

Understanding Individual Decision-Making in Multi-Agent Reinforcement Learning: A Dynamical Systems Approach

James Rudd-Jones

Centre for Artificial Intelligence,
Department of Computer Science,
University College London
London, UK
james.rudd-jones.22@ucl.ac.uk

María Pérez-Ortiz

Centre for Artificial Intelligence,
Department of Computer Science,
University College London
London, UK
maria.perez@ucl.ac.uk

Mirco Musolesi

Centre for Artificial Intelligence,
Department of Computer Science,
University College London
London, UK
Department of Computer Science and
Engineering, University of Bologna
Bologna, Italy
m.musolesi@ucl.ac.uk

ABSTRACT

Analysing learning behaviour in Multi-Agent Reinforcement Learning (MARL) environments is challenging, in particular with respect to *individual* decision-making. Practitioners frequently tend to study or compare MARL algorithms from a qualitative perspective largely due to the inherent stochasticity in practical algorithms arising from random dithering exploration strategies, environment transition noise, and stochastic gradient updates to name a few. Traditional analytical approaches, such as replicator dynamics, often rely on mean-field approximations to remove stochastic effects, but this simplification, whilst able to provide general overall trends, might lead to dissonance between analytical predictions and actual realisations of individual trajectories. In this paper, we propose a novel perspective on MARL systems by modelling them as *coupled stochastic dynamical systems*, capturing both agent interactions and environmental characteristics. Leveraging tools from dynamical systems theory, we analyse the stability and sensitivity of agent behaviour at individual level, which are key dimensions for their practical deployments, for example, in presence of strict safety requirements. This framework allows us, for the first time, to rigorously study MARL dynamics taking into consideration their inherent stochasticity, providing a deeper understanding of system behaviour and practical insights for the design and control of multi-agent learning processes.

KEYWORDS

Multi-Agent Reinforcement Learning, Dynamical Systems, Individual Agent Behaviour, Stability Analysis, Sensitivity Analysis

1 INTRODUCTION

Reinforcement learning (RL) in multi-agent settings routinely exhibits abrupt performance shifts, oscillations, and inconsistent game equilibria during training [23, 39]. RL practitioners observe these phenomena across a sweep of training runs even with the same modelling architectures and hyperparameters, making stability at training time and asymptotic performance both difficult to reason about and costly to validate empirically [27]. Unstable training can lead to brittle policies that fail under small perturbations in initialisation or environment [1, 27]. In safety-critical applications such as autonomous driving, performance oscillations or catastrophic divergence pose unacceptable risks [2, 22]. Assessing a learning

system's stability and sensitivity to parameter changes provides crucial insight into its robustness, generalisation capacity, and long-term adaptability in dynamic environments. In fact, non-linear and chaotic dynamics can lead to oscillatory and inconsistent behaviour that may appear as random [50]. Positively, there is typically some structure in the phase space of the dynamical system that explains the seemingly purely stochastic behaviour [49].

There is a long tradition of framing learning as a dynamical system: cognitive science and neuroscience have advocated dynamical systems views of agents and environments [19]; evolutionary game theory models population adaptation via dynamical systems analysis [38]; and several RL algorithms can be proved to converge by viewing updates as a dynamical system. Multi-Agent Reinforcement Learning (MARL) systems can be studied through a dynamical systems perspective. For example, Cross Learning [14] is closely related to replicator dynamics [11]. Q-learning has also been shown to behave like best-response dynamics, which often align with replicator dynamics [55]. However, these studies either (i) focus on *population level* and *deterministic dynamics* (e.g., replicator or mean-field limits), or (ii) treat *environment as a dynamical system* while representing the learning rule only implicitly.

In this paper, for the first time, we present the study of individual decision-making in MARL systems using tools from dynamical systems theory from an *individual* perspective. In order to do so, we view the environment and agent learning updates as *coupled dynamical systems* [7]. This agent-centric perspective closely reflects the true agent-environment interactions, whereas modelling only a population-level strategy tends to average out many of these interactions and couplings, eliminating the stochastic behaviour inherent in practical MARL implementations. Adding more agents to the system can be interpreted as introducing additional coupled dynamical systems, resulting in increasingly complex non-linear dynamics. In contrast to previous work, we therefore adopt an individual *agent-centric parameter-space* viewpoint that models the learning updates themselves as a discrete-time dynamical system, then study the stability and sensitivity of the coupled learner(s) and environment system. This approach allows us to leverage the mature toolbox of dynamical systems theory, including mathematical methods for analysing stationary (or invariant) distributions [18, 48], limit cycles, quasi-cycles, random and strange attractors, and Lyapunov stability [36] in non-linear, stochastic, coupled systems. In the paper, we will provide a pragmatic description of the

application of these tools in the context of MARL¹ with a focus on the analysis of the emergent dynamics of the interacting agents.

