The Rise of Algorithmic Trading: Implications for Price Elasticity and Market Competitiveness*

Ivan Gufler†

Luiss Guido Carli

November 6, 2024

(Click here for the latest version)

Abstract

Financial markets' demand elasticity has decreased substantially over the last 20 years, leading to increased volatility and undermining price stability. This paper investigates the impact of the use of AI traders that make unsupervised trading decisions on aggregate stock elasticity. Building on Haddad et al. (2021), I estimate investors' demand and then I simulate a fictitious financial market populated by artificial intelligence (AI) traders, whose investment decisions are governed by a neural network-based reinforcement learning algorithm. The introduction of AI traders has non-trivial effects on elasticity: on the one hand, AI traders have a high individual-specific elasticity; thus, conditional on trading the asset they increase aggregate elasticity. On the other hand, their high sensitivity to stock price changes implies that they reduce their exposure on the risky asset in downturns. This can reduce aggregate elasticity by up to 2.5%.. The last is larger when other investors demand is more sensitive to aggregate market elasticity.

Keywords: Reinforcement learning, Financial stability, Market efficiency, Algorithmic trading, Asset pricing

*This project started during my visiting period at HEC Paris during Fall 2023. I am grateful to my supervisor Emanuele Tarantino, Loriana Pelizzon, Andrea Pozzi, Francesco Sangiorgi, Paolo Santucci De Magistris. All remaining errors are my own.

†Email: igufler@luiss.it, Website: https://sites.google.com/view/ivangufler

1 Introduction

The finance literature has extensively studied the price elasticity of demand in financial markets, which measures how sensitive prices are to changes in the quantity demanded. This line of research, pioneered by Koijen and Yogo (2019), has shown that price elasticity is lower than what standard asset pricing models predict. Haddad et al. (2021) attribute the decline in price elasticity to the rise of passive investing. A key part of this phenomenon can be explained by the growing costs associated with acquiring, processing, and translating information into investment decisions. Price elasticity is of first order importance for market efficiency, as low elasticity can lead to substantial price movements even in response to small demand or supply shocks, amplifying volatility and reducing price informativeness.

In recent years, the impact of algorithmic trading on financial markets has become a key focus for researchers and policymakers, particularly with the advancement of new technologies such as Artificial Intelligence (AI). The European Securities and Markets Authority (ESMA) defines algorithmic trading as: trading in financial instruments where a computer algorithm automatically determines individual parameters of orders—such as whether to initiate the order, the timing, price, or quantity of the order, or how to manage the order after its submission—with limited or no human intervention.[1] The use of such technology is pervasive in financial markets: 70% of market orders in the US financial markets are made by computers. The ESMA has further reported that the number of investment funds that publicly discloses their use of artificial intelligence or machine learning has increased from 10 to more than 50 between 2017 and 2022, with up to 3 billion euros of assets under management.

In a context of large passive investors and low elasticity, the impact of algorithmic trading governed by advanced artificial intelligence technologies is not trivial. On the one hand, SEC report on Algorithmic Trading in U.S. Capital Markets (SEC, 2020) claims that the investors

[1] The literature in finance associates algorithmic trading with high frequency trading (HFT). The definition of the ESMA is however more general and does not restrict the use of this technology to HFT only.

using algorithmic technologies are more likely to actively trade due to the high computing power (Bai et al., 2016; Dugast and Foucault, 2018; Farboodi et al., 2022) and speed in executing market orders (Biais et al., 2015; Foucault et al., 2016). Thus, algorithmic traders are potentially more elastic than passive investors. However, the presence of more elastic traders is a necessary but not sufficient condition to increase market elasticity. Whether prices become more elastic ultimately depends on whether these agents actually trade stocks. Sornette and Von der Becke (2011) argue that during boom periods, algorithmic traders participate in the market and provide liquidity. However, during bad times, algorithms withdraw liquidity from the market. Whether the participation of algorithmic traders in financial markets, whose demand is governed by advanced artificial intelligence algorithms, increases or decreases demand elasticity remains an open question.

This paper aims to study the impact of algorithmic traders on price elasticity. To this end, I build on Haddad et al. (2021) to estimate the asset demand of U.S. institutional investors and simulate a fictitious financial market, where prices and aggregate stock elasticity are equilibrium outcomes. The contribution of this paper is twofold: from a methodological point of view, I propose a market micro-structure framework to study the decisions made by artificial intelligence traders (AI traders), which trade with a non-negligible price impact. Secondly, I use this theoretical framework to study the properties of the demand of AI traders. Thus, I estimate the individual-specific elasticity of this type of agents. After providing evidence on the trading aggressiveness of AI traders, I examine whether, and under which conditions, they increase aggregate stock elasticity.

Artificial intelligence traders will be governed by a state-of-the-art reinforcement learning algorithm, the Deep Deterministic Policy Gradient (DDPG). The economics and finance literature has mostly focused on the tabular Q-Learning, because of its implementability and interpretability (Calvano et al., 2020; Colliard et al., 2022, among others). Unfortunately, this algorithm is not suitable in very complex environments such as financial markets for several reasons. First, the tabular Q-Learning cannot handle large and highly dimensional state and action spaces. Second, it is not possible to evaluate the optimal strategy learned by the algorithm in unexplored configurations of the state space. Instead, the DDPG algorithm, a neural network-based reinforcement learning technique accommodates both a large state space (the information used to optimize the strategy) and a continuous action space (the actions chosen by the algorithm). The algorithm is trained on a subset of data, and its strategy is then validated in a previously unexplored market environment. To learn its optimal strategy, the algorithm repeatedly trades in a simulated financial market, observing the state (which contains all relevant information) and taking actions to determine the optimal course of action based on its information set. Once the optimal strategy is learned, I test the algorithm's ability to trade in the market using data it has not previously encountered. The goal is to study how the algorithm behaves in a simulated market that closely resembles real-world conditions, how its investment decisions impact price elasticity compared to a benchmark case without AI trader participation, and to what extent other investors adjust their portfolio allocation in response to the presence of algorithmic traders.

I show that the AI trader's individual-specific demand is larger than that of the average investor in the US financial market. In particular, it trades more actively when the degree of strategic response of the representative investor is lower, i.e. when the demand of the latter is less sensitive to other investors portfolio allocation. This last result suggests that the AI trader actively seeks profit opportunities that the representative investor would have left unexploited. Not surprisingly, the presence of an investor with aggressive trading strategy increases aggregate elasticity compared to a market without algorithmic traders. However, this result holds primarily during market boom periods. As the price of the risky asset falls, the AI trader sharply reduces its exposure to the stock, and aggregate elasticity drops by 2.5%. Such drop in market elasticity will cause prices to be more sensitive to investor-specific demand shocks, causing increased volatility and reduced price informativeness (Gabaix and Koijen, 2021). All in all, these effects undermine financial market efficiency and economy's welfare, especially in bad times.

