order (buy or sell).

The agent's state space includes publicly available information in the market, such as the remaining time in the trading period and time since the last trade. We also include features from the market's order book, such as size, spread, and currently listed order prices. The state must be a constant size, but the order book is dynamic throughout the trading period. We address this problem by fixing a limited depth of book, padding or truncating as necessary. For padding, we set prices at estimated final fundamental, plus or minus three standard deviations of the observation noise.

We also include the omega ratio, a metric that determines the favorability of submitting an order. Lastly, we include the number of transactions and their prices. We pad or truncate the transaction price history as necessary to fit a fixed length, as for the order book.

4.2.4 Action Space. The benchmark manipulator's learned policy selects the price of the order to submit. Upon each arrival, the manipulator perceives the observable state and submits an order. It determines whether to buy or sell by flipping a coin, then submits a single-unit order for the selected side. There is no option to refrain from submitting an order, but the same effect can be achieved by submitting an order price sufficiently far from current quotes.

For the DQN agent, the action space is a discrete set of ZIM strategies, each defined by a setting of the ZI parameters plus the offset parameter χ. On each market arrival, the agent observes state s and evaluates the available actions using the DNN representation of the Q-function. The optimal action $a^* = \arg\max_a Q_\theta(s, a)$—one of the available ZIM strategies—is selected, and applied to the current market state to generate an order for the market.

When the benchmark manipulator uses a policy learned through DDPG to select an action, it directly selects a value $A \in [0, 1]$. Our

agent then maps this action to a price for its order at time t:

$$p_t^{\text{DDPG}} = \hat{r}_t + \big(sign(\psi)\chi - C\big)A,$$

where C is a hyperparameter tuned during training, $sign(\psi)$ is the direction of the agent's contract holdings, and χ is an offset parameter. This mapping function is similar to (2), though rather than randomly selecting a requested profit from a uniform distribution, the agent learns the requested profit directly.

4.2.5 Reward Function. The benchmark manipulator aims to maximize combined profit B from the market and benchmark (1). We thus define reward for time t as the difference between the total profit at its next arrival at time t' and the total profit at t:

$$\rho_t = B(t') - B(t).$$

This reward cannot be immediately calculated, since the order placed at time t can match with another anytime between t and t' (when it is replaced by a new order). Thus, we wait until t' to calculate the reward for the action at time t. At the end of the market at time T, the summation of the rewards is equivalent to the manipulator's final payoff:

$$B(T) = \sum_{t \in Arr} \rho_t, \text{ where } Arr \text{ denotes the agent's market arrivals.}$$

5 EXPERIMENTS

We test the efficacy and implications of benchmark manipulation strategies through agent-based simulation, employing a simplified form of empirical game-theoretic analysis (EGTA) [37, 43] to identify approximate equilibria among the available strategies. The first question is to what extent agents employing benchmark manipulation strategies—hand-crafted or learned—can influence the benchmark to enhance profit. The second is what are the ramifications for market performance and agent welfare. We evaluate these questions in multiple market environments, employing a variety of strategies for the background agents and benchmark manipulator. In each case, we find the combination of strategies that background traders play in equilibrium in the presence or absence of manipulation. We then evaluate the outcomes in each case, from the perspectives of the manipulator, background agents, and aggregate market.

5.1 Market Environment Settings

Our test environments have fifteen background agents and one benchmark manipulator. The market settings are the same as employed by Wright and Wellman [44]. The market fundamental time series has mean $\bar{r} = 10^5$, mean reversion $\kappa = 0.01$, and market shock variance $\sigma_s = 2 \times 10^4$. The maximum number of units all agents can hold at any time is $q_{\max} = 10$. Lastly, the private value variance is $\sigma_{PV}^2 = 2 \times 10^7$. The finite time horizon of the market is $T = 2,000$ time steps. The background agents and manipulator arrive to the market according to a Poisson distribution with rate $\lambda_a = 0.012$.