Using this dynamical-systems perspective, we demonstrate that many seemingly unstable or stochastic learning trajectories in MARL can be explained by low-dimensional dynamical structures emerging from the coupling between agents and their environments. We show that empirical stability and sensitivity metrics derived from dynamical systems theory correspond closely to observed training behaviour across a variety of MARL settings. This connection enables a practical diagnostic framework: one can identify when a learning process transitions from stable to unstable regimes, assess robustness to perturbations, and potentially guide the tuning of hyperparameters to maintain desirable dynamical properties.

Concretely, the contributions of this paper can be summarised as follows:

- We formalise MARL updates as coupled discrete-time dynamical systems in parameter space, both in the deterministic and stochastic settings as a basis for understanding *individual* decision making in MARL.
- We introduce a practical methodology for analysing the stability of MARL systems coupled by applying established tools from dynamical systems theory, including the computation of invariant distributions, Lyapunov exponents, recurrence plots, and fractal dimensions, enabling a comprehensive characterisation of the system’s dynamical regimes, transitions, and resilience properties.
- We utilise these analytical methods in a suite of stateless and state-based MARL games with tabular and deep MARL methods, highlighting the applicability and generality of these methods. We adjust hyperparameters of agents to understand the sensitivity of the underlying dynamical systems to these changes.
- Finally, we discuss potential applications of this toolbox, particularly its use in analysing stability and sensitivity for algorithm design.

2 BACKGROUND

Multi-agent reinforcement learning (MARL) has demonstrated remarkable successes across a wide range of domains, including competitive multiplayer games [9, 56], cooperative and competitive social dilemmas [3, 4, 15, 33], fluid flow and control problems [40, 51], and even environmental policy derivation [43, 44, 60]. Despite these successes, MARL remains substantially more challenging to train and analyse than its single-agent counterpart. Non-stationarity induced by multiple agents learning concurrently often produces oscillatory behaviours or brittle dependence on hyperparameters leading to instability or non-convergence, which complicate reproducibility and theoretical understanding [10]. Unlike single-agent RL, where convergence to optimal policies can often be guaranteed under certain assumptions [57], MARL algorithms frequently fail to converge or converge only under restrictive assumptions such as of a zero-sum game [32] or in the mean field limit [58]. As a result, a general theoretical understanding of MARL convergence remains

elusive, motivating ongoing efforts to reconcile empirical analysis with theoretical insights.

Evolutionary Game Theory & Replicator Dynamics. Game Theory has been extensively used to understand MARL equilibria and convergence [59], providing static views of the system but struggles with the non-stationarity in multi-agent learning. Evolutionary Game Theory, a branch of Game Theory, has emerged as a more suitable alternative as it models adaption through selection and mutation *over time*. Labelled as Replicator Dynamics [30, 38], the agent-centric system is reformulated as a population level dynamical system that denotes the evolving population share of agent strategies over time. Evolutionary game theory introduces evolutionary stable strategies that are an asymptotically stable fixed point of the replicator dynamical system. MARL updates and replicator dynamics are strongly linked as the population share of each strategy can be related to the probability of taking a certain action, as if viewing through the lens of a single agent [10]. For example, Cross learning [14] admits a formal correspondence to replicator dynamics in normal-form games, converging to the replicator dynamics in the continuous time limit [11]. Subsequent work extended these links to more complex scenarios, including stochastic and sequential games [21, 28, 29]. However, there is a disconnect between replicator dynamics theory and practical implementations of MARL. Firstly, an infinite population or mean-field approximation is assumed, abstracting away from the local, trajectory-dependent updates of individual agents. Secondly, algorithm specific details such as exploration, non-myopic updates, bootstrapping, and function approximation that are inherent sources of stochasticity aren’t modelled by replicator dynamic. Consequently, replicator dynamics can provide powerful qualitative and geometric insights for MARL (e.g., why rock-paper-scissors leads to cycling), but fails to capture finite-agent effects, stochastic learning, and other complexities of modern MARL algorithms.

Other Deterministic Methods. Alternatively, Barfuss et al. [6] look at deriving the deterministic limit of three common RL algorithms - Q-learning, SARSA, and actor critic learning. By separating the adaption timescale (learning) from the interaction timescale (game dynamics) they map stochastic update rules found in traditional RL algorithms (e.g., ϵ -greedy exploration) to continuous deterministic flows [6]. Thus enabling them to leverage deterministic dynamical systems theory to understand the attractors of the systems. However, adjusting algorithms so that they act in deterministic ways reduces much of their performance as well as the ability to scale to larger state spaces with function approximation. What is gained in theoretical predictive performance comes at the cost of scalable implementations.

