Algorithmic Collusion in Electronic Markets: The Impact of Tick Size

Álvaro Cartea[a,b], Patrick Chang[a], José Penalva[c,a]

[a] Oxford-Man Institute of Quantitative Finance
[b] Mathematical Institute, University of Oxford
[c] Universidad Carlos III

Abstract

We characterise the stochastic interaction of learning algorithms as a deterministic system of differential equations to understand their long-term behaviour in a repeated game. In a symmetric bimatrix repeated game, we prove that the dynamics of many learning algorithms converge to the outcomes of pure strategy Nash equilibria of the stage game. In market making, we show that the algorithms tacitly collude to extract rents and tick size (coarseness of price grid) matters: a large tick size obstructs competition, while a smaller tick size lowers trading costs for liquidity takers, but slows the speed of convergence to an equilibrium.

Keywords: Artificial Intelligence, Tacit Collusion, Evolutionary Game Theory, Market Making, Limit Order Books, Tick Size

> *[...] perfect competition cannot be taken for granted, even on transparent open limit order books with a very thin pricing grid*
>
> Biais, Bisière and Spatt (2010)

1. Introduction

In electronic markets, collusion can arise in many forms. For instance, Christie and Schultz (1994) showed that market makers avoided odd-eighth quotes, an outcome that was perceived as evidence of tacit collusion (and was a strong force behind the introduction of decimalisation in the year 2000). These days, as markets rely more and more on computerised trading, a novel and growing literature examines how algorithms behave when they compete amongst themselves and learn as they interact with each other. It is not clear if these interactions will result in competitive or collusive outcomes. Recent literature finds that the strategic interactions of the algorithms can lead to tacit collusion, see Calvano et al. (2020); yet, the mechanisms that produce these outcomes are not fully understood.

We thank seminar participants at the Federal Reserve Board, D.C., the University of Oxford and ETH Zürich, the 3rd Annual MSQA London Quant Conference, and the 11th World Congress of the Bachelier Finance Society. We would like to thank Sida Li, Roel Oomen, Clara Vega, and Harrison Waldon for helpful comments. José Penalva would like to acknowledge the hospitality of the universities of Oxford and Georgetown which hosted him while this paper was written. José Penalva acknowledges financial support from Agencia Estatal de Investigación (PID2019-104649RB-I00/AEI/10.13039/501100011033) and the Comunidad de Madrid under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M12), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

Preprint submitted to TBA *This version: October 3, 2022. First version: May 10, 2022*

In this paper, we contribute to the algorithmic collusion literature in the setting of imperfect private information, i.e., algorithms do not observe prices (those of rivals or a common market price) and the rewards received are noisy. We primarily study algorithms that use information only from their own strategies, actions, and payoffs. The interaction of learning algorithms is characterised as a deterministic system of ordinary differential equations (ODEs) in a repeated game environment. We apply this characterisation in the context of a market making game in which a finite number of market makers use algorithms that compete to provide liquidity in the offer side of a limit order book (LOB). Specifically, we construct a stylised model to study whether and how supracompetitive outcomes arise, and the role of the tick size (i.e., the minimum price variation, and the space between prices on the price grid) in facilitating or obstructing competition among competing algorithms.

The characterisation of the dynamics between competing learning algorithms as a system of ODEs is powerful. With this approach, one can study and interpret the dynamics of the interactions, and use the rich literature in evolutionary game dynamics to understand the long-term behaviour of the algorithms in a repeated game. In a symmetric bimatrix repeated game, we prove that the dynamics from Cross learning, the exponential-weight algorithm for exploration and exploitation (EXP3), frequency adjusted Q-learning, and synchronous Q-learning converge to a pure strategy Nash equilibrium of the stage game. On the other hand, we prove that the dynamics of Q-learning can converge to outcomes that are not those of Nash equilibria of the stage game. In particular, Q-learning can converge to asymmetric outcomes where one market maker repeatedly plays an action that is sub-optimal in the sense that it is not a best response (of the stage game) and it is unresponsive to the other player's actions.

To understand how the tick size in a LOB affects competition among algorithms, we characterise the set of pure strategy Nash equilibria from the discrete action space stage game (parameterised by the tick size) and its relation to the continuous action space Bertrand–Nash equilibrium of the stage game. With a finite tick size, the set of pure strategy Nash equilibria of the stage game often has more than one element. In this set, the Nash equilibrium with the lowest offer is referred to as the most competitive equilibrium. However, when algorithms compete to provide liquidity, there is no guarantee that the algorithms will converge to the most competitive equilibrium. Indeed, tacit collusion can and often occurs as the algorithms tacitly coordinate on one of the more profitable Nash equilibrium outcomes. Nonetheless, as the tick size tends to zero, we show that the bounds which define the set of Nash equilibria of the stage game with a discrete action space converge to the Bertrand–Nash equilibrium offer. Therefore, by controlling the size of the tick in the LOB, regulators are able to cap the excess rents the algorithms can extract.

We find that a large tick size obstructs competition, and that a smaller tick size encourages competition and lowers trading costs for liquidity takers. With a smaller tick size, algorithms are more likely to produce supracompetitive offers, but the excess profits are bounded by the range of possible offers of the Nash equilibria of the stage game, which shrinks as the tick size decreases. Also, as the tick size decreases (and with it excess profits) the speed of convergence to a rest point decreases, which may generate costs that limit how much can be gained from making the tick size very small.

Finally, we find that competition between Q-learning algorithms does not lead to stage game Nash equilibrium outcomes. It leads to outcomes that are sub-optimal for both the algorithmic market makers and lead to greater costs for liquidity takers. Specifically, competition between Q-learning algorithms tends to generate asymmetric actions, where one market maker earns less than its counterpart and does not exploit potentially advantageous deviations. In addition, Q-learning algorithms often lead to supracompetitive offers above the most collusive Nash equilibrium outcome, which increases the trading costs for liquidity takers.

Our paper contributes to a range of literature in a number of ways. First, we contribute towards reinforcement learning in games by applying the stochastic approximation framework from Benaïm (1999) to show how one obtains the appropriate ODEs, which we use to formalise existing connections between several learning algorithms and their respective ODEs.[1] Furthermore, we introduce a new connection between synchronous Q-learning and the well-known replicator-mutation dynamics from evolutionary game theory. The approach we take is different from that in Börgers and Sarin (1997) who apply results from Norman (1972) to obtain uniform convergence in probability between the trajectories, whereas we establish uniform convergence almost surely. Our results on convergence to a Nash equilibrium outcome of the stage game is related to the work of Hopkins and Posch (2005), Duffy and Hopkins (2005), and Beggs (2005) who study the dynamics for the Erev and Roth's learning model (Erev and Roth, 1998).

Second, as far as we are aware (see Dorner, 2021, for a thorough review on algorithmic collusion), we are the first to apply an evolutionary game theory approach to study algorithmic collusion between independent learning algorithms. Our approach complements the existing literature.

Asker, Fershtman and Pakes (2021) prove by induction and by contradiction that with no exploration, synchronous Q-learning converges to a Nash equilibrium. They also prove that Q-learning has a non-zero probability of converging to any symmetric action pair. Our results follow from characterising the interaction between the algorithms as a system of ODEs and are analogous to theirs.

On the other hand, Calvano et al. (2020, 2021) find that Q-learning (using algorithms that condition on states, defined by previous stage game actions/outcomes) leads to supracompetitive prices in both a Bertrand oligopoly where opponent prices are observed and a Cournot oligopoly where a market price is observed. We demonstrate that Q-learning (with no states) cannot always learn an appropriate response to the opponent in the setting of imperfect private information, which prevents Q-learning from reaching a Nash equilibrium outcome and results in supracompetitive prices. Calvano et al. (2020, 2021) further demonstrate that Q-learning achieves tacit collusion by punishing deviations under both perfect and imperfect monitoring. In our paper, we use the vector field of policies and the basins of attraction to show that a gradual reward-punishment scheme is achieved under imperfect private information.

Additionally, Hansen, Misra and Pai (2021) prove that UCB-type algorithms will learn to play the collusive action for symmetric 2×2 games when there is no Nature player, while Cartea et al. (2022*b*) study competition between a range of multi-arm bandit algorithms in over-the-counter markets. Both studies find that supracompetitive prices can be achieved in the setting of imperfect private information. We further demonstrate that reaching a supracompetitive outcome also depends on the initial condition of the algorithms.[2]

Finally, our results extend the results in the microstructure literature on market making in open exchanges under conditions of imperfect information, (Kandel and Marx, 1997; Kadan, 2006; Loertscher, 2008; Vives, 2011; Baruch and Glosten, 2019). Our model is closest to the price competition model of Spulber (1995) in a market making environment similar to that in Kandel and Marx (1997). In Spulber (1995) price competition takes place in a setting with uncertainty about rivals' costs. In our setting, uncertainty does not arise from differences in marginal costs (which we assume to be equal to zero), but from the possibility that the order that offers the best price is not necessarily the one that is executed first. This can arise for a number of reasons. For example, if there are differences in realised latency between the market

[1] Tuyls, Verbeeck and Lenaerts (2003) and Sato and Crutchfield (2003) derive the dynamics for Q-learning with a softmax activation for mapping the Q-values to the policy; Gomes and Kowalczyk (2009) derive the dynamics for Q-learning with the ϵ-greedy policy; Kaisers and Tuyls (2010) show that the frequency adjusted Q-learning recovers the replicator-mutation dynamics; and Kasbekar and Proutiere (2010) derive the dynamics for the EXP3 algorithm.

[2] Other related work on algorithmic collusion include: Brown and MacKay (2021) who study the effect of asymmetry (in decision frequency) in pricing technology between online retailers, and Klein (2021) who study sequential pricing where the algorithms take turns setting a price. Both studies find that the algorithms end up quoting supracompetitive prices.

makers, where by realised latency we refer to the time between making the decision of what offer to make and that offer reaching the LOB, relative to the random arrival of the incoming executable order; see Cartea and Sánchez-Betancourt (2021) who provide evidence for stochastic latency. In this context, we capture this uncertainty in the static stage game through a latency parameter that allows us to randomly shuffle the winning order while retaining the analytical simplicity of a simultaneous action stage game. As in our model, in Kandel and Marx (1997), a finite number of market makers compete in price on a discrete price grid and the model gives rise to multiplicity of Nash equilibria. In our paper, the ODEs allow us to resolve the problem of indeterminacy of multiplicity by characterising the basins of attraction of the different equilibria. On the other hand, Kandel and Marx (1997) use focal points arguments to justify traders coordinating on even eighths as an equilibrium selection method.[3]

Our model incorporates the potential effects of imperfect competition to the debate on the tick size. There is an extensive literature on this issue (see Verousis, Perotti and Sermpinis (2017) for an overview). There is also a debate on the current one cent tick size in the U.S. equity market, because a significant number of shares are traded with spreads that are constrained by this tick size. While it is recognised that this constraint leads to significant liquidity frictions (most recently documented by the SEC Tick Pilot program, see Griffith and Roseman (2019), Chung, Lee and Rösch (2020), and Penalva and Tapia (2021)), a recent study by Li and Ye (2021) finds that a company's management can minimise their company's asset's quoted spread (as a percentage of its nominal price) by setting the nominal price in a way that the quoted spread is two ticks wide. Our results highlight the endogenous relationship between the equilibrium spread and the tick size, and the difficulties associated with implementing a policy that aims at setting the tick size, or the nominal value the asset, so the quoted spread is m ticks, especially if m is small.

The remainder of this paper proceeds as follows. Section 2 outlines the stochastic approximation framework to cast the interaction between the learning algorithms as a deterministic system of ODEs, and discusses the various algorithms and their respective dynamics. Section 3 applies results from evolutionary game theory to understand the asymptotic dynamics between the learning algorithms. Section 4 introduces the market making game and the relationship between the Nash equilibria of the stage game with a discrete action space and with a continuous action space. Section 5 explores the role of tick size in facilitating or obstructing competition. Section 6 discusses the policy implications from our findings and relates the results from Calvano et al. (2020) to the market making game as a robustness check. Finally, the Appendices collect some proofs and supplementary experiments.

2. Algorithm dynamics as a system of ordinary differential equations

It is not straightforward to study the repeated interaction between independent learning algorithms that provide liquidity in electronic markets. From a modelling perspective, most learning algorithms require the environment to be stationary. However, in electronic markets, the environment faced by each learning algorithm is non-stationary because the actions of one player affect the payoffs of the other players, and because the algorithms are constantly adapting their strategies. In this section, we outline the intuition behind the stochastic approximation framework of Benaïm (1999) and discuss how to cast the interaction between algorithms as a deterministic system of ordinary differential equations (ODEs).

We consider learning algorithms with two key features. One, the algorithms are relevant in financial markets. Two, the dynamics and outcomes of the algorithms can be formally studied within a game theoretic framework. Electronic markets are high-speed environments where computational speed is important

[3] Kandel and Marx (1997) precedes the general adoption of the term "focal point" in the literature. They use norms and conventions as the criteria for equilibrium selection as in: "Such coordination can be achieved either by overt collusion or through the use of norms and conventions".

because the time-scale between decisions is milliseconds at most. Thus, to be competitive, market participants need to use algorithms that are simple to implement and computationally efficient. Also, to remain competitive, algorithms must continuously adapt to changes in the market environment and, in particular, must be able to react to the actions taken by their adversaries (i.e., competitors). Finally, the dynamics of the algorithms we consider can be cast in the framework of evolutionary game theory. Thus, in Section 3, we use the Nash equilibria of the static stage game to study the long-run behavior of algorithms that repeatedly play the static game.

2.1. Intuition

Consider an n-player (stage) game in normal form $G = \{\mathcal{A}_i, \pi_i(\cdot); i \in \mathcal{I}\}$, where $\mathcal{I} = \{1, \dots, I\}$ is the set of players, $\mathcal{A}_i$ is a finite set of K actions, and π_i is player i's payoff function that maps $\prod_{j=1}^{I} \mathcal{A}_j$ into $\mathbb{R}$. The game includes an additional Nature player, indexed by ν, that takes random actions on $\mathbb{R}$ according to a probability distribution that is a function of the realised vector of actions $\tilde{a} = (a_1, \dots, a_I)$. The stage game is repeated over iterations $n \in \{1, 2, \dots, N\}$ where N can be finite or infinite.

Each player $i \in \mathcal{I}$ chooses a strategy for the repeated game. This strategy is represented by a recursive learning algorithm. At each period n, the algorithm describes a series of computations that generates an action and a series of computations to update variables that the algorithm tracks. The variables tracked by the algorithm are stochastic processes, which in the cases discussed here are the probabilities of playing a certain action (i.e., the policy) or the Q-values from a Q-learning algorithm. Once the algorithm performs an action, it receives feedback in the form of a reward (the player's stage-game payoff) which the algorithm uses to update variables and to instruct the next action.

Specifically, we denote $\{\boldsymbol{\pi}(n)\}_{n\in\mathbb{N}}$ as the sequence of random reward vectors $\boldsymbol{\pi}(n) \in E$ for the I algorithms with a bounded support for all n. The probability law of the rewards is denoted by μ_π, which depends on the actions, and $\{\boldsymbol{x}(n)\}_{n\in\mathbb{N}}$ denotes a sequence of discrete-time stochastic processes generated by the learning algorithms; i.e., a sequence of policies or the sequence of Q-values from a Q-learning algorithm. The increments of the stochastic processes are given by

$$\boldsymbol{x}(n+1) - \boldsymbol{x}(n) = \gamma_n \, f\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) \,, \tag{1}$$

where $f : \mathbb{R}^{I\times K} \times E \to \mathbb{R}^{I\times K}$ is the stochastic update rule of the learning algorithms and $\gamma_n > 0$ is a decreasing learning rate that satisfies

$$\sum_n \gamma_n = \infty \quad \text{and} \quad \sum_n \gamma_n^{1+q/2} < \infty \tag{2}$$

for some $q \geq 2$.

We use stochastic approximation techniques to show that the trajectories of the discrete-time stochastic processes (i.e., the realisations of the stochastic processes) from the learning algorithms can be approximated with the trajectories from a deterministic system of ODEs (i.e., solutions to a system of ODEs); see Benaïm (1999). Formally, one approximates the stochastic update rule f by a deterministic function F that is the average of the update rule. The difference between f and F is the discrepancy between the actual increment and the expected increment of $\boldsymbol{x}$. Thus, a crucial step in the approximation is to verify that the discrepancies become negligible as the value of the learning rate γ_n becomes sufficiently small.

Therefore, the goal is to rewrite the stochastic update rule in (1), so that the increments of the stochastic processes are given by

$$\boldsymbol{x}(n+1) - \boldsymbol{x}(n) = \gamma_n \, \left(F\left(\boldsymbol{x}(n)\right) + \boldsymbol{U}(n)\right) \,. \tag{3}$$

That is, we separate the stochastic update rule into two parts: its expected value, and an innovation component comprised of deviations from this expected value. Specifically, the deterministic vector field $F : \mathbb{R}^{I\times K} \to \mathbb{R}^{I\times K}$ is the expectation of the algorithm's stochastic update rule with respect to the reward distribution,

$$F\left(\boldsymbol{x}\right) = \int f\left(\boldsymbol{x}, \boldsymbol{\pi}\right)\, \mu_\pi(d\boldsymbol{\pi})\,,$$

and $\boldsymbol{U}(n)$ are the deviations that arise when approximating the stochastic update rule f with the deterministic vector field F, that is,

$$\boldsymbol{U}(n) = f\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) - F\left(\boldsymbol{x}(n)\right)\,.$$

If we treat the learning rate parameter γ_n as the time step, the expression in (3) can be seen as a Cauchy–Euler approximation scheme to solve numerically the deterministic system of ODEs

$$d\boldsymbol{x}/dt = \dot{\boldsymbol{x}} = F(\boldsymbol{x})\,,$$

provided that the deviations $\boldsymbol{U}(n)$ are negligible for a small enough value of γ_n. Throughout, the dot over a variable denotes its derivative with respect to time.

Our objective is to cast the dynamics of F as systems of ODEs from evolutionary game theory because evolutionary game dynamics can help to provide theoretical guarantees of optimality in terms of convergence to a Nash equilibrium. Therefore, when necessary, we approximate the stochastic update rule f with $\tilde{f} : \mathbb{R}^{I\times K} \times E \to \mathbb{R}^{I\times K}$, which is a linear function of the reward that simplifies F to a system of ODEs from evolutionary game dynamics. In our setting, the rewards the algorithms receive depend on the action of algorithm i, the actions of the opponents, and Nature's action. Thus, an additional benefit of $\tilde{f}$ being linear in the rewards is that it simplifies the analytical computation of F to obtain the system of ODEs.

The approximation F works for sufficiently small values of the learning rate γ_n because the small steps ensure that the increments of the stochastic processes from the algorithms are small. Consequently, as $\gamma_n \to 0$, the law of large numbers ensures that the *actual* increment is arbitrarily close to the *expected* increment F. In other words, when the value of the learning rate γ_n is sufficiently small, the trajectories of the continuous-time limit of the stochastic processes follow the trajectories of the system of ODEs. See Appendix A for the technical conditions and the formal derivation of the results that we present in the following subsection.

2.2. *Algorithm dynamics*

For the remainder of the paper, with an abuse of notation, $\boldsymbol{x}(n)$ only describes the discrete-time stochastic processes of policies generated from the algorithms.

Q-learning and variations

Q-learning in Algorithm 1 is a model-free reinforcement learning algorithm that learns the value of an action in a particular state. We only consider a singleton state where Q-values represent estimates of the expected discounted rewards for each action. At each iteration n, the algorithm picks an action based on the policy $\boldsymbol{x}_i(n) = (x_{i1}(n), \ldots, x_{iK}(n))$ which describes the probability that algorithm i chooses action k from the set of K possible actions. The algorithm updates the Q-values based on the learning rate γ_n, on the reward received $\pi_i(n)$, and the discount factor δ. We use the softmax activation function in (4) to

Algorithm 1: Q-learning algorithm for player i.

```
Learning rate: $\gamma_n \in (0,1]$. Exploration-exploitation: $\tau \in \mathbb{R}^+$. Discount factor: $\delta \in [0,1)$.
Initialise Q-values: $Q_{ik}(1)$ for $k = 1, \dots, K$.
for $n = 1, 2, \dots$ do
    Pick action $a_{ik}$ with probability
        $x_{ik}(n) = \frac{e^{\tau\, Q_{ik}(n)}}{\sum_\ell e^{\tau\, Q_{i\ell}(n)}}\,.$    (4)
    Let $a_i(n)$ be the selected action and let $\pi_i(n) := \pi_i(a_i(n), a_{-i}(n), a_\nu)$ be the reward received.
    Evolve the Q-values according to
        $Q_{ik}(n+1) = \begin{cases} Q_{ik}(n) + \gamma_n \left[\pi_i(n) + \delta \, \max \boldsymbol{Q}_i - Q_{ik}(n)\right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ Q_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n)\,, \end{cases}$    (5)
    where $\max \boldsymbol{Q}_i = \max_j \{Q_{ij}(n)\}$.
end
```

map the Q-values to the policy, which includes the parameter τ that controls the level of exploration and exploitation.[4]

In addition to Q-learning, we consider two variants of Q-learning: frequency adjusted Q-learning and synchronous Q-learning. In Appendix A, we show that the expected innovation of the policies $F(\boldsymbol{x})$ for both variants are described by the replicator-mutation dynamics, also known as the selection-mutation equation, from evolutionary game dynamics. We proceed by first looking at the replicator-mutation dynamics and how it arises out of the two variants and then return to the more complex dynamics of Q-learning.

The replicator-mutation dynamics are a system of ODEs that consists of two terms: a selection equation and a mutation equation. The selection equation focuses on selecting better policies over time, which is analogous to exploitation, and the mutation equation introduces variety to the policies, which is analogous to exploration. The specific replicator-mutation dynamics that arise from the variants of Q-learning are given by

$$\dot{x}_{ik} = \tau \, \underbrace{x_{ik} \left[\overline{\Pi}_{ik} - \sum_\ell x_{i\ell} \, \overline{\Pi}_{i\ell} \right]}_{\text{selection equation}} + \underbrace{x_{ik} \sum_\ell x_{i\ell} \, \ln\left(\frac{x_{i\ell}}{x_{ik}}\right)}_{\text{mutation equation}}\,, \tag{ODE 1}$$

where

$$\overline{\Pi}_{ij} = \mathbb{E}_{\boldsymbol{x}_{-i}}\left[\mathbb{E}_\nu\left[\pi_i(a_i, a_{-i}, a_\nu) \,|\, a_i = a_{ij}, a_{-i} = a_{-i}\right]\right] \tag{6}$$

is the expected payoff of player i when playing action a_{ij}. Here, $\mathbb{E}_{\boldsymbol{x}_{-i}}[\,\cdot\,]$ is the expectation operator over the actions of the opponents and $\mathbb{E}_\nu[\,\cdot\,]$ is the expectation operator over Nature's action. In the dynamics of

[4] In the reinforcement learning literature, the parameter τ is commonly referred to as the temperature parameter.

the policy described by (ODE 1), a large value of τ places more weight on exploitation and a small value of τ places more weight on exploration. Specifically, if we consider only the mutation equation in (ODE 1), i.e., $\tau = 0$, then the dynamics reach a rest point $\dot{x}_{ik} = 0$ for all k when $x_{ik} = 1/K$ for all k. Thus, the mutation equation drives the dynamics to pure exploration where every action has an equal probability of being played, i.e., towards $x_{ik} = 1/K$ for all k.

Next, we introduce the frequency adjusted Q-learning (FAQ-learning), see Kaisers and Tuyls (2010). The difference between FAQ-learning and Q-learning arises from changing the update rule so that the Q-values from FAQ-learning evolve according to

$$Q_{ik}(n+1) = \begin{cases} Q_{ik}(n) + \min\left\{\frac{\beta}{x_{ik}(n)}, 1\right\} \alpha_n \left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ Q_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n)\,, \end{cases}$$

where $\alpha_n, \beta \in (0, 1]$. The minimum operator ensures that the Q-values do not escape the convex hull of experienced rewards, i.e., escape the space of discounted rewards. With this new rule, the learning rate is $\gamma_n = \alpha_n \beta$ when $x_{ik} \geq \beta$ and is $\gamma_n = \alpha_n$ when $x_{ik} < \beta$. We assume that the value of β is such that $x_{ik} \geq \beta$ to focus on the subspace of policies $[\beta, 1-\beta]^{I\times K}$ where the learning rate $\gamma_n = \alpha_n \beta$ is scaled by the factor $1/x_{ik}$.

The scaling factor $1/x_{ik}$ is an important mechanism in FAQ-learning, it ensures that the trading algorithm remains competitive in electronic markets by adapting to changes in the reward as a result of the actions of adversaries. The algorithm can react to the actions of other players because when the algorithm learns that a specific action is sub-optimal, it will have a low probability of picking the action. However, if it picks that action and receives a large reward, the scaling of $1/x_{ik}$ will lead to a sufficiently large change in the Q-values so that the algorithm chooses that action more frequently. In Appendix A, we show that in the subspace of the policies $[\beta, 1-\beta]^{I\times K}$, the trajectories of policies from FAQ-learning follow the trajectories of policies from the replicator-mutation dynamics in (ODE 1) as $\gamma_n \to 0$.

Next, we look at the second variant of Q-learning, synchronous Q-learning, which updates the Q-values with the same rule as Q-learning in (5). They differ in that synchronous Q-learning conducts counterfactual learning to update the Q-values for every action at every iteration, instead of updating only the Q-values for the action performed. That is, synchronous Q-learning assumes that the learning algorithm constructs a counterfactual reward for any action it would have chosen, given the actions of the opponents. Therefore, synchronous Q-learning has the ability to adapt to changes in the reward as a result of the actions of the adversaries and remain competitive in electronic markets. By constructing counterfactual rewards that depend on the actions of the opponents, synchronous Q-learning requires more information and incorporates it into its Q-values so that it can make more informed decisions relative to the opponents current strategy.

Constructing counterfactual rewards is not always straightforward because it requires knowledge of the system. For example, in a Bertrand model of price competition, one can construct counterfactual rewards if the shape of the demand curve is known. Similarly, in a market making setting, market makers can construct counterfactual rewards given the actions of the other market makers because they can reverse engineer the price-time priority in the LOB.[5] In Appendix A, if counterfactual rewards are available, we show that as in FAQ-learning, the trajectories of policies from synchronous Q-learning follow the trajectories of policies from the replicator-mutation dynamics in (ODE 1) as $\gamma_n \to 0$.

The key consideration for including these variants (in addition to their suitability for electronic markets) is their relationship to the replicator-mutation dynamics. From this relationship, we obtain theoretical guarantees of their optimality in terms of convergence to Nash equilibrium in the next section.

[5] In reality, we do not need to know the actions of every market maker. The aggregate behaviour is enough to know the price level and the queue position of our order in the LOB.

Finally, we return to the dynamics of Q-learning in Algorithm 1. This algorithm is important because of its popularity in practical applications. We remark three points. First, Q-learning does not have the mechanisms like those of FAQ-learning and synchronous Q-learning to ensure the algorithm remains competitive when the opponents adapt their strategies. Second, the dynamics of the policies from Q-learning do not recover the replicator-mutation dynamics in (ODE 1). Third, when we take the appropriate expectations and write the dynamics of Q-learning in terms of the policies, the dynamics of the policies still depend on Q-values. However, if we focus on the dynamics of the Q-values (and not the policies), then we can fully characterise the dynamics of Q-values without the policies. Once we have the trajectories of the Q-values, we obtain the trajectories of the policies through the softmax activation.

Hence, in Appendix A, we show that for sufficiently small values of the learning rate γ_n, the dynamics of Q-learning in terms of Q-values are given by

$$\dot{Q}_{ik} = x_{ik}\left[\overline{\Pi}_{ik} + \delta\, \max \boldsymbol{Q}_i - Q_{ik}\right], \tag{ODE 2}$$

where x_{ik} is fully characterised by the Q-values and it is the probability that player i plays the action a_{ik} given in (4). As before, $\overline{\Pi}_{ik}$ denotes the expected payoff of player i playing action a_{ik} in (6).

Other learning algorithms

Although Q-learning is popular in finance (see Spooner et al., 2018; Ning, Lin and Jaimungal, 2021, for an application of Q-learning in market making and an application of double deep Q-learning in optimal execution, respectively), it is not the only type of learning algorithm. There are different classes of learning algorithms such as learning automata and multi-arm bandit algorithms. Learning automata are characterised as policy iterators where the updates from the learning rule are performed directly on the policy. Multi-arm bandit algorithms, on the other hand, are designed to address the trade-off between exploration and exploitation by minimising regret.

We consider one algorithm from each of these two classes. Specifically, we consider Cross learning from Cross (1973) as an example of a learning automaton, and the exponential-weight algorithm for exploration and exploitation (EXP3 algorithm) from Auer et al. (2002) as an example of a multi-arm bandit algorithm. In Appendix A, we show that for sufficiently small values of the learning rate γ_n, the trajectories of the policies from both algorithms follow the trajectories from the replicator dynamics given by

$$\dot{x}_{ik} = x_{ik}\left[\overline{\Pi}_{ik} - \sum_{\ell} x_{i\ell}\,\overline{\Pi}_{i\ell}\right], \tag{ODE 3}$$

where $\overline{\Pi}_{ij}$ denotes the expected payoff of player i playing action a_{ij} in (6).[6] The replicator dynamics describe the expected innovation of the policies $F(\boldsymbol{x})$ from Cross learning and the EXP3 algorithm.

Cross learning, described in Algorithm 2, evolves the policy according to (7) where γ_n is the learning rate and $\pi_i(n)$ denotes the reward received by the player at iteration n. The rewards lie within the interval $[0, 1]$ to ensure that the policy is a probability distribution over the set of K actions.

Cross learning can adapt to changes in the reward (provided there is a non-zero probability of playing every action) compared with Q-learning because the inputs from the reward directly change the policy. Therefore, changes to the reward are immediately reflected in the policies unlike Q-learning where changes to the policies require a sufficient change in the Q-values.