The remaining part of the paper is structured as follows: Section 2 review the relevant literature on the estimation of price elasticity and reinforcement learning for financial applications, Section 3 discusses the methodology to estimate assets demand following Haddad et al. (2021) and simulate the fictitious financial market, and the deep deterministic policy gradient algorithm that governs the decisions of the AI traders, Section 4 describes the results of the simulation and the impact of algorithmic traders. Section 5 concludes.

2 Related literature

This paper aims to contribute to the recent literature on market elasticity of demand and performance by leveraging investor holding data. Koijen and Yogo (2019) propose a structural model that links investor positions in the market with stock characteristics, allowing for heterogeneity across investors. Assuming a fixed and exogenous supply within each period, the model incorporates market clearing (i.e., no excess demand or supply), which enables the construction of counterfactual prices to answer questions such as: How much of volatility and predictability is explained by stock characteristics? What is the expected change in prices if a specific type of investor exits the market? To what extent do unobserved factors influence investors' portfolios? Another example of exercises that can be performed using this class of model is studying the impact of introducing a "large" artificial intelligence trader (i.e., one with significant assets under management). Gabaix and Koijen (2021) further supports the idea that prices in financial markets are relatively inelastic. The study extensively outlines the reasons why low price elasticity is detrimental to financial market performance. When elasticity is low, prices become more sensitive to demand and supply shocks, amplifying fluctuations and increasing volatility. Unlike Koijen and Yogo (2019), who focus on the level of investors' holdings, van der Beck et al. (2022) estimates a first-difference version of the model to investigate the determinants of trades. The paper highlights that demand elasticity varies with the trading horizon and is larger than previously reported. van der Beck (2021) explores the implications of low price elasticity in the context of ESG investing, showing that flows into ESG stocks increase their prices, causing realized returns to deviate from expected returns. A counterfactual experiment demonstrates that, in the absence of flow-driven price pressure on ESG stocks, these stocks would have underperformed during the sample period.

More recently, Haddad et al. (2021) links the decline in aggregate stock elasticity to the rise of passive investing. They propose a two-layer equilibrium model where not only prices but also aggregate elasticity is an equilibrium outcome. Investors trade differently depending on whether they encounter more or less aggressive traders within their investment universe. This feature introduces another layer of heterogeneity: elasticity is stock-specific and depends on the set of investors trading that particular stock. In this paper, I build on Haddad et al. (2021) by simulating a fictitious financial market populated with one or more AI traders and studying how their portfolio decisions affect aggregate stock elasticity.

The role of algorithmic and artificial intelligence traders in financial markets has caught the attention of several scholars. Algorithmic traders are found to be faster than humans in discovering profit opportunities and executing orders. Bai et al. (2016) and Farboodi et al. (2022) examine how these recent trends have impacted price informativeness. Foucault et al. (2016) investigates the extent to which algorithmic trading affects price changes. These types of traders exploit incoming news and profit from it in the short run. Similarly, Biais et al. (2015) emphasizes the negative externalities generated (and not internalized) by institutions using algorithmic trading.

Algorithmic trading in recent years has certainly benefited from advances in artificial intelligence techniques, such as machine learning and reinforcement learning. The latter has been employed to address various financial problems, including finding arbitrage opportunities, pricing securities in market-making frameworks, and solving optimal portfolio problems. Colliard et al. (2022) design a Q-learning algorithm to set prices in a Glosten and Milgrom (1985) model. They find that the algorithm learns to set prices in the presence of adverse selection but fails to increase competitiveness for risky assets and in states it rarely explores.

A growing literature has instead studied the ability of neural network-based reinforcement learning to solve portfolio problems. This class of algorithms is particularly suitable for portfolio optimization as they can: (i) handle very large state spaces, (ii) choose from a large or even continuous action space, and (iii) be validated on out-of-sample data (Yang et al., 2020; Zhang et al., 2020; Vyetrenko and Xu, 2019). However, these contributions typically consider a partial-equilibrium framework where the algorithm does not impact prices in any way.

This work differs from other studies in several aspects. First, the focus is not merely on the performance of the algorithm, nor does it aim to find and calibrate the best-performing algorithm. Instead, the goal is to investigate the impact of AI traders on price elasticity and how changes in price elasticity induce adjustments in the demand for assets by other institutional investors. To this end, I relax the assumption that the AI trader is a marginal investor and allow it to trade with price impact. Second, I do not rely solely on data or theory to shed light on the role of artificial intelligence; rather, I follow the recent literature on structural estimation of asset demand and simulate a financial market with heterogeneous assets and investors, aiming for a representation as close to reality as possible. Then, I populate it with one or more AI trader, whose demand has non-negligible impact on prices.

3 Methodology

In this section, I describe the methodology to construct the fictitious financial market that will be augmented with the presence of AI traders. First, I estimate the demand for assets of US investors following Haddad et al. (2021). Second, I derive equilibrium prices and aggregate elasticity considering a market populated by a representative investor and J AI traders. Third, I describe the AI trader objective function and the deep deterministic policy gradient (DDPG) algorithm that governs its portfolio decisions.

3.1 Estimating individual-specific elasticity

I estimate the demand of stocks for each investor $i = 1, ..., I$ following Haddad et al. (2021):

$$\log \delta_{i,n,t} - p_{nt} - \chi \mathcal{E}_{agg,n,t} p_{nt} = \underline{d}_{0,i,t} + \underline{d}'_{1,i,t} X_{nt} \\ - \left(\underline{\mathcal{E}}_{0,i,t} + \underline{\mathcal{E}}'_{1,i,t} X^e_{n,t}\right) p_{nt} + \epsilon_{int} \tag{1}$$

where $\delta_{i,n,t}$ is the portfolio allocation of investor i on asset n at time t with respect to the outside asset (cash, bond, ...), $p_{n,t}$ is the log price of stock n at time t, $X^d_{n,t}$ is a matrix of stock's n characteristics including log book equity and log book equity squared, $X^e_{n,t}$ is the log book equity of stock n at time t and $\epsilon_{i,n,t}$ is the latent demand capturing unobserved factors. The aggregate elasticity is $\mathcal{E}_{agg,n,t}$ and is defined as:

$$\mathcal{E}_{agg,n,t} = \frac{\sum_{i=1}^{N} w_{i,n,t} \, A_{i,t} \, \underline{\mathcal{E}}_{i,n,t}}{P_{n,t} + \chi \sum_{i=1}^{N} w_{i,n,t} A_{i,t}} \tag{2}$$

where $\underline{\mathcal{E}}_{i,n,t} = \underline{\mathcal{E}}_{0,i,t} + \underline{\mathcal{E}}_{1,i,t} X^e_{n,t}$ denotes the the individual-specific elasticity.