Chaos in MARL. One of the major difficulties in MARL is the prevalence of chaotic dynamics, which makes proving convergence particularly challenging. For example, Sato et al. [47] demonstrate how replicator dynamics can generate complex orbits as well as chaotic attractors and Galla et al. [20] and Sanders et al. [46] show that two player and many player games respectively can exhibit fixed points, limit cycles, or chaotic behaviour dependent on algorithm parameter choices. These findings raise the fundamental question of whether fixed points are consistently attainable in such systems at all, or whether approximations and assumptions used to enforce convergence inadvertently strip away the very dynamics

¹For a more in-depth description of dynamical systems theory, we refer the reader to the existing excellent resources in the areas, e.g., [50].

that could be essential for capturing the richness of agent interactions. In this sense, the ‘‘chaos’’ observed in MARL is not merely a technical obstacle, but a core characteristic of the domain that complicates both theory and practice.

3 OUR APPROACH AT A GLANCE

In this work we focus on understanding individual decision-making of MARL agents, an agent-centric rather than population level viewpoint. Our perspective is therefore complementary to population based approaches: rather than modelling population averages, we directly analyse the learning dynamics of individual agents.

3.1 Definitions and the Concept

Dynamical Systems. A dynamical system is a mathematical framework for describing the evolution of a system over time, formally consisting of a state space X together with an evolution rule ϕ_t that describes how the state changes. The state space is typically in \mathbb{R}^n or a manifold, and the evolution rule can be either continuous or discrete in time as well as either deterministic or stochastic. Continuous time deterministic dynamics are expressed by an Ordinary Differential Equation (ODE):

$$\dot{x} = f(x(t)), \quad x(0) = x_0, \quad (1)$$

where $f : X \rightarrow \mathbb{R}^n$ is a vector field (or flow field), and the solution is the flow $\phi_t(x_0)$ that represents a trajectory of the system starting from the initial state. Discrete time deterministic dynamics are described by an iterated map:

$$x_{t+1} = F(x_t), \quad x_0 \in X, \quad (2)$$

where $F : X \rightarrow X$ is a transformation and the system evolves by iteration of F .

One can visualise the state space (aka phase space) as a landscape of possible states that the dynamical system evolves through. Trajectories (realisations) of the system emit structure in the phase space, known as a phase plot/portrait, that illustrate the behaviour of the system. Attractors are locations in phase space the system evolves towards regardless of initial conditions, indicating asymptotic behaviour. There are three main types of attractors: fixed points - point locations in which the system converges as the vector field vanishes or $F(x^*) = x^*$, limit cycles - attractors the system periodically loops or orbits around, strange attractor - bounded region of phase space where the system exhibits irregular behaviour leading to a fractal structure.

Coupled dynamical systems exhibit the same structure, but are comprised of multiple dynamical systems that have cross or coupling terms within f or F , such as the canonical bidirectionally coupled Logistic Map [52]:

$$\begin{aligned} x_{t+1} &= x_t(r_x - r_x x_t - \beta_{xy} y_t) \\ y_{t+1} &= y_t(r_y - r_y y_t - \beta_{yx} x_t), \end{aligned} \quad (3)$$

where $r_x, r_y, \beta_{xy}, \beta_{yx}$ are parameters that adjust the chaotic behaviour and coupling strength.

Markov Decision Process/Markov Game. RL utilises the Markov Decision Process (MDP) framework, defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. Where \mathcal{S} is the set of states, \mathcal{A} the set of actions, $P(s' | s, a)$ the transition probability of moving to state s' after taking action a in state s , $R(s, a)$ the reward function, and $\gamma \in [0, 1)$ a discount factor.

At each timestep t , the agent observes $s_t \in \mathcal{S}$, chooses $a_t \in \mathcal{A}$, receives reward $r_t = R(s_t, a_t)$, and transitions to $s_{t+1} \sim P(\cdot | s_t, a_t)$. To generalise MDPs for multiple agents, the framework becomes a stochastic game or Markov Game (MG), a multi-agent MDP with N agents defined by the tuple $\langle \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, P, \{R^i\}_{i=1}^N, \gamma \rangle$. Where \mathcal{S} is the shared state space, \mathcal{A}^i is the action space of agent i , $P(s' | s, a^1, \dots, a^N)$ is the transition probability dependent on the joint action $a = (a^1, \dots, a^N)$, and $R^i(s, a)$ is the reward function for agent i . Unlike the single-agent case, where the environment is stationary under a fixed policy, multi-agent systems are inherently non-stationary since the dynamics depend on the evolving policies of all agents.