[6] The policies from the EXP3 algorithm converge to (ODE 3) up to a constant re-scaling of time.

```
Algorithm 2: Cross learning algorithm for player $i$.
    Learning rate: $\gamma_n \in (0,1]$.
    Initialise the policy: $x_{ik}(1) \in (0,1)$ such that $\sum_k x_{ik}(1) = 1$ for $k = 1, \dots, K$.
    for $n = 1, 2, \dots$ do
        Pick action $a_{ik}$ with probability $x_{ik}(n)$.
        Let $a_i(n)$ be the selected action and let $\pi_i(n) := \pi_i(a_i(n), a_{-i}(n), a_\nu)$ be the reward received.
        Evolve the policy according to
            $x_{ik}(n+1) = \begin{cases} \gamma_n \, \pi_i(n) + \big(1 - \gamma_n \, \pi_i(n)\big) \, x_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ \big(1 - \gamma_n \, \pi_i(n)\big) \, x_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n)\,. \end{cases} \quad (7)$
    end
```

```
Algorithm 3: EXP3 algorithm for player $i$.
    Learning rate: $\gamma_n \in (0,1]$.
    Initialise: $w_{ik}(1)$ for $k = 1, \dots, K$.
    for $n = 1, 2, \dots$ do
        Pick action $a_{ik}$ with probability $x_{ik}(n) = (1 - \gamma_n) \frac{w_{ik}(n)}{\sum_\ell w_{i\ell}(n)} + \frac{\gamma_n}{K}$.
        Let $a_i(n)$ be the selected action and let $\pi_i(n) := \pi_i(a_i(n), a_{-i}(n), a_\nu)$ be the reward received.
        Evolve the weights according to
            $w_{ik}(n+1) = \begin{cases} w_{ik}(n) \, \exp\left(\dfrac{\gamma_n \, \pi_i(n)}{x_{ik}(n) \, K}\right) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ w_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n)\,. \end{cases} \quad (8)$
    end
```

The exponential-weight algorithm for exploration and exploitation (EXP3), described in Algorithm 3, is a multi-arm bandit algorithm from Auer et al. (2002). At iteration n, the policy is a combination of importance-weight estimators of the rewards and a term that contributes towards exploration. The update rule in (8) is applied to the exponentially weighted reward estimate $\boldsymbol{w}$ which depends on the reward received $\pi_i(n)$ and on the parameter γ_n that acts as both the learning rate and the exploration parameter.

The EXP3 algorithm is suited for the adversarial bandit setting because the EXP3 algorithm makes no statistical assumptions about the reward distributions, as opposed to the traditional bandit algorithms such as the family of Upper Confidence Bound algorithms which assumes that the reward received from each action is independent and identically distributed. Similar to FAQ-learning, the EXP3 algorithm also has the scaling factor $1/x_{ik}$ so that it remains competitive in electronic markets by adapting to changes in the reward as a result of the actions of adversaries.

2.3. Discussion

First, in Appendix A, we show uniform convergence almost surely between the trajectories from the algorithms and the trajectories from the system of ODEs when γ_n satisfies (2). The convergence achieved in the asymptotics when γ_n is sufficiently small. An alternative choice is a constant learning rate that is sufficiently small, i.e., $\gamma_k = \gamma$ for all k. In this case, the strongest type of convergence the learning algorithms can achieve is uniform convergence in probability (see Benveniste, Métivier and Priouret, 1990, Theorem 1, Chapter 2.2 of Part I) for an arbitrary but finite time horizon $T > 0$. When γ_n satisfies (2), there is a transient phase of initial randomness in the trajectories of the algorithms before γ_n becomes sufficiently small so that the trajectories of the algorithms becomes "locked into" the trajectories from the ODEs. Hence, for the remainder of the paper, we assume that γ_n is a decreasing sequence with γ_1 sufficiently small so that we skip the transient phase. We are interested in the size of the basins of attraction to discuss the proportion of initial conditions that converge to certain rest points. Thus, the assumption allows us to interpret the likelihood of converging to a particular rest point without addressing the complexities associated with the lock-in probability (see Chapter 4 of Borkar, 2008).

Second, we only state results for the symmetric case when algorithms of the same type, with the same number of actions, play against each other. Our results can easily be extended to combinations of algorithms with different number of actions. Suppose that I players use one of the aforementioned algorithms, where player i has K_i actions. Then, we need to compute the deterministic vector field $F : \mathbb{R}^{K_1 \times \ldots \times K_I} \rightarrow \mathbb{R}^{K_1 \times \ldots \times K_I}$ where the component function F_{ik} is the appropriate deterministic vector field for the stochastic process of player i's algorithm. Verifying that the deviations are negligible when $\gamma_n \rightarrow 0$ remains the same as in the symmetric cases, which we do in Appendix A. Furthermore, Appendix B includes a demonstration that the stochastic approximation works with two players who use two different algorithms and the algorithms have a different number of actions to select from.

Finally, we only show results for the case when the state is a singleton. Extending the stochastic approximation to multiple states is possible, provided that for a fixed policy, the state process forms an ergodic Markov chain with a unique stationary distribution. The stationary distribution allows us to compute an appropriate deterministic vector field F. The extension is beyond the scope of the current paper, which we address in a follow-up paper (see Cartea et al., 2022*a*).

3. Asymptotic dynamics of algorithms

The expected innovation of policies $F(\boldsymbol{x})$ for Cross learning, the EXP3 algorithm, frequency adjusted Q-learning, and synchronous Q-learning are described by systems of ODEs from evolutionary game theory. Thus, we use existing results from evolutionary game theory to show that the asymptotic behavior of algorithmic competition in our market making game leads to a Nash equilibrium outcome of the stage game.

More specifically, with $I = 2$ players and a finite number of actions K, the market making game we study belongs to the subset of symmetric two-player bimatrix games where both players have the same payoff matrix $\boldsymbol{\Pi}_1 = \boldsymbol{\Pi}_2^\top$ and the players are interchangeable with policies $\boldsymbol{x} = \boldsymbol{x}_1 = \boldsymbol{x}_2$, which are the probabilities of playing each action. To study the symmetric setting, we fix the payoff matrix as the payoff from player one $\boldsymbol{\Pi} = \boldsymbol{\Pi}_1$, where the entries $\Pi_{k\ell}$ correspond to the conditional expected payoff of player one playing action a_k and player two playing action a_ℓ, as in Section 2. Bimatrix games where the players have the same payoff matrix are also known as partnership games or potential games; see Section 6 in Hofbauer (2011).

Before diving into results for this class of games, we recall that in evolutionary game theory a Nash equilibrium is related to the fitness of the policy $\boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x}$. Specifically, a policy $\hat{\boldsymbol{x}} \in S_K$, where $S_K = \{\boldsymbol{x} \in \mathbb{R}^K : x_k \geq 0, \sum_{k=1}^K x_k = 1\}$ is the $(K-1)$-dimensional simplex, is a Nash equilibrium if

$$\hat{\boldsymbol{x}}^\top \boldsymbol{\Pi}\, \hat{\boldsymbol{x}} \geq \boldsymbol{x}^\top \boldsymbol{\Pi}\, \hat{\boldsymbol{x}}$$

for all $\boldsymbol{x} \in S_K$. The Nash equilibrium is strict if equality holds only for $\hat{\boldsymbol{x}} = \boldsymbol{x}$. A pure strategy Nash equilibrium is a Nash equilibrium at the corner of the simplex, while a mixed Nash equilibrium is a Nash equilibrium in the interior of the simplex. We show that the dynamics of the algorithms (which represent the players' strategies) in a repeated game will only reach a stable outcome that is a pure strategy Nash equilibrium of the stage game. The dynamics of the algorithms are unstable around the mixed Nash equilibria and non-Nash pure strategies of the stage game, and therefore not a stable outcome.

3.1. Replicator dynamics

In Appendix A, we prove that the policies from Cross learning and the EXP3 algorithm converge to the replicator dynamics, also known as the selection equation. The dynamics of (ODE 3) in a symmetric game are given by

$$\dot{x}_k = x_k \left[(\boldsymbol{\Pi}\, \boldsymbol{x})_k - \boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x} \right] , \tag{9}$$

where $(\boldsymbol{\Pi}\, \boldsymbol{x})_k = \sum_{\ell=1}^{K} \Pi_{k\ell}\, x_\ell$.

To understand the long-term behaviour of the replicator dynamics in (9), we introduce some terminology. The rest points of the replicator dynamics are the policies $\boldsymbol{x} \in S_K$ for which the right-hand side of (9) is equal to zero for all k. A rest point in the interior of S_K is a solution to the system of linear equations $(\boldsymbol{\Pi}\, \boldsymbol{x})_1 = \cdots = (\boldsymbol{\Pi}\, \boldsymbol{x})_K$. In general, there exists at most one solution for (9) in the interior of S_K. Additionally, every corner of the simplex is also a rest point. A rest point $\hat{\boldsymbol{x}}$ is stable (also known as Lyapunov stable) if for every neighbourhood U of $\hat{\boldsymbol{x}}$, there exists a neighbourhood V of $\hat{\boldsymbol{x}}$ such that $\boldsymbol{x} \in V$ means that $\boldsymbol{x}(t) \in U$ for all $t \geq 0$. A rest point $\hat{\boldsymbol{x}}$ is asymptotically stable if it is stable and has a neighbourhood U such that $\boldsymbol{x}(t) \to \hat{\boldsymbol{x}}$ for $t \to \infty$. A rest point $\hat{\boldsymbol{x}}$ is globally stable if it is stable and $\boldsymbol{x}(t) \to \hat{\boldsymbol{x}}$ for $t \to \infty$ whenever $x_k > 0$ with $\hat{x}_k > 0$ for all k.

Proposition 1 *In a partnership game (where the players have the same payoff matrix), if γ_n satisfies (2), then every interior trajectory of policies from Cross learning and the EXP3 algorithm converges to a pure strategy Nash equilibrium of the stage game, with probability one.*

Proof First, for partnership games, replicator dynamics are (myopic) adjustment dynamics that satisfy $\dot{\boldsymbol{x}}^\top \boldsymbol{\Pi}\, \boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \in S_K$, with equality only at equilibria, which means that the policy always moves towards a better reply than the current policy until reaching an equilibrium. Moreover, from the Folk theorem of evolutionary game theory, we know that if the rest point $\hat{\boldsymbol{x}}$ is stable, then it is a Nash equilibrium. Then, by Theorem 6.1 in Hofbauer (2011), for every partnership game, the strict Lyapunov function $\boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x}$ increases along trajectories, and a strict local maximiser of $\boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x}$ is asymptotically stable for replicator dynamics. Therefore, every interior trajectory of the replicator dynamics converges to a Nash equilibrium. Second, mixed equilibria and non-Nash pure strategies of partnership games are unstable under the replicator dynamics, which means the replicator dynamics will only converge to a pure strategy Nash equilibrium; see Lemma 1 in Duffy and Hopkins (2005).

Finally, because partnership games have a strict Lyapunov function $\boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x}$ for the replicator dynamics, we conclude from Corollary 6.6 in Benaïm (1999) and Theorem 1 in Pemantle (1990) that the continuous-time affine interpolated processes of policies from Cross learning and the EXP3 algorithm converge to a pure strategy Nash equilibrium of the stage game, with probability one. □

Proposition 1 is analogous to Proposition 4 in Hopkins and Posch (2005) and Proposition 1 in Duffy and Hopkins (2005) who prove a similar result for Erev and Roth's learning model (Erev and Roth, 1998).

3.2. Replicator-mutation dynamics

In Appendix A, we prove that the policies from frequency adjusted and synchronous Q-learning converge to the replicator-mutation dynamics, also known as the selection-mutation equation. The dynamics of (ODE 1) in a symmetric game are given by

$$\dot{x}_k = \tau\, x_k \left[(\mathbf{\Pi}\, \boldsymbol{x})_k - \boldsymbol{x}^\top \mathbf{\Pi}\, \boldsymbol{x} \right] + x_k \sum_{\ell} x_\ell \,\ln\left(x_\ell / x_k \right) \tag{10}$$

with $(\mathbf{\Pi}\, \boldsymbol{x})_k = \sum_{\ell=1}^{K} \Pi_{k\ell}\, x_\ell$. To understand the long-term behaviour of the replicator-mutation dynamics in (10), we divide it by τ to re-scale the time step and write it as

$$\dot{x}_k = x_k \left[(\mathbf{\Pi}\, \boldsymbol{x})_k - \boldsymbol{x}^\top \mathbf{\Pi}\, \boldsymbol{x} \right] + \varepsilon_k - x_k \sum_{\ell} \varepsilon_\ell \,, \tag{11}$$

where the term

$$\varepsilon_j = -\frac{1}{\tau}\, x_j \ln\left(x_j \right) > 0$$

describes the mutation rates and depends on the exploration-exploitation parameter from the learning algorithms. Thus, the replicator-mutation dynamics from frequency adjusted and synchronous Q-learning satisfy the special mutation rates of Section 20.3 in Hofbauer and Sigmund (1998), a condition we require in the proof of the following proposition.

Proposition 2 *In a partnership game (where the players have the same payoff matrix), if γ_n satisfies (2) and the value of the exploration-exploitation parameter τ is sufficiently large, then every trajectory of policies from frequency adjusted Q-learning and synchronous Q-learning converges to a pure strategy Nash equilibrium of the stage game, with probability one.*

Proof The replicator-mutation dynamics in (11) satisfy the special mutation rates; thus, we have that for every partnership game, the potential function

$$P(\boldsymbol{x}) = \frac{1}{2}\, \boldsymbol{x}^\top \mathbf{\Pi}\, \boldsymbol{x} + \sum_{\ell} \varepsilon_\ell \,\ln\left(x_\ell \right) \,, \tag{12}$$

is a Lyapunov function for (11), see Theorem 6.2 in Hofbauer (2011). Therefore, every solution of (11) converges to a connected set of rest points. These rest points are not Nash equilibria because of the second term on the right-hand side of (12), but approximate Nash equilibria. This is because the local maximiser of (12) is not necessarily the same local maximiser of $\boldsymbol{x}^\top \mathbf{\Pi}\, \boldsymbol{x}$. However, for sufficient exploitation (a large value of τ), the mutation rates are sufficiently small and we recover the same local maximiser of $\boldsymbol{x}^\top \mathbf{\Pi}\, \boldsymbol{x}$.

Furthermore, with sufficient exploitation (a sufficiently large value of τ), mixed equilibria and non-Nash pure strategies of partnership games are unstable under the replicator-mutation dynamics, so the replicator-mutation dynamics will only converge to a pure strategy Nash equilibria. This result follows analogously from the proof of Proposition 2 in Duffy and Hopkins (2005). For sufficient exploitation, the results of Lemma 1 in Duffy and Hopkins (2005) hold.

Finally, the potential function in (12) is a strict Lyapunov function in the sense of Corollary 6.6 in Benaïm (1999). Therefore, along with Theorem 1 in Pemantle (1990), we conclude that the continuous-time affine interpolated processes of policies from frequency adjusted and synchronous Q-learning converge to a pure strategy Nash equilibrium of the stage game, with probability one. □

Proposition 2 is analogous to Proposition 2 in Duffy and Hopkins (2005) who prove a similar result for Erev and Roth's learning model (Erev and Roth, 1998) with an exponential choice rule.

3.3. *Q-learning dynamics*

Hart and Mas-Colell (2003) show that in general, uncoupled dynamics do not lead to a Nash equilibrium. In their paper, uncoupled means that the adjustment of a player's strategy does not depend on the payoff function of the other players. It can, however, depend on the strategies of other players, as well as on the player's own payoff function. All the dynamics in our paper are uncoupled according to the definition of Hart and Mas-Colell (2003), however, the replicator dynamics and the replicator-mutation dynamics in a partnership game are exceptions because their properties allow their dynamics to converge to a Nash equilibrium (see Section IV point (f) in Hart and Mas-Colell, 2003). On the other hand, we show that the dynamics from Q-learning can converge to an equilibrium that is not a Nash equilibrium. Specifically, we show that the strategies of two Q-learning algorithms, each with two actions, converge to asymmetric actions that they play for an arbitrary time horizon $T > 0$. For some initialisation of Q-values with no exploration ($\tau \to \infty$), the strategies become "stuck"; the algorithms are unable to learn an appropriate response to the actions of the opponents so they stick to the same action. Of course, the algorithms can get stuck in any of the four possible joint-action combination, but the joint-actions that are asymmetric are sufficient to demonstrate convergence to a non-Nash equilibrium in a symmetric game; we show this in the next proposition.

Proposition 3 *Assume that players use Q-learning in a two-player symmetric game with positive entries in the payoff matrix $\mathbf{\Pi}$. If γ_n satisfies (2) with γ_1 sufficiently small (to skip the transient phase) and there is no exploration ($\tau \to \infty$), then there are starting conditions of Q-values for which the strategies of the players converge to asymmetric actions, which are not a Nash equilibrium of the stage game (for an arbitrary time horizon $T > 0$).*

Proof Without loss of generality, assume that each player has only two actions and fix the discount parameter $\delta = 0$. First, we show that with no exploration ($\tau \to \infty$), player i using Q-learning is unable to learn the best response to an opponent playing a fixed action at every iteration. Let the index b denote the best response and let the index k denote the sub-optimal action. Then, the actions of player i are a_b and a_k with rewards Π_b and Π_k, respectively, and, of course, $\Pi_k < \Pi_b$.

With γ_1 sufficiently small to skip the transient phase, the dynamics for Q-values follow

$$\dot{Q}_\ell = x_\ell \left[\Pi_\ell - Q_\ell\right]$$

for $0 \le t \le T$, so the dynamics for the Q-values of player i are given by

$$\dot{Q}_k = x_k \left[\Pi_k - Q_k\right] \quad \text{and} \quad \dot{Q}_b = x_b \left[\Pi_b - Q_b\right]. \tag{13}$$

If the Q-values for player i's algorithm are initialised such that $Q_b(1) < Q_k(1) < \Pi_k$, then for $\tau \to \infty$ we have that $x_k = 1$ and $x_b = 0$, so the solution to (13) is given by

$$Q_k(t) = C\, e^{-t} + \Pi_k \quad \text{and} \quad Q_b(t) = Q_b(1)$$

for all $0 \le t \le T$, where C is the constant of integration. Therefore, in this setting, when the opponent plays a fixed action, player i's Q-learning algorithm is unable to learn the best response.

In a symmetric game with no exploration, if players initialise their Q-learning algorithms to play asymmetric actions, then the algorithms will converge to asymmetric actions for an arbitrary time horizon $T > 0$. Hence, the long-term outcome is a non-Nash equilibrium pair of strategies as at least one of the two players is not playing a best response to the other's action. □

Proposition 3 is similar to Proposition 1 in Asker, Fershtman and Pakes (2021) who show that Q-learning with no exploration has a positive probability to converge to any symmetric action pair. Proposition 3 extends their results to show that asymmetric outcomes are also possible, and both proofs rely on random initial conditions with no exploration.

Furthermore, unlike Propositions 1 and 2, Proposition 3 does not guarantee convergence in the asymptotics. To our knowledge, there is no clear candidate for a Lyapunov function for the dynamics of Q-learning. Therefore, asymptotic convergence is not guaranteed. The best we could hope to achieve in this case is uniform convergence between the trajectories of the Q-values and the trajectories of the associated ODEs for a finite horizon $T > 0$, which can be arbitrarily long.

One issue that arises is whether the resulting sub-optimal behavior could be part of a Nash equilibrium of the repeated game, where asymmetry can arise, as we know from the Folk theorem. An implication of the proof of Proposition 3 is that the algorithm becomes insensitive to the other algorithm's action. In particular, if algorithm one is not playing a best response to algorithm two's action, and algorithm two is stuck on action a_s, then algorithm one could deviate to her best response without fear of retribution, provided that algorithm two's new stage game payoff is not too low.[7] Hence, the observed algorithmic behavior is driven by factors unrelated to the Folk theorem, and may not be a Nash equilibrium of the stage or the repeated game.

4. Market making with algorithms

We apply the results from Sections 2 and 3 to a stylised model of market making where a finite number of players use algorithms to compete to provide liquidity. Specifically, we study the role of tick size (the minimum price variation in the LOB) in facilitating or hindering competition among competing algorithms. We present the model in Section 4.1, after which we characterise and compare the theoretical properties of the equilibria of the discrete action space stage game with the continuous action space Bertrand–Nash equilibrium in Section 4.2. Finally, in Sections 4.3 and 4.4 we study a specific tick size configuration with two algorithmic market makers that reduces the stage game to a 2 × 2 symmetric game and find that a large tick size can restrict competition.

4.1. The stage game

Our analysis uses a repeated game of Bertrand competition among I strategic market making algorithms with zero marginal expected cost from trading.[8] The strategic algorithms compete by setting the price of liquidity in a market with a large number of smaller, less efficient, competitive market makers. The less efficient market makers are competitive in the sense that they have zero expected profit and offer liquidity at a price equal to their expected cost, which is normalised to be equal to two and provides an upper bound on the price of liquidity.[9]

[7] To be precise, algorithm two will not respond to algorithm one's deviation as long as the post deviation payoff keeps the Q-value of algorithm two's current action above those of the Q-values of algorithm two's other actions. Note that despite the fact that algorithm two may obtain higher payoffs from another action, this does not imply that the Q-value for that other action will (eventually) be higher than the current one, as we saw in the proof of Proposition 3.

[8] This is not a necessary assumption. The key implicit assumption is that the I market makers have the same expected cost which we normalise to be equal to zero.

[9] The difference in expected costs between the different type of market makers can arise for a number of reasons, e.g., lower administrative costs, economies of scale, better technology, better information or information processing ability, etc. These modelling assumptions are discussed more extensively in Section 6.2.

More precisely, $\mathcal{I} = \{1, \ldots, I\}$ players repeatedly play a stage game defined by $G = \{\mathcal{A}, \pi_i(\cdot); i \in \mathcal{I}\}$. Each player i chooses an algorithm to implement her strategy over repetitions of G over N periods indexed by n. In each round, the algorithm selects actions on a grid, $\mathcal{A} = \{0, \vartheta, \ldots, K\vartheta = 2\}$, and the players receive the stage game payoff, π_i, that depends on the actions taken by the algorithms in that stage game. The parameter $\vartheta \in (0, 1]$ represents the fineness of the price grid, i.e., the tick size of the LOB. For simplicity, and in keeping with the literature (e.g., Anshuman and Kalay, 1998; Cordella and Foucault, 1999), we focus on the ask side of the LOB, so the algorithms compete to provide liquidity by posting offer prices (algorithm i posts offer a_i) to potential liquidity takers who want to purchase one unit of the asset.[10] The resulting best price represents the quoted half-spread, independent of whether the order was filled.

At every iteration, only one liquidity provider wins the trade and the winner is determined by the fill probabilities. For player i, the probability distribution of filling an incoming trade when posting an order a_i is denoted by $P(a_i, a_{-i})$. This fill probability depends on the level of the offer a_i in the LOB relative to where the other players posted their offers, a_{-i}. The ex-ante probability of filling the posted offer with an incoming buy trade is a decreasing function of the distance to the midprice and all posted offers in the LOB have a non-zero probability of being executed.[11] Specifically, the probability that an offer a' that is worse (further away from the midprice) than offer a_i (i.e., $a' > a_i$) is filled, is decreasing in the difference of the offer prices, $a' - a_i$, and in the latency parameter μ.[12] The parameter μ represents the importance of gaining execution priority in the LOB which we associate with latency. For simplicity, we assume that orders posted at the same price are equally likely to be executed.

The expected payoff from offer $a = a_i$ conditional on other offers, a_{-i}, is symmetric across players and given by

$$\pi(a_i, a_{-i}) = a_i \, P(a_i, a_{-i}) = a_i \, \frac{\exp(-\mu \, a_i)}{\sum_j \exp(-\mu \, a_j)} \, . \tag{14}$$

With the fill probability $P(a_i, a_{-i})$ in (14), the algorithm offering the best offer price (i.e., lowest offer in the LOB) is more likely to fill an incoming buy trade, albeit at a lower profit. This fill probability is parameterised by μ, and as $\mu \to \infty$, there is no uncertainty in execution so the probability that the lowest priced offer captures the whole market (i.e., fills the incoming buy orders) converges to one, as in the classic model of Bertrand competition.

4.2. *Symmetric Nash equilibria of the stage game*

The set of pure strategy Nash equilibria of the stage game provides a natural benchmark to study the long-term properties of competing algorithms (which represent the strategies of the players in the repeated game), specifically because in Section 3 we showed that in a symmetric bimatrix game, the dynamics from the algorithms we study (except Q-learning) converge to a pure strategy Nash equilibrium outcome of the stage game. Given the symmetry of the stage game, we focus on the symmetric pure strategy Nash equilibria (the equilibria for short) of the game.

In the following propositions, we show that for I market makers and a given value of the latency parameter μ, the set of equilibria depends on the price grid which is parameterised by the tick size $\vartheta > 0$. We

[10] Alternatively, one can interpret the model for the bid side of the LOB, so that $-a_i \leq 0$ is the bid price, and a lower $a_i \geq 0$ represents a price improvement in the bid.

[11] There are a number of circumstances that would allow orders posted at worse prices to be executed. For example, random latency between order submission and arrival to the matching engine, as documented in Aquilina, Budish and O'Neill (2021). This is discussed in greater detail in Section 6.2.

[12] We also assume that the strategic algorithms will gain price priority over the smaller competitive market makers if they offer liquidity at $a_i = 2$.

also show that the set of equilibria is strongly tied to the outcome of the unique Bertrand–Nash equilibrium of the continuous action space game; that is, the game with $\vartheta \to 0$ so the market makers can quote an offer anywhere in the interval $[0, 2]$.

Proposition 4 *The stage game with continuous action space $\mathcal{A} \in [0, 2]$ has a unique, globally stable, pure strategy Nash equilibrium at*

$$a_{BN} = \frac{I}{\mu\,(I-1)} \quad \text{for} \quad \mu > \frac{I}{2\,(I-1)} \,. \tag{15}$$

Conversely, the unique pure strategy Nash equilibrium is the corner point monopoly price

$$a_{BN} = a_M = 2 \quad \text{for} \quad \mu \leq \frac{I}{2\,(I-1)} \,.$$

Finally, when $\mu < \infty$, the perfectly competitive price $a_{PC} = 0$ is strictly dominated; that is, it is never optimal for a market maker to post an offer at the midprice of the asset.

For a proof see Appendix D.

The Bertrand–Nash equilibrium offer has very natural properties. First, a_{BN} is decreasing in competition as proxied by the number of players. Second, as the value of the latency parameter $\mu \to \infty$, a_{BN} converges to zero profit ($a_{PC} = 0$). As uncertainty about execution falls and the best price captures the incoming trade with increased certainty, the model's equilibrium approximates that of the traditional Bertrand pricing model of zero profit. Finally, we note that the zero profit, perfectly competitive, outcome is always dominated for finite I and finite μ, as there is always a non-zero probability of obtaining strictly positive profits at a higher price, see (14).

The Bertrand–Nash equilibrium offer assumes that market makers can quote an offer anywhere in the interval $[0, 2]$. However, the stage game is played with a set of discrete actions $\mathcal{A} = \{\vartheta, 2\,\vartheta, \ldots, 2\}$, so the Bertrand–Nash equilibrium offer may not be a point on the price grid of the LOB.[13] Therefore, in Proposition 5 below we derive the set of pure strategy Nash equilibria for the stage game with a discrete action space, and we show that the set is non-empty. We also obtain bounds that define the set of pure strategy Nash equilibria of the game, and show that the bounds converge to the Bertrand–Nash equilibrium offer as the size of the tick in the LOB approaches zero.

Proposition 5 *Assume that the number of players I is finite. Let the latency parameter $\mu \in (I/\big(2\,(I - 1), \infty)$ and the tick size $\vartheta \in (0, 1]$, so the action set $\mathcal{A} = \{\vartheta, 2\,\vartheta, \ldots, 2\}$ is non-empty. Then the following three statements hold:*

5.1 An offer a_k in the LOB is a pure strategy symmetric Nash equilibrium of the stage game with discrete actions if

$$L^* := \frac{\vartheta\, I}{(I-1)\,(\exp(\mu\,\vartheta) - 1)} \leq a_k \leq \frac{\vartheta\, I}{(I-1)\,(1 - \exp(-\mu\,\vartheta))} =: U^* \,. \tag{16}$$

5.2 The set $[L^*, U^*] \cap \mathcal{A}$ is non-empty.

[13] We ignore pricing at the midprice, $a_i = 0$, as it is a strictly dominated strategy.

5.3 *The Bertrand–Nash equilibrium offer satisfies the bounds in* (16)*, i.e.,* $L^* \leq a_{BN} \leq U^*$*, and the bounds* L^* *and* U^* *converge to* a_{BN} *in* (15) *as the size of the tick* $\vartheta \to 0$.

For a proof see Appendix D.

In the parameterised versions of the model we consider, we find that the set of pure strategy Nash equilibria $[L^*, U^*] \cap \mathcal{A}$ often has more than one element. Therefore, to distinguish the Nash equilibria in the set, we call $a_L = \min\,[L^*, U^*] \cap \mathcal{A}$ and $a_U = \max\,[L^*, U^*] \cap \mathcal{A}$ the most competitive and the most collusive pure strategy Nash equilibrium of the stage game with a discrete action space, respectively. Of course, $a_L = a_U$ when the set of pure strategy Nash equilibria is a singleton.

In consonance with the results in Section 3, the dynamics from all our algorithms (except Q-learning) converge to one of the pure strategy Nash equilibria in $[L^*, U^*] \cap \mathcal{A}$, under appropriate conditions. Given Proposition 5.3, as the tick size decreases, the bounds that define the set of Nash equilibria in the stage game with a discrete action space shrinks around the Bertrand–Nash equilibrium offer. Therefore, the maximum rents that the algorithms can extract also decrease when the size of the tick in the LOB decreases.[14]

4.3. Equilibria selection and tacit collusion

Often, the stage game with a discrete action space has more than one pure strategy Nash equilibrium. There is no guarantee that the algorithms will select the most competitive equilibrium. In Section 2, we characterised the behavior of algorithms as a deterministic system of ODEs which we use to address the problem of equilibrium selection. When there are multiple Nash equilibria in a game between competing algorithms, the likelihood of these algorithms converging to a particular equilibrium depends on the initial conditions of the policies from those algorithms, i.e., it depends on the basin of attraction of the Nash equilibrium in which the initial conditions falls. For any prior on the space of initial strategies for the algorithms, we can determine the probability of converging to the different Nash equilibrium outcomes using their basins of attraction. In particular, we use a uniform prior to provide an estimate for the probability of reaching each equilibrium. Recall that in Section 2.3 we assumed that γ_n satisfies (2) and γ_1 is sufficiently small to skip the transient phase so that the trajectories of the algorithms immediately follow the trajectories of the ODEs.[15]

This is most clearly illustrated by setting $\vartheta = 1$. In our model, a tick size $\vartheta = 1$ with two market makers ($I = 2$) reduces the stage game to a 2×2 symmetric game with interesting properties given by the following proposition.