The regression in Eq. 2 suffers from endogeneity as prices appear both on the left- and right-hand side. Therefore, I estimate Eq. 2 using 2SLS, where $p_{n,t}$ is instrumented by $\hat{p}_{n,t}$. The instrument for investor i is the log market capitalization as if all other investors j held an equal weighted portfolio within their investment universe. This counterfactual price is not affected by investor's i investment decisions. Using the same reasoning, I construct an instrument for $\mathcal{E}_{agg,n,t}$.

I estimate the coefficients $\underline{d}_{0,i,t}$, $\underline{d}_{1,i,t}$, $\underline{\mathcal{E}}_{0,i,t}$ and $\underline{\mathcal{E}}_{1,i,t}$ conditional on χ, that represents the degree of strategic interaction. This coefficient of the interaction between aggregate stock elasticity and log price tells how sensitive are the portfolio choices of investors relative to aggregate market elasticity. To gain more insights on how the AI trader trades under different market conditions, I estimate Eq. 1 conditional on $\chi = 1$ and $\chi = 2.5$. The latter is similar to the average degree of strategic response reported by Haddad et al. (2021).

Individual-specific coefficients are estimated following the procedure of Haddad et al.

(2021): I start with a guess for $\mathcal{E}_{agg,n,t}$ and a value for χ and run the cross sectional regression in Eq. 1. Given the coefficients $\underline{\mathcal{E}}_{0,i,t}$ and $\underline{\mathcal{E}}_{1,i,t}$ I update the value of $\mathcal{E}_{agg,n,t}$ using Eq. 2. I repeat these two steps until the value of aggregate stock elasticity converges. This ensures that the coefficients are consistent with the estimated aggregate elasticity.

Data I estimate the model on the US equity market from 2000:Q1 to 2022:Q4. Data on stocks are from CRSP and Compustat. Investor holdings have been retrieved from Thomson Reuters Institutional Holdings Database at quarterly frequency and include all US investors with at least one hundred thousand dollars assets under management. CRSP-Compustat data are merged using CUSIP number. Active investors that hold fewer than 1000 stocks in a quarter are pooled together, in groups that hold 2000 stocks on average. Furthermore, shares outstanding are normalized to be equal to one, so that the price $P_{n,t}$ represents the market capitalization.[2]

3.2 Asset Prices with a Representative Agent

I consider a market for N risky assets, with price and dividends denoted by $P_{n,t}$ and $D_{n,t}$ and shares outstanding equal to $S_{n,t}$. I normalize the outstanding shares to one, so that they are constant over time. A risk-free bond is also available with exogenous and constant gross return R_f.

The market is populated by a representative investor and J AI traders. The $j-th$ AI trader have shares held on asset n equal to $S^j_{n,t}$. The residuals shares are bought by the representative investor and denoted by $\tilde{S}_{n,t} = S_n - \sum_{j=1}^{J} S^j_{n,t}$.

The representative investor has demand:

$$\log \delta_{n,t} - p_{n,t} = d_0 + d_1 X^d_{n,t} - (\mathcal{E}_0 + \mathcal{E}_1 X^e_{n,t} - \chi \mathcal{E}_{agg,n,t}) p_{n,t} \tag{3}$$

$$w_{n,t} = \frac{\delta_{n,t}}{1+\sum_{m=1}^{N} \delta_{n,t}} \tag{4}$$

[2]The dataset is constructed following Koijen and Yogo (2019) and I apply the weighting scheme during the estimation as in Haddad et al. (2021).

where $w_{n,t}$ is its portfolio weight on asset n.

At every point in time market clearing requires that the total amount invested in each stock by the representative investor and by the AI traders equals the stock's market capitalization. Formally:

$$A_t w_{n,t} + P_{n,t} \sum_{j=1}^{J} S_{n,t}^{j} = P_{n,t} S_n, \, \forall n, t \tag{5}$$

The previous condition can be expressed in terms of residual demand of the representative investor:

$$A_t w_{n,t} = P_{n,t} \tilde{S}_{n,t} \tag{6}$$

To derive the equilibrium prices I use the market clearing condition and Eq. 3:

$$\underline{d}_0 + \underline{d}_1 X_{n,t}^{d} + [(1 - \underline{\mathcal{E}}_0) + \chi \mathcal{E}_{agg,n,t}] p_{n,t} = p_{n,t} + \tilde{s}_{n,t} - \log(A_t - \sum_{m=1}^{N} p_{n,t} \tilde{s}_{n,t}) \tag{7}$$

The wealth of the representative agent evolves according to the law of motion:

$$A_t = A_{t-1} R_f + \sum_{m=1}^{N} \tilde{S}_{m,t-1} (P_{m,t} + D_{m,t} - R_f P_{m,t-1}) \tag{8}$$

Substituting the latter into 7 and rearranging terms:

$$\begin{aligned} 0 = \underline{d}_0 + \underline{d}_1 X_{n,t}^{d} + (1 - \underline{\mathcal{E}}_0) p_{n,t} + \chi \mathcal{E}_{agg,n,t} p_{n,t} - p_{n,t} - \tilde{s}_{n,t} + \epsilon_{n,t} + \\ \log(A_{t-1} R_f + \sum_{m=1}^{N} \tilde{S}_{m,t} (D_{m,t} - R_f P_{m,t-1}) - \sum_{m=1}^{N} P_{m,t} \Delta \tilde{S}_{m,t}) \end{aligned} \tag{9}$$

Eq. 9 is a system of N equation in N unknowns $(p_{1,t}, p_{2,t}, ..., p_{N,t})$ that can be solved numerically and represents equilibrium price after accounting for the representative agent and AI traders demand.[3]

[3] For the numerical computation I impose the following bounds in order to meet the existence conditions

Equilibrium computation Notably, the demand of the representative investor (Eq. 3) is function of the aggregate elasticity, which in turn depends on the allocation chosen by investors. The solution of equilibrium prices in Eq. 9 embeds a fixed point problem for the weights chosen by the representative investor. To overcome this issue, I start with a guess for the portfolio weights of the representative investor, $w_{n,t}$ and for the current price $P_{n,t}$. Conditionally on these values, I compute the current period assets under management of the representative investor and the aggregate elasticity, $\mathcal{E}_{agg,n,t}$. Subsequently, I solve for the equilibrium prices and compute the representative investor demand as in Eq. 3. I repeat this procedure until the value of $w_{n,t}$ converge. The algorithm for the equilibrium computation is reported below.