Single-Agent RL as a Dynamical System. Consider a discrete-time dynamical system for a singular agent. We assume for now that everything is deterministic and highlight later the sources of potential stochasticity.

$$s_{t+1} = f(s_t, a_t), \quad (4)$$

where a_t is the agent action, s_t is the state at time step t , and $f(\cdot, \cdot)$ is our transition function that is deterministic at time t . This defines the environmental dynamical system where agent actions have an effect on the transition dynamics. Similarly to Beer [7] we model a coupled dynamical system of the learning RL based agent and the environmental dynamical system:

$$\theta_{h+1} = g(\theta_h, s_h, a_h), \quad (5)$$

where h is a learning update step, which relates to a certain number of time steps t . θ_h (a scalar or vector) is the state of the agent at update h , perhaps the model parameters or similar, and an update depends on the trajectory of states and actions gained in that update window. Actions are generated from a policy π using the environment state s_t and the agent state θ_h : $a_t = \pi(x_t; \theta_h)$. We want to understand the stability of the function g , which can be done by analysing what kind of attractor it settles upon. This definition represents the simplest form, but it can be extended to incorporate practical implementations such as replay buffers, target networks, or off-policy learning, where the samples used in the updates may originate from different behavioural policies.

MARL as a Dynamical System. For clarity in this section we assume just two agents in this system but the definition can be expanded to general agents. Upon introducing multiple agents our discretised transition dynamics become:

$$s_{t+1} = f(s_t, a_t^1, a_t^2), \quad (6)$$

where the superscript identifies the agent. This creates a coupled dynamical system between the updates of the individual agents’ dynamical systems as the trajectories are dependent on the updates of other agents, defined as:

$$\begin{aligned} \theta_{h+1}^1 &= g^1(\theta_h^1, s_h, a_h^1) \\ \theta_{h+1}^2 &= g^2(\theta_h^2, s_h, a_h^2), \end{aligned} \quad (7)$$

where an agent updates its internal representation given a set of historical states and its own actions. Different MARL algorithms change g^i (e.g., independent learners or opponent modelling), altering the coupling strength/structure. Additional variables can be added creating further coupling, such as opponent actions or other feature representations used for agent updates. In the following, we

focus on the simplest case where an agent makes decisions about the system only from global state information and its own action, defined in agent parameter space:

$$\begin{aligned}\theta_{h+1}^1 &= g^1(\theta_h^1, f(s_t, \pi_{\theta^1}(s_t), \pi_{\theta^2}(s_t)), \pi_{\theta^1}(s_t)) \\ \theta_{h+1}^2 &= g^2(\theta_h^2, f(s_t, \pi_{\theta^1}(s_t), \pi_{\theta^2}(s_t)), \pi_{\theta^2}(s_t)).\end{aligned}\quad (8)$$

Our goal is to *measure and compare* stability and sensitivity of g across MARL algorithms, games, and hyperparameters, beyond only fixed-point existence.

The Impacts of Stochasticity. Equation 8 presents the simplest coupled dynamical system for a fully deterministic setting. In practical settings, multiple sources of stochasticity arise within the system, stemming from environmental transitions, exploratory mechanisms such as random dithering, sampling from policy distributions, and stochastic gradient updates. Traditionally, these stochastic elements have been approximated deterministically to enable analytical tractability. In contrast, we explicitly model these sources of randomness to more faithfully capture the behaviour of MARL agents as a coupled stochastic dynamical system:

$$\begin{aligned}\theta_{h+1}^1 &= g^1(\theta_h^1, f(s_t, \pi_{\theta^1}(s_t) + \xi_t, \pi_{\theta^2}(s_t) + \xi_t) + \eta_t, \pi_{\theta^1}(s_t) + \xi_t) + \zeta_h \\ \theta_{h+1}^2 &= g^2(\theta_h^2, f(s_t, \pi_{\theta^1}(s_t) + \xi_t, \pi_{\theta^2}(s_t) + \xi_t) + \eta_t, \pi_{\theta^2}(s_t) + \xi_t) + \zeta_h\end{aligned}\quad (9)$$

where ξ_t represents stochasticity in exploration strategies or policy sampling, η_t the environment transition noise, and ζ_h the stochastic gradient updates. Below, we present a version that subsumes all sources of stochasticity into a single term:

$$\begin{aligned}\theta_{h+1}^1 &= g^1(\theta_h^1, f(s_t, \pi_{\theta^1}(s_t), \pi_{\theta^2}(s_t)), \pi_{\theta^1}(s_t)) + v_h \\ \theta_{h+1}^2 &= g^2(\theta_h^2, f(s_t, \pi_{\theta^1}(s_t), \pi_{\theta^2}(s_t)), \pi_{\theta^2}(s_t)) + v_h,\end{aligned}\quad (10)$$

where v_h is the combined stochastic effect term.

Comparison to Replicator Dynamics. For clarity, we distinguish our agent-centric perspective from the population-level replicator dynamics. The latter can be formulated as a continuous-time dynamical system defined by:

$$\dot{u}_i = u_i [w_i(\mathbf{u}) - \bar{w}(\mathbf{u})], \quad (11)$$

where $\mathbf{u} = (u_1, u_2, \dots, u_n)$ represents the population state vector representing the percentage of the population belong to each of the n strategies, $w_i(\cdot)$ is the fitness function of a specific strategy, and $\bar{w}(\cdot)$ is the average fitness of the total population. As the $\bar{w}(\cdot)$ term is an expectation over the whole population, this averages out many of the stochastic effects.