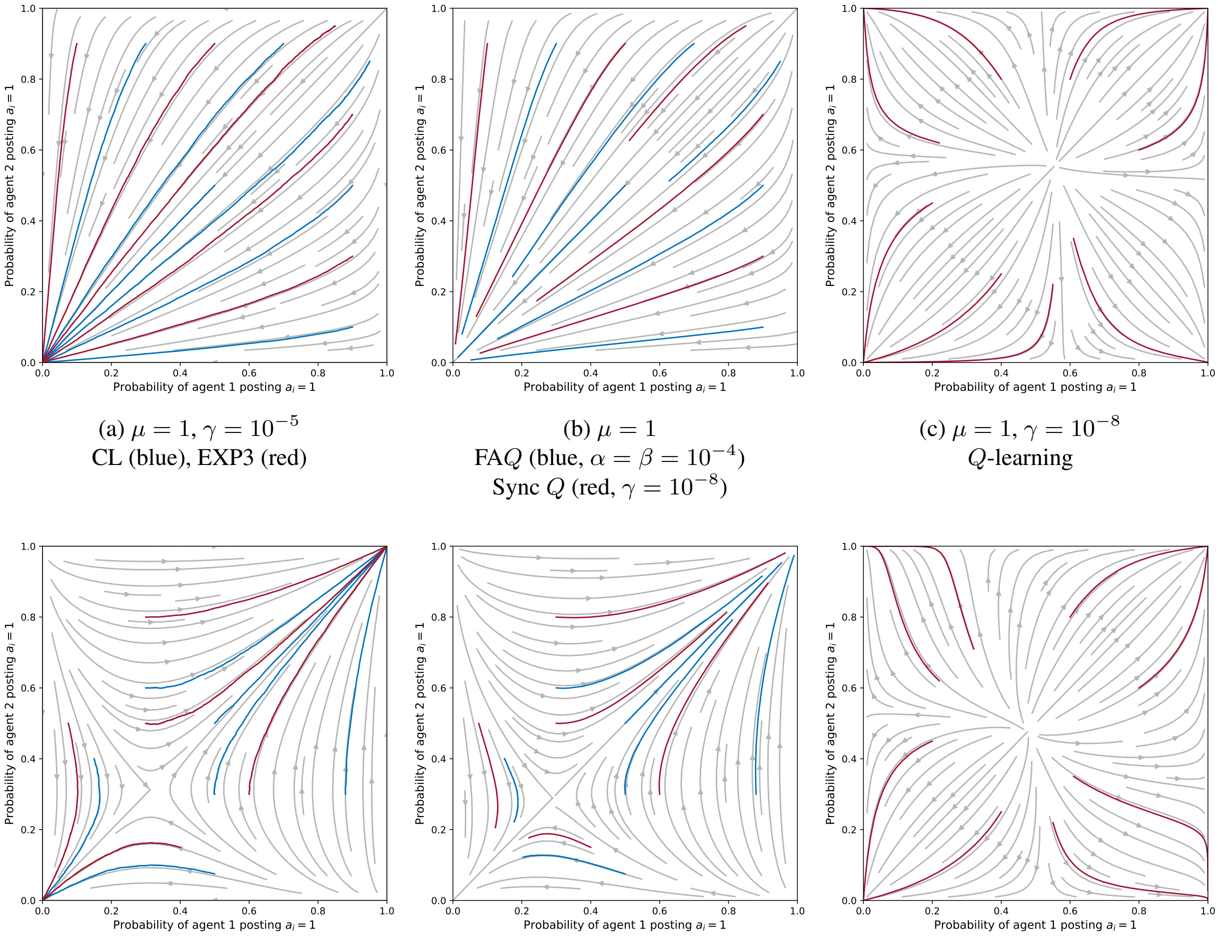
Proposition 6 *Let the number of market makers be* $I = 2$ *and let the tick size be* $\vartheta = 1$ *so that the action space is* $\mathcal{A} = \{1, 2\}$*. If the value of the latency parameter is such that* $\mu \leq \ln 3$*, then the outcome* $(a_1, a_2) = (2, 2)$ *is the only pure strategy Nash equilibrium of the stage game. Conversely, if the value of the latency parameter is such that* $\mu > \ln 3$*, then the game has two pure strategy Nash equilibria,* $(a_1, a_2) = (2, 2)$ *and* $(a_1, a_2) = (1, 1)$.

[14] Given the discrete nature of the grid, the expected rents may not be a strictly decreasing function of ϑ. For example, for some particular value of ϑ the highest equilibrium may be the Bertrand–Nash equilibrium offer, which will be lower than one with some smaller ϑ which does not have the Bertrand–Nash equilibrium offer as part of the resulting grid. More precisely stated, the expected rents are bounded by a strictly decreasing function of ϑ.

[15] Alternatively, the assumption that γ_1 is small can be removed provided that one can construct an appropriate prior to capture the behaviour of the transient phase before the trajectories lock-in.

Proof The outcome $(2, 2)$ gives each player an expected profit of $a_i/2 = 1$. A deviation to a lower offer price $a_i = 1$ leads to an expected profit of $1/(1 + \exp(-\mu))$, which for finite μ is strictly less than one. Hence $(2, 2)$ is a Nash equilibrium for all values of the latency parameter μ.

The outcome $(1, 1)$ is an equilibrium if deviating to the higher offer $a_i = 2$ generates a lower expected payoff than quoting at the lower offer $a_i = 1$. The expected payoff from the deviation is worse if $1/2 > 2\,(1 + \exp(\mu))^{-1}$; i.e., if $\mu > \ln 3$. □



(a) $\mu = 1$, $\gamma = 10^{-5}$
CL (blue), EXP3 (red)

(b) $\mu = 1$
FAQ (blue, $\alpha = \beta = 10^{-4}$)
Sync Q (red, $\gamma = 10^{-8}$)

(c) $\mu = 1$, $\gamma = 10^{-8}$
Q-learning

(d) $\mu = 2$, $\gamma = 10^{-5}$
CL (blue), EXP3 (red)

(e) $\mu = 2$
FAQ (blue, $\alpha = \beta = 10^{-4}$)
Sync Q (red, $\gamma = 10^{-8}$)

(f) $\mu = 2$, $\gamma = 10^{-8}$
Q-learning

Figure 1: Theoretical and actual trajectories with different values of μ for all the algorithms in 2×2 games with tick size $\vartheta = 1$ and action space $\mathcal{A} = \{1, 2\}$.

Figure 1 depicts the trajectories of the different algorithms with $\mu \in \{1, 2\}$. This allows us to compare the behavior of the algorithms for the case where there is one Nash equilibrium with the case when there are two Nash equilibria as shown in Proposition 6.

The figure describes the dynamics of the algorithms $i \in \{1, 2\}$ through the probability x_{i1} that the algorithm offers $a_i = 1$; clearly, the probability that the algorithm offers $a_i = 2$ is $x_{i2} = 1 - x_{i1}$. That is, i's policy is fully characterised by x_{i1}. The figure shows the theoretical trajectories of policies in grey and the actual trajectories of policies from the algorithms in blue and red (for the different algorithms that recover the same dynamics) as they evolve from different starting points. For all the variations of Q-learning, we set the exploration-exploitation parameter $\tau = 100$ and the discount factor $\delta = 0.75$. The actual trajectories from the algorithms are obtained by playing the stage game for 2×10^7 iterations. We choose values of the learning rate γ (see sub-captions) that achieve an almost perfect approximation. The simulations use a constant learning rate for computational convenience because the trajectories of the algorithms move very slowly when γ_n satisfies (2). Furthermore, a constant learning rate acts as a proxy for the case where γ_n satisfies (2) with γ_1 sufficiently small because both achieve uniform convergence immediately.

The figure shows six panels, the top three correspond to $\mu = 1$, where our theoretical results tell us there is only one Nash equilibrium represented by $(x_{11} = 0, x_{21} = 0)$ so both market makers post offers at $a_i = 2$ with probability one. The bottom three correspond to $\mu = 2$, where our theoretical results tell us there are two equilibria, the more collusive outcome represented by $(x_{11} = 0, x_{21} = 0)$ with offers at $a_i = 2$, and the more competitive outcome represented by $(x_{11} = 1, x_{21} = 1)$ with offers at $a_i = 1$. The two figures on the left represent the replicator dynamics (for the Cross learning and EXP3 algorithm), the middle two are the replicator-mutation dynamics (for the FAQ-learning and synchronous Q-learning algorithms), and the two figures on the right represent the dynamics of Q-learning once we map the trajectories of Q-values to the trajectories of policies.

We see that the trajectories of Cross learning and the EXP3 algorithm converge to a pure strategy Nash equilibrium of the stage game; and when the value of τ is sufficiently large (i.e., little to no exploration), the trajectories of FAQ-learning and synchronous Q-learning converge to a pure strategy Nash equilibrium of the stage game. Furthermore, we see that the trajectories of policies from Q-learning converge to asymmetric actions in a symmetric game, which is not optimal for one of the market makers and therefore not a Nash equilibrium of the stage game. Hence, the results from Propositions 1–3 validate our numerical results.

When multiple equilibria exist (i.e., for high values of μ in Proposition 6), the equilibrium reached depends on the initial conditions of the policies. When $\mu = 2$, the basin of attraction of the competitive outcome covers 75.65% of the area of the unit square for the replicator dynamics, and 77.2% for the replicator-mutation dynamics. In this setting, assuming that the initial policies are uniformly distributed over the unit square, tacit collusion though possible, is less likely than the competitive outcome.

When the stage game has multiple Nash equilibria, we use the term tacit collusion to describe the situation where the algorithms tacitly coordinate on a Nash equilibrium of the stage game that is more profitable than the most competitive Nash equilibrium of the stage game. Our choice of terminology is based on the argument that the players' strategies embodied in the algorithms include a reward-punishment scheme that allows supracompetitive offers to be sustained under certain circumstances.[16] In economic theory, collusion embodies a reward-punishment scheme (Harrington, 2018). That is, if a player cooperates, then the player is rewarded in the future by other players who continue to cooperate; while if a player defects, then the player is punished in the future by the other players to reduce the deviating player's profits. The algorithms' vector fields illustrate the reward-punishment scheme in our setting of imperfect private information where the algorithms do not observe prices and the rewards received are noisy.[17]

[16] We describe tacit collusion as supracompetitive offers relative to the most competitive Nash equilibrium of the stage game, which we consider as the baseline non-collusive oligopolistic outcome. Another definition for supracompetitive outcomes could use price equal to marginal cost as the reference competitive outcome of the repeated game. As the marginal cost is zero, all outcomes would then be collusive.

[17] As seen in the discussion after Proposition 3, the behaviour of Q-learning is different from that of the other algorithms. However, for a streamlined discussion, we keep the same definition of collusion across all algorithms.

Consider Figure 1d where player one's algorithm is cooperative and starts quoting the collusive offer with a high probability (e.g., $x_{11} = 0.2$). If player two's algorithm is cooperative and also starts quoting the collusive offer with a high probability, then both algorithms will gradually learn and adapt to further cooperate and both players will be rewarded. On the other hand, if player two's algorithm starts quoting the competitive offer with a higher probability, player one's algorithm cannot detect the deviation immediately because of the setting of imperfect private information. However, player one's algorithm will gradually learn and will adapt to the deviation by increasing the probability of quoting the competitive offer and punish player two. Thus, the boundary between the basins of attraction between the collusive outcome and the competitive outcome can be thought of as the trigger for the algorithms' cooperation or retaliation.

Tacit collusion in our context can also be interpreted as "blundering into tacit coordination" where the algorithms blunder into a more profitable Nash equilibrium outcome of the stage game (see Green, Marshall and Marx, 2014). However, the algorithms represent a player's strategy in the repeated game, and the player's strategy is further characterised by the algorithm's parameters and initial conditions. Thus, blundering into a more profitable Nash equilibrium of the stage game may be a result of the players' strategic selection of parameters, including initial conditions, that are more likely to converge to one of the more profitable Nash equilibria.

Appendix B provides additional supporting material on the 2×2 game. Figure B.8 studies the effect of the exploration-exploitation parameter τ for frequency adjusted and synchronous Q-learning; Figure B.9 studies the effect of the discount parameter δ for Q-learning; Figure B.10 compares the trajectories for different configurations of the learning rate γ; and finally, Figure B.11 demonstrates that the stochastic approximation works for asymmetric algorithms with asymmetric actions and asymmetric learning rates.

4.4. Discretisation, tacit collusion, and the quoted half-spread

In the algorithmic collusion literature, a common way to measure tacit collusion is through the excess rents extracted relative to the Bertrand–Nash equilibrium, i.e., $a_* - a_{BN}$ where a_* is the equilibrium offer that the algorithms reach (see Calvano et al., 2020). In this section, we break down the effect of competition on the half-spread into three components: the Bertrand–Nash equilibrium offer when the tick size $\vartheta \to 0$; the mechanical effect of the discrete grid defined as the difference between the most competitive Nash equilibrium offer and the Bertrand–Nash equilibrium offer, i.e., $a_L - a_{BN}$; and the effect from tacit collusion defined as $a_* - a_L$. We illustrate our proposal in the context of our model with $I = 2$ and $\vartheta = 1$, and introduce a novel component of the spread generated by collusive behavior.

In our model, the quoted half-spread is determined by the interaction between competing algorithms, and depends on the tick size. When the action space is continuous, we showed that the quoted half-spread should be equal to the Bertrand–Nash equilibrium offer a_{BN}. This is the first component of the half-spread and is determined by the number of competitors I and the value of the latency parameter μ; see Proposition 4. In an ideal setting, the reference for setting the tick size should be the Bertrand–Nash equilibrium offer a_{BN}. Yet, it is not reasonable to assume that one can set the tick size exactly at $\vartheta = a_{BN}$ for a number of reasons: model and market parameters may not be known with exact precision, they may not be constant, etc.

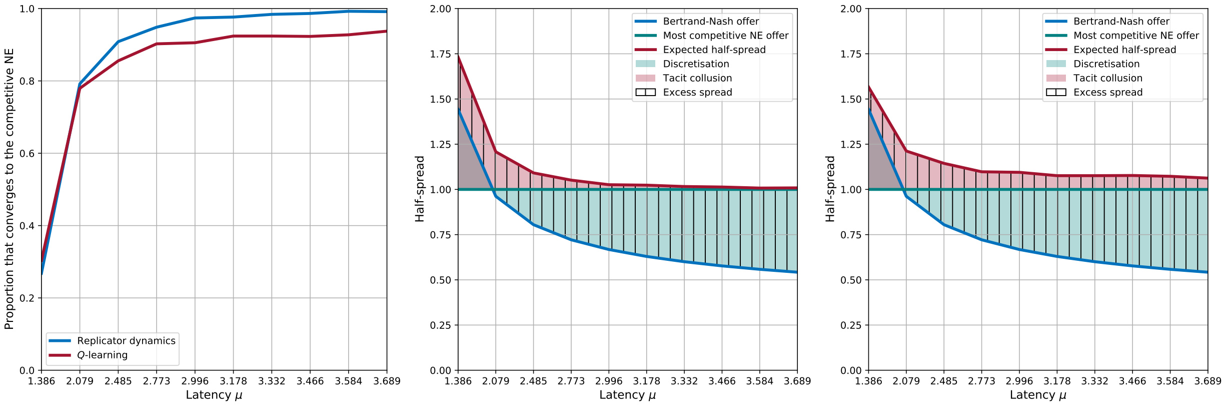
The other two components are due to the discreteness of the grid parameterised by ϑ. The second component of the half-spread is due to the mechanical effect of a discrete grid, namely that the half-spread from the Bertrand–Nash equilibrium offer a_{BN} may not be part of the price grid. This wedge between the observed and "real" price has been identified and its implication studied since at least the early work of Gottlieb and Kalay (1985). To illustrate this in our model, consider a very large tick size $\vartheta = 2$. Then, the algorithms can only select one of two offers: $a_i = 0$ or $a_i = 2$. As the algorithms have no incentive to post $a_i = 0$ (it is strictly dominated for $\mu < \infty$) there is only one Nash equilibrium at $a_i = 2$ and the resulting half-spread contains additional profits over the Bertrand–Nash equilibrium offer a_{BN}, which is due exclusively to the mechanical effect of a coarse grid.

One of our contributions is to introduce a third component in the half-spread due to the possibility of tacit collusion. The outcome of the equilibrium half-spread is determined by which equilibrium the algorithms converge to. By imposing a tick size, the set of equilibria changes, thus even if one could set $\vartheta = a_{BN}$ the interaction between the algorithms may result in a equilibrium half-spread that is not the offer a_{BN}.

With tick size $\vartheta = 1$, a low value of the latency parameter $\mu \leq \ln 3$ leads to a single Nash equilibrium at $a_i = 2$, so the algorithms generate the maximum profit. When the value of μ is low, there is little competitive pressure so a large tick size drives up the half-spread because the probability that the best offer is filled is lower.

Note that in our model, the second component of the half-spread is not always positive, it can also be negative. In the literature, the quoted spread is usually set by the zero profit condition as in Glosten and Milgrom (1985) and Li and Ye (2021), so whenever the equilibrium half-spread is not on the discrete price grid, the quoted spread increases. To see this, note that when the zero profit condition is between two price levels, the equilibrium half-spread cannot drop to the lower price point, as that generates strictly negative profits, so the offer is posted at the higher price on the grid. In our setting, with $\vartheta = 1$ and if $a_{BN} \in (1, 2)$, the most competitive Nash equilibrium offer would be at the higher offer $a_i = 2$ when $\mu \leq \ln 3$, or the lower offer $a_i = 1$ when $\mu > \ln 3$. Thus in our setting the second factor, the mechanical effect of a discrete price grid, can be either positive or negative.

Another contrast between our results and those in the extant literature is that in our setting there may be more than one possible Nash equilibrium of the stage game, see Proposition 6. Whenever there is more than one Nash equilibrium, we define the effect from tacit collusion as the difference between the equilibrium offer and the most competitive Nash equilibrium offer. With tick size $\vartheta = 1$, the effect from tacit collusion occurs with higher latency values $\mu > \ln 3$. A higher value of μ leads to more competitive pressure because there is more certainty that the lowest offer wins the trade, which generates the possibility of a more competitive Nash equilibrium offer that is closer to a_{BN}.



(a) Convergence to the most competitive NE

(b) Replicator dynamics

(c) Q-learning

Figure 2: Equilibrium selection and expected equilibrium half-spread with $\vartheta = 1$.

In Figure 2, we study the model for several values of μ with $\vartheta = 1$ to further understand how the equilibrium spread from competing algorithmic market makers and its components change with the value of the latency parameter μ when multiple equilibria exists, i.e., $\mu > \ln 3$. The results are taken from Section 5.1 where the details surrounding the experiment are discussed in detail. Figure 2a illustrates how the basin of attraction of the competitive outcome increases as the value of μ increases, and with it the corresponding likelihood of tacit collusion decreases.

To complement the analysis, Figures 2b and 2c show the expected equilibrium half-spread generated by the algorithms. The figure shows the three components of the expected equilibrium half-spread: the Bertrand–Nash equilibrium offer, the most competitive Nash equilibrium offer, and the cost of tacit collusion. The vertical lines in the figure indicate the excess spread relative to the Bertrand–Nash equilibrium offer. The excess spread (related to the excess profit is a common measure for tacit collusion used in the literature) is the sum of the mechanical effect of the discrete grid and the cost from tacit collusion in our decomposition.

Figure 2b looks at the breakdown of the spread into its three components for the replicator dynamics, which is representative of all the algorithms with the exception of Q-learning. Initially, for the lowest values of μ, the component due to the tick size in the LOB is negative because the most competitive Nash equilibrium offer is below the Bertrand–Nash equilibrium offer. However, tacit collusion occurs quite often and its effect is greater than the effect due to a discrete action space. As the value of μ increases, the likelihood of reaching the most competitive Nash equilibrium increases so the effect of tacit collusion decreases. However, the increase in the value of μ leads to a decrease in the Bertrand–Nash equilibrium offer, which falls below the most competitive Nash equilibrium; thus, the increase in the discreteness component is larger than the reduction in the effect of tacit collusion. Overall, the net effect is that the level of excess spread increases, and the increase is due to the discreteness in the LOB. In Figure 2c, the decomposition for Q-learning is similar to those from the replicator dynamics. The primary difference is that the effect of tacit collusion does not fall as rapidly and stays positive even as the value of μ becomes large.

Our analysis suggests that a large tick size ($\vartheta = 2$) runs the risk of amplifying trading costs to liquidity takers by creating a barrier for competition and allowing the algorithms to sustain wider spreads.[18] A slight decrease in the tick size ($\vartheta = 1$), can substantially improve competition and reduce trading costs by allowing the more competitive offer $a_i = 1$ to become the equilibrium one, which reduces the spread down to two ticks if the outcome of competition leads to the more competitive outcome. Our analysis underlines the complexity of implementing an m-tick policy (see for example Li and Ye, 2021) as the size of the spread is endogenously determined with the tick size. This makes it extremely difficult for the regulator or the firm's managers to determine in advance the exact number of ticks the equilibrium spread would achieve after setting the size of the tick or the nominal value of the asset. In particular, a two-tick spread, or more precisely, a one-tick half-spread can be the most competitive outcome or the most collusive outcome, and we may not know which one it is because the spread cannot shrink further.[19] A slightly smaller tick size, one that generates more ticks within the spread, allows the possibility for the algorithms to compete, and is thus less likely to help sustain uncompetitively wide spreads.

5. Tick size and tacit collusion among algorithms

We saw that a large tick size can be a source of uncompetitive profits among market making algorithms, and saw how the competitive pressure that results from a larger value of the latency parameter μ (where

[18] When $\vartheta = 2$, the spread is guaranteed to remain at two ticks because $a_i = 2$ is the only Nash equilibrium in this setting as $a_i = 0$ is strictly dominated for $\mu < \infty$.

[19] We observe one tick spreads, i.e., half-tick half-spreads, but that is equivalent to our one tick half-spread when the mid price is at 0.5 instead of at zero.

a better offer has a higher certainty of filling the trade) reduces the opportunity for tacit collusion while increasing the gains from discretisation with a large tick size. In this section, we explore the effect of competition on the half-spread as we reduce the tick size. We study the relationship between our variables of interest using two metrics. The first is the likelihood of tacit collusion measured as the proportion of non-competitive outcomes relative to the most competitive Nash equilibrium offer. The second metric looks at excess profits relative to the Bertrand–Nash equilibrium offer. In terms of the decomposition of the spread we saw above, this corresponds to the expected profits from the sum of the discreteness and tacit collusion components of the spread.

Based on the theoretical results from Sections 2 and 3, and Proposition 5, we know that a smaller tick size shrinks the bounds that define the set of Nash equilibrium outcomes and hence limits the potential profits from tacit collusion for all the algorithms we consider, except Q-learning. The upper bound in Proposition 5.1 defines an upper bound for the excess profits relative to the Bertrand–Nash equilibrium offer for a given tick size ϑ and latency μ. Nonetheless, the results from simulating the trajectories provide further insight into the relative size of the basins of attraction for the set of pure Nash equilibria, and enable us to study the outcomes of Q-learning in a (semi-) deterministic manner.[20]

We explore the effect of the tick size ϑ and the latency parameter μ on the excess profits obtained with $I = 2$ market makers. For combinations of ϑ and μ, we find that a smaller tick size facilitates competition, but the gains from an increasingly smaller tick size become very small and may not be compensated by other costs not explicitly considered in the model. An example of such costs is the time it takes for the algorithms to reach the pure Nash equilibrium (as in Cordella and Foucault, 1999). These are not captured by our two metrics that are computed using only the asymptotic behaviour of the algorithms.

We find that reducing the tick size facilitates competition and reduces the excess rents extracted by the algorithms. However, we show that very often algorithms converge to actions that do not usually correspond with those of the most competitive Nash equilibrium. Specifically, tacit collusion from not quoting the most competitive Nash offer is prevalent for a small tick size. Furthermore, Q-learning results in a sub-optimal outcome for both the market makers and liquidity takers. First, competition between Q-learning algorithms tends to generate asymmetric actions, where one market maker earns less than its counterpart and does not exploit potentially advantageous deviations. Second, Q-learning algorithms often lead to supracompetitive offers above the most collusive Nash equilibrium outcome, which increases the trading costs for liquidity takers. Finally, we find that a smaller tick size reduces the speed of convergence to a rest point, which may lead to situations where a tick size of zero is never optimal as in Cordella and Foucault (1999).

5.1. Asymptotic learning outcomes

Setup

When there are more than two market makers ($I > 2$) and more than two actions ($K > 2$), it is difficult to depict the policies from the algorithms in two-dimensions because the vector fields of policies are defined on an $I \times (K - 1)$ domain. From Proposition 5, we know that the set of Nash equilibria is bounded and the set of equilibria may contain more than one element. We also know that the bounds, which define the set of Nash equilibria, converge to the Bertrand–Nash equilibrium offer a_{BN} as the tick size decreases. Another advantage of characterising the behaviour of the algorithms as a deterministic system of ODEs is that we can study the size of the basins of attraction of the rest points (which is the set of Nash equilibria except for Q-learning) to understand the likelihood of reaching a particular offer. For a combination of parameter values, we simulate the trajectories of the system of ODEs from random initial conditions until the ODEs

[20] The trajectories from the dynamics are deterministic, but subject to the randomness of initial conditions.

reach a rest point. The initial conditions are randomly sampled with $1000 \times K$ starting conditions for each tick size. We increase the sample size with K to compensate the increase in dimensionality of the space of policies as the number of actions increases.[21]

We focus our study on the replicator dynamics and the dynamics of Q-learning. We omit the replicator-mutation dynamics because the replicator-mutation dynamics resemble the replicator dynamics for large values of τ, i.e., sufficient exploitation, which is required for the replicator-mutation dynamics to converge to a pure strategy Nash equilibrium. We restrict our analysis to the case with two market makers ($I = 2$) because the theoretical results that guarantee convergence to a pure strategy Nash equilibrium of the stage game in Section 3 only hold for two market makers.

The initial conditions for the replicator dynamics are sampled from a multivariate flat Dirichlet distribution for each player, which is equivalent to a uniform distribution over the $K - 1$ simplex. If we uniformly sample the Q-values over a large range with a large value of τ, the policy will likely start near a corner point. Therefore, the initial conditions for the dynamics of Q-learning are sampled from a uniform distribution between $[1 - 1/(2\,\tau), 1 + 1/(2\,\tau)]$ for each Q_{ik}.

Convergence to a rest point (long-term stable state) for the replicator dynamics is achieved when each player has more than 99% probability of playing the same action, i.e., convergence is achieved once the policies "essentially" reach a pure strategy Nash equilibrium. We have no analytical characterisation of where the dynamics of Q-learning will converge, so we apply a different convergence criterion: convergence to a rest point for the dynamics of Q-learning is achieved when $\dot{\boldsymbol{Q}} \approx \boldsymbol{0}$, i.e., each component of $\dot{Q}_{ik}$ is less than 10^{-10}. Both the replicator dynamics and the dynamics of Q-learning are simulated with a step size of $\Delta t = 0.01$. The dynamics of Q-learning are simulated through the Q-values with a discount factor $\delta = 0.75$, the Q-values are then mapped to the policy with the exploration-exploitation parameter $\tau = 100$.

We study combinations of tick size $\vartheta = 1, 1/2, 1/3, \ldots, 1/10$ and latency $\mu = \ln(4\,c)$ for $c = 1, \ldots, 10$. For each pair (ϑ, μ), we simulate the trajectories of the replicator dynamics and the dynamics of Q-learning from random initial conditions. Once each trajectory converges, we record the action that has the highest probability of being played.[22] We use the convergent policies ($\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$) to compute the profits of each player given by $\hat{\boldsymbol{x}}_1^\top\, \boldsymbol{\Pi}\, \hat{\boldsymbol{x}}_2$ and $\hat{\boldsymbol{x}}_2^\top\, \boldsymbol{\Pi}\, \hat{\boldsymbol{x}}_1$. We repeat this for every initial condition and compute the average of all the profits to estimate the expected profits $\bar{\pi}_*$. The excess profit $\bar{\pi}_* - \pi_{BN}$ is the difference between the expected profit and the expected profit from the Bertrand–Nash equilibrium offer.

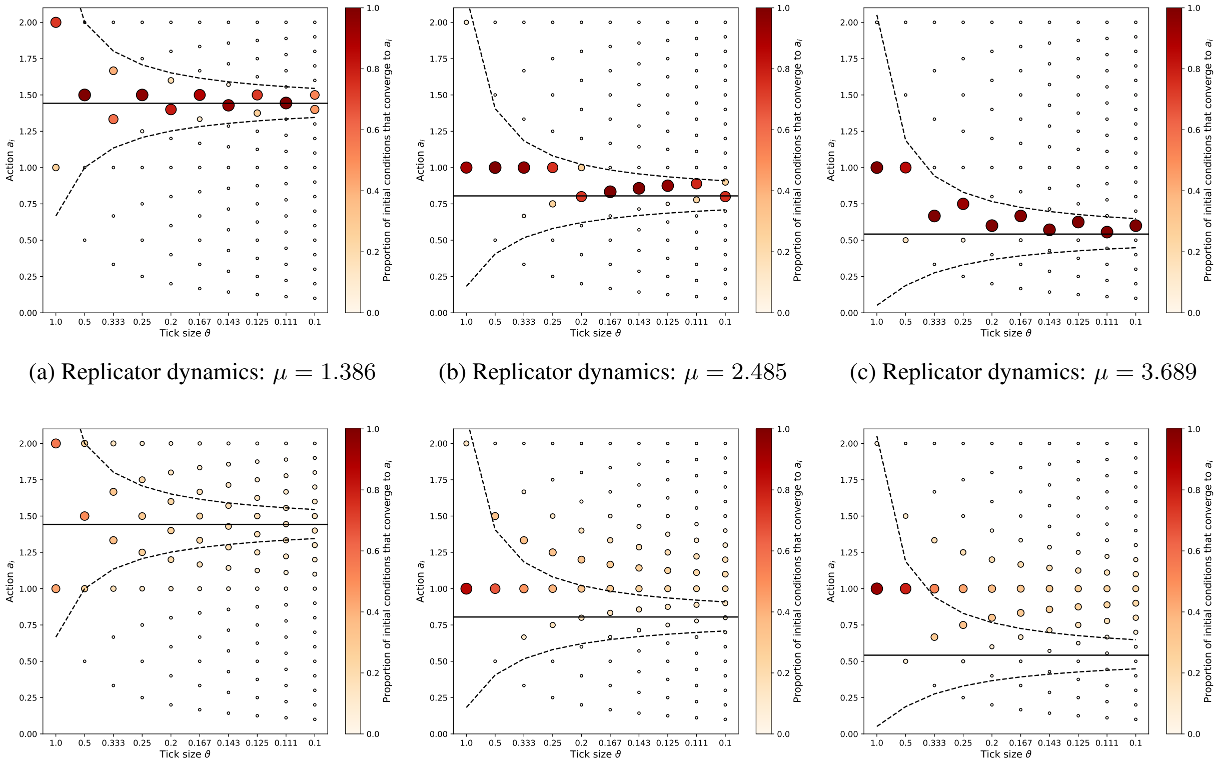
Asymptotic actions

Figure 3 plots the proportion of initial conditions that converges to a particular action as a function of the tick size for three values of the latency parameter $\mu = \ln(4\,c)$ where $c = 1, 3, 10$ with $I = 2$ market makers. The black horizontal line is the Bertrand–Nash equilibrium offer a_{BN} of the stage game with a continuous action space and determines the first component of the spread. The offers between the bounds (dashed lines) are the elements of the set of symmetric pure strategy Nash equilibria from the stage game with a discrete action space characterised in Proposition 5.1. The most competitive Nash equilibrium offer is the lowest offer within the bounds, and the distance between the most competitive Nash equilibrium offer and the Bertrand–Nash equilibrium offer determines the second component of the spread, the one due to the discreteness of the grid. The size of the basin of attraction is depicted by the filled-in circles representing

[21] The number of samples grows linearly in K even though the volume of the space of policies grows exponentially in the number of actions, to keep computational tractability.