Are the prices indeed equilibrium prices? To answer this question, I verify that prices satisfy the market clearing condition at any time. In other words, I compute the difference between the amount of dollars invested in each stock by the representative investor and (eventually) the AI trader and the market capitalization. The difference of these two quantities is always in the order of 10^{-10}.[4]

for the two log terms in Eq. 9

$$\begin{cases} P_{n,t} > \dfrac{\sum_{j=1}^{J} \theta_{n,t}^{j} AUM_t^j}{S_n} \\ \sum_{m=1}^{N} P_{m,t} < \dfrac{A_{t-1} R_f + \sum_{m=1}^{N} \tilde{S}_{m,t-1} \left(D_{m,t} - R_f P_{m,t-1}\right) + \sum_{m=1}^{N} \sum_{j=1}^{J} \theta_{m,t}^{j} AUM_t^j}{\sum_{j=1}^{J} S_{m,t-1}^{j}} \end{cases}$$

[4]The distribution of the difference between allocation of the agents and the stock's market clearing is reported in Figure 4.

Algorithm 1 Equilibrium computation

Initialize starting values w^0 and p^0
h $\leftarrow 0$
while $(\|w^h - w^{h-1}\| > \text{tol})$ **or** $(h = 0)$ **do**
  Compute A_t^h conditional on p^h
  Compute $\mathcal{E}_{agg,n,t}$ conditional on w^h, p^h and A_t^h
  Solve equilibrium prices p^{h+1} (Eq. 9)
  Compute RI's portfolio weights as $w^* \leftarrow \frac{\delta_{n,t}}{1+\sum_{m=1}^{N} \delta_{n,t}}$
  $w^{h+1} \leftarrow (w^* \times 0.5 + w^h \times 0.5)$
  $h \leftarrow h + 1$
end while
return p^{h+1}, $\mathcal{E}_{agg,n,t}^h$

Calibration of the representative investor and stock characteristics To calibrate the demand coefficients of the representative investor I aggregate the coefficients $\underline{d}_{0,i,t}$, $\underline{d}_{1,i,t}$, $\underline{\mathcal{E}}_{0,i,t}$ and $\underline{\mathcal{E}}_{1,i,t}$ estimated in the previous section using an AUM-weighted average across investors. Then, I obtain time-invariant coefficient by the average over time. Formally, the coefficients of the representative investor are given by:

$$\bar{x} = \frac{1}{T} \sum_{i=1}^{I} \frac{AUM_{i,t}}{\sum_{i=1}^{I} AUM_{i,t}} x_{i,t}$$

where x is alternatively $\underline{d}_{0,i,t}$, $\underline{d}_{1,i,t}$, $\underline{\mathcal{E}}_{0,i,t}$ and $\underline{\mathcal{E}}_{1,i,t}$. The stock characteristic log book equity is simulated from an AR(1) process fitted on data. Further details are provided in Appendix A.

3.3 AI trader

The AI trader maximizes the flow profit:

$$\mathbb{E} \sum_{t=1}^{\infty} \delta^t \log(\pi_t) \tag{10}$$

where $\delta^t \in (0, 1]$ is the discount factor and π_t is the portfolio return, defined as:

$$\pi_t = R_f + \sum_{n=1}^{N} \theta_{n,t-1}(R^*_{n,t} - R_f) \tag{11}$$

where $R^*_{n,t} = \frac{P^*_{n,t}+D_{n,t}}{P_{n,t-1}}$. At the time the AI trader decides the portfolio allocation $\theta_{n,t-1}$ the price $P_{n,t}$ has not realized yet. Consequently, the portfolio return is calculated using $P^*_{n,t}$, i.e. the price that would prevail at time t if the AI trader does not change the number of shares held. I refer to $P^*_{n,t}$ as the "beginning of period price". $\theta_{n,t}$ is the portfolio weight on the N risky assets. The AI trader's wealth evolves according to:

$$A^j_t = A^j_{t-1} \times \pi_t \tag{12}$$

The AI trader optimizes the portfolio weights $\theta_{n,t}$ and $\theta_{R_f,t}$, namely the weight on the risk-free bond, such that $\theta_{n,t} + \theta_{R_f,t} = 1$ and $\theta_{n,t}, \theta_{R_f,t} \geqslant 0 \; \forall n, t$. In other words, the AI trader allocates its AUM on the risky and risk-free asset and is not allowed to take short positions on any asset.[5]

3.4 Deep deterministic policy gradient

Following Yang et al. (2020), the AI trader is governed by a deep deterministic policy gradient (DDPG) algorithm (Lillicrap et al., 2015), which incorporates the advantages of the actor-critic algorithm and the deep q-learning (DQN) approach. The choice is driven by the fact that the DDPG is particularly suitable to solve financial problems, such as trading or portfolio optimization, as it can handle very large state spaces and optimize over a continuous action space (Yang et al., 2020).

The DDPG algorithm is composed by two neural networks, a *critic network*, Θ^Q, which takes as input the state and approximates the Q-values, and an *actor network*, Θ^μ, which

[5]The short-selling constraint is consistent with the data used to estimate Eq. 1. The 13F filings of the SEC only include long positions.

uses the Q-values to optimize the continuous actions. For each of the two a *target* copy of the network is constructed and slowly updated, in order to achieve more stability during the training phase. The target networks are denoted by $\Theta^{Q'}$ and $\Theta^{\mu'}$ for the critic and actor, respectively. Moreover, it uses a replay buffer to train off-policy from batches randomly drawn from this set. Following Ioffe and Szegedy (2015) and Lillicrap et al. (2015) we add batch normalization layers to the networks.

At each point in time the DDPG observes the current state ω, chooses action(s) a and receives a reward π. Finally, the next state ω' is revealed. In order to explore the environment a noise is added to the action chosen by the agent. For the exploration, noise is drawn from Dirichlet, such that actions are non-negative and sum up to one.[6]

The transition $\{\omega, a, \pi, \omega'\}$ is stored in the replay buffer $\mathcal{B}$ and a batch of size $|B|$ of transitions is randomly drawn from $\mathcal{B}$.

The target Q-values are computed on the batch as follows:

$$y\left(\pi, \omega'\right) = \pi + \gamma \Theta^{Q'}\left(\omega', \Theta^{\mu'}\left(\omega'\right)\right)$$

where γ is the discount factor.

The loss for the critic network is then given by:

$$L\left(\theta^{Q}\right) = \left[\left(y - Q\left(\omega, a \mid \Theta^{Q}\right)\right)^{2}\right]$$

To update the critic network we minimize the loss between the Q-values computed using the target network, $y\left(\pi, \omega'\right)$ and those obtained from the online one, with respect to the critic parameters.