3.2 Dimensions of our Analysis

Equation 10 provides a general definition of a coupled dynamical system for two agents, which during learning generates a resultant phase portrait in (θ^1, θ^2) for any combination of environment and MARL agents. Leveraging stochastic dynamical systems theory, we can diagnose and quantify the behaviour of the underlying system dynamics. Below are three primary avenues opened by this framework, which we explore in more detail in subsequent sections:

- *Stability analysis:* Understanding the asymptotic performance of a MARL system allows us to rigorously validate the resulting game equilibria.

- *Sensitivity analysis:* Once the phase-space structure can be analysed, we can assess how parameter changes affect stability, providing insight into the system's sensitivity.
- *Control:* By quantifying stability and sensitivity in the coupled MARL dynamical system, we can inform strategies for improved control.

4 EXPERIMENTAL SETTINGS

To explore the three avenues above we utilise the same environment and experimental conditions. For the individual components, such as specific tooling or methods required for each avenue (e.g., dynamical systems theory methods), we introduce them in the relevant section. We consider four canonical stateless repeated games and one larger multi-state environment. Prisoner’s Dilemma: A standard social dilemma with a unique Nash equilibrium (defection), though cooperation can arise under repeated interaction [5, 41]. Matching Pennies: A zero-sum game with a mixed-strategy equilibrium. Stag Hunt: A coordination game with multiple equilibria (cooperative “stag” or risk-dominant “hare”). Chicken: A cooperative/competitive game where mutual aggression is costly, with two pure-strategy equilibria (one player swerves while the other does not) and a mixed-strategy equilibrium. Overcooked: A cooperative multi-agent environment in which two players must coordinate to prepare and deliver dishes in a shared kitchen. The environment’s high-dimensional, partially observable state space make it a challenging test bed for studying learning dynamics [13]. Appendix A presents the detailed descriptions and configurations of these environments.

We use two independent learner algorithms that closely align with replicator dynamics. In tabular Q-learning, each agent learns its own Q -function independently, typically with ϵ -greedy exploration. However, in stateless repeated games, deterministic Q-learning approximates replicator dynamics in the continuous-time, infinitesimal step-size limit if it uses Boltzmann (Softmax) exploration [10]. In policy gradient, using the REINFORCE trick reduces to replicator dynamics in the mean-field limit, since subtracting the baseline corresponds to subtracting the average payoff [8]. Further, we look at Independent Deep Q Networks (IDQN) [53], which, by utilising neural network function approximation, scales to the larger state space environments such as Overcooked, but inherently has many more stochastic elements.

5 UNDERSTANDING INDIVIDUAL DECISION-MAKING IN MARL

5.1 Stability Analysis

Overview. Analysing the stability of a MARL system relates to understanding the asymptotic performance, indicating potential game equilibria. Traditional game theory and evolutionary game theory provide static equilibria, a singular fixed point that the system as a whole tends towards, but which is not always seen with practical MARL algorithms [30]. When working with environments and agents that are amenable to replicator dynamics, closed form solutions arise and thus one can plot the gradient vector plot signalling the flow of the replicator dynamical system. Since our approach is data-driven, we visualise realisations of the dynamical system in parameter space, also known as policy traces. Figure

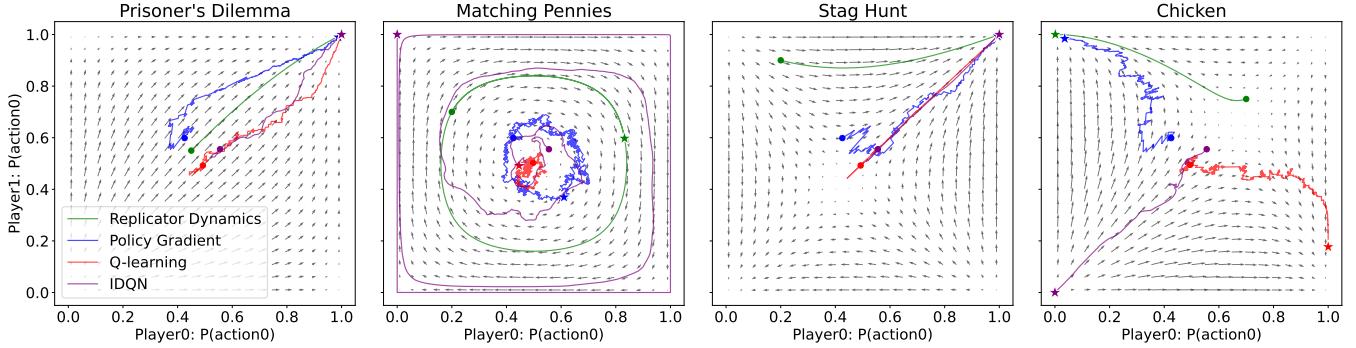


Figure 1: Comparison between replicator dynamics and realisations of Policy Gradient, Tabular Q-learning, and IDQN with Boltzmann exploration in the four stateless environments. Replicator dynamics are represented by the analytical vector field as well as a realisation in blue from an arbitrary initial condition. Other algorithms realisations are defined by the legend. Realisations start from circular points and end at stars.