[22] Converging to an action is clear for the replicator dynamics because the action will have more than a 99% probability of being played. On the other hand, converging to an action for Q-learning is computed by mapping the long-term stable state of the Q-values to the policy, then the action that has the largest probability of being played is the action recorded.

the price grid. The size of the grid points, as well as the colour with which they are filled represents the relative proportion of initial conditions that converge to that (symmetric) offer. For example, in Figure 3a with $\vartheta = 1$, we obtain that 26.85% and 73.15% of the initial conditions converge to the symmetric offer $a_i = 1$ and $a_i = 2$, respectively. The distance from $a_i = 1$ to the line (at 1.443) represents the (negative) component of the spread due to the discreteness of the grid. Therefore, 26.85% of the initial conditions will result in an equilibrium offer that is better than the Bertrand–Nash equilibrium offer.



(a) Replicator dynamics: $\mu = 1.386$

(b) Replicator dynamics: $\mu = 2.485$

(c) Replicator dynamics: $\mu = 3.689$

(d) Q-learning: $\mu = 1.386$

(e) Q-learning: $\mu = 2.485$

(f) Q-learning: $\mu = 3.689$

Figure 3: Proportion of initial conditions that converges to a certain action as a function of tick size. The horizontal black line is the Bertrand–Nash equilibrium offer. The offers between the bounds (dashed lines) are the elements of the set of symmetric pure strategy Nash equilibria from the stage game with a discrete action space characterised in Proposition 5.1. The circles indicate the price grid, and the size and colour represent the proportion of initial conditions converging to the equilibrium offer.

However, the dynamics of Q-learning may not converge to symmetric actions. In those cases, we record the actions that have the largest probability of being played by each player and report the average proportion between the two players.[23]

The top three panels of Figure 3 depict the replicator dynamics and show how they always converge to an action that is a pure strategy Nash equilibrium of the stage game with a discrete action space. The majority of initial conditions converges to the action that is nearest to a_{BN}. A relatively coarse grid (parameterised by ϑ) is sufficient to obtain outcomes that are close to what is attainable with an infinitely fine grid, and the fineness of the grid roughly determines how close the offers are to a_{BN}. Most of the time, the most

[23] The proportion of initial points that converge to asymmetric actions varies with latency and the tick size. We discuss this in greater detail below.

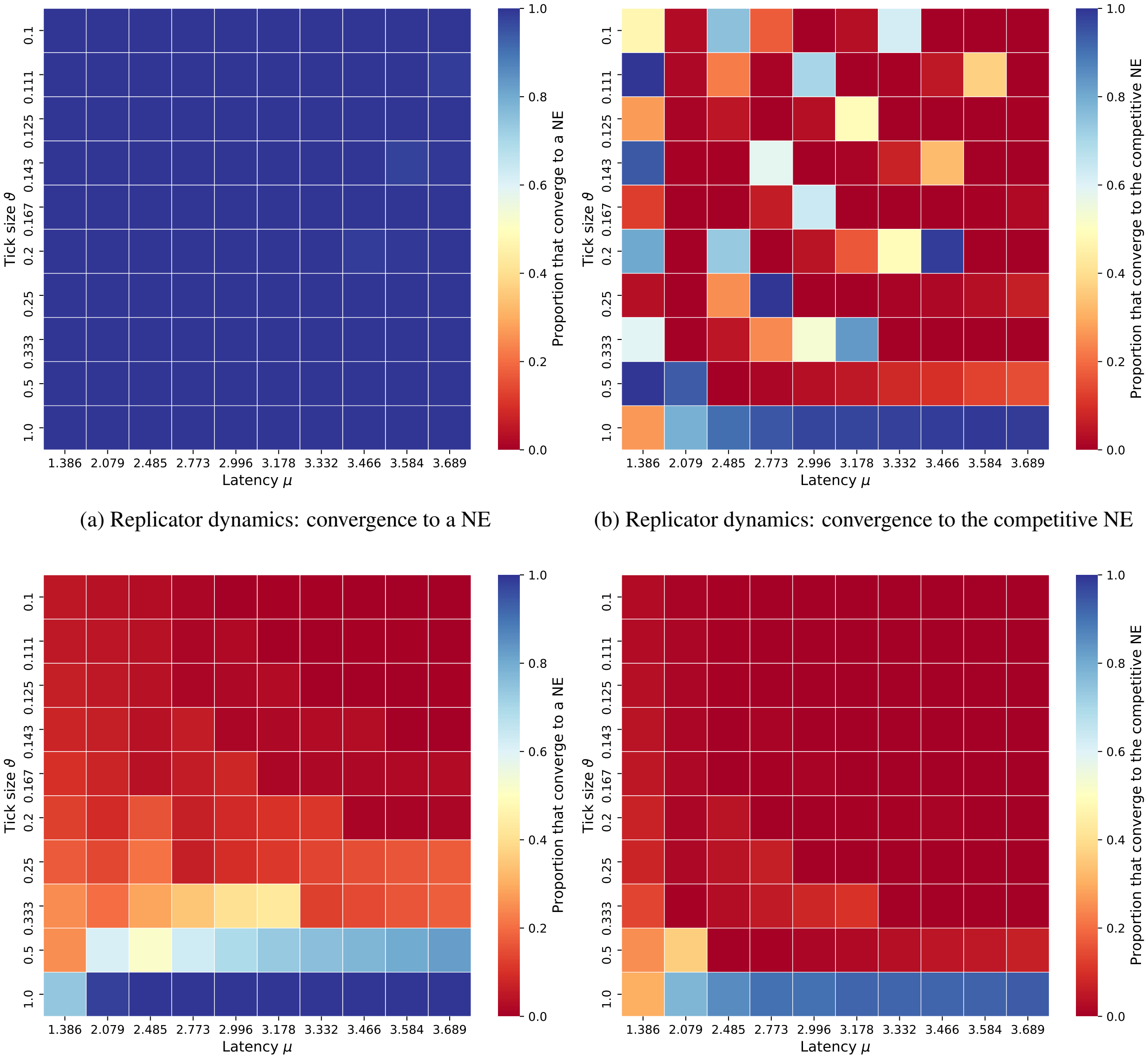
competitive outcome lies below a_{BN} but on occasion it lies above it. The excess profits (the sum of the discreteness and tacit collusion effects) are bounded by the distance between a_{BN} and the upper bound that defines the set of Nash equilibria depicted by the dashed line. We see that this distance becomes very small very quickly. To limit tacit collusion in terms of excess profits relative to the Bertrand–Nash equilibrium offer, we do not necessarily need a tick size of zero; a small but finite grid size will suffice to approximate the Bertrand–Nash outcomes.

On the other hand, the lower three panels depict the dynamics of Q-learning, where we find that the dynamics converge to actions that are not part of any pure strategy Nash equilibrium and tend to converge to supracompetitive offers above the most collusive Nash equilibrium, especially for larger values of the latency parameter. The behavior of the dynamics of Q-learning makes it less predictable and more likely to lead to supracompetitive outcomes above the most collusive Nash equilibrium. The observed tendency to converge to supracompetitive outcomes means that for these algorithms it might not be possible to place a bound on the level of tacit collusion in terms of the Bertrand–Nash equilibrium offer by reducing the tick size.

In Figure 4 we look at the importance of Nash outcomes as predictors of the outcome of algorithmic competition, and within the ones that converge to Nash, which converge to the most competitive ones. The figure illustrates the proportion of initial conditions that converge to a pure strategy Nash equilibrium and the proportion that converge to the most competitive pure strategy Nash equilibrium for the replicator dynamics and the dynamics of Q-learning for all combinations of (ϑ, μ) with $I = 2$. Given that the game is symmetric, the long-term stable state (a rest point) is a pure strategy Nash equilibrium if the actions between the players are symmetric and the action itself is a pure strategy Nash equilibrium.

The upper left panel is unsurprising, every trajectory of the replicator dynamics converges to a pure strategy Nash equilibrium. The tick size ϑ becomes important as the replicator dynamics converges to offers around the Bertrand–Nash equilibrium offer. The location of the Bertrand–Nash equilibrium offer relative to the grid of offers induced by the tick size affects the frequency of convergence to the most competitive pure strategy Nash equilibrium of the market making game. The dynamics are more likely to reach the most competitive pure strategy Nash equilibrium when the most competitive Nash offer is the closest offer to the Bertrand–Nash equilibrium offer. However, the most competitive outcome is not a frequent outcome for most combinations of (ϑ, μ). The most competitive pure strategy Nash equilibrium is a frequent outcome only for a large tick size because the most competitive Nash offer is usually the closest offer to the Bertrand–Nash equilibrium offer for most values of the latency parameter μ due to the coarse grid in the LOB. Therefore, tacit collusion from not quoting the most competitive Nash offer is prevalent for a small tick size. However, tacit collusion in the form of market makers not quoting the most competitive offer is not necessarily very costly to traders seeking liquidity, as the most collusive pure strategy Nash equilibrium can be very close to the Bertrand–Nash equilibrium offer.

The lower two panels of Figure 4 capture the dynamics of Q-learning, which are quite different from those of the replicator dynamics. The dynamics of Q-learning rarely converge to a pure strategy Nash equilibrium. Convergence to a pure strategy Nash equilibrium is more likely if the grid is coarser, although this could be a purely statistical phenomenon because the set of Nash equilibria represents a larger proportion of the points on the grid. Additionally, almost none of the trajectories from the dynamics of Q-learning converge to the most competitive pure strategy Nash equilibrium, except when the tick size is large. Furthermore, when combining Figures 3 and 4, we see that the dynamics of Q-learning have a tendency to converge to supracompetitive offers above the most collusive pure strategy Nash equilibrium. Thus, the dynamics of Q-learning can lead to significantly worse prices for liquidity takers and increasing trading costs.



(a) Replicator dynamics: convergence to a NE

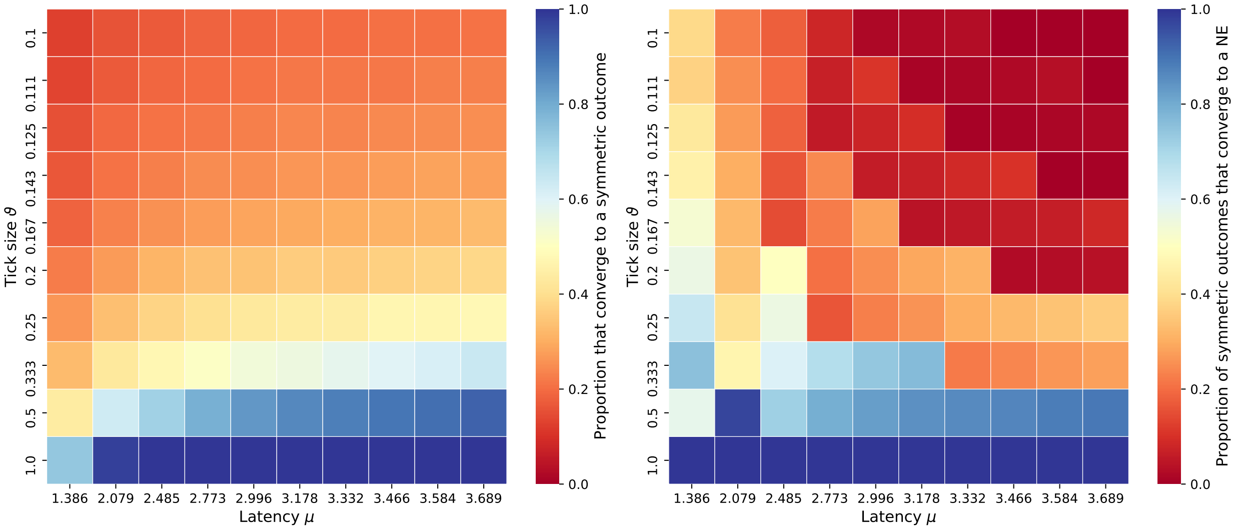
(b) Replicator dynamics: convergence to the competitive NE

(c) Q-learning: convergence to a NE

(d) Q-learning: convergence to the competitive NE

Figure 4: Proportion of replicator dynamics and the dynamics of Q-learning that converges to a pure strategy Nash equilibrium and the most competitive pure strategy Nash equilibrium of the stage game with a discrete action space as a function of (ϑ, μ).

Figure 5 depicts the outcomes from the dynamics of Q-learning in more detail. On the left panel we see that the proportion of starting points that leads to symmetric actions is relatively small, decreases as the tick size falls, and increases with the value of μ (where a better offer has a higher certainty of filling the trade). Similarly, on the right panel, we look at the convergence to a Nash equilibrium, conditional on converging to symmetric actions. In this case the pattern is reversed, once the market makers converge to a symmetric action pair, it is more likely to be a Nash equilibrium for large tick size and a small value of μ. The majority of the dynamics of Q-learning result in asymmetric actions, and when they result in symmetric actions, the actions are usually supracompetitive offers that are significantly greater than those from the most collusive



(a) Proportion of symmetric outcomes

(b) Proportion of symmetric outcomes converging to a NE

Figure 5: Proportion of the dynamics of Q-learning that converges to a symmetric outcome, and the proportion of the symmetric outcomes that converges to a pure strategy Nash equilibrium as a function of (ϑ, μ).

pure strategy Nash equilibrium. Therefore, if algorithmic market markers use the Q-learning algorithm, the high level of asymmetric outcomes (where both offers are usually supracompetitive) leads to situations where one market maker earns less than its counterpart. Moreover, if the marker makers end up making symmetric offers, then trading costs to liquidity takers will be higher than if they were using the other algorithms because the dynamics of Q-learning tend to converge to supracompetitive prices.

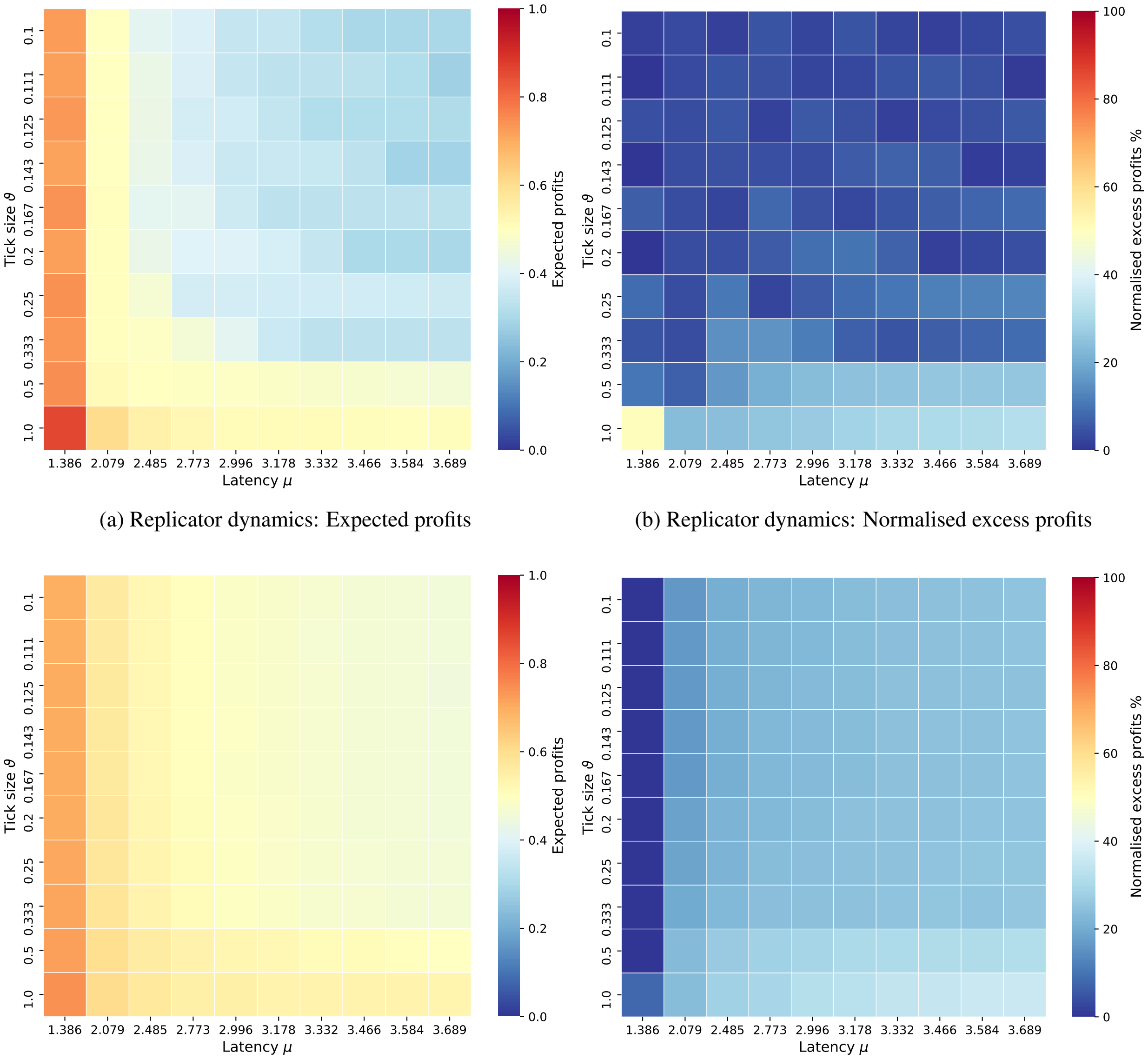
Asymptotic profits

We saw that players can collude by choosing algorithms with theoretical guarantees to reach a pure strategy Nash equilibrium that tacitly coordinate on offers above those of the most competitive Nash equilibrium. However, the additional costs from such collusion may not be substantial. Here, we look at the second metric of excess rents, the sum of the expected profits from discreteness and tacit collusion, which the algorithms extract from their equilibrium offers relative to the Bertrand–Nash equilibrium offer as a more appropriate measure of trader welfare.[24]

Figure 6 shows the expected profits at the rest points (long-term stable state) reached by the algorithms averaged over the different initial conditions, and the excess profits obtained relative to the Bertrand–Nash equilibrium offer. Excess profits are normalised as follows: denote the expected profits as $\bar{\pi}_*$, let π_{BN} denote the expected profits from the Bertrand–Nash equilibrium offer, and π_M those from the monopolistic offer (where the monopoly offer is $a_M = 2$). Then, the normalised excess profits are given by $(\bar{\pi}_* - \pi_{BN})/(\pi_M - \pi_{BN})$. For example, the expected profits are close to the monopoly offer when this metric is close to one; and the expected profits are close to the Bertrand–Nash equilibrium offer when this metric is close to zero.

The results in Figure 6 with $\vartheta = 1$ illustrate that a smaller tick size can substantially improve competition and reduce trading costs compared with $\vartheta = 2$. We see this from the expected profits that are less

[24] As discussed earlier, we measure tacit collusion from an initial level with oligopolistic profits. A more stringent measure of competition would lead to higher excess profits.



(a) Replicator dynamics: Expected profits

(b) Replicator dynamics: Normalised excess profits

(c) Q-learning: Expected profits

(d) Q-learning: Normalised excess profits

Figure 6: Expected profits and normalised excess profits across the various initial conditions for the replicator dynamics and the dynamics of Q-learning as a function of (ϑ, μ).

than one (which are the expected profits for $\vartheta = 2$) for all values of the latency parameter μ considered. Higher latency values of μ increase the likelihood that the algorithms will quote the most competitive Nash equilibrium. However, at the same time, as the value of μ decreases, the Bertrand–Nash equilibrium offer decreases; thus the algorithms could converge to a spread lower than the smallest possible offer imposed by $\vartheta = 1$. Hence, welfare can be improved by further reducing the tick size.

As we reduce the tick size ϑ, both the replicator dynamics and the dynamics of Q-learning lead to lower offers and profits. The normalised excess profits for the dynamics of Q-learning drop from the range of 30–40% to 20–30% when the tick size is reduced. On the other hand, the normalised excess profits from the replicator dynamics drop from the range of 20–30$\%$ to zero when the tick size is reduced. The lower panels depict the results for the dynamics of Q-learning, which generate higher levels of expected and excess profits, because they do not react aggressively to competing bids and ignore (by getting stuck) the best response dynamics that would bring it to a pure strategy Nash equilibrium.

Overall, we find that reducing the tick size gives regulators and managers a tool to limit or reduce the excess rent extracted by algorithmic market makers. Specifically, for a fixed value of the latency parameter μ, regulators can control the level of excess profits by reducing the tick size, because by Proposition 5.3, the bounds that define the set of pure strategy Nash equilibria move closer to the Bertrand–Nash equilibrium offer as the tick size decreases. However, the type of algorithm used by the market maker is important. In our model, the lack of responsiveness of algorithms such as Q-learning allows the algorithms to sustain supracompetitive outcomes that are inconsistent with the predictions from the equilibria of the static game.

From the point of view of market makers, the higher profit levels achieved from the use of Q-learning may appear advantageous. However, we propose two reasons to explain why this may not be optimal for market makers. First, as demonstrated in Figure 5, Q-learning often leads to asymmetric outcomes. The average profits depicted in Figure 6 obscure the fact that in the asymmetric outcomes one market maker will always trade less and earn significantly lower profits than its counterpart. For example, if we look at any path in Figure 1f that converges to the asymmetric outcome ($x_{11} = 0$, $x_{21} = 1$), that is to algorithm one playing $a_1 = 2$ and algorithm two playing $a_2 = 1$, the asymptotic profit for this realisation is 0.24 for algorithm one and 0.88 for algorithm two. As shown in Proposition 3, algorithm one is stuck on action $a_1 = 2$ when it could obtain a greater expected payoff (0.5) by switching to $a_1 = 1$ without fear of retribution, either in the short- or in the long-run. Such realisations may, and would most probably drive some market makers to abandon these algorithms, or to readjust them to move to a different dynamic path.

Second, Q-learning cannot always learn the optimal response when the opponents change their strategy, as we have just illustrated. This causes Q-learning to get stuck in a sub-optimal action and possibly be taken advantage of. This is illustrated in Figure B.11 in the appendix, where we demonstrate that the stochastic approximation works for asymmetric algorithms with asymmetric actions. In this asymmetric setting, Q-learning is “fooled” into playing a sub-optimal action by all the other non Q-learning algorithms.[25]

5.2. *Convergence times*

We saw that a smaller tick size limits the excess profits for the algorithms that converge to pure strategy Nash equilibria of the market making game. We also find, as in Cordella and Foucault (1999), that a small tick size reduces the speed of convergence to a rest point, i.e., a long-term stable state. The slower convergence may lead to situations where a tick size of zero may never be optimal. That is, the reduction in excess profits from a smaller tick size may not be compensated by other costs that are not explicitly accounted for in the model, like those associated with a slower speed of convergence to equilibrium. Despite the similarity of the result to that in Cordella and Foucault (1999), it is important to point out that slower convergence rates arise for different reasons. In Cordella and Foucault (1999) the speed of convergence is determined by the combination of the tick size and the equilibrium bidding strategies. In their model, a larger tick size allows for faster convergence because in equilibrium convergence takes place from a tick by tick undercutting strategy. In our model, algorithms do not follow such strategies and the faster convergence comes from the smaller dimensionality of the action space, so that with a larger tick size the algorithms need to figure out the best action from a set with fewer actions. Below, we illustrate an example for how a tick size of zero might not be optimal.

[25] See Appendix B for a more detailed discussion.

We first look at convergence speeds as a function of the tick size. We find that a smaller tick size leads to longer convergence times, see Figure C.12 in Appendix C. Practically speaking, the calendar time it takes for the dynamics of the algorithm to reach convergence depends on the learning rate γ_n and the calendar time between each iteration of the algorithm. The time unit from the ODEs $t = \sum_{i=1}^{n} \gamma_i$ is related to calendar time $t_c = n\,s$, where s is the calendar time between each iteration of the algorithm. For convenience, we illustrate the point with a constant learning rate so that $t = n\,\gamma$. If $\gamma = s$, the time from the ODEs t corresponds to the calendar time t_c.

As an example, consider the replicator dynamics. For a tick size $\vartheta = 1$, the ODEs take around $t = 20$ units to reach convergence to a pure strategy Nash equilibrium. Let us further suppose the algorithms use $\gamma = 10^{-5}$ as the learning rate. It means the algorithms need $n = t/\gamma = 2 \times 10^6$ iterations to reach a pure strategy Nash equilibrium. Therefore, if the algorithms post an offer every millisecond then it will take 2×10^6 milliseconds (roughly 33 minutes) to reach a pure strategy Nash equilibrium. Now, with a tick size of $\vartheta = 0.1$, the dynamics take around $t =$ 2,000 units to reach convergence. If the algorithm uses the same learning rate and the same decision frequency, then it takes around 2×10^8 milliseconds (roughly 8.5 trading days if we assume that each trading day is 6.5 hours) to reach a pure strategy Nash equilibrium. The differences in convergence times are very significant.[26]

This slower convergence needs to be evaluated with the actions chosen by the algorithms on their way to the pure strategy Nash equilibrium. If these actions involve a sequence of best responses from supracompetitive prices, the slower convergence exacerbates the costs of trading prior to reaching the Nash equilibrium outcomes. Conversely, if the actions involve a sequence of best responses from prices that are more competitive than the Bertrand–Nash equilibrium offer, the slower convergence improves the costs of trading before reaching a Nash equilibrium offer. We provide an example in Appendix C to illustrate that slower convergence can lead to higher trading costs in a finite horizon to demonstrate that a very small tick size may not be optimal. Overall, a reasonable tick size is one that is small enough to facilitate competition while at the same time being sufficiently large that convergence to a rest point is achieved within a reasonable horizon.

6. Discussion and conclusions

In this paper we looked at several algorithms and studied how they compete among themselves in a stylised market making model. First, we selected algorithms that are representative of most of the learning algorithms in use today. We characterised the stochastic dynamics of these algorithms in a repeated game setting with a deterministic system of ODEs, which we used to determine the algorithms' long-term behavior. Under suitable conditions, we found that the strategies of most algorithms converge to a pure strategy Nash equilibria of the stage game. We applied these insights to study: the outcome of competition between algorithms in a stylised market making model; how the outcome of competition can lead to tacit collusion; and how these outcomes depend on the size of the tick in the LOB.

We found that competition between Q-learning algorithms (i) does not lead to a Nash equilibrium, (ii) can lead to supracompetitive pricing and outcomes that are sub-optimal, (iii) and that reducing the tick size only helps reduce excess profits when the initial tick size is large.

For the other learning algorithms (all of which converge to Nash equilibria of the stage game), we verified that they behave as predicted by the theory, and used the asymptotic behavior of the ODEs to study the basins of attraction of the different Nash equilibria of the stage game and the speed of convergence, both

[26] In our model, we assume that market orders arrive at every iteration of the repeated game. This assumption can be relaxed by adjusting the fill probability to incorporate the arrival rate of market orders. Note that the iteration n is independent of whether markets orders are filled, the iteration increases by one whenever the algorithms perform an action.

as a function of the tick size. For these algorithms, tacit collusion can and does arise, but the excess profits are bounded by the range of possible Nash equilibria, which shrinks with the tick size. However, as the tick size decreases (and with it excess profits) the speed of convergence also decreases which may generate costs that limit how much can be gained from making the tick size very small. Overall, a large tick size can generate barriers to competition, while a smaller tick size encourages competition and lowers trading costs.

6.1. *The choice of algorithms*

The algorithms considered are specifically chosen because they are suitable for a market making setting. First, the algorithms are computationally inexpensive and fast; the time-scale between decisions in electronic markets is milliseconds at most. Second, most of the algorithms have the ability to adapt to changes in the reward as a result of the actions of adversaries. The algorithms achieve this by either updating the policies directly based on the reward (e.g., Cross learning) or they scale the reward according to the probability of playing an action (e.g., EXP3 and FAQ-learning). Finally, all but one of the algorithms we considered have dynamics that converge to well-understood dynamics which allow us to provide theoretical guarantees for convergence to Nash equilibria. The exception is Q-learning, which we include because it is one of the most popular reinforcement learning algorithms. The inclusion of Q-learning also illustrates that the stochastic approximation technique applies generally to any discrete-time stochastic process generated by the learning algorithm, which can be used to study the dynamics of other algorithms not considered here.

However, more research is required for a more exhaustive range of algorithms. There are certain algorithms with theoretical guarantees to reach a collusive outcome. For example, Hansen, Misra and Pai (2021) prove that UCB-type algorithms will always play the collusive action for symmetric 2×2 games when there is no Nature player. Therefore, it is worth investigating the case of asymmetric algorithms in more detail. Specifically, pairing an (optimal) algorithm which has theoretical guarantees to reach a pure strategy Nash equilibrium with an algorithm that is known to achieve highly collusive outcomes. Understanding if the optimal algorithm will induce the collusive algorithm to a Nash equilibrium, or if the collusive algorithm will induce the optimal algorithm to a collusive outcome will provide more insight into what regulatory oversights are required. In a related work, Cartea et al. (2022*b*) find that pairing UCB-type algorithms with an EXP3 algorithm reduces the excess rents extracted compared with the case when only UCB-type algorithms compete to provide liquidity in over-the-counter markets.

Nonetheless, we demonstrate that the deterministic approach using stochastic approximation techniques provides a more intricate and advanced understanding behind the dynamics of algorithms and its impact on tacit collusion and competition. Therefore, we argue that the deterministic approach should be considered in future research as a useful tool for the field of algorithmic collusion.

6.2. *Modelling assumptions*

Although the policy conclusions we reach are not obvious, they naturally arise out of the properties of the model we study. Rent extraction in equilibrium arises from the assumptions that (i) we model competition as taking place between large, more competitive (I) players within the bounds set by the (implicit) zero profit condition of smaller, less competitive players; and (ii) we assume that price undercutting does not lead to the capturing of the entire market (by assuming $\mu < \infty$). The first assumption allows for oligopoly and collusion while imposing an exogenous limit on rent extraction. The second allows us to regulate the degree of aggressiveness of price competition.

By making these assumptions, our starting point moves away from the common assumption in the microstructure literature generally, and in market making models in particular, that competition between market makers drives their expected profits to zero as proposed in the classic model of price competition in Bertrand (1993) (see Glosten and Milgrom, 1985; Anshuman and Kalay, 1998; Admati and Pfleiderer, 1988;

Diamond and Verrecchia, 1981; Kyle, 1985; Grossman and Stiglitz, 1980, among many others). However, as discussed in the introduction, our work falls within the literature that looks at exchanges where market makers have market power (Kandel and Marx, 1997; Kadan, 2006; Loertscher, 2008; Vives, 2011; Baruch and Glosten, 2019), where price competition need not necessarily drive prices to the point where market makers make zero profits. Our model is closest to the price competition models of Spulber (1995); Kandel and Marx (1997). In Spulber (1995) price competition takes place in a setting with uncertainty about rival's costs. In our setting, the uncertainty does not arise from differences in marginal costs (which we assume to be equal to zero), but from the possibility that the order that offers the best price is not necessarily the one that is executed first. This can arise for a number of reasons. For example, if there are differences in realised latency between the market makers, where by realised latency we refer to the time between making the decision of what offer to make and that offer reaching the LOB, relative to the random arrival of the incoming executable order; see Cartea and Sánchez-Betancourt (2021) who provide evidence for stochastic latency. In this context, we capture this uncertainty in the static stage game through the latency parameter μ which allows us to randomly shuffle the winning order while retaining the analytical simplicity of a simultaneous action stage game.