We consider instances of this market with and without a market maker. If the MM is present, its arrival rate is $\lambda_{mm} = 0.05$. The market maker submits 100 buy orders and 100 sell orders at each market arrival. The spread the market maker uses is 1024 and each order is spaced by 100. The market maker is not considered a player in the market game as its parameters are fixed.

ZI strategies available to the background traders are the same as those employed by Wright and Wellman [44]. We use the pure-strategy equilibrium among background traders found by these authors as the baseline no-manipulation case.

In each environment the benchmark manipulator is assigned contract holdings $\psi = 40$. The ZIM agent draws its requested surplus $\zeta_t \sim U[380, 420]$ and chooses among strategies with $\eta \in \{0.5, 1.0\}$, and possible offsets $\chi \in \{0, 250, 500, 750\}$. Selecting $\chi = 0$ is tantamount to not manipulating. We also examine environments where the manipulator learns trading strategies with DQN or DDPG, using the methods described in §4.2.

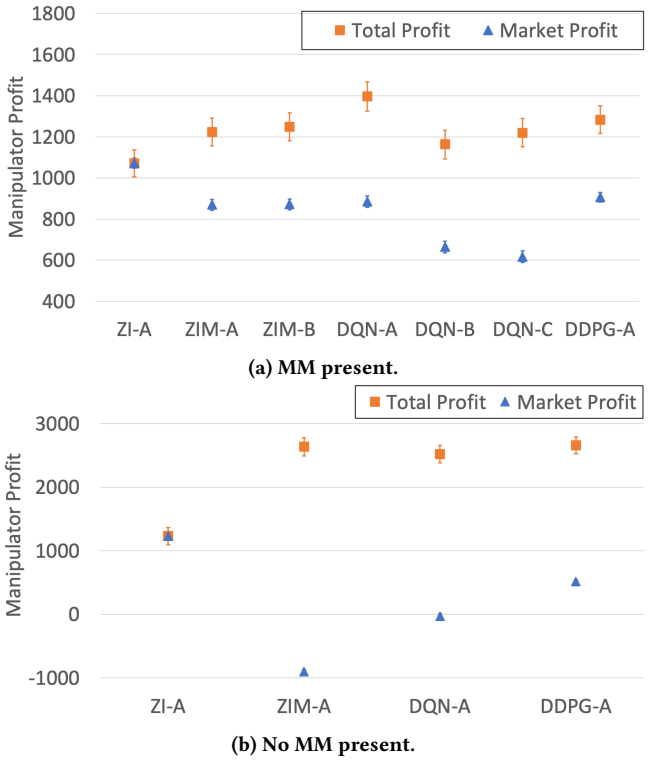
5.2 Simplified EGTA Process

We model the market as a role-symmetric game and partition the agents into two roles: background traders and a single benchmark manipulator. Starting with the baseline no-manipulation equilibrium identified by Wright and Wellman [44], we replace one of the background traders with a manipulator: implemented as a ZIM, DQN, or DDPG agent. For the ZIM agent, we try each ZIM candidate against the baseline equilibrium and select the most profitable. For the DRL (DQN or DDPG) agents, we likewise train in the context of this baseline.

Once the benchmark manipulation strategy is selected, it is likely that the background traders are no longer in equilibrium. Therefore, we test single-player strategy deviations of the background traders, holding the manipulator strategy fixed. If there is a beneficial single-player deviation, we test a variety of mixed strategies containing the original equilibrium strategy and the best deviation. Fixing the new distribution of background traders, we repeat the manipulator strategy optimization (enumerated selection for ZIM, or retraining for the DRL agents). We then repeat the process with another single-player deviation for background traders, followed if applicable by another optimization of the manipulator.