1 compares the replicator dynamics phase plot and a realisation, with an agent-centric realisation of Q-learning, Policy Gradient, and IDQN agents all using Boltzmann exploration so that the parameters relate to action probabilities. In general, the replicator dynamics flow field is closely followed by the true realisations of each agent. Both Q-learning and the Policy Gradient approach are tuned to follow replicator dynamics by using very small learning rates and Boltzmann exploration to reduce the effect of stochasticity. However, although IDQN also uses Boltzmann exploration, there is still stochasticity in the function approximation, gradient updates, and other features such as target networks. The rapidly expanding cycle in Matching Pennies and the move towards defection in Chicken against the replicator dynamics flow can be attributed to these additions.

Instead, by focusing on an agent-centric stochastic coupled dynamical system framework, we aim to understand agent behaviour directly from data. This comes at the cost of precise prediction, but it avoids the pitfalls of assuming purely deterministic dynamics. We can analyse the true structure in parameter phase space of arbitrary combinations of environments and algorithms, in a way that complements the predictive findings from replicator dynamic analysis where they apply. The challenge is that when stochasticity is present, fixed points and limit cycles are no longer well-defined in the strict sense, since process noise perturbs trajectories. Classical terminology (e.g. fixed points, attractors, stability) can still be used, but it refers to these noisy counterparts.

Formally, let the update rule for one agent be written as a Markov process:

$$\theta_{h+1} = g(\theta_h, s_t, \pi) + v_h. \quad (12)$$

We define a stationary distribution ρ^θ that satisfies:

$$\theta_h \sim \rho^\theta \Rightarrow \theta_{h+1} \sim \rho^\theta. \quad (13)$$

While exact solutions are often intractable, empirical approximations of ρ^θ can be obtained by simulating trajectories for long horizons (or batched across initial conditions), yielding an ergodic estimate. The shape and properties of ρ^θ determine whether the system admits noisy analogues of fixed points, cycles, or chaotic attractors. If the distribution focuses on a contained area of the

phase space then we may have a fixed point. Quasi-cycles define a situation where the underlying deterministic behaviour would converge to a fixed point or limit cycle but the stochasticity forces it to enter a limit cycle with noise. These manifest not as a true closed orbit, but as sustained oscillatory behaviour, leading to a ‘‘smeared’’ ring in ρ^θ . To make this concrete, we introduce several quantitative and qualitative diagnostics:

- *Stationary (invariant) distributions* [18, 48]: Probability distribution over θ . If the Frobenius norm $\|\Sigma\|_F$ of the covariance of ρ^θ is low it indicates convergence to a fixed point.
- *Lyapunov exponents* [36]: Lyapunov exponents quantify how quickly two trajectories diverge or converge, a hallmark of stability or chaos. For trajectories θ_h and θ'_h starting ϵ apart, $\lambda = \lim_{h \rightarrow \infty} \frac{1}{h} \log \frac{\|\theta_h - \theta'_h\|}{\|\theta_0 - \theta'_0\|}$. Negative exponents suggest convergence to a noisy fixed point, near-zero values indicate cycles or neutral stability, and positive exponents are evidence of chaos.
- *Recurrence plots* [16]: Dynamical systems over time can visit states recurrently, visualising the pattern of these revisits indicates ordered, periodic, or chaotic behaviour. Defined as a binary matrix R_{ij} where $R_{ij} = 1$ if $\|\theta_i - \theta_j\| < \epsilon$, else 0.
- *Fractal dimensions of attractors* [54]: Chaotic attractors often have fractal geometry occupying a fractional dimension between integer values. Using states θ_i , calculating: $C(r) = \frac{1}{N^2} \sum_{i,j} \mathbf{1}\{\|\theta_i - \theta_j\| < r\}$ denotes the probability two points are within distance r . For small r , $C(r) \sim r^{D_2}$ where D_2 is the correlation dimension. $D_2 \approx 0$ is a fixed point, $D_2 \approx 1$ a limit cycle, and D_2 non-integer > 1 a fractal attractor (signal of chaos).