However, the presence of multiplicity of Nash equilibria in the Bertrand game played on a finite grid is not driven by the parameter μ. Kandel and Marx (1997) introduce a model of price competition with a finite number of market makers competing in price on a discrete price grid, and the model also gives rise to multiplicity of Nash equilibria. We can address this multiplicity using the ODEs. The ODEs allow us to resolve the problem of indeterminacy of multiplicity by characterising the basins of attraction of the different equilibria. Kandel and Marx (1997) addressed the issue of multiplicity in the context of the odd eighths debate, by arguing that even eights acted as a focal point to coordinate on the most preferred among the multiple equilibria.

The assumption about the presence of heterogeneous market makers with different profit structures is consistent with empirical evidence and existing models of market making. In our setting, this heterogeneity is reduced to two types of market makers: those that compete to set the best prices and the rest. We see this in the existing literature, for example in models that have fast market makers co-existing with slow market makers, as in Cartea and Penalva (2012), Hoffmann (2014), and Foucault, Kozhan and Tham (2017). In our setting, the faster market makers are able to extract rents, as in Cartea and Penalva (2012), whereas in Foucault, Kozhan and Tham (2017) these rents are driven to zero by free entry.[27] The model can be extended to include the costs of the speed advantage both by free entry (increasing I) or by making μ a (costly) choice parameter. Allowing heterogeneity in speed via the choice of μ captures the importance of speed in queue competition (as proposed in Yao and Ye, 2018).[28] We do not model the choice of μ but it is straightforward to extrapolate how repeated competition between traders for lower latency (higher values of the latency parameter μ) can easily lead to over-investment in speed as described in Biais, Foucault and Moinas (2015).

However, speed need not be the only source of competitive (dis)advantage. Market makers can have a competitive advantage in maintaining costs of inventory down, for example via brokerage arrangements with retail brokers or institutional investors, or from processing publicly available data, both from other

[27] The difference in behavior across traders in Foucault, Kozhan and Tham (2017) and Hoffmann (2014) exists because traders with a speed advantage impose costs on slower traders by using their speed to pick off stale quotes. In equilibrium these extra profits exactly offset the costs of the investments necessary to gain the speed advantage that generates those extra profits.

[28] Yao and Ye (2018) find that faster traders are better able to establish time priority and extract rents even in contexts of constrained price competition where the fundamental competitive factor is queue position. Differences in the value of μ capture this by allocating a greater share of profit to the faster trader even when setting the same price.

assets in the same market or from information across different markets.[29] Competition and heterogeneity along those dimensions can be incorporated into the model at the cost of increased complexity without changing the key insights from our analysis. Similarly, the extraction of rents from different sources of competitive advantages can be justified as economic profits that help recoup fixed costs as in Anshuman and Kalay (1998).

Other assumptions of the model are made to clarify the analysis. For example, we assume that the I market makers face the same (zero) marginal cost. This marginal cost can be changed without loss of generality by reinterpreting a_i as the difference between the offered price and the zero-profit offer. Introducing differences in costs (or expected profitability) complicates the model as it gives the lower cost market maker a range of prices at which the other market makers cannot profitably post offers. This is similar to relabeling I as the lowest cost market makers and setting $K\vartheta$ equal to the zero profit condition of the second lowest cost market makers.[30] As demonstrated by the model in Kandel and Marx (1997), the model can be extended to incorporate additional institutional characteristics and be made more realistic without changing the essential properties of the results.

6.3. Implications for regulation and financial impact on traders

When interpreting our model for regulatory purposes one needs to distinguish between two types of rents. First, the oligopolistic rents extracted by the market makers from their market power (either through speed or informational advantage to capture oligopoly trades), which is represented by the Bertrand–Nash equilibrium offer of the game with a continuous action space. Second, the additional excess rents that can be obtained in the game with a discrete action space or from collusive behavior.

The first source of rents is determined by the level of competition between market makers defined by I, as well as by the structural aggressiveness of price competition parameterised by μ. Above, we discussed how I can be determined by the costs of gaining a competitive advantage in a context of free-entry. Similarly, μ can also be determined as the outcome of investments (as in Biais, Foucault and Moinas (2015) or Foucault, Kozhan and Tham (2017)). Thus, the resulting rents need not be excess profits but could be competitive rents to compensate for fixed costs in technology or information acquisition which could be welfare enhancing. The regulator can influence these rents through interventions that affect the costs of entry by regulating key elements of trading such as fees, data access, clearing, or compliance requirements.

Additional sources of rents are the excess rents from the discreteness of the price grid and those from tacit collusion, and both sources are directly linked to the regulated size of the tick. In our analysis, we combined the two sources and measured them as excess profits. We found that a smaller tick size reduces the rents from both sources, although it comes at the cost of a longer convergence time for the algorithms to reach a rest point. Regulators should account for the resulting trade-off between reducing the two sources of excess rents and the slower convergence rates when determining the optimal tick size as proposed in Cordella and Foucault (1999).

Finally, our analysis presents a challenge for proposals that use the quoted spread as a reference to set the tick size (see for example Li and Ye, 2021, who change the effective tick size by adjusting the nominal price of an asset). This challenge arises from the endogenous relationship between the tick size and the resulting spread, which may be exacerbated by the possibility of tacit collusion between imperfectly competitive market makers. In particular, trying to set a tick size such that the spread has a fixed (and small) number of ticks can create a vicious cycle of ever increasing spreads and tick sizes. Consider an asset that

[29] Schmickler and Tremacoldi-Rossi (2021) find that competitive market makers interact across a substantial number of assets as a way to hedge their inventory positions.

[30] The cost heterogeneity is equivalent to the proposed reinterpretation up to the expected execution probability at $K\vartheta$.

for the current tick size has a spread with multiple ticks, say x, and one wants to increase the tick size so that the spread is fewer ticks wide, say $y < x$. Suppose the tick size is increased proportionately to achieve the desired number of ticks, i.e. increased by a factor of x/y (or the nominal price reduced is reduced by a factor of y/x). Then, with the new and greater tick size, the set of equilibria of the market making game changes, and with it, the possibility that the market making algorithms coordinate on a supracompetitive outcome. This would lead to an observed spread that is wider than anticipated, e.g. it could be $y+1$ (of the now larger) ticks wide. Insisting on a y tick spread would then lead to an even wider tick size until either (i) the tick size reaches an upper bound imposed by the less competitive market makers (a spread equal to $K\vartheta$ in our model), or (ii) the tick is so large that it ensures a very significant discreteness wedge between the observed and the Bertrand–Nash outcome.

6.4. *Learning with states*

The economics literature primarily studies algorithmic collusion by simulating the strategic interaction between competing learning algorithms (see Calvano et al., 2020). We differ from that literature in two key aspects: one, we substitute the analysis of simulations with the analysis of the ODEs that describe the trajectories of the learning algorithms, and two, while Calvano et al. (2020) allows the learning algorithm to condition the current strategy on the joint actions in the previous stage game, our learning algorithms play unconditionally. To link our results with the literature, we extend our analysis to consider strategies that condition on the joint actions in the previous stage game. Characterising the ODEs that describe the dynamics of the conditional strategies is beyond the scope of the current paper, but, as in Calvano et al. (2020), we simulate the interactions between the algorithms and describe the results. We find that the simulated behavior of the algorithms in our setting displays similar qualitative properties as those in Calvano et al. (2020); that is, supracompetitive prices are achieved and sustained through a reward-punishment mechanism.

We simulate the interaction between two Q-learning algorithms as in Calvano et al. (2020). The learning algorithm conditions on the past actions through a common state process. At iteration n, the Q-values for algorithm i represent the discounted payoff of playing action a_i in state s_n, which we denote by $Q^i_{a_i,s_n}(n)$. Following Calvano et al. (2020), we consider perfect monitoring where the state at iteration n is the joint action pair at iteration $n-1$, i.e., $s_n = (a_1(n-1), a_2(n-1)))$. The Q-values are updated according to the following learning rule

$$Q^i_{a_i,s_n}(n+1) = Q^i_{a_i,s_n}(n) + \gamma_n \left[\pi_i(n) + \delta \max_{a'} Q^i_{a',s_{n+1}}(n) - Q^i_{a_i,s_n}(n)\right]$$

for the action chosen at iteration n in state s_n.[31] As before, the Q-values are mapped to the policy using the softmax activation function

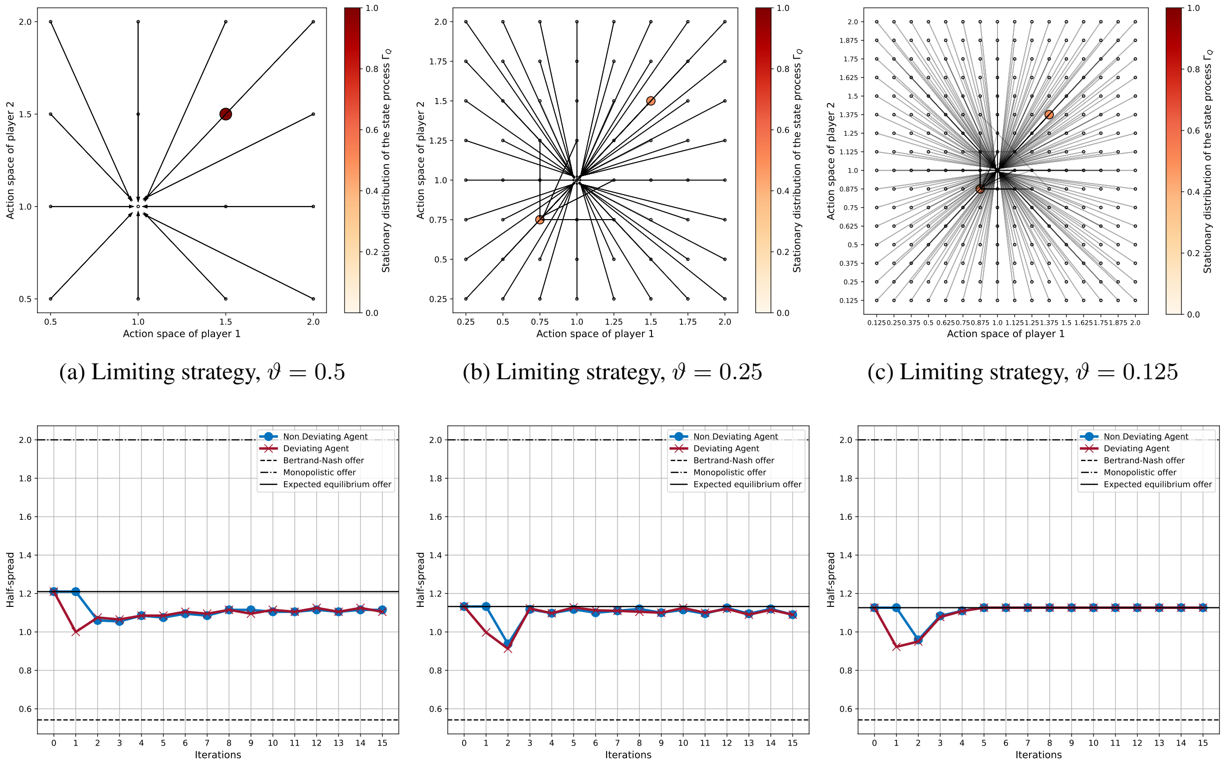
$$x^i_{a_i,s_n}(n) = \frac{e^{\tau\, Q^i_{a_i,s_n}(n)}}{\sum_{a'} e^{\tau\, Q^i_{a',s_n}(n)}}\,.$$

Hence, $x^i_{a_i,s_n}(n)$ denotes the probability of algorithm i playing action a_i at iteration n in state s_n.[32]

We run the algorithms for 100 million iterations which is more than sufficient for the algorithms to converge. The simulation uses a constant learning rate $\gamma = 0.01$, a discount factor $\delta = 0.95$, and the

[31] For consistency with Calvano et al. (2020), we remove the Nature player in these simulations so that the reward received at each iteration is the expected payoff.

[32] Calvano et al. (2020) maps the Q-values to a policy using the ϵ-greedy policy instead of the softmax activation.



(a) Limiting strategy, $\vartheta = 0.5$ (b) Limiting strategy, $\vartheta = 0.25$ (c) Limiting strategy, $\vartheta = 0.125$

(d) Deviation-response, $\vartheta = 0.5$ (e) Deviation-response, $\vartheta = 0.25$ (f) Deviation-response, $\vartheta = 0.125$

Figure 7: Limiting strategy and deviation-response functions for tick size $\vartheta = 0.5, 0.25, 0.125$. For the limiting strategy, the arrows are the largest transition probability from a particular state for the associated one-step transition probability of the states $P_Q(s; s')$ from the limiting Q-values. The circles indicate the various states, and the size and colour represent the stationary distribution of the state process Γ_Q induced by $P_Q(s; s')$. The deviation-response functions depicts the average evolution of the half-spread following a one-period deviation to the static best-response to the rival's pre-deviation price.

exploration-exploitation parameter $\tau = 20$. The latency parameter is $\mu = \log(40)$ and the game is played for tick size $\vartheta = 0.5, 0.25, 0.125$. As in Calvano et al. (2020), the starting Q-values uses the discounted payoff that would accrue if opponents played every action with equal probability. The simulation leads to limiting Q-values for each tick size. This is repeated 50 times to obtain 50 limiting strategies for each tick size, which leads to a total of 150 limiting strategies.

Notice that for the limiting Q-values that are fixed, the state process s_n is Markov with one-step transition probabilities given by

$$\mathbb{P}(s_{n+1} = s' \mid s_n = s) := P_Q(s; s') := x^1_{a'_1,s}\, x^2_{a'_2,s}\,,$$

where $s' = (a'_1, a'_2)$. Furthermore, the state process forms an aperiodic Markov chain with a single recurrent class under P_Q. Therefore, the Markov chain P_Q induces a unique stationary distribution Γ_Q on the states and satisfies the following system of equations

$$\Gamma_\theta(s) = \sum_{s'} \Gamma_\theta(s')\, P_\theta(s'; s)\,, \quad \sum_{s} \Gamma_\theta(s) = 1$$

for all s. The transition probabilities and stationary distribution are useful to visualise the limiting strategy for a particular set of limiting Q-values.

The top panel of Figure 7 plots the limiting strategy for one set of limiting Q-values. The circles indicate each possible state that the algorithms can observe, and the size and colour represent the stationary distribution of the state process Γ_Q induced by $P_Q(s; s')$. The arrows denote the largest transition probability from a particular state. We only plot one limiting strategy, however, the different runs mostly lead to the same qualitative behaviour. From most states, the algorithms coordinate to a particular joint action, such as $s = (1, 1)$ in Figure 7a, before settling on supracompetitive offers or offer cycles that are supracompetitive. Furthermore, almost all the supracompetitive offers are above the most collusive Nash equilibrium of their respective tick size.

The bottom panel of Figure 7 plots the deviation-response functions averaged over the 50 limiting strategies for each tick size. For each limiting strategy, we start from the equilibrium half-spread and we force one algorithm to defect for one-period to the static best-response to the rival's pre-deviation price. From there, we monitor the subsequent evolution of the half-spread.[33] Indeed, we observe the same reward-punishment mechanism as in Calvano et al. (2020). After an initial price war, the algorithms gradually coordinate back towards their pre-deviation behaviour.

Overall, we can replicate the simulated behaviour of the algorithms found in Calvano et al. (2020) with our market making model. We expect that the methods used in our paper extended to state-contingent algorithms will provide solid theoretical foundations for the results obtained from the simulations with state dependent actions.

References

Admati, Anat R, and Paul Pfleiderer. 1988. "A theory of intraday patterns: Volume and price variability." *The Review of Financial Studies*, 1(1): 3–40.

Anshuman, V Ravi, and Avner Kalay. 1998. "Market making with discrete prices." *The Review of Financial Studies*, 11(1): 81–109.

Aquilina, Matteo, Eric Budish, and Peter O'Neill. 2021. "Quantifying the High-Frequency Trading "Arms Race"." *The Quarterly Journal of Economics*, 137(1): 493–564.

Asker, John, Chaim Fershtman, and Ariel Pakes. 2021. "Artificial Intelligence and Pricing: The Impact of Algorithm Design." National Bureau of Economic Research, Inc NBER Working Papers 28535.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. "The nonstochastic multiarmed bandit problem." *SIAM journal on computing*, 32(1): 48–77.

Baruch, Shmuel, and Lawrence R. Glosten. 2019. "Tail expectation and imperfect competition in limit order book markets." *Journal of Economic Theory*, 183: 661–697.

Beggs, A.W. 2005. "On the convergence of reinforcement learning." *Journal of Economic Theory*, 122(1): 1–36.

Benaïm, Michel. 1999. "Dynamics of stochastic approximation algorithms." 1–68. Berlin, Heidelberg:Springer Berlin Heidelberg.

Benveniste, Albert, Michel Métivier, and Pierre Priouret. 1990. *Adaptive Algorithms and Stochastic Approximations.* Heidelberg:Springer Berlin.

Bertrand, J. 1993. "Review of "Theorie mathematique de la richesse sociale" and "Recherche sur les principes mathematiques de la theorie des richesses"." *Journal des Savants*, 499–508.

Biais, Bruno, Christophe Bisière, and Chester Spatt. 2010. "Imperfect competition in financial markets: An empirical study of Island and Nasdaq." *Management Science*, 56(12): 2237–2250.

Biais, Bruno, Thierry Foucault, and Sophie Moinas. 2015. "Equilibrium fast trading." *Journal of Financial Economics*, 116(2): 292 – 313.

Borkar, Vivek. 2008. *Stochastic approximation: A dynamical systems viewpoint.* Hindustan Book Agency.

Brown, Zach Y, and Alexander MacKay. 2021. "Competition in Pricing Algorithms." National Bureau of Economic Research Working Paper 28860.

Börgers, Tilman, and Rajiv Sarin. 1997. "Learning Through Reinforcement and Replicator Dynamics." *Journal of Economic Theory*, 77(1): 1–14.

[33] Following Calvano et al. (2020), when the equilibrium behaviour is a cycle, we consider deviations starting from every point of the cycle and take the average of all of them.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello. 2020. “Artificial intelligence, algorithmic pricing, and collusion.” *American Economic Review*, 110(10): 3267–97.

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicoló, and Sergio Pastorello. 2021. “Algorithmic collusion with imperfect monitoring.” *International Journal of Industrial Organization*, 79: 102712.

Cartea, Álvaro, and José Penalva. 2012. “Where is the Value in High Frequency Trading?” *Quarterly Journal of Finance*, 2(3): 1–46.

Cartea, Álvaro, and Leandro Sánchez-Betancourt. 2021. “Optimal Execution with Stochastic Delay.” *SSRN: 3812324*.

Cartea, Álvaro, Patrick Chang, José Penalva, and Harrison Waldon. 2022*a*. “The Algorithmic Learning Equations: Evolving Strategies in Dynamic Games.” *Available at SSRN 4175239*.

Cartea, Álvaro, Patrick Chang, Mateusz Mroczka, and Roel Oomen. 2022*b*. “AI driven liquidity provision in OTC markets.” *SSRN: 4111152*.

Christie, William G, and Paul H Schultz. 1994. “Why do NASDAQ market makers avoid odd-eighth quotes?” *The Journal of Finance*, 49(5): 1813–1840.

Chung, Kee H, Albert J Lee, and Dominik Rösch. 2020. “Tick size, liquidity for small and large orders, and price informativeness: Evidence from the Tick Size Pilot Program.” *Journal of Financial Economics*, 136(3): 879–899.

Cordella, Tito, and Thierry Foucault. 1999. “Minimum Price Variations, Time Priority, and Quote Dynamics.” *Journal of Financial Intermediation*, 8(3): 141–173.

Cross, John G. 1973. “A Stochastic Learning Model of Economic Behavior.” *The Quarterly Journal of Economics*, 87(2): 239–266.

Diamond, Douglas W, and Robert E Verrecchia. 1981. “Information aggregation in a noisy rational expectations economy.” *Journal of Financial Economics*, 9(3): 221–235.

Dorner, Florian E. 2021. “Algorithmic collusion: A critical review.”

Duffy, John, and Ed Hopkins. 2005. “Learning, information, and sorting in market entry games: theory and evidence.” *Games and Economic Behavior*, 51(1): 31–62.

Erev, Ido, and Alvin E. Roth. 1998. “Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria.” *The American Economic Review*, 88(4): 848–881.

Foucault, Thierry, Roman Kozhan, and Wing Wah Tham. 2017. “Toxic arbitrage.” *The Review of Financial Studies*, 30(4): 1053–1094.

Glosten, Lawrence R, and Paul R Milgrom. 1985. “Bid, ask and transaction prices in a specialist market with heterogeneously informed traders.” *Journal of Financial Economics*, 14(1): 71–100.

Gomes, Eduardo Rodrigues, and Ryszard Kowalczyk. 2009. “Dynamic Analysis of Multiagent Q-Learning with ϵ-Greedy Exploration.” *ICML ’09*, 369–376. New York, NY, USA:Association for Computing Machinery.

Gottlieb, Gary, and Avner Kalay. 1985. “Implications of the discreteness of observed stock prices.” *The Journal of Finance*, 40(1): 135–153.

Green, Edward J., Robert C. Marshall, and Leslie M. Marx. 2014. “Tacit Collusion in Oligopoly.” In *The Oxford Handbook of International Antitrust Economics, Volume 2.* , ed. Roger D. Blair and D. Daniel Sokol, 464–497. Oxford University Press.

Griffith, Todd G., and Brian S. Roseman. 2019. “Making cents of tick sizes: The effect of the 2016 U.S. SEC tick size pilot on limit order book liquidity.” *Journal of Banking & Finance*, 101: 104–121.

Grossman, Sanford J, and Joseph E Stiglitz. 1980. “On the impossibility of informationally efficient markets.” *The American Economic Review*, 70(3): 393–408.

Hansen, Karsten T, Kanishka Misra, and Mallesh M Pai. 2021. “Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms.” *Marketing Science*, 40(1): 1–12.

Harrington, Joseph E. 2018. “Developing Competition Law for Collusion by Autonomous Artificial Agents.” *Journal of Competition Law & Economics*, 14(3): 331 – 363.

Hart, Sergiu, and Andreu Mas-Colell. 2003. “Uncoupled Dynamics Do Not Lead to Nash Equilibrium.” *The American Economic Review*, 93(5): 1830–1836.

Hofbauer, Josef. 2011. “Deterministic evolutionary game dynamics.” *Proceedings of Symposia in Applied Mathematics Volume*, 69.

Hofbauer, Josef, and Karl Sigmund. 1998. *Evolutionary Games and Population Dynamics.* Cambridge University Press.

Hoffmann, Peter. 2014. “A dynamic limit order market with fast and slow traders.” *Journal of Financial Economics*, 113(1): 156 – 169.

Hopkins, Ed, and Martin Posch. 2005. “Attainability of boundary points under reinforcement learning.” *Games and Economic Behavior*, 53(1): 110–125.

Kadan, Ohad. 2006. “So who gains from a small tick size?” *Journal of Financial Intermediation*, 15(1): 32–66.

Kaisers, Michael, and Karl Tuyls. 2010. “Frequency Adjusted Multi-Agent Q-Learning.” *AAMAS ’10*, 309–316. Richland, SC:International Foundation for Autonomous Agents and Multiagent Systems.

Kandel, Eugene, and Leslie M Marx. 1997. "Nasdaq market structure and spread patterns." *Journal of Financial Economics*, 45(1): 61–89.

Kasbekar, Gaurav, and Alexandre Proutiere. 2010. "Opportunistic medium access in multi-channel wireless systems: A learning approach." 1288–1294.

Klein, Timo. 2021. "Autonomous algorithmic collusion: Q-learning under sequential pricing." *The RAND Journal of Economics*, 52(3): 538–558.

Kyle, Albert S. 1985. "Continuous auctions and insider trading." *Econometrica: Journal of the Econometric Society*, 1315–1335.

Li, Sida, and Mao Ye. 2021. "The Optimal Price of a Stock: A Tale of Two Discretenesses." *SSRN: 3763516*.

Loertscher, Simon. 2008. "Market making oligopoly." *The Journal of Industrial Economics*, 56(2): 263–289.

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

Norman, M. Frank. 1972. *Markov Processes and Learning Models. Mathematics in Science and Engineering: a series of monographs and textbooks*, Academic Press.

Pemantle, Robin. 1990. "Nonconvergence to Unstable Points in Urn Models and Stochastic Approximations." *The Annals of Probability*, 18(2): 698–712.

Penalva, José, and Mikel Tapia. 2021. "Heterogeneity and Competition in Fragmented Markets: Fees Vs Speed." *Applied Mathematical Finance*, 28(2): 143–177.

Sato, Yuzuru, and James P. Crutchfield. 2003. "Coupled replicator equations for the dynamics of learning in multiagent systems." *Phys. Rev. E*, 67: 015206.

Schmickler, Simon N.M., and Pedro Tremacoldi-Rossi. 2021. "In Good Times and in Bad: High-Frequency Market Making Design, Liquidity, and Asset Prices."

Spooner, Thomas, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. "Market Making via Reinforcement Learning." *AAMAS '18*, 434–442. Richland, SC:International Foundation for Autonomous Agents and Multiagent Systems.

Spulber, Daniel F. 1995. "Bertrand competition when rivals' costs are unknown." *The Journal of Industrial Economics*, 1–11.

Tuyls, Karl, Katja Verbeeck, and Tom Lenaerts. 2003. "A Selection-Mutation Model for Q-Learning in Multi-Agent Systems." *AAMAS '03*, 693–700. New York, NY, USA:Association for Computing Machinery.

Verousis, Thanos, Pietro Perotti, and Georgios Sermpinis. 2017. "One size fits all? High frequency trading, tick size changes and the implications for exchanges: market quality and market structure considerations." *Review of Quantitative Finance and Accounting*, 1–40.

Vives, Xavier. 2011. "Strategic supply function competition with private information." *Econometrica*, 79(6): 1919–1966.

Yao, Chen, and Mao Ye. 2018. "Why trading speed matters: A tale of queue rationing under price controls." *The Review of Financial Studies*, 31(6): 2157–2183.

Appendix A. Stochastic approximation for independent learning algorithms

This appendix provides the formal definitions and technical conditions required to approximate the discrete-time stochastic processes from the algorithms with a system of deterministic ODEs. We show how one computes the appropriate dynamics that describe the trajectories of the stochastic processes from the algorithms and we verify the required conditions so that the trajectories of the stochastic processes follow the trajectories from the system of deterministic ODEs.

Appendix A.1. Framework and methodological approach

Recall that $\{\boldsymbol{\pi}(n)\}_{n\in\mathbb{N}}$ denotes the sequence of random reward vectors $\boldsymbol{\pi}(n) \in E$ for the I players with a bounded support for all n, and $\{\boldsymbol{x}(n)\}_{n\in\mathbb{N}}$ denotes the sequence of discrete-time stochastic processes generated by the learning algorithms. The increments of the stochastic processes are given by

$$\boldsymbol{x}(n+1) - \boldsymbol{x}(n) = \gamma_n \, f\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) \, , \tag{A.1}$$

where the learning rate $\gamma_n > 0$ satisfies (2) and $f : \mathbb{R}^{I\times K} \times E \to \mathbb{R}^{I\times K}$ is the stochastic update rule of the learning algorithms. For many algorithms, it is not straightforward to compute F because f is a non-linear function of the reward. To address this, we write the increments in (A.1) as

$$\boldsymbol{x}(n+1) - \boldsymbol{x}(n) = \gamma_n \, \tilde{f}\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) + \boldsymbol{O}(\gamma_n^2) \, , \tag{A.2}$$

where the linear function of the reward $\tilde{f} : \mathbb{R}^{I\times K} \times E \to \mathbb{R}^{I\times K}$ is an approximation to the update rule f, and $\boldsymbol{O}(\gamma_n^2)$ are approximation errors from writing f as $\tilde{f}$.

The goal is to rewrite the stochastic update rule in (A.1) so that the increments of the stochastic processes are given by

$$\boldsymbol{x}(n+1) - \boldsymbol{x}(n) = \gamma_n \left(F\left(\boldsymbol{x}(n)\right) + \boldsymbol{U}(n) + \boldsymbol{O}(\gamma_n)\right) . \tag{A.3}$$

The deviations and approximation errors in (A.3) depend on whether f or $\tilde{f}$ is used to compute the vector field F. Specifically, if f is used, then we have

$$F\left(\boldsymbol{x}\right) = \int f\left(\boldsymbol{x}, \boldsymbol{\pi}\right)\, \mu_\pi(d\boldsymbol{\pi})\,, \quad \boldsymbol{U}(n) = f\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) - F\left(\boldsymbol{x}(n)\right)\,, \quad \text{and} \quad \boldsymbol{O}(\gamma_n) \equiv \boldsymbol{0}\,. \tag{A.4}$$

On the other hand, if $\tilde{f}$ is used, then we have

$$F\left(\boldsymbol{x}\right) = \int \tilde{f}\left(\boldsymbol{x}, \boldsymbol{\pi}\right)\, \mu_\pi(d\boldsymbol{\pi})\,, \quad \boldsymbol{U}(n) = \tilde{f}\left(\boldsymbol{x}(n), \boldsymbol{\pi}(n)\right) - F\left(\boldsymbol{x}(n)\right)\,, \tag{A.5}$$

and appropriate approximation errors $\boldsymbol{O}(\gamma_n)$.

To formalise the stochastic approximation, we first provide the definition of an asymptotic pseudotrajectory to describe the convergence of the trajectories from the stochastic processes to those of the system of ODEs. Roughly speaking, the stochastic processes are an asymptotic pseudotrajectory of the system of ODEs if the trajectories of the continuous-time limit of the stochastic processes follow the trajectories of the ODEs. Before defining an asymptotic pseudotrajectory we require the notion of semiflow, which we provide below.

Definition 1 *A semiflow Φ on a metric space (M, d) is a continuous map*

$$\Phi : \mathbb{R}^+ \times M \to M\,, \quad (t, x) \mapsto \Phi(t, x) = \Phi_t(x)\,,$$

such that

$$\Phi_0 = \text{Identity}\,, \quad \Phi_{t+s} = \Phi_t \circ \Phi_s\,,$$

for all $(t, s) \in \mathbb{R}^+ \times \mathbb{R}^+$ and the operator $\circ$ is the composition function.

To compare a discrete-time stochastic process to a semiflow, we interpolate the discrete-time stochastic process to construct a continuous-time affine interpolated process $\boldsymbol{X}$. Specifically, let $t_n = \sum_{i=1}^n \gamma_i$ and write the entries of $\boldsymbol{X}$ as

$$X_{ik}\left(t_n + s\right) = x_{ik}(n) + s\,\frac{x_{ik}(n+1) - x_{ik}(n)}{t_{n+1} - t_n}\,,$$

for all $n \in \mathbb{N}$ and $0 \leq s < \gamma_{n+1}$. In this way, $\boldsymbol{X}$ is continuous and describes the trajectories of the stochastic processes from the learning algorithms.