6 RESULTS

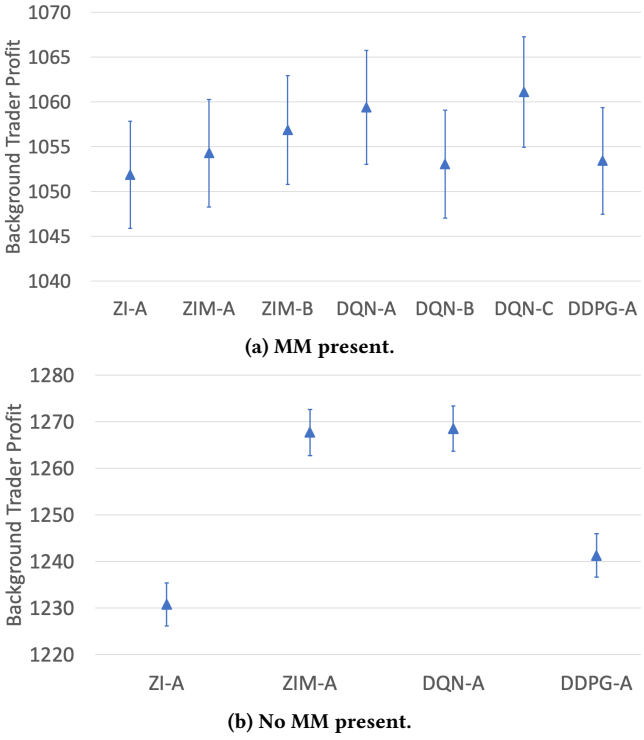
We analyze the performance of the various manipulators in multiple environments. Environment **A** is the market environment where the background agents are equilibrated for no manipulation in a pure strategy equilibrium found by Wright and Wellman [44]. Environment **B** refers to the market environment where the background agents are calibrated to the ZIM manipulator using the single-player deviation method. Environment **C** denotes the market environment where the background agents are calibrated to the DQN manipulator using the single-player deviation method. We include the label "ZI" to signify the case when the agent does not manipulate. We study the welfare impacts of the three manipulators—ZIM, DQN, and DDPG—by examining agent and aggregate market payoffs. Specifically, we calculate the market profit and total profit of the benchmark manipulator where total profit aggregates the profit from market trading (i.e., market profit) and profit from the benchmark holdings. We also find the profit of the background traders. The total profit and market profit are the same for the background traders because they are indifferent to the final benchmark calculation. Lastly, we study the aggregate market profit and aggregate total profit. Aggregate market profit is the sum of the background



(a) MM present.

(b) No MM present.

Figure 3: Profit of the manipulator. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average payoff of the manipulator with standard error bars.



(a) MM present.

(b) No MM present.

Figure 4: Aggregate market profit of the fifteen background traders. The x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average with standard error bars.

traders' profit, MM profit (if present), and the benchmark manipulator's market profit. Aggregate total profit replaces the third term with benchmark manipulator's total profit.

Fig. 3 depicts the total and market profit of the benchmark manipulator. In all cases, the total profit of the benchmark manipulator increases when it manipulates the benchmark. When a MM is present, DQN and DDPG agents in Environment **A** significantly increase profits. The ZIM agent and DQN agent in **B** and **C** increased their average total profit from the non-manipulative case, but not by as much. The manipulators' market profit decreases from the non-manipulative case. It is worthwhile for the successful manipulator to endure the decrease in market profit because its profits from the change in benchmark more than cover the loss. Without MM, all of the manipulators significantly increase total profit and decrease market profit. It is easier for the manipulator to profit when there is no MM because it does not need to trade through the MM's many orders in the book to change the price.