Please refer to Appendix B for implementation details of the above quantities. Figure 2 compares the stationary distribution for Prisoner’s Dilemma and Matching Pennies, and also between IDQN agents that use Boltzmann and ϵ -greedy exploration. The former relates to replicator dynamics, and as expected the stationary distribution has most mass at the sink points matching the replicator dynamics vector field in Figure 1. In Prisoner’s Dilemma this looks like a tightly concentrated stationary distribution in the top right

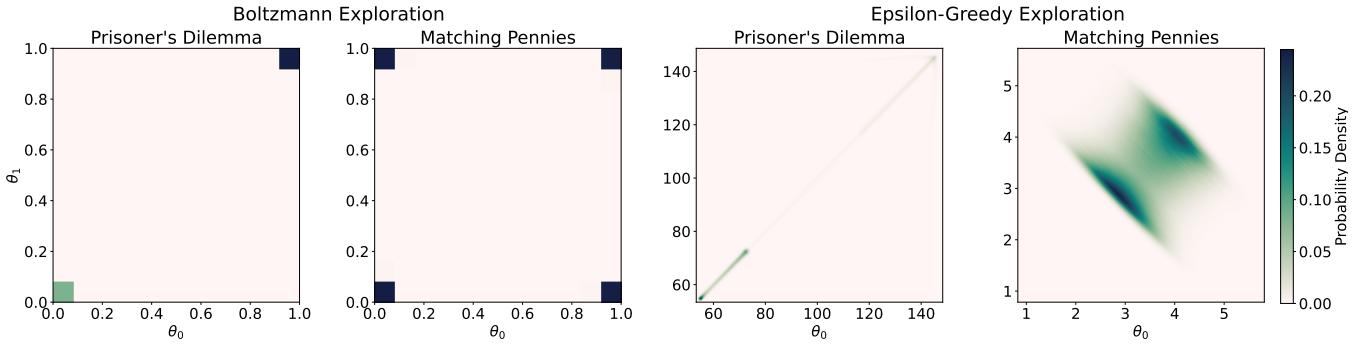


Figure 2: Stationary distributions calculated from realisations of training for two IDQN agents in Prisoner’s Dilemma and Matching Pennies. The two figures on the left are agents using Boltzmann exploration, therefore the parameters can be interpreted as probabilities of taking action 0. On the Boltzmann plots bin counts are intentionally very low so it is clear where the stationary distribution has density. The two figures on the right are agents using ϵ -greedy exploration. Parameters cannot be interpreted as action probabilities which is why their scale can be much larger.

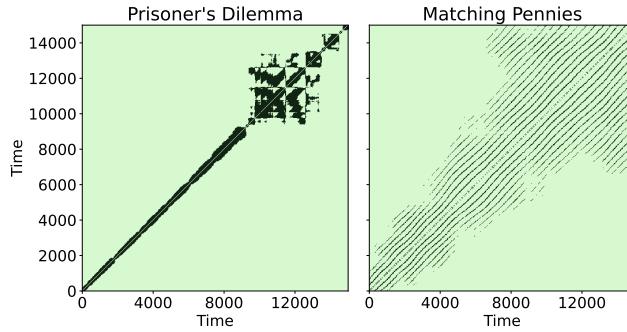


Figure 3: Recurrence plots from a realisation of training two IDQN agents in Prisoner’s Dilemma and Matching Pennies. Figure indicates times when the coupled dynamical system of all agents visits the same area in phase space at the time on the x -axis and y -axis. Intuitively this means locations are marked when $\theta_i \approx \theta_j$ when $i = x, j = y$. The identity band is masked out as when $i = j$ it is always recurrent.

corner, and since we are using Boltzmann exploration this relates to an interpretable action probability. Some mass in the bottom left indicates that for some initial conditions convergence does not match replicator dynamics. In Matching Pennies the distribution covers the four corners of parameter space since the cycling quickly diverges, as seen in Figure 1. For IDQN with stochastic ϵ -greedy exploration strategy the behaviour is much more varied, parameters are no longer bounded between 0 and 1 and thus cannot be interpreted as action probabilities. For Prisoner’s Dilemma the stationary distribution has the most mass around the top right and bottom left vertices, indicating convergence of the parameters to one of two fixed points, even with the stochasticity causing “trails”. However, in Matching Pennies we observe an approximate limit cycle, indicating some quasi-cycle in parameter space.

Further quantitative evidence in Table 1 supports these claims. In Prisoner’s Dilemma $\lambda_{\max} \approx 0$ and a fractal dimension $D_2 \approx 0$

Table 1: Diagnostic quantities for distinguishing stochastic fixed points, limit/quasi-cycles, and chaotic attractors.