Definition 2 *A continuous function $\boldsymbol{X} : \mathbb{R}^+ \to M$ is an asymptotic pseudotrajectory for the semiflow Φ if*

$$\lim_{t\to\infty} \sup_{t\leq h\leq T} \left|\boldsymbol{X}(t+h) - \Phi_h\left(\boldsymbol{X}\left(t\right)\right)\right| = 0$$

for any $T > 0$ and $|\cdot|$ denotes the euclidean norm.

To show that the continuous-time affine interpolated processes $\boldsymbol{X}$ is an asymptotic pseudotrajectory of the flow induced by the continuous globally integrable vector field F with probability one, it is sufficient to verify that the following conditions hold:

$$\sup_n \mathbb{E}\left(|\boldsymbol{U}(n)|^q\right) < \infty\,, \tag{C1}$$

$$\sup_n |\boldsymbol{x}(n)| < \infty\,, \tag{C2}$$

$$\lim_{n\to\infty} \boldsymbol{O}(\gamma_n) = 0\,, \quad \text{a.s.} \tag{C3}$$

for the same q required in (2).[34] See Propositions 4.1, 4.2, and Remark 4.5 in Benaïm (1999).

Appendix A.2. Q-learning and variations

Recall that Q-learning updates the Q-values based on the learning rate γ, on the reward received $\pi_i(n)$, and the discount factor δ. The Q-values are then mapped to the policy $\boldsymbol{x}_i(n) = (x_{i1}(n), \ldots, x_{iK}(n))$ which describes the probability that player i chooses action k from the set of K possible actions at each iteration n. The mapping uses a softmax activation function given by $x_{ik}(n) = e^{\tau\, Q_{ik}(n)} / \sum_\ell e^{\tau\, Q_{i\ell}(n)}$.

First, we present the results for frequency adjusted Q-learning to illustrate why the replicator-mutation dynamics do not describe the dynamics of the policies from Q-learning. We then present the results for the dynamics of Q-learning in terms of Q-values, and finally, we present the results for synchronous Q-learning which is new in the literature.

Frequency adjusted Q-learning

Our approach to derive the ODEs, which is different from that in Tuyls, Verbeeck and Lenaerts (2003), Sato and Crutchfield (2003) and Kaisers and Tuyls (2010) who take derivatives to obtain the dynamics, allows us to demonstrate that the scaling factor $1/x_{ik}(n)$ is essential to recover the replicator-mutation dynamics.

Without loss of generality, we drop the minimum operator and focus only on the case when the Q-values for FAQ-learning evolve according to

$$Q_{ik}(n+1) = \begin{cases} Q_{ik}(n) + \frac{\gamma_n}{x_{ik}(n)}\left[\pi_i(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ Q_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n)\,. \end{cases}$$

Below, we introduce Lemma 1 to write the update rule of the policy as a linear function of the reward. The approximation simplifies the computation of the deterministic vector field F in Proposition 7 and reduces the dynamics of the policies to the replicator-mutation dynamics in (ODE 1). Finally, in Proposition 8, we show that the trajectories of the policies for I independent players each using the FAQ-learning algorithm are approximated with the trajectories from the replicator-mutation dynamics.

Lemma 1 *When γ_n is sufficiently small, the increment of the policy for FAQ-learning is given by*

$$\begin{aligned} \Delta x_{ik}(n) &= x_{ik}(n+1) - x_{ik}(n) \\ &= \begin{cases} \gamma_n\, \tau\, \big(1 - x_{ik}(n)\big)\big[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\big] + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -\gamma_n\, \tau\, x_{ik}(n)\big[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\big] + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,, \end{cases} \end{aligned} \tag{A.6}$$

which is in the form of (A.2) and the reward from playing a_{ij} is $\pi_{ij}(n) = \pi_i(a_i(n) = a_{ij}, a_{-i}(n), a_\nu)$.

[34] The vector field is continuous if its component functions are continuous, and globally integrable if it has unique trajectories.

Proof Use

$$\frac{x_{ik}(n+1)}{x_{ik}(n)} = \frac{e^{\tau\, Q_{ik}(n+1)}}{\sum_\ell e^{\tau\, Q_{i\ell}(n+1)}} \frac{\sum_\ell e^{\tau\, Q_{i\ell}(n)}}{e^{\tau\, Q_{ik}(n)}} = \frac{e^{\tau\, \Delta Q_{ik}(n)}}{\sum_\ell x_{i\ell}(n)\, e^{\tau\, \Delta Q_{i\ell}(n)}}\,,$$

to write

$$x_{ik}(n+1) = x_{ik}(n)\, \frac{e^{\tau\, \Delta Q_{ik}(n)}}{\sum_\ell x_{i\ell}(n)\, e^{\tau\, \Delta Q_{i\ell}(n)}}\,,$$

where

$$\Delta Q_{ij}(n) = Q_{ij}(n+1) - Q_{ij}(n) = \begin{cases} \frac{\gamma_n}{x_{ij}(n)} \left[\pi_{ij}(n) + \delta \max \boldsymbol{Q}_i - Q_{ij}(n)\right] & \text{for } j \text{ s.t. } a_{ij} = a_i(n)\,, \\ 0 & \text{for } j \text{ s.t. } a_{ij} \neq a_i(n)\,. \end{cases}$$

Next, consider the case when $a_{ik} = a_i(n)$. Write the increment of the policy as

$$x_{ik}(n+1) - x_{ik}(n) = x_{ik}(n) \left(\frac{e^{\tau\, \Delta Q_{ik}(n)} - \sum_\ell x_{i\ell}(n)\, e^{\tau\, \Delta Q_{i\ell}(n)}}{\sum_\ell x_{i\ell}(n)\, e^{\tau\, \Delta Q_{i\ell}(n)}} \right)\,,$$

and let $B^{\gamma_n}_{j,n} = \exp\left(\tau\, \Delta Q_{ij}(n)\right)$ to write

$$\begin{aligned} \Delta x_{ik}(n) &= x_{ik}(n) \left(\frac{B^{\gamma_n}_{k,n} - \sum_\ell x_{i\ell}(n) - x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)}{\sum_\ell x_{i\ell}(n) + x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)} \right) \\ &= x_{ik}(n) \left(\frac{\left(B^{\gamma_n}_{k,n} - 1\right) - x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)}{1 + x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)} \right) \\ &= \frac{x_{ik}(n)\left(1 - x_{ik}(n)\right)\, \left(B^{\gamma_n}_{k,n} - 1\right)}{1 + x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)}\,. \end{aligned}$$

Geometric expand the denominator to obtain

$$\begin{aligned} \Delta x_{ik}(n) &= x_{ik}(n)\left(1 - x_{ik}(n)\right)\, \left(B^{\gamma_n}_{k,n} - 1\right)\, \left[1 - \sum_{m=1}^{\infty} \left(x_{ik}(n)\, \left(B^{\gamma_n}_{k,n} - 1\right)\right)^m\right] \\ &= x_{ik}(n)\left(1 - x_{ik}(n)\right)\, \left(B^{\gamma_n}_{k,n} - 1\right) + O(\gamma_n^2)\,. \end{aligned}$$

Expand the exponential function with the power series $B^{\gamma_n}_{k,n} = 1 + \frac{\tau\,\gamma_n}{x_{ik}(n)}\left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] + O(\gamma_n^2)$ to obtain

$$\begin{aligned} \Delta x_{ik}(n) &= x_{ik}(n)\left(1 - x_{ik}(n)\right) \frac{\tau\,\gamma_n}{x_{ik}(n)}\left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] + O(\gamma_n^2) \\ &= \left(1 - x_{ik}(n)\right) \tau\,\gamma_n \left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] + O(\gamma_n^2)\,. \end{aligned}$$

Similarly, consider the case when $a_{ik} \neq a_i(n) = a_{i\ell}$, let h be an additional indexing variable to write

$$\begin{aligned}
x_{ik}(n+1) - x_{ik}(n) &= x_{ik}(n) \left(\frac{e^{\tau\, \Delta Q_{ik}(n)} - \sum_h x_{ih}(n)\, e^{\tau\, \Delta Q_{ih}(n)}}{\sum_h x_{ih}(n)\, e^{\tau\, \Delta Q_{ih}(n)}} \right) \\
&= x_{ik}(n) \left(\frac{1 - \sum_h x_{ih}(n) - x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right)}{\sum_h x_{ih}(n) + x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right)} \right) \\
&= -\frac{x_{ik}(n)\, x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right)}{1 + x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right)} \\
&= -x_{ik}(n)\, x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right) \left[1 + \sum_{m=1}^{\infty} \left(x_{i\ell}(n)\, \left(B_{\ell,n}^{\gamma_n} - 1\right)\right)^m \right] \\
&= -x_{ik}(n)\, x_{i\ell}(n)\, \frac{\tau\, \gamma_n}{x_{i\ell}(n)} \left[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\right] + O(\gamma_n^2) \\
&= -x_{ik}(n)\, \tau\, \gamma_n \left[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\right] + O(\gamma_n^2)\,.
\end{aligned}$$

□

Proposition 7 *The component functions of the deterministic vector field F for the policies of I independent learning algorithms who use FAQ-learning are given by*

$$F_{ik}\left(\boldsymbol{x}\right) = \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] + x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln \left(\frac{x_{i\ell}(n)}{x_{ik}(n)} \right), \qquad \text{(A.7)}$$

where the expected payoff of algorithm i playing action a_{ij} is given by

$$\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}} \Big[\mathbb{E}_{\nu} \left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i} \right] \Big]\,.$$

Proof Apply Lemma 1 to obtain $\tilde{f}$, which is the approximation to the update rule in the form of (A.2). To compute F, take the expectation of (A.6) with respect to the rewards that can be received. We do this in three steps and neglect terms of order $O(\gamma_n^2)$: take the expectation of Nature's action conditional on the actions of all the other players; take the expectation over the actions of the opponents conditional on the action of algorithm i; and take expectations over the actions of algorithm i.

Step 1. The conditional expected payoff of algorithm i that plays action $a_i(n) = a_{ij}$ and the remaining algorithms play a_{-i} is given by $\Pi_{ij}(a_{-i}, n) := \mathbb{E}_\nu\left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i}\right]$. Next, write the expected increment of the policies conditional on actions a_i and a_{-i} as

$$\mathbb{E}\Big[\Delta x_{ik}(n) \,|\, a_i(n)\,,\, a_{-i}(n)\Big] = \begin{cases} \gamma_n\, \tau\, \left(1 - x_{ik}(n)\right) \left[\Pi_{ik}(a_{-i}, n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] \text{ for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -\gamma_n\, \tau\, x_{ik}(n) \left[\Pi_{i\ell}(a_{-i}, n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\right] \text{ for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \end{cases}$$

Step 2. The conditional expected payoff for algorithm i that plays action $a_i(n) = a_{ij}$ depends on the actions of the opponents, and the actions of the opponents depend on their policy which varies with n. Therefore, take the expectation over the actions of the opponents to obtain the conditional expected payoff

$\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}}\big[\Pi_{ij}(a_{-i}, n)\big]$ for algorithm i playing action $a_i(n) = a_{ij}$. Thus, the expected increment of the policies conditional on the action of algorithm i is

$$\mathbb{E}\Big[\Delta x_{ik}(n) \,|\, a_i(n)\Big] = \begin{cases} \gamma_n\, \tau\, \big(1 - x_{ik}(n)\big) \Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -\gamma_n\, \tau\, x_{ik}(n) \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big] & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \end{cases} \tag{A.8}$$

Step 3. Take the expectation over the action of algorithm i and use the two cases in (A.8) to write the expected increment of the policies:

$$\begin{aligned} \mathbb{E}\Big[\Delta x_{ik}(n)\Big] &= \gamma_n\, \tau\, x_{ik}(n)\big(1 - x_{ik}(n)\big)\Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] \\ &\quad - \gamma_n\, \tau \sum_{\ell \neq k} x_{i\ell}(n)\, x_{ik}(n)\, \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big] \\ &= \gamma_n\, \tau\, x_{ik}(n) \Bigg[\big(1 - x_{ik}(n)\big)\Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] \\ &\qquad - \sum_{\ell \neq k} x_{i\ell}(n) \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big]\Bigg] \\ &= \gamma_n\, \tau\, x_{ik}(n) \left[\Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] - \sum_{\ell} x_{i\ell}(n) \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big]\right] \\ &= \gamma_n\, \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n) \overline{\Pi}_{i\ell}(n)\right] + \gamma_n\, \tau\, x_{ik}(n) \left[\delta \max \boldsymbol{Q}_i - \sum_{\ell} x_{i\ell}(n)\, \delta \max \boldsymbol{Q}_i\right] \\ &\quad + \gamma_n\, \tau\, x_{ik}(n) \left[\sum_{\ell} x_{i\ell}(n)\, Q_{i\ell}(n) - Q_{ik}(n)\right]. \end{aligned} \tag{A.9}$$

The second term on the right-hand of (A.9) is zero because the quantity $\delta \max \boldsymbol{Q}_i$ is a constant for fixed n, so the term in brackets sums to zero. Next, to simplify the last term of (A.9), use $Q_{ik}(n) = \sum_{\ell} x_{i\ell}(n)\, Q_{ik}(n)$ to take $\sum_{\ell} x_{i\ell}(n)$ as a common factor. Then $\tau\,\big(Q_{i\ell}(n) - Q_{ik}(n)\big) = \ln\left(x_{i\ell}(n)/x_{ik}(n)\right)$, because $e^{\tau\, Q_{i\ell}(n)}/e^{\tau\, Q_{ik}(n)} = x_{i\ell}(n)/x_{ik}(n)$, so we have that

$$\tau \sum_{\ell} x_{i\ell}(n)\big(Q_{i\ell}(n) - Q_{ik}(n)\big) = \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)}\right).$$

Hence,

$$\mathbb{E}\left[\Delta x_{ik}(n)\right] = \gamma_n\, \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right] + \gamma_n\, x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)}\right)$$

for each algorithm-action pair ik. Therefore, (A.7) describes the component functions of $F\left(\boldsymbol{x}\right)$ given by (A.5). □

We see that the continuous globally integrable vector field F in (A.7) is the replicator-mutation dynamics. Moreover, we see that the scaling factor $1/x_{ik}(n)$ plays an important role so that we end up with the correct degree of $x_{ik}(n)$ in (A.7) after taking the expectation over of algorithm i's own actions. Finally, it remains to verify that conditions (C1)–(C3) are satisfied in terms of the policies to show that the trajectories of the policies follow the trajectories from the replicator-mutation dynamics.

Proposition 8 *Assume that the value of the exploration-exploitation parameter τ is finite and that the Q-values remain in the convex hull of experienced rewards. Let F be the continuous globally integrable vector field $F : [0,1]^{I\times K} \to [0,1]^{I\times K}$ with component functions*

$$F_{ik}(\boldsymbol{x}) = \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] + x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)} \right) .$$

Set $t_n = \sum_{i=1}^{n} \gamma_i$ and let $\boldsymbol{X}$ be the continuous-time affine interpolated processes of policies for I independent learning players who use FAQ-learning with entries

$$X_{ik}(t_n + s) = x_{ik}(n) + s\, \frac{x_{ik}(n+1) - x_{ik}(n)}{t_{n+1} - t_n}$$

for all $n \in \mathbb{N}$ and $0 \leq s < \gamma_{n+1}$. Then, $\boldsymbol{X}$ is an asymptotic pseudotrajectory of the flow induced by F with probability one.

Proof Condition (C2) is trivially satisfied given that $\boldsymbol{x}(n) \in [0,1]^{I\times K}$ for all $n \in \mathbb{N}$. For condition (C1), notice that the entries of the deviation term in (A.5) are given by

$$U_{ik}(n) = \begin{cases} \Bigg(\tau\left(1 - x_{ik}(n)\right)\left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] \\ \quad - \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ \quad - x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)} \right) \Bigg) \\ \Bigg(- \tau\, x_{ik}(n) \left[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\right] \\ \quad - \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \\ \quad - x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)} \right) \Bigg) \end{cases}$$

Now, the Q-values and τ are finite by assumption, therefore there exists a lower bound $0 < x_m \leq x_{ij}(n)$ for all i, j, n such that the logarithm of $x_{i\ell}(n)/x_{ik}(n)$ remains finite. Hence, (C1) is satisfied because $\pi_{ij}(n)$ is bounded for all i, j, n by assumption.

Finally for (C3), we verify that the approximation errors converge to zero. Consider k such that $a_{ik} = a_i(n)$, then the approximation errors from the geometric expansion are

$$O(\gamma_n) = -\frac{1}{\gamma_n}\, x_{ik}(n)\, \left(1 - x_{ik}(n)\right) \left(B_{k,n}^{\gamma_n} - 1\right) \sum_{m=1}^{\infty} \left(x_{ik}(n) \left(B_{k,n}^{\gamma_n} - 1\right)\right)^{m} ,$$

where $B_{k,n}^{\gamma_n} = \exp\big(\tau \, \Delta Q_{ik}(n)\big)$ and the approximation errors from the power series expansion are

$$O(\gamma_n) = \frac{1}{\gamma_n} \, x_{ik}(n) \, \big(1 - x_{ik}(n)\big) \sum_{m=2}^{\infty} \frac{1}{m!} \left(\frac{\tau \, \gamma_n}{x_{ik}(n)} \big[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\big] \right)^m .$$

By assumption, there exists a lower bound x_m, so $\gamma_n \ll x_m$ as $\gamma_n \to 0$ because the Q-values and τ are finite. Therefore, for small enough value of γ_n, $|B_{k,n}^{\gamma_n} - 1| < 1$ so that the infinite sum from the geometric expansion is finite and equal to zero when multiplied by $\left(B_{k,n}^{\gamma_n} - 1\right)/\gamma_n$ because $\left(B_{k,n}^{\gamma_n} - 1\right)/\gamma_n$ converges to zero as $\gamma_n \to 0$ by L'Hôpital's rule. Rewrite the approximation errors from the power series expansion as

$$\begin{aligned} O(\gamma_n) &= \frac{1}{\gamma_n} \, x_{ik}(n) \, \big(1 - x_{ik}(n)\big) \sum_{m=2}^{\infty} \frac{1}{m!} \left(\frac{\tau \, \gamma_n}{x_{ik}(n)} \big[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\big] \right)^m \\ &= \frac{1}{\gamma_n} \, x_{ik}(n) \, \big(1 - x_{ik}(n)\big) \left(B_{k,n}^{\gamma_n} - 1 - \frac{\tau \, \gamma_n}{x_{ik}(n)} \big[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\big] \right) , \end{aligned}$$

which converges to zero as $\gamma_n \to 0$ by L'Hôpital's rule. Therefore, the approximation errors from both the geometric and power series expansions converge to zero as $\gamma_n \to 0$. We follow the same logic for k such that $a_{ik} \neq a_i(n) = a_{i\ell}$. Thus, (C3) is satisfied. □

Q-learning

Previously, in frequency adjusted Q-learning, we saw that the scaling factor $1/x_{ik}(n)$ is crucial to recover the replicator-mutation dynamics. Q-learning does not have the scaling factor, so the dynamics of Q-learning in terms of the policies have extra dependencies on Q-values. Hence, for Q-learning, we approximate the dynamics of Q-values instead of approximating both the policies and the Q-values.

It is straightforward to compute the deterministic vector field F because the update rule for Q-values is a linear function of the rewards. The expectations depend on the probability of the algorithms performing certain actions, which is immediately obtained from the Q-values.

Proposition 9 *The components of the deterministic vector field F for the Q-values of I independent learning algorithms who use Q-learning is*

$$F_{ik}\left(\boldsymbol{Q}\right) = x_{ik}(n) \, \left[\overline{\Pi}_{ik}(n) + \delta \, \max \boldsymbol{Q}_i - Q_{ik}(n)\right] , \tag{A.10}$$

where the probability that player i takes action k, i.e., action a_{ik}, is given by

$$x_{ik}(n) = \frac{e^{\tau \, Q_{ik}(n)}}{\sum_\ell e^{\tau \, Q_{i\ell}(n)}} ,$$

and the expected payoff of algorithm i playing action a_{ij} is given by

$$\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}}\Big[\mathbb{E}_\nu\big[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i}\big]\Big] .$$

Proof Rearrange the evolution of the Q-values in (5) into the form of (A.1) and write the increment of the Q-values as

$$\Delta Q_{ik}(n) = Q_{ik}(n+1) - Q_{ik}(n) = \begin{cases} \gamma_n \, \left[\, \pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n) \,\right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n), \\ 0 & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n), \end{cases} \tag{A.11}$$

to obtain the update rule f. Next, we take the expectation of (A.11) with respect to the rewards to compute F in three steps: take the expectation of the Nature player's action conditional on the actions of all others; take the expectation over the actions of the opponents conditional on the action of algorithm i; and take expectations over the actions of algorithm i to obtain

$$\mathbb{E}\Big[\Delta Q_{ik}(n)\Big] = \gamma_n\, x_{ik}(n)\,\Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big]\,,$$

for each algorithm-action pair ik. Therefore, (A.10) describes the components of the continuous globally integrable vector field F in (A.4). □

We see that the continuous globally integrable vector field F in (A.10) is in terms of Q-values. Thus, conditions (C1)–(C3) are also verified in terms of Q-values.

Proposition 10 *Assume that the Q-values remain in the convex hull of experienced rewards. Let F be the continuous globally integrable vector field $F : \mathbb{R}^{I\times K} \to \mathbb{R}^{I\times K}$ with component functions*

$$F_{ik}\left(\boldsymbol{Q}\right) = x_{ik}(n)\left[\overline{\Pi}_{ik}(n) + \delta\, \max \boldsymbol{Q}_i - Q_{ik}(n)\right].$$

Set $t_n = \sum_{i=1}^{n} \gamma_i$ and let $\boldsymbol{Q}$ be the continuous-time affine interpolated processes of Q-values for I independent learning algorithms who use Q-learning with entries

$$Q_{ik}\left(t_n + s\right) = Q_{ik}(n) + s\,\frac{Q_{ik}(n+1) - Q_{ik}(n)}{t_{n+1} - t_n}$$

for all $n \in \mathbb{N}$ and $0 \le s < \gamma + n + 1$. Then, $\boldsymbol{Q}$ is an asymptotic pseudotrajectory of the flow induced by F with probability one.

Proof By assumption, $\boldsymbol{Q}(n)$ is finite for all $n \in \mathbb{N}$, therefore condition (C2) is satisfied. Now, the entries of the deviations are given by

$$U_{ik}(n) = \begin{cases} \big[\,\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\,\big] - x_{ik}(n)\ \left[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -x_{ik}(n)\ \left[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \end{cases}$$

Now, (C1) is satisfied because $\pi_{ij}(n)$ is bounded for all i, j, n by assumption. Finally, (C3) is trivially true from (A.4). □

Synchronous Q-learning

We do not take the expectation over the action of algorithm i when computing F, because synchronous Q-learning "performs" every action through counterfactuals; instead, the increment of the policy is the increment across all the actions. Therefore, we model the dynamics of the policies because we obtain the correct degree of $x_{ik}(n)$ to reduce the dynamics of the policies to the replicator-mutation dynamics.

Lemma 2 *When γ_n is sufficiently small, the increment of the policy from updating a specific action $a_i(n)$ for FAQ-learning is given by*

$$\begin{aligned} &\Delta x_{ik}(n) = x_{ik}(n+1) - x_{ik}(n) \\ &= \begin{cases} \gamma_n\, \tau\, x_{ik}(n)\, \big(1 - x_{ik}(n)\big)\big[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\big] + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -\gamma_n\, \tau\, x_{ik}(n)\, x_{i\ell}(n)\big[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\big] + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,, \end{cases} \end{aligned} \tag{A.12}$$

which is in the form of (A.2) and the reward from playing a_{ij} is $\pi_{ij}(n) = \pi_i(a_i(n) = a_{ij}, a_{-i}(n), a_\nu)$.

Proof An immediate result from the proof for Lemma 1 when using the appropriate update rule for Q-values. □

Lemma 2 describes the increment of the policy for algorithm-action pair ik from a specific action $a_i(n)$, but synchronous Q-learning updates and learns from every action. Thus, the increment for the algorithm-action pair ik is the sum of increments from (A.12) across every action.

Proposition 11 *The component functions of the deterministic vector field F for the policies of I independent learning algorithms using synchronous Q-learning are given by*

$$F_{ik}\left(\boldsymbol{x}\right) = \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right] + x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)}\right), \tag{A.13}$$

where the expected payoff of algorithm i playing action a_{ij} is given by

$$\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}}\Big[\mathbb{E}_{\nu}\left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i}\right]\Big].$$

Proof Lemma 2 provides the increment of the policy from updating a specific action $a_i(n)$, but synchronous Q-learning updates and learns from all the actions at every iteration. Therefore, the increment of the policy is given by

$$\begin{aligned}\Delta x_{ik}(n) = \gamma_n\, \tau\, x_{ik}(n) \Bigg(& \big(1 - x_{ik}(n)\big)\left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right] \\ & - \sum_{\ell \neq k} x_{i\ell}(n)\left[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\right]\Bigg) + O(\gamma_n^2),\end{aligned} \tag{A.14}$$

for all algorithm-action pair ik. To compute F, take the expectation of (A.14) with respect to the rewards that can be received without $O(\gamma_n^2)$ terms. We do this in two steps. First, take the expectation of the Nature player's action conditional on the actions of all others. Second, take the expectation over the actions of the opponents. For each of these steps, we condition on the action of algorithm i when conducting counterfactuals for action $a_i(n)$ so that the algorithms can learn from every action, but there is no need to take the expectation over the action of algorithm i because synchronous Q-learning conducts counterfactuals to update all possible actions.

Specifically, in the first step, define the expected payoff conditional on algorithm i playing action $a_i(n) = a_{ij}$ and the remaining algorithms playing a_{-i} as: $\Pi_{ij}(a_{-i}, n) := \mathbb{E}_\nu\left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i}\right]$. The expected increment of the policies conditional on actions a_{-i} is given by

$$\begin{aligned}\mathbb{E}\Big[\Delta x_{ik}(n) \,|\, a_{-i}(n)\Big] = \gamma_n\, \tau\, x_{ik}(n) \Bigg(& \big(1 - x_{ik}(n)\big)\Big[\Pi_{ik}(a_{-i}, n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] \\ & - \sum_{\ell \neq k} x_{i\ell}(n)\Big[\Pi_{i\ell}(a_{-i}, n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big]\Bigg).\end{aligned}$$

In the second step, take the expectation over the actions of the opponents, and the actions of the opponents depend on their policy which varies with n. Therefore, the conditional expected payoff for algorithm i

playing action $a_i(n) = a_{ij}$ is given by $\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}}\left[\Pi_{ij}(a_{-i}, n)\right]$. Thus, the expected increment of the policies is given by

$$\begin{aligned}\mathbb{E}\Big[\Delta x_{ik}(n)\Big] &= \gamma_n\, \tau\, x_{ik}(n) \left(\big(1 - x_{ik}(n)\big) \Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] \right. \\ &\qquad \left. - \sum_{\ell \neq k} x_{i\ell}(n) \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big] \right) \\ &= \gamma_n\, \tau\, x_{ik}(n) \left(\Big[\overline{\Pi}_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\Big] - \sum_{\ell} x_{i\ell}(n) \Big[\overline{\Pi}_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n)\Big] \right) ,\end{aligned}$$

for each algorithm-action pair ik. Similar to the proof of Proposition 7, use the fact that for fixed n, the quantity $\delta \max \boldsymbol{Q}_i$ is a constant, and the following relationship $\tau\,\big(Q_{i\ell}(n) - Q_{ik}(n)\big) = \ln\left(x_{i\ell}(n)/x_{ik}(n)\right)$ to get

$$\mathbb{E}\Big[\Delta x_{ik}(n)\Big] = \gamma_n\, \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right] + \gamma_n\, x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)}\right) ,$$

for each algorithm-action pair ik. Therefore, (A.13) describes the component functions of $F\left(\boldsymbol{x}\right)$ given by (A.5). □

Notice that the approximation errors $O(\gamma_n)$ are a finite sum of approximation errors from the increments of every action from an algorithm. Furthermore, a finite sum of terms that individually converge to zero also converges to zero. Thus, as before, it is sufficient to verify that the approximation errors $O(\gamma_n)$ in (A.12) converge to zero as $\gamma_n \to 0$.

Proposition 12 *Assume that the value of the exploration-exploitation parameter τ is finite and that the Q-values remain in the convex hull of experienced rewards. Let F be the continuous globally integrable vector field $F : [0,1]^{I\times K} \to [0,1]^{I\times K}$ with component functions*

$$F_{ik}\left(\boldsymbol{x}\right) = \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right] + x_{ik}(n) \sum_{\ell} x_{i\ell}(n)\, \ln\left(\frac{x_{i\ell}(n)}{x_{ik}(n)}\right) .$$

Set $t_n = \sum_{i=1}^{n} \gamma_i$ and let $\boldsymbol{X}$ be the continuous-time affine interpolated processes of policies for I independent learning algorithms using synchronous Q-learning with entries

$$X_{ik}\left(t_n + s\right) = x_{ik}(n) + s\, \frac{x_{ik}(n+1) - x_{ik}(n)}{t_{n+1} - t_n}$$

for all $n \in \mathbb{N}$ and $0 \leq s < \gamma_{n+1}$. Then, $\boldsymbol{X}$ is an asymptotic pseudotrajectory of the flow induced by F with probability one.

Proof Condition (C2) is trivially satisfied given that $\boldsymbol{x}(n) \in [0,1]^{I\times K}$ for all $n \in \mathbb{N}$. For condition (C1), notice that the entries of the deviation term in (A.5) are given by

$$U_{ik}(n) = \Bigg(\tau\, x_{ik}(n)\, \big(1 - x_{ik}(n)\big)\, \left[\pi_{ik}(n) + \delta \max \boldsymbol{Q}_i - Q_{ik}(n)\right]$$

$$- \tau\, x_{ik}(n) \sum_{\ell \neq k} x_{i\ell}(n) \left[\pi_{i\ell}(n) + \delta \max \boldsymbol{Q}_i - Q_{i\ell}(n) \right]$$

$$- \tau\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] - x_{ik}(n) \sum_{\ell} x_{i\ell}(n) \ln \left(\frac{x_{i\ell}(n)}{x_{ik}(n)} \right) \Bigg) ,$$

for each algorithm-action pair ik because every action is updated in synchronous Q-learning. As before, the Q-values and τ are finite by assumption so there exists a lower bound $0 < x_m \leq x_{ij}(n)$ for all i, j, n such that the logarithm of $x_{i\ell}(n)/x_{ik}(n)$ remains finite. Hence, (C1) is satisfied because $\pi_{ij}(n)$ is bounded for all i, j, n by assumption.

Finally, the approximation errors in (A.14) are a finite sum of approximation errors in (A.12). Furthermore, the approximation errors in (A.12) are those from the proof of Proposition 8 without the factor of $1/x_{ik}(n)$, and converge to zero as $\gamma_n \to 0$. Therefore, (C3) is satisfied. □

Appendix A.3. Other learning algorithms

Cross learning

Cross learning is the first algorithm used to make the connection between reinforcement learning and evolutionary game theory. Börgers and Sarin (1997) make the formal connection using stochastic approximation techniques by applying Theorem 1.1 in Chapter 8 of Norman (1972). Here, we demonstrate an alternative technique to make the formal connection using the approach in Appendix A.1.