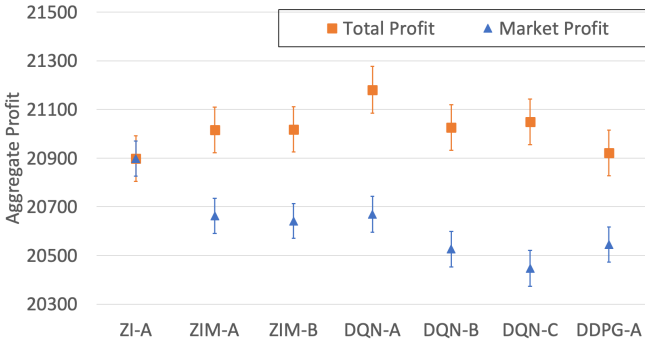
Fig. 4 shows the profit of the background agents. The background agents benefit from benchmark manipulation in all cases. The background agents are especially better off when there is no MM; this is likely due to an increase in direct trades with the manipulator. The manipulator's orders are priced to influence the benchmark, which tends to divorce them from market values and in many cases make them more attractive to background traders. Background traders benefit from the manipulative activity, both from the increase in profitable trades, and from the opportunity to demand higher surplus on orders that would have traded anyway.

Fig. 5 shows the aggregate total and market profit. Aggregate total profit increases with benchmark manipulation. Aggregate market profit decreases, as the market becomes less efficient when the manipulator is more successful. Manipulation impacts the benchmark enough that the manipulator's gain from the benchmark exceeds its losses from trading in the market. The background traders gain at most the manipulator's loss from the market, but the manipulator's resulting gain from the external contract exceeds that of the background traders. The implicit loser is the counterparty to the manipulator in the benchmark contract, who effectively pays the price of successful benchmark manipulation.

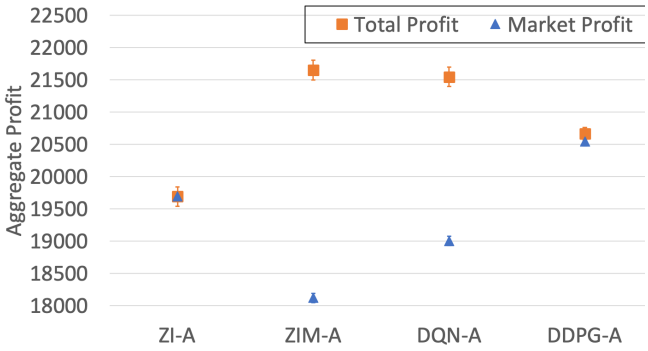
Fig. 6 depicts the VWAP benchmark in each market environment. As expected (particularly given observed profit effects), the benchmark increases significantly when there is manipulation compared to when there is no manipulation. The manipulator is able to successfully shift the benchmark in the direction of its contract holdings regardless of MM presence, though the magnitude of the shift is larger without MM.

7 POLICY ANALYSIS

Following the LIBOR scandal, regulators investigated other benchmarks that had allegedly been manipulated and imposed some of

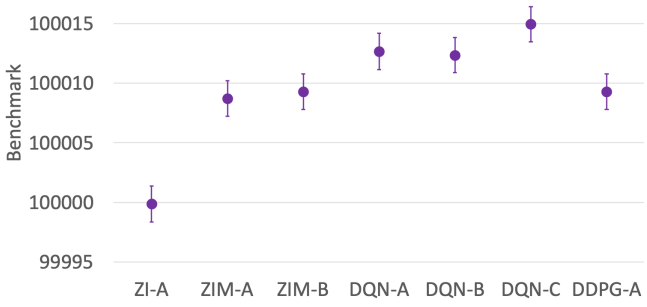


(a) MM present.

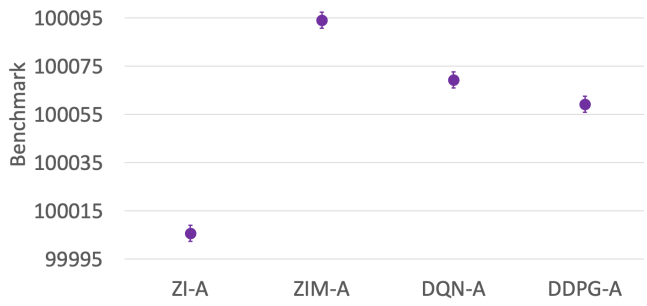


(b) No MM present.

Figure 5: Aggregate total and market profit of all agents. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average with standard error bars.