[6]The Dirichlet distribution of a multivariate generalization of the Beta distribution. Formally $\mathrm{Dic}(\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$ and $B(\alpha)$ is the multivariate Beta function: $B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}$. Sampling random actions from this distribution guarantees that *i*) $\bar{x}_i \in [0, 1]$ and *ii*) $\sum_{i=1}^{K} x_i = 1$.

$$\nabla_Q \frac{1}{|B|} \sum_{(\omega,a,\pi,\omega',d)\in B} -L\left(\Theta^Q\right)$$

For the actor network we maximize the expected return, defined as:

$$\mathbb{J}(\theta) = \mathbb{E}[Q(\omega,a)|_{\omega=\omega_i,a_t=\mu(\omega_t)}]$$

To do this we compute the gradient of $\mathbb{J}(\Theta)$ with respect to the policy parameters:

$$\nabla_{\Theta^\mu}\mathbb{J}(\Theta) \approx \nabla_a Q(\omega,a)\nabla_{\Theta^\mu}\mu(\omega|\Theta^m u)$$

Finally, the two target networks are soft updated using the parameters of the online actor and critic networks:

$$\Theta^{\mu'} \leftarrow \tau\Theta^{\mu'} + (1-\tau)\Theta^{\mu}$$
$$\Theta^{Q'} \leftarrow \tau\Theta^{Q'} + (1-\tau)\Theta^{Q}$$

where $\tau \ll 1$ controls the rate at which the target networks are updated.

The calibration of hyperparameters reported in Table 1 is in line with Lillicrap et al. (2015):

Table 1: Calibration of DDPG hyperparameters

Parameter	**Value**	**Description**
τ	0.001	target update rate
α	0.0001	actor learning rate
β	0.001	critic learning rate
B	64	batch size
$\bar{B}$	10^5	replay buffer size
γ	0.99	discount rate

Notes: This table reports the hyperparameters of the Deep Deterministic Policy Gradient (DDPG) algorithm.

4 Simulation

In this section I describe the results of the simulation exercise when introducing one AI trader. Within the simulation, prices and aggregate stock elasticity are endogenously determined in equilibrium. To compute the aggregate stock elasticity one must know the demand coefficients $\mathcal{E}_{0,i}$ and $\mathcal{E}_{1,i}$. As first exercise, I estimate these demand for the AI trader, assuming it to be small enough not to have an impact on prices and aggregate stock elasticity. In the subsequent section, I investigate the effect on aggregate stock elasticity when introducing a "large" AI trader, i.e. with sufficient assets under management to impact equilibrium quantities with its portfolio decisions.

4.1 Simulation design

Each simulation runs for 500 episodes during which the AI trader observes the current state, takes an action, observes the reward and finally the next state is revealed. An episode involves 65 time-steps. During a simulation, the AI trader explores the environment over and over, updating its optimal strategy in order to maximize the objective function.

At the start of each simulation, the initial parameters of the neural network are randomly initialized. Additionally, the exploration of the environment involves randomness. To ensure that this randomness does not influence the results, I conduct 20 simulations and report the average statistics along with a confidence interval.

4.2 AI traders individual-specific elasticity

To compute the aggregate stock elasticity $\mathcal{E}_{agg,n,t}$, it is necessary to know the individual-specific elasticity of each investor. However, to estimate the latter, one must know the value of $\mathcal{E}_{agg,n,t}$. For the representative investor, these coefficients are identified by following the iterative procedure of Haddad et al. (2021) described in Section 3. For the AI trader, I simulate a fictitious financial market and assume the AI trader has no impact on prices, and thus on aggregate stock elasticity. In other words, the AI trader has zero AUM. I let the AI trader train for 500 episodes. Using the portfolio allocation, equilibrium prices, and aggregate stock elasticity at the last episode, I estimate the AI trader by running the following time-series regression:

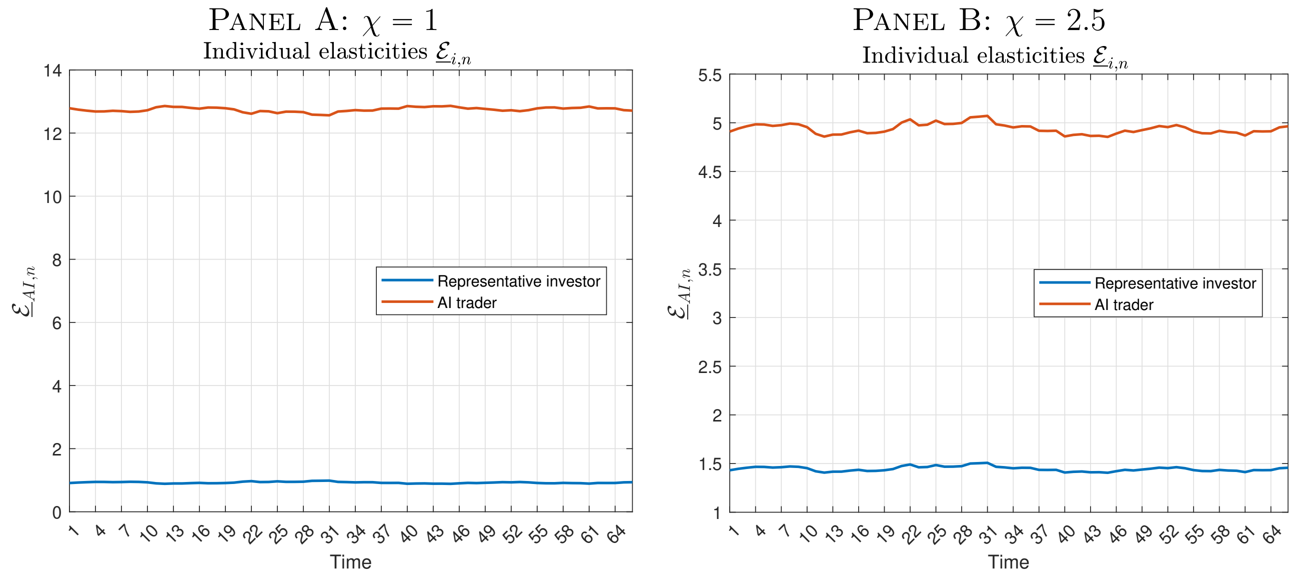
$$\log \delta_t - p_t - \chi \mathcal{E}_{agg,t} p_t = \underline{d}_{0,i,t} + \underline{d}_1' X_t^d - \left(\underline{\mathcal{E}}_0 + \underline{\mathcal{E}}_1' X_t^e \right) p_t + \epsilon_t \tag{13}$$

The individual-specific elasticity $\underline{\mathcal{E}}_{i,n,t} = \underline{\mathcal{E}}_{0,i,t} + \underline{\mathcal{E}}_{1,i,t} X_{n,t}^e$ for the AI trader and representative investor are shown in Figure 1. I estimated the parameters of demand for both investors conditional on a rather low value of strategic response, $\chi = 1$ and on a more realistic value, $\chi = 2.5$, as reported in Haddad et al. (2021). Representative investor and AI trader have positive $\underline{\mathcal{E}}_{i,n,t}$. No matter how competitive is the environment, i.e. how large the degree of strategic response, the AI trader is always more elastic than the representative investor, especially when the degree of strategic response is lower. Thus, conditional on trading the AI trader is likely to increase market elasticity. However, precisely because of its aggressive trading strategy the AI trader might reduce its exposure during market downturns, causing a drop in market elasticity.