Below, in Proposition 13, we compute the deterministic vector field F; this is straightforward because the rewards are a linear function of the update rule f. Finally, in Proposition 14, we show how to approximate the trajectories of the policies for I independent Cross learning algorithms with the trajectories from the replicator dynamics.

Proposition 13 *The component functions of the deterministic vector field F for the policies of I independent learning players using Cross learning are given by*

$$F_{ik}(\boldsymbol{x}) = x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right] , \tag{A.15}$$

where the expected payoff of algorithm i playing action a_{ij} is given by

$$\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}} \Big[\mathbb{E}_{\nu} \left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \,|\, a_i(n) = a_{ij}, a_{-i}(n) = a_{-i} \right] \Big] .$$

Proof Rearrange the evolution of the policy in (7) into the form of (A.1) and write the increment of the policies as

$$\Delta x_{ik}(n) = x_{ik}(n+1) - x_{ik}(n) = \begin{cases} \gamma_n\, \pi_{ik}(n) \left(1 - x_{ik}(n)\right) & \text{for } k \text{ s.t. } a_{ik} = a_i(n) , \\ -\gamma_n\, \pi_{i\ell}(n)\, x_{ik}(n) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell} , \end{cases} \tag{A.16}$$

to obtain f. Compute F by taking the expectation of (A.16) with respect to the rewards that can be received. We do this in three steps: take the expectation of the Nature player's action conditional on the actions of all the other players; take the expectation over the actions of the opponents conditional on the action of algorithm i; and take expectations over the actions of algorithm i to obtain

$$\mathbb{E}\Big[\Delta x_{ik}(n) \Big] = \gamma_n\, x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n) \right]$$

for each algorithm-action pair ik. Therefore, (A.15) describes the component functions of $F(\boldsymbol{x})$ given by (A.5). □

The continuous globally integrable vector field F in (A.15) is the replicator dynamics in (ODE 3). Therefore, to prove that Cross learning can be approximated by the replicator dynamics, we verify conditions (C1)–(C3) to obtain Proposition 14.

Proposition 14 *Let F be the continuous globally integrable vector field $F : [0,1]^{I\times K} \to [0,1]^{I\times K}$ with component functions*

$$F_{ik}(\boldsymbol{x}) = x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right] .$$

Set $t_n = \sum_{i=1}^{n} \gamma_i$ and let $\boldsymbol{X}$ be the continuous-time affine interpolated processes of policies for I independent learning algorithms using Cross learning with entries

$$X_{ik}(t_n + s) = x_{ik}(n) + s\, \frac{x_{ik}(n+1) - x_{ik}(n)}{t_{n+1} - t_n}$$

for all $n \in \mathbb{N}$ and $0 \leq s < \gamma_{n+1}$. Then, $\boldsymbol{X}$ is an asymptotic pseudotrajectory of the flow induced by F with probability one.

Proof Condition (C2) is trivially satisfied because $\boldsymbol{x}(n) \in [0,1]^{I\times K}$ for all $n \in \mathbb{N}$. The entries of the deviations are given by

$$U_{ik}(n) = \begin{cases} \left(\pi_{ik}(n)\ (1 - x_{ik}(n)) - x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right]\right) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ \left(-\pi_{i\ell}(n)\, x_{ik}(n) - x_{ik}(n) \left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\, \overline{\Pi}_{i\ell}(n)\right]\right) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \end{cases}$$

Now, (C1) is satisfied because $\pi_{ij}(n)$ is bounded for all i, j, n by assumption. Finally, (C3) is trivially true from (A.4). □

EXP3

Below, we introduce Lemma 3 to write the update rule of the policy as a linear function of the reward. The approximation simplifies the computation of the deterministic vector field F in Proposition 15 and reduces the dynamics of the policies to the replicator dynamics in (ODE 3). Finally, in Proposition 16, we show that the trajectories of the policies for I independent algorithms each using the EXP3 algorithm is approximated with the trajectories from the replicator dynamics.

Lemma 3 *When γ_n is sufficiently small, the increment of the policy for the EXP3 algorithm is given by*

$$x_{ik}(n+1) - x_{ik}(n) = \begin{cases} \gamma_n\, \dfrac{\pi_{ik}(n)}{K}\, (1 - x_{ik}(n)) + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ -\gamma_n\, \dfrac{\pi_{i\ell}(n)}{K}\, x_{ik}(n) + O(\gamma_n^2) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,, \end{cases} \tag{A.17}$$

which is in the form of (A.2) and the reward from playing a_{ij} is $\pi_{ij}(n) = \pi_i(a_i(n) = a_{ij}, a_{-i}(n), a_\nu)$.

Proof First, consider the case when $a_{ik} = a_i(n)$. Write the increment of the policy as

$$x_{ik}(n+1) - x_{ik}(n) = \frac{(1-\gamma_n)\; w_{ik}(n)\, A^{\gamma_n}_{k,n}}{\sum_\ell w_{i\ell}(n) + w_{ik}(n)\, \left(A^{\gamma_n}_{k,n} - 1\right)} - \frac{(1-\gamma_n)\; w_{ik}(n)}{\sum_\ell w_{i\ell}(n)}\,, \tag{A.18}$$

where $A^{\gamma_n}_{j,n} = \exp\left(\frac{\gamma_n\, \pi_{ij}(n)}{x_{ij}(n)\, K}\right)$. Substitute $(1-\gamma_n)\; w_{ik}(n) = \sum_\ell w_{i\ell}(n)\; [x_{ik}(n) - \gamma_n/K]$ in (A.18) and write

$$\begin{aligned}
\Delta x_{ik}(n) &= \frac{\sum_\ell w_{i\ell}(n)\; \left[x_{ik}(n) - \frac{\gamma_n}{K}\right]\, A^{\gamma_n}_{k,n}}{\sum_\ell w_{i\ell}(n) + \sum_\ell w_{i\ell}(n)\, \frac{x_{ik}(n) - \frac{\gamma_n}{K}}{1-\gamma_n}\, \left(A^{\gamma_n}_{k,n} - 1\right)} - \frac{\sum_\ell w_{i\ell}(n)\; \left[x_{ik}(n) - \frac{\gamma_n}{K}\right]}{\sum_\ell w_{i\ell}(n)} \\
&= \frac{(1-\gamma_n)\; \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, A^{\gamma_n}_{k,n}}{(1-\gamma_n) + \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)} - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \\
&= \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \left[\frac{(1-\gamma_n)\; A^{\gamma_n}_{k,n} - \left[(1-\gamma_n) + \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)\right]}{1 - \gamma_n + \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)}\right] \\
&= \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right) \left[\frac{(1-\gamma_n) - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)}{1 - \gamma_n + \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)}\right].
\end{aligned}$$

Use geometric sums to obtain

$$\begin{aligned}
\Delta x_{ik}(n) = &\left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)\, \left[(1-\gamma_n) - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\right] \\
&\times \left[1 + \gamma_n - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right) + O(\gamma_n^2)\right] \\
= &\left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right)\, \left[1 - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\right] + O(\gamma_n^2) \\
= &\left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{k,n} - 1\right) - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)^2\, \left(A^{\gamma_n}_{k,n} - 1\right) + O(\gamma_n^2) \\
= &\, x_{ik}(n)\, \left(A^{\gamma_n}_{k,n} - 1\right) - x_{ik}^2(n)\, \left(A^{\gamma_n}_{k,n} - 1\right) + O(\gamma_n^2)\,.
\end{aligned}$$

Finally, expand the exponential function with the power series $A^{\gamma_n}_{k,n} = 1 + \frac{\gamma_n\, \pi_{ik}(n)}{x_{ik}(n)\, K} + O(\gamma_n^2)$ to obtain

$$\begin{aligned}
\Delta x_{ik}(n) &= \frac{\gamma_n\, \pi_{ik}(n)}{K} - \frac{\gamma_n\, x_{ik}(n)\, \pi_{ik}(n)}{K} + O(\gamma_n^2) \\
&= \frac{\gamma_n\, \pi_{ik}(n)}{K}\, \left(1 - x_{ik}(n)\right) + O(\gamma_n^2)\,.
\end{aligned}$$

Similarly, consider the case when $a_{ik} \neq a_i(n) = a_{i\ell}$, and write

$$\begin{aligned}
\Delta x_{ik}(n) &= \frac{x_{ik}(n) - \frac{\gamma_n}{K}}{1 + \frac{x_{i\ell}(n) - \frac{\gamma_n}{K}}{1-\gamma_n}\, \left(A^{\gamma_n}_{\ell,n} - 1\right)} - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \\
&= \frac{(1-\gamma_n)\; \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)}{(1-\gamma_n) + \left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\, \left(A^{\gamma_n}_{\ell,n} - 1\right)} - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \left[\frac{(1-\gamma_n) - \left[(1-\gamma_n) + \left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\left(A_{\ell,n}^{\gamma_n} - 1\right)\right]}{(1-\gamma_n) + \left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\left(A_{\ell,n}^{\gamma_n} - 1\right)}\right] \\
&= -\left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\left(A_{\ell,n}^{\gamma_n} - 1\right)\left[1 + \gamma_n - \left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\left(A_{\ell,n}^{\gamma_n} - 1\right) + O(\gamma_n^2)\right] \\
&= -\left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\left(x_{i\ell}(n) - \frac{\gamma_n}{K}\right)\left(A_{\ell,n}^{\gamma_n} - 1\right) + O(\gamma_n^2) \\
&= -x_{ik}(n)\, x_{i\ell}(n)\left(A_{\ell,n}^{\gamma_n} - 1\right) + O(\gamma_n^2) \\
&= -\frac{\gamma_n\, \pi_{i\ell}(n)}{K}\, x_{ik}(n) + O(\gamma_n^2)\,.
\end{aligned}
$$

□

The expectation of $\tilde{f}$ is straightforward to compute because $\tilde{f}$ is linear in the rewards. Therefore, we take the expectation over Nature's actions, the opponents' actions, and algorithm i's own actions to compute F.

Proposition 15 *The component functions of the deterministic vector field F for the policies of I independent learning algorithms using the EXP3 algorithm is given by*

$$
F_{ik}(\boldsymbol{x}) = \frac{x_{ik}(n)}{K}\left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\,\overline{\Pi}_{i\ell}(n)\right], \tag{A.19}
$$

where the expected payoff of algorithm i playing action a_{ij} is given by

$$
\overline{\Pi}_{ij}(n) = \mathbb{E}_{\boldsymbol{x}_{-i}}\left[\mathbb{E}_{\nu}\left[\pi_i(a_i(n), a_{-i}(n), a_\nu) \mid a_i(n) = a_{ij}, a_{-i}(n) = a_{-i}\right]\right].
$$

Proof Apply Lemma 3 to obtain the approximation $\tilde{f}$ to the update rule in the form of (A.2). To compute F, take the expectation of (A.17) with respect to the rewards that can be received. We do this in three steps and neglect terms of order $O(\gamma_n^2)$: take the expectation of the Nature player's action conditional on the action of all the other players; take the expectation over the actions of the opponents conditional on the action of algorithm i; and take expectations over the actions of algorithm i to obtain

$$
\mathbb{E}\left[\Delta x_{ik}(n)\right] = \frac{\gamma_n}{K}\, x_{ik}(n)\left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\,\overline{\Pi}_{i\ell}(n)\right]
$$

for each algorithm-action pair ik. Therefore, (A.19) describes the component functions of $F(\boldsymbol{x})$ given by (A.5). □

The continuous globally integrable vector field F in (A.19) is the replicator dynamics in (ODE 3) (up to a constant re-scaling of time). Thus, the final step is to verify conditions (C1)–(C3) to obtain Proposition 16.

Proposition 16 *Let F be the continuous globally integrable vector field $F : [0,1]^{I\times K} \to [0,1]^{I\times K}$ with component functions*

$$
F_{ik}(\boldsymbol{x}) = \frac{x_{ik}(n)}{K}\left[\overline{\Pi}_{ik}(n) - \sum_{\ell} x_{i\ell}(n)\,\overline{\Pi}_{i\ell}(n)\right].
$$

Set $t_n = \sum_{i=1}^{n} \gamma_i$ and let $\boldsymbol{X}$ be the continuous-time affine interpolated processes of policies for I independent learning algorithms using the EXP3 algorithm with entries

$$X_{ik}\left(t_n + s\right) = x_{ik}(n) + s\, \frac{x_{ik}(n+1) - x_{ik}(n)}{t_{n+1} - t_n}$$

for all $n \in \mathbb{N}$ and $0 \leq s < \gamma_{n+1}$. Then, $\boldsymbol{X}$ is an asymptotic pseudotrajectory of the flow induced by F with probability one.

Proof Condition (C2) is trivially satisfied because $\boldsymbol{x}(n) \in [0,1]^{I \times K}$ for all $n \in \mathbb{N}$. For condition (C1), the entries of the deviations from (A.5) are given by

$$U_{ik}(n) = \begin{cases} \dfrac{1}{K} \left(\pi_{ik}(n) \left(1 - x_{ik}(n)\right) - x_{ik}(n) \, \left[\overline{\Pi}_{ik}(n) - \displaystyle\sum_{\ell} x_{i\ell}(n) \, \overline{\Pi}_{i\ell}(n) \right] \right) & \text{for } k \text{ s.t. } a_{ik} = a_i(n)\,, \\ \dfrac{1}{K} \left(-\pi_{i\ell}(n) \, x_{ik}(n) - x_{ik}(n) \, \left[\overline{\Pi}_{ik}(n) - \displaystyle\sum_{\ell} x_{i\ell}(n) \, \overline{\Pi}_{i\ell}(n) \right] \right) & \text{for } k \text{ s.t. } a_{ik} \neq a_i(n) = a_{i\ell}\,. \end{cases}$$

Now, (C1) is satisfied because $\pi_{ij}(n)$ is bounded for all i, j, n by assumption.

Next, we verify that the approximation errors converge to zero. Consider k such that $a_{ik} = a_i(n)$, then the approximation errors from the geometric expansion are

$$\begin{aligned} O(\gamma_n) = &- \frac{1}{K} \left(A_{k,n}^{\gamma_n} - 1\right) + \frac{2}{K}\, x_{ik}(n) \, \left(A_{k,n}^{\gamma_n} - 1\right) - \frac{\gamma_n}{K^2} \left(A_{k,n}^{\gamma_n} - 1\right) - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \left(A_{k,n}^{\gamma_n} - 1\right) \\ &+ \frac{1}{\gamma_n} \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \left(A_{k,n}^{\gamma_n} - 1\right) \left(1 - \gamma_n - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right)\right) \sum_{m=1}^{\infty} \left(\gamma_n - \left(x_{ik}(n) - \frac{\gamma_n}{K}\right) \left(A_{k,n}^{\gamma_n} - 1\right)\right)^m \,, \end{aligned}$$

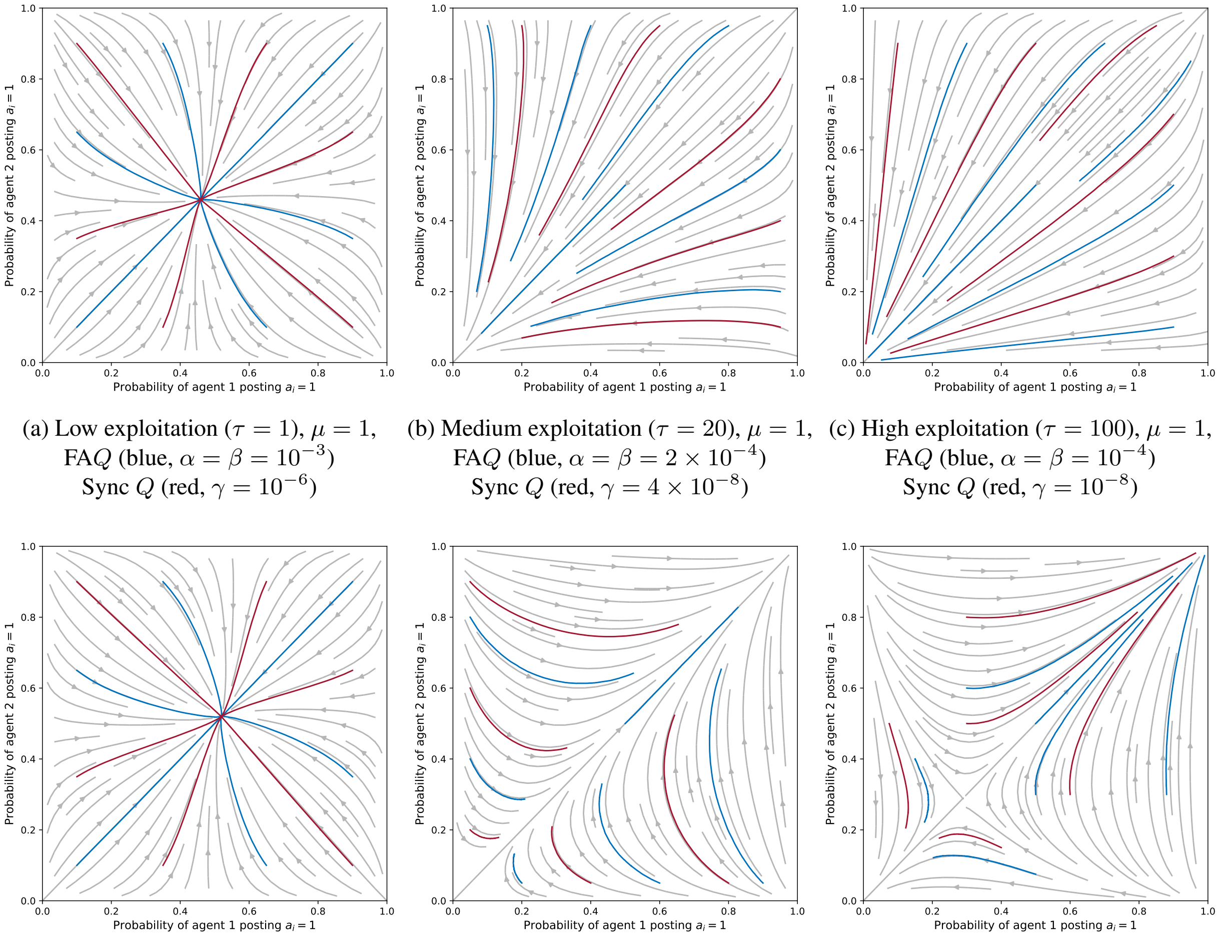
where $A_{k,n}^{\gamma_n} = \exp\left(\frac{\gamma_n \, \pi_{ik}(n)}{x_{ik}(n)\, K}\right)$ and the approximation errors from the power series expansion are

$$\begin{aligned} O(\gamma_n) &= \frac{1}{\gamma_n} \left(x_{ik}(n) - x_{ik}^2(n)\right) \left(\sum_{m=2}^{\infty} \frac{1}{m!} \left(\frac{\gamma_n \, \pi_{ik}(n)}{x_{ik}(n)\, K} \right)^m \right) \\ &= \frac{1}{\gamma_n} \left(x_{ik}(n) - x_{ik}^2(n)\right) \left(A_{k,n}^{\gamma_n} - 1 - \frac{\gamma_n \, \pi_{ik}(n)}{x_{ik}(n)\, K} \right) \,. \end{aligned}$$

When $\gamma_n \ll x_{ik}$, the infinite sum from the geometric expansion is finite and therefore zero when multiplied by $\left(A_{k,n}^{\gamma_n} - 1\right)/\gamma_n$ because $\left(A_{k,n}^{\gamma_n} - 1\right)/\gamma_n$ converges to zero as $\gamma_n \to 0$ by L'Hôpital's rule. The remaining terms also converge to zero as $A_{k,n}^{\gamma_n} - 1$ converges to zero as $\gamma_n \to 0$. Finally, the approximation errors from the power series expansion converges to zero as $\gamma_n \to 0$ by L'Hôpital's rule. We follow the same logic for k such that $a_{ik} \neq a_i(n) = a_{i\ell}$. Thus, (C3) is satisfied. □

Appendix B. Additional results

This appendix provides additional understanding into the dynamics of the algorithms from Section 4.3 and the stochastic approximation technique. It includes additional comparisons for different values of the exploration-exploitation parameter τ and the discount parameter δ; the effect of different configurations of the learning rate γ; and a verification that the stochastic approximation works for asymmetric algorithms with asymmetric actions and asymmetric learning rates.



(a) Low exploitation ($\tau = 1$), $\mu = 1$, FAQ (blue, $\alpha = \beta = 10^{-3}$) Sync Q (red, $\gamma = 10^{-6}$)

(b) Medium exploitation ($\tau = 20$), $\mu = 1$, FAQ (blue, $\alpha = \beta = 2 \times 10^{-4}$) Sync Q (red, $\gamma = 4 \times 10^{-8}$)

(c) High exploitation ($\tau = 100$), $\mu = 1$, FAQ (blue, $\alpha = \beta = 10^{-4}$) Sync Q (red, $\gamma = 10^{-8}$)

(d) Low exploitation ($\tau = 1$), $\mu = 2$, FAQ (blue, $\alpha = \beta = 10^{-3}$) Sync Q (red, $\gamma = 10^{-6}$)

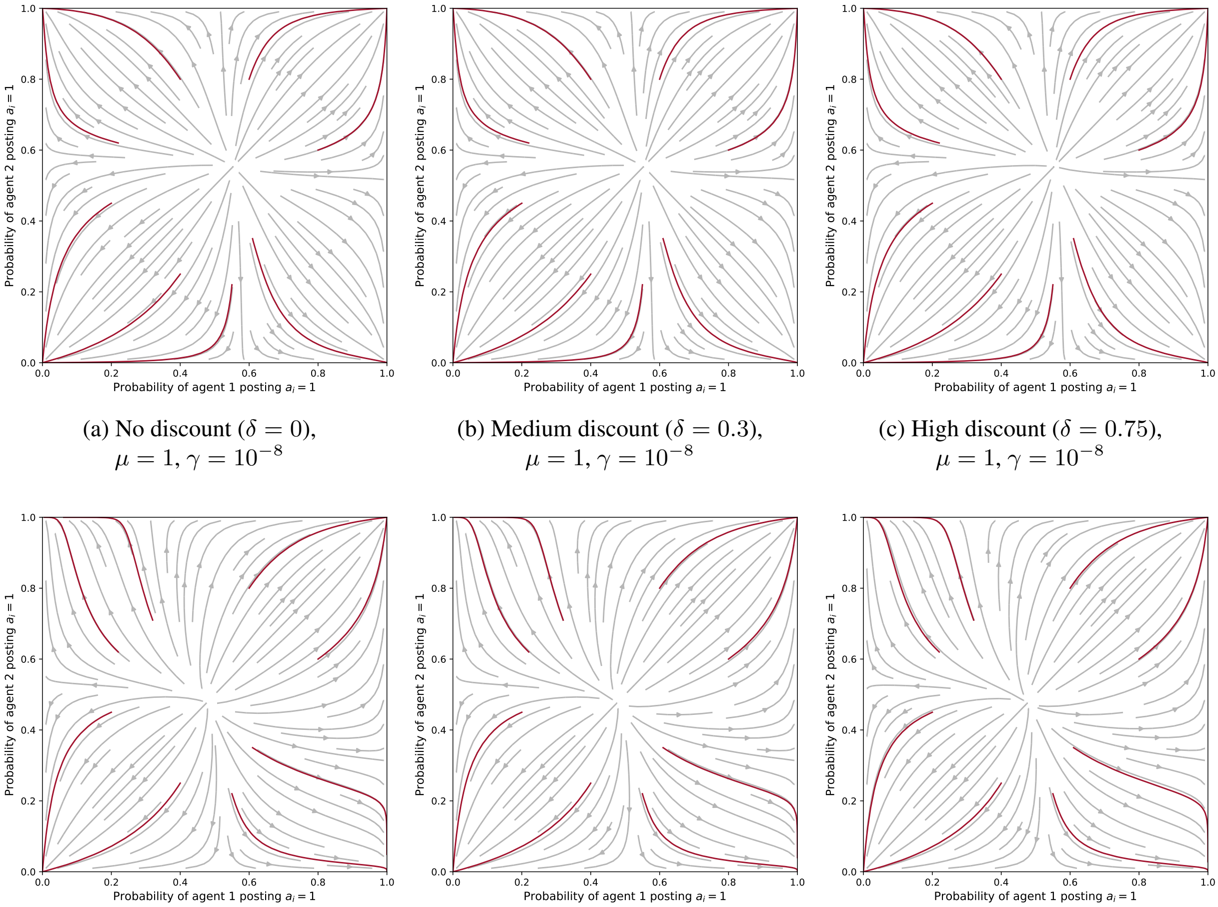
(e) Medium exploitation ($\tau = 20$), $\mu = 2$, FAQ (blue, $\alpha = \beta = 2 \times 10^{-4}$) Sync Q (red, $\gamma = 4 \times 10^{-8}$)

(f) High exploitation ($\tau = 100$), $\mu = 2$, FAQ (blue, $\alpha = \beta = 10^{-4}$) Sync Q (red, $\gamma = 10^{-8}$)

Figure B.8: Theoretical and actual trajectories for frequency adjusted and synchronous Q-learning in a 2×2 game with different values of μ and τ.

First, we discuss how the value of the exploration-exploitation parameter τ influences the policies of the players. Figure B.8 looks at how the exploration-exploitation parameter τ affects the learning dynamics for frequency adjusted and synchronous Q-learning using the same setup of the 2×2 game in Section 4.3 with $a_i = (1, 2)$ for $\mu = 1$ and $\mu = 2$. For a wide range of values of the discount parameter δ the dynamics show a similar behaviour; here we set $\delta = 0.75$. We consider three cases of τ: $\tau = 1$, i.e., little to no exploitation, $\tau = 20$, i.e., there is exploitation, $\tau = 100$, i.e., almost pure exploitation. The grey lines indicate the theoretical trajectories of policies while the blue and red lines are the actual trajectories of polices from the algorithms for x_{i1} with $i \in \{1, 2\}$. We specifically pick values of the learning rate to achieve an accurate approximation and the specific learning rate used are given in the sub-captions of each plot.

We see that the value of the exploration-exploitation parameter τ affects the equilibria reached. In particular, several cases stand out. First, $\tau = 1$ leads to a convergence to a mixed equilibrium where posting



(a) No discount ($\delta = 0$), $\mu = 1, \gamma = 10^{-8}$

(b) Medium discount ($\delta = 0.3$), $\mu = 1, \gamma = 10^{-8}$

(c) High discount ($\delta = 0.75$), $\mu = 1, \gamma = 10^{-8}$

(d) No discount ($\delta = 0$), $\mu = 2, \gamma = 10^{-8}$

(e) Medium discount ($\delta = 0.3$), $\mu = 2, \gamma = 10^{-8}$

(f) High discount ($\delta = 0.75$), $\mu = 2, \gamma = 10^{-8}$

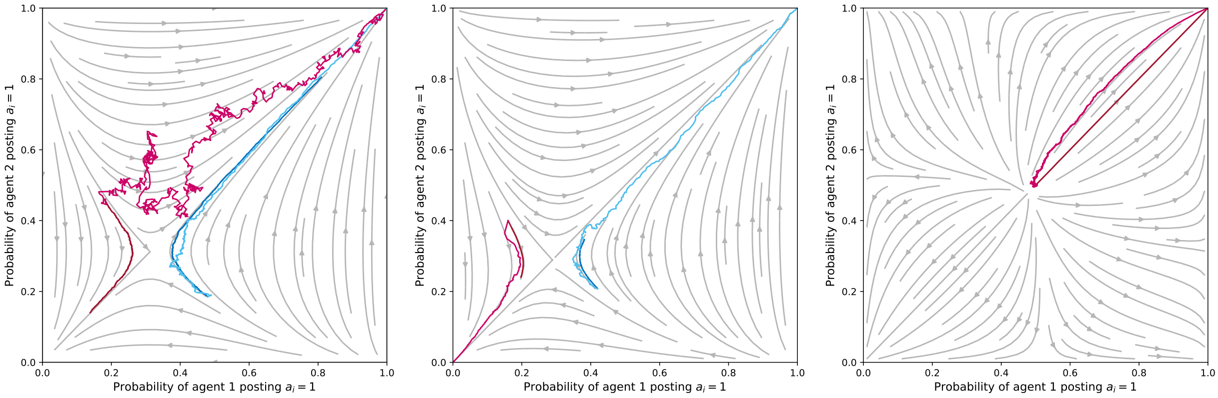
Figure B.9: Theoretical and actual trajectories of policies for Q-learning in a 2×2 game with different values of μ and δ.

an offer $a_i = 1$ happens around 50% of the time for both $\mu = 1$ and $\mu = 2$ which corresponds to the equilibrium from the mutation equation for $K = 2$. Second, when $\tau = 20$ and $\mu = 2$, there is a global convergence to a probability of one for posting an offer $a_i = 1$ when there should be two symmetric pure strategy Nash equilibria in the stage game. This discrepancy is a result of the mutation term contributing a sufficient push towards the center of the simplex, and this push is sufficient to drive the overall dynamics away from the collusive pure Nash equilibrium of playing $a_i = 1$ with zero probability. Finally, when the value of τ is sufficiently large, the dynamics from the mutation term provides minimal contribution to the overall dynamics, hence the dynamics resemble the replicator dynamics.

Figure B.9 compares the theoretical trajectories and actual trajectories of the policies for Q-learning through the probability of each learning algorithm posting an offer $a_i = 1$. We fix the value of the exploration-exploitation parameter $\tau = 100$ and compare the dynamics for three different discount values of $\delta = 0, 0.3, 0.75$ and for $\mu = 1, 2$. The trajectories for Q-learning are approximated through Q-values, and with the Q-values we compute the trajectories of the policies $\boldsymbol{x}_i(n)$ through the softmax activation. This allows us to easily visualise and understand the dynamics of Q-learning that would otherwise be difficult to visualise through the space of the four-dimensional Q-values.

Interestingly, for a wide range of values of the discount parameter δ the dynamics show a similar behaviour, neither does it address the fact that Q-learning converges to asymmetric actions in a symmetric game. Instead, the latency parameter μ plays a larger role in changing the dynamics of Q-learning. This is seen more clearly in Figure 5. For a tick size $\vartheta = 1$, we see that a larger latency value leads to more symmetric outcomes.

Intuitively, a partial explanation as to why Q-learning gets stuck in asymmetric actions is because Q-learning is poor at adapting to changes in the reward as a result of the actions of adversaries. Unlike policy iterators such as Cross learning, Q-learning and its variations track a separate value estimate which then gets mapped to the policy. However, the variations of Q-learning have additional advantages that make them more adaptable. Synchronous Q-learning receives more information through counterfactuals, thus it can make more informed decisions compared with Q-learning. On the other hand, recall that frequency adjusted Q-learning has a scaling factor of $1/x_{ik}$ which allows the algorithms to adapt to changes in the reward. This is because if the algorithm learns that a specific action is sub-optimal, it will have a low probability of picking the action. However, if it picks that action and receives a large reward, the scaling of $1/x_{ik}$ will lead to a sufficiently large change to the Q-values so that the algorithm chooses that action more frequently, and therefore able to adapt. Q-learning does not have this scaling factor, which means that it is unable to adapt its strategy appropriately to changes in the reward.