(a) MM present.



(b) No MM present.

Figure 6: The VWAP benchmark under different manipulation strategies and environments. Each point shows the average VWAP with standard error bars.

the largest penalties ever paid by financial institutions. Given the important role of benchmarks as financial infrastructure, regulators also turned to potential policy measures to avoid manipulation. The International Organization of Securities Commissions published its Principles for Financial Benchmarks [19] and the European Union adopted its Benchmarks Regulation. Both documents stress the governance obligations of benchmark administrators, the quality of benchmark data, and most relevantly, robust methodological design of benchmarks. Nonetheless, regulators have neither suggested, nor mandated benchmark design features at a microstructure level of granularity. International regulators' interest in developing best practices for benchmark methodology means there should be substantial interest in the implications of DRL for benchmark manipulation.

The role of regulation is also important because we should not expect markets to produce optimal benchmarks themselves. Index providers do not generally operate in fully competitive markets or internalize the full costs and benefits of the indices they produce. There are several reasons. First, indices are subject to network effects that can cause them to gain a significant degree of lock-in, giving the index provider market power. Second, benchmarks are often produced as a side effect of other financial activity and do not provide their administrators with a robust revenue stream, notwithstanding the benchmark's significant effects on the welfare of counterparties [32]. To illustrate, LIBOR originally arose to serve as a reference rate for banks' own lending activities, but came to play a pivotal role in the enormous interest rate derivatives market, without generating any direct revenue for the LIBOR panel banks. As a result of these forces, administrators' private incentives to ensure optimal benchmark design are weaker than what would be socially desirable.

8 CONCLUSION

We analyze the effectiveness and impact of financial benchmark manipulation, in a simulated market with a single traded security. The manipulator's objective is to shift the benchmark up or down, in order to profit from holdings of a contract tied to the benchmark. The benchmark is transaction-based (VWAP in this study), so potentially influenced by market actions. These actions are costly in that they entail reduced profits or even losses in the primary market. We design and implement three types of benchmark manipulator: one simple hand-crafted strategy, and two derived using deep reinforcement learning.

We find that all three strategies succeed in profitable benchmark manipulation. Presence of a market maker makes manipulation more difficult, and reduces but does not eliminate the manipulative effect. With or without MM, the manipulative activity increases profits of background traders, who thus have no incentives to help mitigate this type of manipulation. Though the aggregate *total* profit of the market participants increases when the benchmark is

manipulated, the aggregate *market* profit decreases. As the profit of all market participants increases, it is the non-market counterparties to the benchmark contracts who bear the burden of the manipulation costs. All of these results hold consistently across a range of experimental market environments.

The DRL agents (DQN and DDPG) effectively learn to manipulate, even though they are not given direct instructions to manipulate, or objectives with explicit reference to manipulation. The manipulative strategies emerge naturally from the selection of standard market actions to maximize profits. These market actions include the possibility of offsetting prices in the direction of benchmark holdings, along with a host of other strategic parameters that can be conditioned on a complex set of state variables. This learning takes place in an environment of other rationally derived trading strategies, and subjected to adjustment based on presence of the manipulator. To our knowledge, this is the first such demonstration of automated learning of market manipulation strategies.

The apparent ease of learning to manipulate presents serious challenges for financial market regulation. Current manipulation law in the US stock market requires establishment of intent to manipulate, which is arguably not present in this scenario. Given the growing accessibility of DRL technique, it may be worth revisiting these laws to address what might be a seen as a "machine learning loophole" for manipulation [2].

REFERENCES

[1] Selim Amrouni, Aymeric Moulin, Jared Vann, Svitlana Vyetrenko, Tucker Balch, and Manuela Veloso. 2021. ABIDES-Gym: Gym environments for multi-agent discrete event simulation and application to financial markets. In *2nd International Conference on Artificial Intelligence in Finance* (New York).