4.3 Impact on aggregate stock elasticity

Once the individual-specific elasticity of the representative investor and the AI trader have been estimated I simulate a financial environment where the AI trader has impact on equilibrium quantities to study how its portfolio allocation shapes aggregate stock elasticity. I compare $\mathcal{E}_{agg,n,t}$ with the benchmark case where only the representative investor populates the market. When introducing the AI trader, I assume the total asset under management of the market remain unchanged. Put it differently, around 10% of the AUm of the representative investor are now managed by the AI trader. This exercise represents the situation in which institutional investor let an algorithm take investment decision without affecting the size of

Figure 1: Individual-specific elasticity



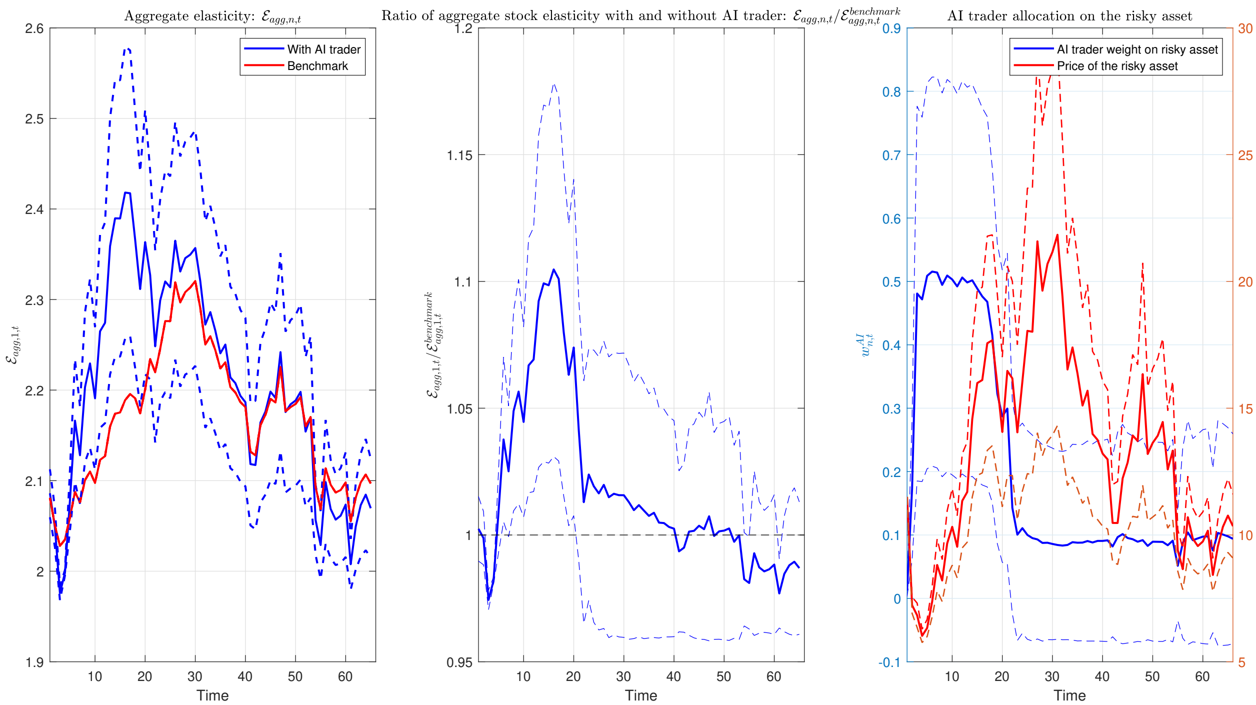
Notes: This figure reports the individual-specific elasticity of the representative investor (blue solid line) and of the AI trader (orange solid line),conditional on a low value of strategic response ($\chi = 1$) and a mid value of ($\chi = 2.5$). The coefficients of the representative investor are estimated investor-by-investor following Haddad et al. (2021) and aggregated by AUM-weighted average. The coefficients of the AI trader are estimated from simulation results assuming no price impact and according to Eq. 13

the market.

Figure 2 reports the aggregate stock elasticity with and without AI trader (left panel), the ratio of the two (mid panel) and the AI trader allocation on the risky asset (right panel) when the degree of strategic response, χ, is relatively low and equal to one. The aggregate elasticity increases in both cases, with and without AI trader, as the log book equity of the stock rises. However, in the first part the of the sample, the rise is significantly more pronounced when the AI trader participates to the market. Indeed, the allocation on the risky asset is around 50% up to time t. As the AI trader reduce its exposure to the stock, aggregate elasticity drops. Recall from 3, the representative investor demand depends positively on $\mathcal{E}_{agg,n,t}$: when this reduces it shifts its AUM to the risk-less asset, further amplifying the reduction of elasticity. This is indeed the case when the price of the stock drops (red line in the right panel) and the AI trader increases its portfolio weight on the risk-less bond. By the end of the simulation period aggregate elasticity is 2.6% lower than in the benchmark case.

In figure 3 I report the simulation results when χ is equal to 2.5, a value in line with the