(a) CL: $\gamma = 10^{-5}$ (blue) vs $\gamma_n = 0.01\, n^{-1/3}$ (cyan) EXP3: $\gamma = 10^{-5}$ (red) vs $\gamma_n = 0.01\, n^{-1/3}$ (pink)

(b) FAQ: $\alpha = \beta = 10^{-4}$ (blue) vs $\alpha_n = 0.01\, n^{-1/3}, \beta = 10^{-4}$ (cyan) Sync Q: $\gamma = 10^{-8}$ (red) vs $\gamma_n = 10^{-4}\, n^{-1/3}$ (pink)

(c) Q-learning, $\gamma = 10^{-8}$ (red) vs $\gamma_n = 10^{-4}\, n^{-1/3}$ (pink)

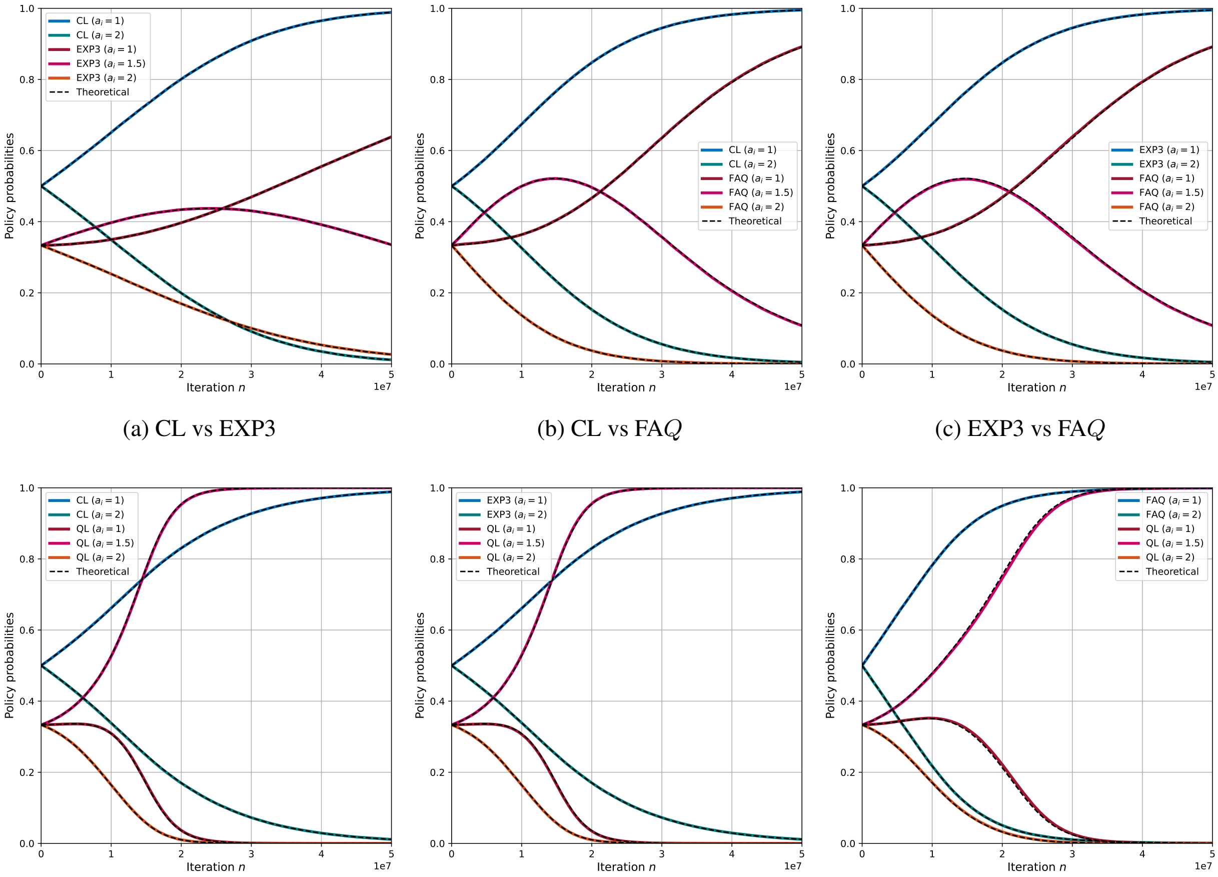
Figure B.10: Theoretical and actual trajectories for all the algorithms in 2×2 games with different configurations of the learning rate γ_n.

Time step

Throughout the paper, we assumed that γ_1 is sufficiently small so that we skip the transient phase. Figure B.10 compares the trajectories for a decreasing learning rate that has yet to skip the transient phase with a constant learning rate which we use to proxy the behaviour when we skip the transient phase. The configurations of γ_n are provided in the sub-captions. The game is played with a latency parameter value $\mu = 2$ and $a_i = (1, 2)$. We set the exploration-exploitation parameter $\tau = 100$ and the discount factor $\delta = 0.75$ for Q-learning and its variations.

For the decreasing learning rate that has yet to skip the transient phase, we see that the approximation is not accurate for the initial phase of the trajectory, but as γ_n becomes smaller, the actual trajectories line up with the theoretical trajectories. In particular, Figure B.10a demonstrates that during the transient phase, the random movements may cause the trajectory to become locked into a different basin of attraction.

Asymmetric games



(a) CL vs EXP3

(b) CL vs FAQ

(c) EXP3 vs FAQ

(d) CL vs Q-learning

(e) EXP3 vs Q-learning

(f) FAQ vs Q-learning

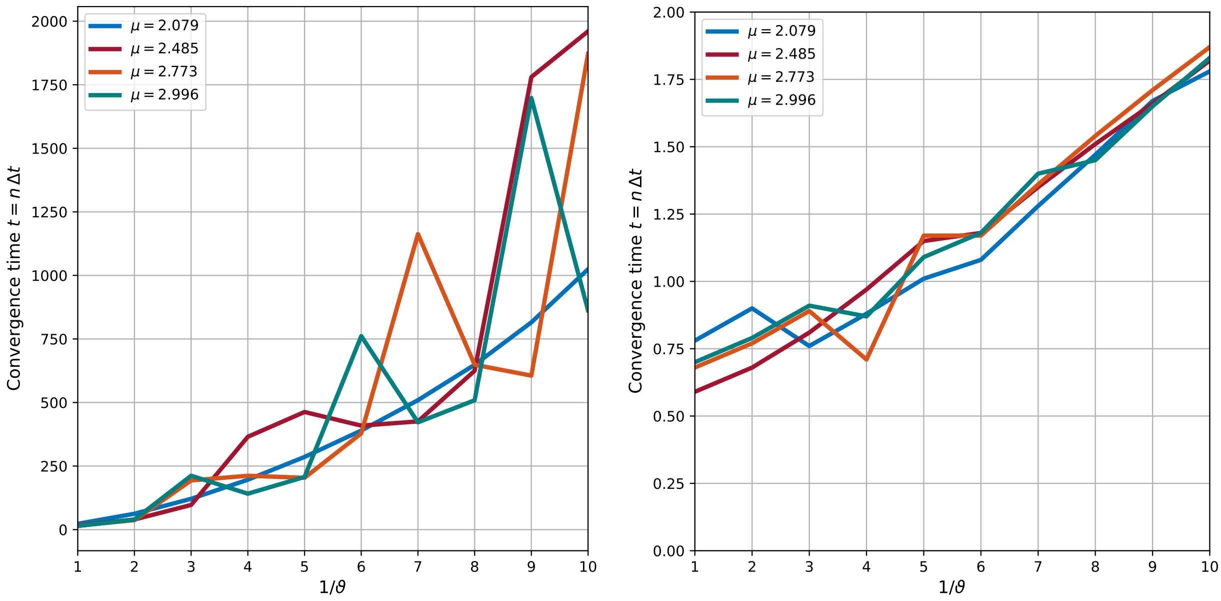
Figure B.11: Theoretical and actual trajectories from a single path with asymmetric algorithms, asymmetric actions and asymmetric learning rates.

Finally, we verify that the stochastic approximation can be extended to players who use different algorithms with asymmetric actions. The first algorithm has two actions $a_1 = (1, 2)$ and the second algorithm has three actions $a_2 = (1, 1.5, 2)$. The game has a latency parameter of $\mu = 2$, thus, the symmetric pure Nash equilibria of the stage game is $(1, 1)$ and $(2, 2)$. We consider all combinations of the algorithms excluding synchronous Q-learning. We fix the learning rate $\gamma = 10^{-6}$ for Cross learning and the EXP3 algorithm, $\alpha = \beta = 10^{-4}$ for frequency adjusted Q-learning, and $\gamma = 10^{-8}$ for Q-learning. For Q-learning and frequency adjusted Q-learning, we set the exploration-exploitation parameter $\tau = 100$ and the discount factor $\delta = 0.75$. We chose a smaller learning rate for Q-learning and its variations to compensate the speedup in the dynamics caused by τ in the replicator-mutation dynamics, but it conveniently allows us to verify that the stochastic approximation also holds for asymmetric learning rates.

Figure B.11 compares a single path of the theoretical and actual trajectories of the policies for the various combinations starting from the initial condition where all the actions have an equal probability of being played. Specifically, for Q-learning, we set $Q_{2k}(1) = 1$ for all k. We see that the actual trajectories from all the combinations recover the theoretical trajectories.

Furthermore, we see that combinations of algorithms that behave like replicator and replicator-mutation dynamics converge to the pure Nash equilibrium of $(1, 1)$. On the other hand, when Cross learning, the EXP3 algorithm, and FAQ-learning are paired with Q-learning (where Q-learning has three actions), the algorithms converge to $a_1 = 1$, whereas Q-learning converges to $a_2 = 1.5$. This result is explained by the inability for Q-learning to respond to changes in the reward. Specifically, in the initial stage when the algorithm is playing each action with an equal probability, the optimal action for Q-learning is to play $a_2 = 1.5$. As the algorithms adapt to the actions of Q-learning, they start to undercut and favor $a_1 = 1$ and they start playing the competitive action with a higher probability. Q-learning is unable to respond to this change and therefore gets stuck in a sub-optimal action. Therefore, in this asymmetric setting, Q-learning is "fooled" into playing a sub-optimal action.

Appendix C. Convergence times

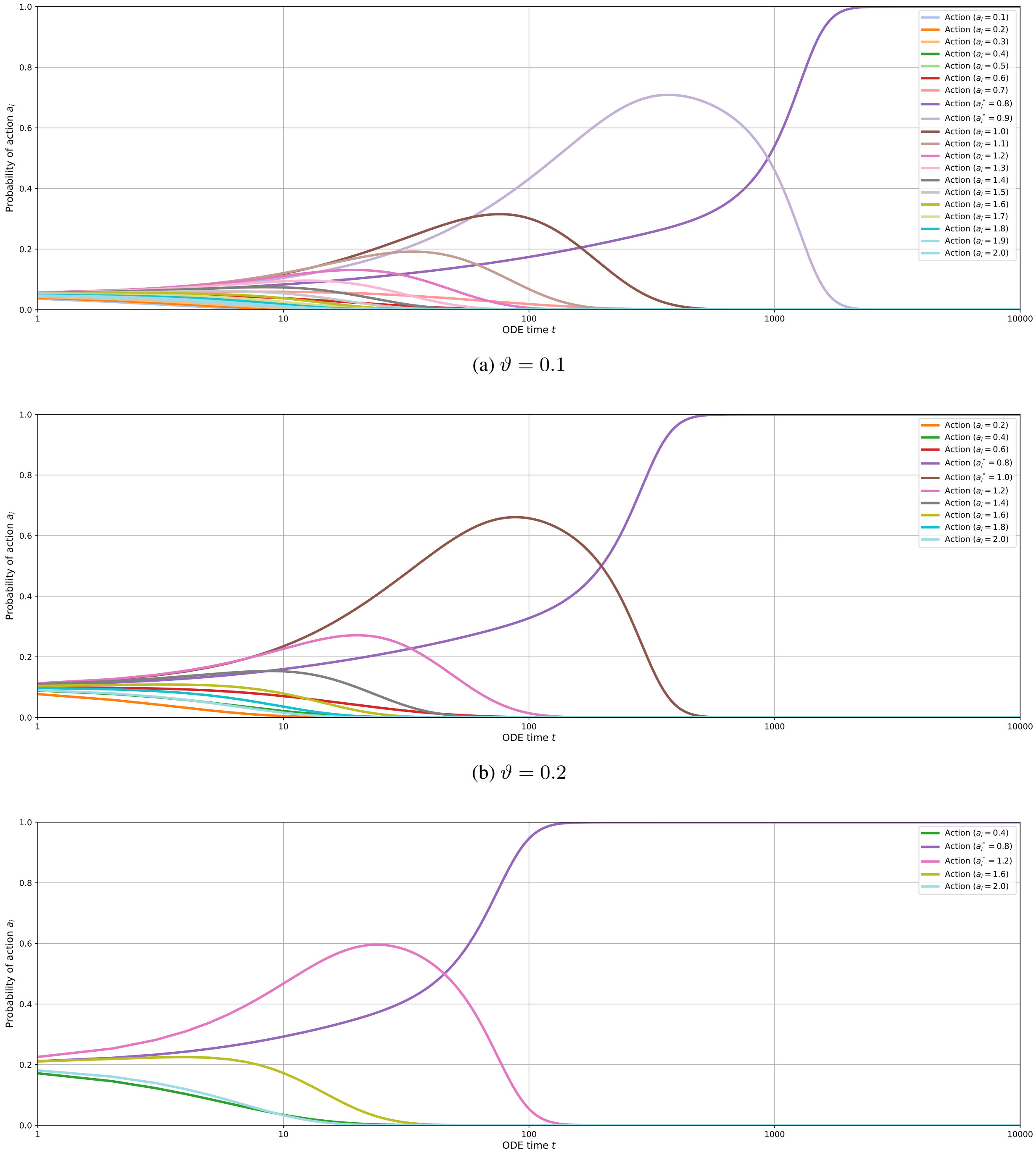


(a) Replicator dynamics

(b) Q-learning

Figure C.12: Time taken for the trajectories of the ODEs to converge to a rest point as a function of the reciprocal of tick size for different values of the latency parameter μ.

Figure C.12 plots the time it takes for the system of ODEs to reach a rest point as a function of the reciprocal of tick size ϑ^{-1} for four values of the latency parameter $\mu = \ln(4\,c)$, where $c = 2, 3, 4, 5$. For the replicator dynamics, we run a single trajectory from the center of the simplex (where every action has an equal probability of being played) until convergence; and for the dynamics of Q-learning, we run a single



(a) $\vartheta = 0.1$

(b) $\vartheta = 0.2$

(c) $\vartheta = 0.4$

Figure C.13: The trajectories of policies for the replicator dynamics with $\mu = 2.485$ and $\vartheta = 0.1, 0.2, 0.4$. The pure strategy Nash equilibria are denoted by a_i^*.

trajectory of Q-values from $Q_{ik}(1) = 1$ for all i, k until convergence. As above, we say that the replicator dynamics converge when each player has more than 99% probability of playing an action. Convergence for the dynamics of Q-learning is achieved when every component $\dot{Q}_{ik}$ is less than 10^{-10}. Both the replicator dynamics and the dynamics of Q-learning are iterated with a step size of $\Delta t = 0.01$, and the dynamics of Q-learning are simulated with the discount factor $\delta = 0.75$, which is mapped to the policy with the exploration-exploitation parameter $\tau = 100$.

We see that smaller tick sizes lead to longer convergence times, with the convergence times for the dynamics of Q-learning being orders of magnitudes smaller than the convergence times for the replicator dynamics. The substantial differences between algorithms arise because the dynamics have different convergence criteria and because Q-learning is less responsive to changes of strategies by opponents, as outlined in the proof of Proposition 3.

We provide an example to illustrate that slower convergence can lead to higher trading costs in a finite horizon to demonstrate that a very small tick size may not be optimal. In Figure C.13 we plot the trajectories for the replicator dynamics starting from the center of the simplex for each player for $\vartheta = 0.1, 0.2, 0.4$ with $\mu = 2.485$ ($c = 2$). We plot the policies for only one player because the players are symmetric. In the three cases, all the algorithms reach the pure Nash equilibrium offer of $a_i = 0.8$ and have a finite horizon where they quote supracompetitive offers above the offer of $a_i = 0.8$. The faster convergence in this case improves the welfare for traders. Specifically, the finite horizon average profits for each market maker over the interval $[0, 2000]$ is 0.403319 for $\vartheta = 0.4$, 0.408585 for $\vartheta = 0.2$, and 0.422347 for $\vartheta = 0.1$.[35] The upper bound of the interval is chosen to correspond to around when the policies of $\vartheta = 0.1$ reaches a pure strategy Nash equilibrium.

The example we provide is a particular stylised case where the initial condition for $\vartheta = 0.1, 0.2, 0.4$ converges to the pure strategy Nash equilibrium offer of $a_i = 0.8$. In general, there are a variety of factors that influence the trading costs in a finite horizon. First, the initial conditions of the trajectories determine which pure strategy Nash equilibrium the algorithms reach. Second, different tick sizes and latency values lead to different sets of pure strategy Nash equilibria. Third, the cutoff that defines a finite horizon is arbitrary and different cutoffs will lead to different trading costs. Finally, the significance of slower convergence times also depends on the learning rate used and the physical time between each iteration of the algorithm, which also influences the choice of the cutoff.

Nonetheless, we demonstrate that the algorithms that converge to pure strategy Nash equilibria of the market making game may also quote supracompetitive prices for a long period of time. Accounting for the trade-off may lead to a conclusion as in Cordella and Foucault (1999) where a tick size of zero may never be optimal.

Appendix D. Proofs

Proof of Proposition 4

The monopolist price is

$$a_M = \arg\max_{a \in [0,2]} \pi(a, a_{-i} = a)\,,$$

which leads to the corner point solution $a_M = 2$.

The Bertrand–Nash equilibrium of the stage game with a continuous action space solves the set of simultaneous equations

$$0 = \frac{\partial}{\partial a_i} \pi\left(a_i, a_{-i}\right), \quad i \in \{1, \ldots, I\}\,,$$

[35] We obtain the finite horizon average profits by computing the expected profit $\boldsymbol{x}^\top \boldsymbol{\Pi}\, \boldsymbol{x}$ at every $\Delta t = 0.01$ and averaging across all the samples of expected profits.

where

$$\frac{\partial}{\partial a_i}\pi\left(a_i, a_{-i}\right) = \left[1 + \sum_{j \neq i} e^{\mu\left(a_i - a_j\right)}\right]^{-1} - \mu\, a_i \sum_{j \neq i} e^{\mu\left(a_i - a_j\right)} \left[1 + \sum_{j \neq i} e^{\mu\left(a_i - a_j\right)}\right]^{-2} .$$

The solution is obtained when all market makers post the same depth of $a_{BN} = a_i = a_{-i}$ for all i. Thus, all players post

$$a_{BN} = \frac{I}{\mu\,(I-1)}\,,$$

which requires $a_{BN} < 2$, i.e., $\mu > I/\big(2\,(I-1)\big)$.

For $a_i = 0$ and finite μ, the marginal profit

$$\left.\frac{\partial}{\partial a_i}\pi\left(a_i, a_{-i}\right)\right|_{a_i=0} = \left[1 + \sum_{j \neq i} e^{\mu\left(-a_j\right)}\right]^{-1} > 0$$

is positive for all $a_{-i} \geq 0$. Therefore, the perfectly competitive price $a_{PC} = 0$ is strictly dominated. □

To prove Proposition 5 we establish a simple but useful lemma and corollary. For readability, we denote $\pi(a_i, a_{-i})$ as $\pi(a_i; a_{-i})$, and the fill probability as

$$P(a_{ik}; a') = \frac{\exp(-\mu\, a_{ik})}{\exp(-\mu\, a_{ik}) + (I-1)\,\exp(-\mu\, a')}\,,$$

where a_{ik} is action k for player i and a' are the actions of the opponents.

Lemma 4

4.1 $\frac{\partial}{\partial a'}\pi_i(a_{ik}; a') = a_{ik}\,\mu\, P(a_{ik}; a')\,\big(1 - P(a_{ik}; a')\big)$.

4.2 $\frac{\partial}{\partial a_{ik}}\pi_i(a_{ik}; a') = P(a_{ik}; a') - a_{ik}\,\mu\, P(a_{ik}; a')\,\big(1 - P(a_{ik}; a')\big)$.

4.3 $\frac{\partial}{\partial a_{ik}}P(a_{ik}; a') > 0$.

4.4 If $a_{ik} > \frac{I}{I-1}\frac{1}{\mu}$, *then* $\frac{\partial}{\partial a_{ik}}\pi_i(a_{ik}; a' = a_{ik}) < 0$.

Proof of Lemma 4

Lemma 4.1:

$$\begin{aligned}
\frac{\partial}{\partial a'}\pi_i(a_{ik}; a') &= \frac{\partial}{\partial a'}\frac{a_{ik}\,\exp(-\mu\, a_{ik})}{\exp(-\mu\, a_{ik}) + (I-1)\,\exp(-\mu\, a')} \\
&= a_{ik}\,\frac{\exp(-\mu\, a_{ik})}{(\exp(-\mu\, a_{ik}) + (I-1)\,\exp(-\mu\, a'))^2}\,\mu\,(I-1)\,\exp(-\mu\, a') \\
&= a_{ik}\,\mu\,\frac{\exp(-\mu\, a_{ik})}{\exp(-\mu\, a_{ik}) + (I-1)\,\exp(-\mu\, a')}\,\frac{(I-1)\,\exp(-\mu\, a')}{\exp(-\mu\, a_{ik}) + (I-1)\,\exp(-\mu\, a')}
\end{aligned}$$

$$= a_{ik}\,\mu\,P(a_{ik};a')\,\big(1 - P(a_{ik};a')\big)\ .$$

Lemma 4.2:

$$\begin{aligned}
\frac{\partial}{\partial a_{ik}}\pi_i(a_{ik};a') &= \frac{\partial}{\partial a_{ik}} a_{ik}\,P(a_{ik};a') = \frac{\partial}{\partial a_{ik}}\frac{a_{ik}\,\exp(-\mu\,a_{ik})}{\exp(-\mu\,a_{ik}) + (I-1)\,\exp(-\mu\,a')} \\
&= P(a_{ik};a') + a_{ik}\,\frac{\partial}{\partial a_{ik}}\frac{1}{1+\exp(\mu\,a_{ik})\,(I-1)\,\exp(-\mu\,a')} \\
&= P(a_{ik};a') - a_{ik}\,\frac{1}{(1+\exp(\mu\,a_{ik})\,(I-1)\,\exp(-\mu\,a'))^2}\,\mu\,(I-1)\,\exp(-\mu\,a')\,\exp(\mu\,a_{ik}) \\
&= P(a_{ik};a') - a_{ik}\,\mu\,\frac{\exp(-\mu\,a_{ik})}{\exp(-\mu\,a_{ik}) + (I-1)\,\exp(-\mu\,a')}\,\frac{(I-1)\,\exp(-\mu\,a')}{\exp(-\mu\,a_{ik}) + (I-1)\,\exp(-\mu\,a')} \\
&= P(a_{ik};a') - a_{ik}\,\mu\,P(a_{ik};a')\,\big(1 - P(a_{ik};a')\big)\ .
\end{aligned}$$

Lemma 4.3:

$$\begin{aligned}
\frac{\partial}{\partial a_{ik}}P(a_{ik};a') &= \frac{\partial}{\partial a_{ik}}\frac{1}{1+\exp(\mu\,a_{ik})\,(I-1)\,\exp(-\mu\,a')} \\
&= \mu\,\frac{\exp(\mu\,a_{ik})\,(I-1)\,\exp(-\mu\,a')}{\left(1+\exp(\mu\,a_{ik})\,(I-1)\,\exp(-\mu\,a')\right)^2} > 0\,.
\end{aligned}$$

Lemma 4.4:

$$\frac{\partial}{\partial a_{ik}}\pi_i(a_{ik};a'=a_{ik}) = P(a_{ik};a')\,\Big(1 - a_{ik}\,\mu\,\big(1 - P(a_{ik};a')\big)\Big)\ .$$

Now, $P(a_{ik};a'=a_{ik}) = 1/I$, so that

$$\frac{\partial}{\partial a_{ik}}\pi_i(a_{ik};a'=a_{ik}) = \frac{1}{I}\left(1 - a_{ik}\,\mu\,\left(1-\frac{1}{I}\right)\right) = \frac{1}{I}\left(1 - a_{ik}\,\mu\,\frac{I-1}{I}\right)\ .$$

By hypothesis, $a_{ik} > \dfrac{I}{I-1}\dfrac{1}{\mu}$ implies that $a_{ik}\,\mu\,\dfrac{I-1}{I} > 1$, therefore, $\dfrac{\partial}{\partial a_{ik}}\pi_i(a_{ik};a'=a_{ik}) < 0$. □

Corollary 1 $\dfrac{\partial}{\partial a_{ik}}\pi_i(a_{ik};a')$ *changes sign at most once on $\mathcal{A}$, and if it does it changes from positive to negative.*

Proof of Proposition 5

Proposition 5.1:

By Corollary 1, an action a_k that satisfies the following two inequalities $\pi(a_k+\vartheta;a'=a_k) \le \pi(a_k;a'=a_k)$ and $\pi(a_k-\vartheta;a'=a_k) \le \pi(a_k;a'=a_k)$ is a pure strategy Nash equilibrium.

When $\pi(a_k+\vartheta;a'=a_k) \le \pi(a_k;a'=a_k)$ we have that

$$\frac{a_k+\vartheta}{1+(I-1)\,\exp(\mu\,\vartheta)} \le \frac{a_k}{I}$$

$$\begin{aligned} I\,(a_k+\vartheta) &\le a_k + I\,a_k\,\exp(\mu\,\vartheta) - a_k\,\exp(\mu\,\vartheta) \\ I\,\vartheta &\le a_k - I\,a_k + (I-1)\,a_k\,\exp(\mu\,\vartheta) \\ I\,\vartheta &\le (I-1)\,a_k\,(\exp(\mu\,\vartheta)-1) \\ \frac{\vartheta}{(I-1)\,(\exp(\mu\,\vartheta)-1)} &\le \frac{a_k}{I}\,, \end{aligned}$$

and when $\pi(a_k-\vartheta; a'=a_k) \le \pi(a_k; a'=a_k)$ we have that

$$\begin{aligned} \frac{a_k-\vartheta}{1+(I-1)\,\exp(-\mu\,\vartheta)} &\le \frac{a_k}{I} \\ a_k\,I - \vartheta\,I &\le a_k + (I-1)\,a_k\,\exp(-\mu\,\vartheta) \\ a_k\,I - a_k - (I-1)\,a_k\exp(-\mu\,\vartheta) &\le \vartheta\,I \\ a_k\,(I-1)\,(1-\exp(-\mu\,\vartheta)) &\le \vartheta\,I \\ \frac{a_k}{I} &\le \frac{\vartheta}{(I-1)\,(1-\exp(-\mu\,\vartheta))}\,. \end{aligned}$$

Combine the two inequalities shows that a_k is a pure strategy symmetric Nash equilibrium of the stage game with discrete actions if

$$L^* := \frac{\vartheta\,I}{(I-1)\,(\exp(\mu\,\vartheta)-1)} \le a_k \le \frac{\vartheta\,I}{(I-1)\,(1-\exp(-\mu\,\vartheta))} =: U^*\,.$$

Proposition 5.2:

We show that $U^* - L^* > \vartheta$.

$$\begin{aligned} \frac{\vartheta\,I}{(I-1)\,(1-\exp(-\mu\,\vartheta))} - \frac{\vartheta\,I}{(I-1)\,(\exp(\mu\,\vartheta)-1)} &> \vartheta \\ \frac{\vartheta\,I}{(I-1)}\left(\frac{1}{1-\exp(-\mu\,\vartheta)} - \frac{1}{\exp(\mu\,\vartheta)-1}\right) &> \vartheta \\ \frac{1}{1-\exp(-\mu\,\vartheta)} - \frac{1}{\exp(\mu\,\vartheta)-1} &> \frac{I-1}{I} \\ \frac{\exp(\mu\,\vartheta)-1-1+\exp(-\mu\,\vartheta)}{(1-\exp(-\mu\,\vartheta))\,(\exp(\mu\,\vartheta)-1)} = 1 &> \frac{I-1}{I}\,. \end{aligned}$$

Therefore, there exists at least one $a_k \in [L^*, U^*]$.

Proposition 5.3:

We first show that the Bertrand–Nash action lies in between the two bounds, and then show that the bounds converge to the Bertrand–Nash action.

(1) We verify that $a_{BN} \in [L^*, U^*]$. Substitute $a_k = a_{BN} = I/\big(\mu\,(I-1)\big)$ into the inequalities to get

$$\begin{aligned} \frac{\vartheta\,I}{(I-1)\,(\exp(\mu\,\vartheta)-1)} \le \frac{I}{\mu\,(I-1)} &\le \frac{\vartheta\,I}{(I-1)\,(1-\exp(-\mu\,\vartheta))} \\ \frac{\mu\,\vartheta}{\exp(\mu\,\vartheta)-1} \le 1 &\le \frac{\mu\,\vartheta}{1-\exp(-\mu\,\vartheta)}\,. \end{aligned}$$

From the first inequality we have $1 \leq \exp(\mu\,\vartheta) - \mu\,\vartheta$ from $\mu\,\vartheta \leq \exp(\mu\,\vartheta) - 1$. From the second inequality we have $1 \leq \mu\,\vartheta + \exp(-\mu\,\vartheta)$ from $1 - \exp(-\mu\,\vartheta) \leq \mu\,\vartheta$. Both conditions are true because $\exp(\mu\,\vartheta) - \mu\,\vartheta$ and $\mu\,\vartheta + \exp(-\mu\,\vartheta)$ are increasing functions for $\mu\,\vartheta > 0$, and both attain a minimum value of one when $\mu\,\vartheta = 0$. Therefore, $a_{BN} \in [L^*, U^*]$.

(2) It is easy to see that

$$\lim_{\vartheta \to 0} \frac{\vartheta\, I}{(I-1)\,(\exp(\mu\,\vartheta) - 1)} = \frac{1}{\mu}\frac{I}{I-1} \quad \text{and} \quad \lim_{\vartheta \to 0} \frac{\vartheta\, I}{(I-1)\,(1 - \exp(-\mu\,\vartheta))} = \frac{1}{\mu}\frac{I}{I-1}$$

follows from

$$1 = \lim_{\vartheta \to 0} \frac{\mu\,\vartheta}{\exp(\mu\,\vartheta) - 1} = \lim_{\vartheta \to 0} \frac{\mu\,\vartheta}{1 - \exp(-\mu\,\vartheta)}\,.$$

□