[2] Alessio Azzutti, Wolf-Georg Ringe, and H. Siegfried Stiehl. 2021. Machine learning, market manipulation, and collusion on capital markets: Why the "black box" matters. *University of Pennsylvania Journal of International Law* 43, 1 (2021), 79–135.

[3] Gunjan Banerji. 2018. Regulator looks into alleged manipulation of VIX, Wall Street's 'fear index'. *Wall Street Journal* (2018).

[4] Aurelio F. Bariviera, Belén Guercio, Lisana B. Martinez, and Osvaldo A. Rosso. 2016. Libor at crossroads: Stochastic switching detection using information theory quantifiers. *Chaos, Solitons & Fractals* 88 (2016), 172–182.

[5] Daan Bloembergen, Daniel Hennes, Peter McBurney, and Karl Tuyls. 2015. Trading in markets with noisy information: An evolutionary analysis. *Connection Science* 27, 3 (2015), 253–268.

[6] Philip Bond, Alex Edmans, and Itay Goldstein. 2012. The real effects of financial markets. *Annual Review of Financial Economics* 4, 1 (2012), 339–360.

[7] Catherine Boyle. 2014. Forex manipulation: How it worked. CNBC.

[8] Erik Brinkman. 2018. *Understanding Financial Market Behavior through Empirical Game-Theoretic Analysis.* Ph. D. Dissertation. University of Michigan.

[9] David Byrd. 2022. Learning not to spoof. In *3rd International Conference on Artificial Intelligence in Finance* (New York). 139–147.

[10] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. 2017. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems* 28, 3 (2017), 653–664.

[11] Darrell Duffie. 2018. *Compression Auctions with an Application to LIBOR-SOFR Swap Conversion.* Working Paper 3727. Stanford Graduate School of Business.

[12] Darrell Duffie and Piotr Dworczak. 2021. Robust benchmark design. *Journal of Financial Economics* 142 (2021), 775–802.

[13] Darrell Duffie and Jeremy C. Stein. 2015. Reforming LIBOR and other financial benchmarks. *Journal of Economic Perspectives* 29, 2 (2015), 191–212.

[14] Alexander Eisl, Rainer Jankowitsch, and Marti G. Subrahmanyam. 2017. The manipulation potential of Libor and Euribor. *European Financial Management* 23 (2017), 604–647.

[15] Tyler Gellasch and Chris Nagy. 2019. Benchmark-Linked Investments: Managing Risks and Conflicts of Interest. Healthy Markets Association.

[16] Dhananjay K. Gode and Shyam Sunder. 1993. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* 101, 1 (1993), 119–137.

[17] John M. Griffin and Amin Shams. 2018. Manipulation in the VIX? *Review of Financial Studies* 31 (2018), 1377–1417.

[18] ICE Benchmark Administration Limited. 2019. U.S. Dollar ICE Bank Yield Index. Intercontinental Exchange.

[19] IOSCO. 2013. Principles for Financial Benchmarks. The Board of the International Organization of Securities Commissions.

[20] Yang Li, Wanshan Zheng, and Zibin Zheng. 2019. Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access* 7 (2019), 108014–108022.

[21] Timothy Lillicrap, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations.*

[22] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. 2020. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *34th AAAI Conference on Artificial Intelligence* (New York). 2128–2135.

[23] Enrique Martínez-Miranda, Peter McBurney, and Matthew J. W. Howard. 2016. Learning unfair trading: A market manipulation analysis from the reinforcement learning perspective. In *IEEE Conference on Evolving and Adaptive Intelligent Systems.* 103–109.

[24] James McBride. 2016. Understanding the Libor Scandal. Council on Foreign Relations.

[25] Takanobu Mizuta. 2020. Can an AI perform market manipulation at its own discretion? A genetic algorithm learns in an artificial market simulation. In *IEEE Symposium Series on Computational Intelligence.* 407–412.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.

[27] John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell. 1998. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting* 17, 5–6 (1998), 441–470.