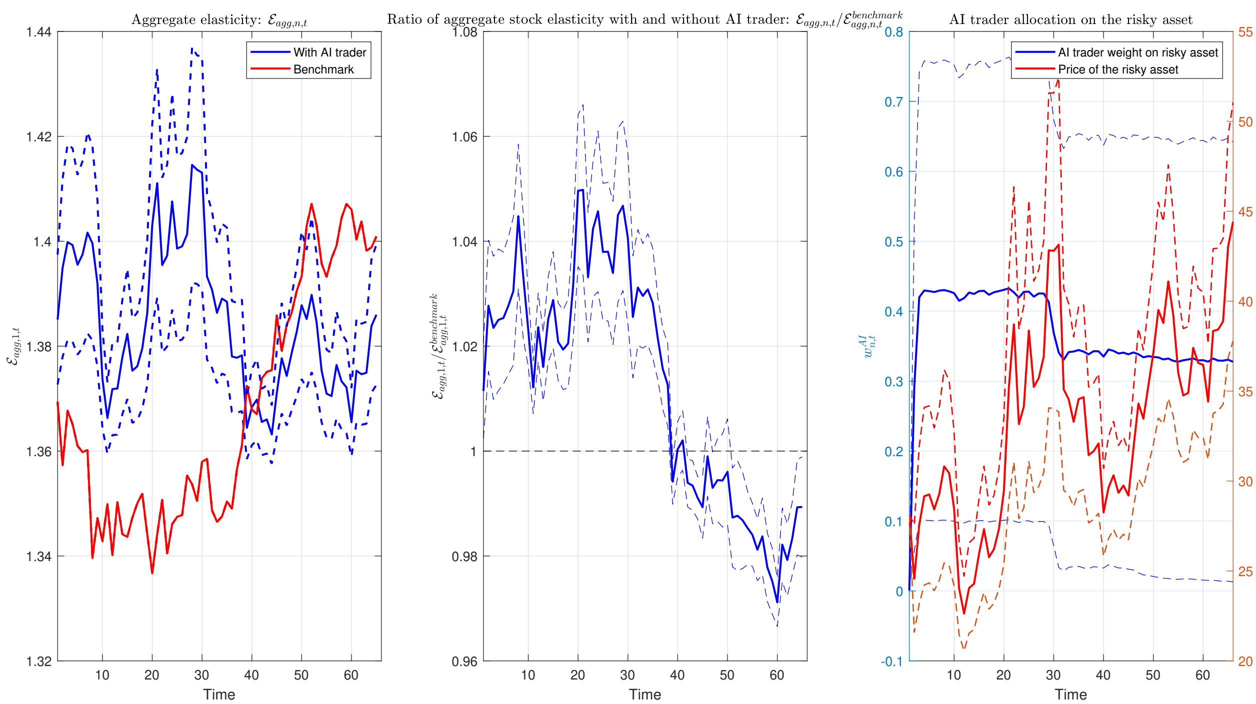
Figure 2: Aggregate stock elasticity and AI trader portfolio allocation ($\chi = 1$)



Notes: This figure reports *i*) the aggregate stock elasticity with and without AI trader (left panel), *ii*)the ratio of aggregate stock elasticity with and without AI trader (mid panel), the allocation of the AI trader and price of the risky asset (right panel). Individual-specific elasticity is estimated conditional on $\chi = 1$. Dashed lines denote the 90% confidence interval across 20 simulations.

average degree of strategic response in Haddad et al. (2021). Aggregate elasticity is higher by up to 5% with respect to the benchmark case as the AI trader is active in the market, i.e. when invests in the risky asset. From period 29, the AI trader allocates less on the stock causing a drop in aggregate elasticity. As this quantity reduces the representative investor responds to this changes by investing more in the outside asset. Compared to Fig. 2, the drop in aggregate elasticity is more pronounced (-3% vs. -2.6%): the parameter χ in the demand function of the representative investor controls the sensitivity of portfolio allocation to changes in aggregate stock elasticity. Even though the AI trader reduces its allocation to the risky asset less compared to the case of low strategic response, the impact on aggregate elasticity is amplified by the parameter χ.

Figure 3: Aggregate stock elasticity and AI trader portfolio allocation ($\chi = 2.5$)



Notes: This figure reports *i*) the aggregate stock elasticity with and without AI trader (left panel), *ii*)the ratio of aggregate stock elasticity with and without AI trader (mid panel), the allocation of the AI trader and price of the risky asset (right panel). Individual-specific elasticity is estimated conditional on $\chi = 2.5$. Dashed lines denote the 90% confidence interval across 20 simulations.

5 Conclusions

In the last 20 years, U.S. financial markets have been characterized by reducing aggregate elasticity, leading to increased volatility and lower price informativeness (Haddad et al., 2021). This phenomenon can be partly explained by the rise of passive investing, which, in turn, is a consequence of increased costs of information processing. Almost simultaneously, algorithmic trading has become more pervasive in financial markets, accounting for around 70% of market orders in recent years (SEC, ESMA). This technology overcomes the costs of acquiring, processing, and translating information into investment decisions and is thus used (also) for active trading.

Whether financial markets would benefit from increased aggregate elasticity due to the presence of algorithmic trading remains an open question for both researchers and policymakers. In this paper, I study the impact of artificial intelligence (AI) trading on financial markets' aggregate elasticity to shed light on whether and under which conditions AI traders provide liquidity to the market or withdraw it instead. Building on Haddad et al. (2021), I simulate a simplified financial market populated by a representative investor, whose demand is calibrated from the data, and an AI trader. The impact on aggregate elasticity is not obvious a priori: on the one hand, the individual-specific elasticity of the AI trader (i.e., how aggressively it trades) is positive and mostly larger than that of the representative investor. On the other hand, the ultimate effect on aggregate elasticity depends on whether and to what extent the AI trader allocates to the risky asset.

Results show that during boom market periods, the AI trader increases the aggregate elasticity of the market compared to the benchmark simulation where only the representative investor trades. However, as soon as prices decline, the AI trader reduces its exposure to the risky asset and thus the market elasticity. In turn, the representative investor reacts to the change in aggregate elasticity, further reducing its demand. This effect is amplified when the degree of strategic response is large, i.e., when agents react sharply to changes in aggregate elasticity. Overall, the estimated reduction is around 3%.

The model illustrates the channels that determine the level of aggregate elasticity, the individual-specific elasticity, and the allocation to the risky asset. Next steps in this work would include enriching the model with a larger number of assets and AI traders. The goal would be to shed light on the interactions between AI traders' allocations and how these would impact aggregate elasticity. In these exercises, the degree of strategic response is taken as given, and together with the assumption of the presence of a representative investor, this limits the realism of the model as well as the study of how AI traders shape market competitiveness. To this end, the next step is to simulate a financial market that, without AI traders, replicates the market statistics we observe in the data. Within this framework, I will study the role of AI traders for aggregate elasticity and how these agents impact market competitiveness through the degree of strategic response.

References

Bai, J., T. Philippon, and A. Savov (2016): "Have financial markets become more informative?" Journal of Financial Economics, 122, 625–654.

Biais, B., T. Foucault, and S. Moinas (2015): "Equilibrium fast trading," Journal of Financial economics, 116, 292–313.

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2020): "Artificial Intelligence, Algorithmic Pricing, and Collusion," American Economic Review, 110, 3267–3297.

Colliard, J.-E., T. Foucault, and S. Lovo (2022): "Algorithmic Pricing and Liquidity in Securities Markets," HEC Paris Research Paper.

Dugast, J. and T. Foucault (2018): "Data abundance and asset price informativeness," Journal of Financial economics, 130, 367–391.

Farboodi, M., A. Matray, L. Veldkamp, and V. Venkateswaran (2022): "Where has all the data gone?" The Review of Financial Studies, 35, 3101–3138.

Foucault, T., J. Hombert, and I. Roşu (2016): "News trading and speed," The Journal of Finance, 71, 335–382.

Gabaix, X. and R. S. Koijen (2021): "In search of the origins of financial fluctuations: The inelastic markets hypothesis," Tech. rep., National Bureau of Economic Research.

Glosten, L. R. and P. R. Milgrom (1985): "Bid, ask and transaction prices in a specialist market with heterogeneously informed traders," Journal of financial economics, 14, 71–100.