[28] Abhishek Nan, Anandh Perumal, and Osmar R. Zaiane. 2022. Sentiment and knowledge based algorithmic trading with deep reinforcement learning. In *33rd International Conference on Database and Expert Systems Applications.* 167–180.

[29] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. 2006. Reinforcement learning for optimized trade execution. In *23rd International Conference on Machine Learning* (Pittsburgh). 673–680.

[30] E. S. Ponomarev, I. V. Oseledets, and A. S. Cichocki. 2019. Using reinforcement learning in the algorithmic trading problem. *Journal of Communications Technology and Electronics* 64, 12 (2019), 1450–1457.

[31] Bernhard Rauch, Max Goettsche, and Florian El Mouaaouy. 2013. LIBOR manipulation: Empirical analysis of financial market benchmarks using Benford's law. *SSRN Electronic Journal* (2013).

[32] Gabriel Rauterberg and Andrew Verstein. 2013. Index theory: The law, promise and failure of financial indices. *Yale Journal on Regulation* 30 (2013), 1.

[33] L. Julian Schvartzman and Michael P. Wellman. 2009. Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *8th International Conference on Autonomous Agents and Multiagent Systems* (Budapest). 249–256.

[34] Megan Shearer, David Byrd, Tucker H. Balch, and Michael P. Wellman. 2021. Stability effects of arbitrage in exchange traded funds: An agent-based model. In *2nd International Conference on Artificial Intelligence in Finance* (New York).

[35] Alexander A. Sherstov and Peter Stone. 2004. Three automated stock-trading agents: A comparative study. In *AAMAS-04 Workshop on Agent-Mediated Electronic Commerce.* New York, 173–187.

[36] Thibaut Théate and Damien Ernst. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications* 173 (2021).

[37] Karl Tuyls, Julien Perolat, Marc Lanctot, Edward Hughes, Richard Everett, Joel Z. Leibo, Csaba Szepesvári, and Thore Graepel. 2020. Bounds and dynamics for empirical game-theoretic analysis. *Autonomous Agents and Multi-Agent Systems* 34, 7 (2020).

[38] Andrew Verstein. 2015. Benchmark manipulation. *Boston College Law Review* 56 (2015), 215–272.

[39] Elaine Wah. 2016. *Computational Models of Algorithmic Trading in Financial Markets.* Ph. D. Dissertation. University of Michigan.

[40] Elaine Wah, Mason Wright, and Michael P Wellman. 2017. Welfare effects of market making in continuous double auctions. *Journal of Artificial Intelligence Research* 59 (2017), 613–650.

[41] Xintong Wang, Chris Hoang, Yevgeniy Vorobeychik, and Michael P. Wellman. 2021. Spoofing the limit order book: A strategic agent-based analysis. *Games* 12, 2 (2021).

[42] Xintong Wang and Michael P. Wellman. 2020. Market manipulation: An adversarial learning framework for detection and evasion. In *29th International Joint Conference on Artificial Intelligence.* 4626–4632.

[43] Michael P. Wellman. 2016. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems* 30 (2016), 1175–1189.

[44] Mason Wright and Michael P. Wellman. 2018. Evaluating the stability of non-adaptive trading in continuous double auctions. In *17th International Conference*

on Autonomous Agents and Multiagent Systems (Stockholm). 614–622.

[45] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. 2020. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences* 538 (2020), 142–158.

[46] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. 2018. Practical deep reinforcement learning approach for stock trading. arXiv:1811.07522 [cs.LG]

[47] Carter Yagemann, Pak Ho Chung, Erkam Uzun, Sai Ragam, Brendan Saltaformaggio, and Wenke Lee. 2021. Modeling large-scale manipulation in open stock markets. *IEEE Security and Privacy* 19, 6 (2021), 58–65.

[48] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *1st International Conference on Artificial Intelligence in Finance* (New York).

[49] Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2020. Deep reinforcement learning for trading. *Journal of Financial Data Science* 2, 2 (2020), 25–40.