Haddad, V., P. Huebner, and E. Loualiche (2021): "How competitive is the stock market? theory, evidence from portfolios, and implications for the rise of passive investing," Tech. rep.

Ioffe, S. and C. Szegedy (2015): "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, pmlr, 448–456.

Koijen, R. S. J. and M. Yogo (2019): "A Demand System Approach to Asset Pricing," Forthcoming at *Journal of Political Economy*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015): "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971.

SEC (2020): "Staff report on algorithmic trading in US Capital Markets," Staff of the US Securities and Exchange Commission.

Sornette, D. and S. Von der Becke (2011): "Crashes and high frequency trading," Swiss Finance Institute Research Paper.

van der Beck, P. (2021): "Flow-driven ESG returns," Swiss Finance Institute Research Paper.

van der Beck, P. et al. (2022): "On the Estimation of Demand-Based Asset Pricing Models," Tech. rep., Swiss Finance Institute.

Vyetrenko, S. and S. Xu (2019): "Risk-Sensitive Compact Decision Trees for Autonomous Execution in Presence of Simulated Market Response," ArXiv, abs/1906.02312.

Yang, H., X.-Y. Liu, S. Zhong, and A. Walid (2020): "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," Tech. rep.

Zhang, Z., S. Zohren, and S. Roberts (2020): "Deep reinforcement learning for trading," The Journal of Financial Data Science, 2, 25–40.

A Simulating stock characteristics

For the purpose of constructing a fictitious financial market I select 10 stocks that are alive from 1982Q1 to 2022Q4. Details on the stocks are provided in Table 2:

Table 2: Stock names and industry

Ticker	**Name**	**Industry**
IBM	International Business Machines Corporation	Technology & Consulting
AXP	American Express	Financial Services
CPB	Campbell Soup Company	Food & Beverage
GE	General Electric Company	Conglomerate / Industrial
XOM	Exxon Mobil Corporation	Oil & Gas
INTC	Intel Corporation	Technology & Semiconductors
KO	The Coca-Cola Company	Food & Beverage
PFE	Pfizer Inc.	Pharmaceuticals
WMT	Walmart Inc.	Retail
XRX	Xerox Holdings Corporation	Technology & Document Solutions

Stock characteristics, log book equity and dividends are simulated stock-by-stock from the data. The simulated series are then used to train (and validate) the algorithm.

Log book equity For each each I consider the time period from 1982Q1 to 2022Q4 and fit an AR(1) process. Using the estimated autoregressive coefficient and the variance of innovation, I simulate the evolution of log book equity. Summary statistics are provided in Table 3.

Dividends To maintain the dividend-to-price ratio consistent with the data, I compute model $D_{n,t}/P_{n,t-1}$. For each stock, I account for the frequency of dividend payments and model it from a uniform distribution. The dividend-to-price payment is modelled as a truncate normal. Table 3 reports the mean and standard deviation of dividend-to-price of the data and simulated series.

Table 3: Characteristics, latent demand and returns empirical moments

log book equity	**IBM**	**AXP**	**CPB**	**GE**	**XOM**	**INTC**	**KO**	**PFE**	**WMT**	**XRX**
Data (mean)	6.420733	10.98275	6.819517	7.879842	5.751793	5.10696	8.974074	8.935129	5.614576	4.445884
Simulation (mean)	4.825207	9.801349	5.803029	7.094325	3.375961	2.903987	8.226762	7.751597	5.405748	2.76012
Data (std. dev.)	0.927985	0.939825	0.560494	0.609443	0.845046	1.681084	0.93541	0.577574	0.693133	0.892253
Simulation (std. dev.)	0.902681	1.95134	2.370549	2.316138	1.160787	1.09987	3.509188	1.5197	0.976696	0.897199
Dividends to P_{t-1}	**IBM**	**AXP**	**CPB**	**GE**	**XOM**	**INTC**	**KO**	**PFE**	**WMT**	**XRX**
Data (mean)	0.089816	0	0.002855	0.038576	0	0.082899	0.568473	0	0.001596	0.00343
Simulation (mean)	0.090149	0	0.002637	0.038171	0	0.172441	0.623209	0	0.001709	0.003302
Data (std. dev.)	0.222077	0	0.03645	0.17225	0	0.463557	0.68346	0	0.014612	0.043796
Simulation (std. dev.)	0.22094	0	0.034932	0.170494	0	0.493821	0.675759	0	0.015512	0.042841

Notes: This table reports the mean and standard deviation of log book equity and dividend-to-price ratio of data and simulated series. Sample: 1982Q1 – 2022Q4.

B Additional results

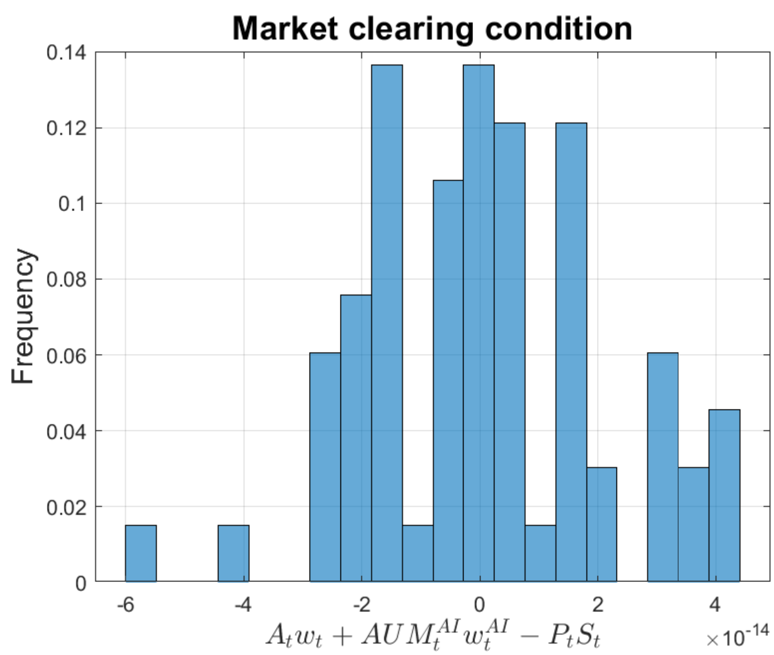
B.1 Market clearing condition

At every point in time I compute the market clearing condition, defined as:

$$A_t w_t + AUM_t^{AI} w_t^{AI} - P_t S_t$$

Put it differently, I verify that sum of allocation on each stock equals the market capitalization. Figure 4 reports the distribution of the market clearing condition. Notice that the scale of the figure is 10^{-14}, which is within machine precision.

Figure 4: Market clearing



Notes: This figure reports the distribution of the difference between sum of allocation on each stock and the market capitalization.