Environment	$\ \Sigma\ _F$	λ_{\max}	D_2
Prisoners’ Dilemma	0.102	≈ 0	0.438
Matching Pennies	2.351	0.039	1.154
Stag Hunt	0.553	≈ 0	0.628
Chicken	0.118	≈ 0	0.760
Overcooked	0.956	≈ 0	0.441

indicate a stable fixed point is reached. Conversely, in Matching Pennies, values for λ_{\max} and D_2 indicate cyclical behaviour. Figure 3 qualitatively compares the recurrence plots for Prisoner’s Dilemma and Matching Pennies, the identity band is masked out as when $i = j$ it is always recurrent so adds no extra information. In Prisoner’s Dilemma the most prominent features are the long solid bands that run parallel to the identity line, these indicate segments of the system’s trajectory are running parallel to each for a period of time. This is a strong indicator of determinism and predictability: a deterministic process produces long diagonal lines, whereas a stochastic process exhibits almost none [37]. The pattern here suggests the system follows a regular, repeating path through its state space. If it were strictly periodic, the plot would show evenly spaced diagonal lines, as seen for Matching Pennies. The arc-like structures, typically arising from oscillations whose frequency or amplitude vary over time, indicate quasi-periodicity [17].

So far we have focussed on environments and number of agents that are amenable to plotting. When working with higher agent parameter counts or more agents this quickly becomes impossible. The empirical quantities described do not need visualisation to be understood, but ideally we would like a visual aid for simpler intuition and qualifiable results. Higher-dimensional phase spaces can only be visualised if projected onto a two- or three-dimensional subspace, which inherently leads to information loss (e.g., using Principal Component Analysis [31]). However, recurrence plots yield a two-dimensional representation of a dynamical system [16].

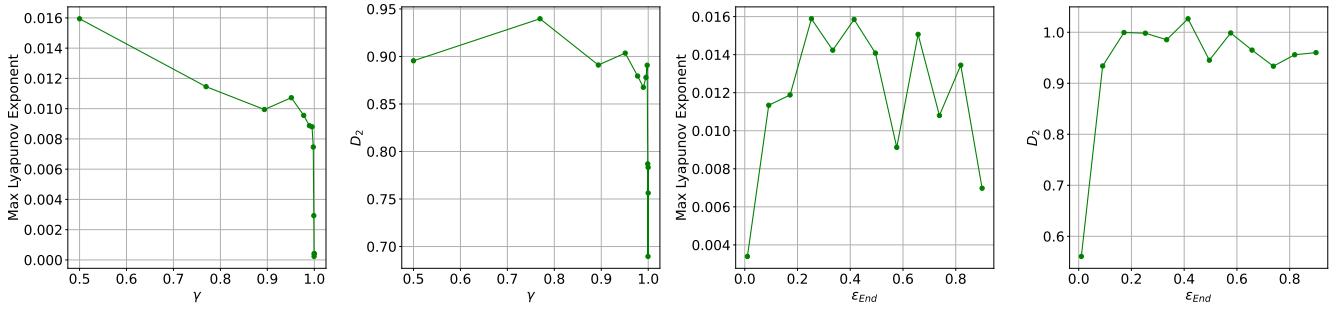


Figure 4: Varying γ , the discounting parameter in IDQN, and ϵ_{End} the end value for ϵ -greedy exploration in IDQN, to understand their impact on the coupled dynamical system attractor via the Max Lyapunov Exponent and fractal dimension D_2 .

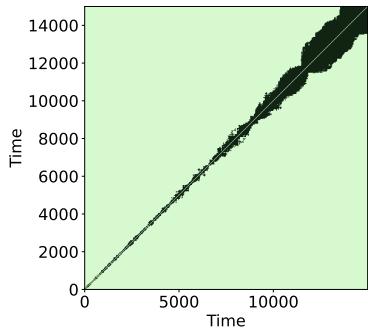


Figure 5: Recurrence plot calculated from a realisation of training two IDQN agents in Overcooked.

Figure 5 highlights a recurrence plot of Overcooked environment, indicating determinism and convergence to a fixed point.

To summarise, analysing the MARL system as an agent-centric coupled dynamical system enables a practitioner to clearly understand the stability and asymptotic performance of an agent’s *true* behaviour. Further, our quantitative and qualitative insights are able to scale to environments and agent numbers that can’t be easily visualised, opening the door for analysis in any environment. When working with classical state-based games, the insights from replicator dynamics generally align with our findings, indicating the usefulness of a paired approach for understanding MARL stability.

5.2 Sensitivity Analysis

So far, we have analysed the phase-space structure of the MARL dynamical system to understand its stability. This serves as a useful post-hoc diagnostic tool, helping practitioners assess the asymptotic performance arising from specific combinations of environments and agent types. Given that we can analyse the phase-space structure, can we also understand how it varies with particular dynamical system inputs? In particular, we seek to characterise the topological changes in the phase space induced by varying hyperparameters or by employing different learning functions g .

To address this, we conduct a sensitivity analysis by sweeping over γ , the discounting value in IDQN, as well as ϵ_{End} . When using ϵ -greedy we instantiate with $\epsilon = 0.9$ and exponentially decay during learning until ϵ_{End} . For each setting, we simulate ensembles of trajectories across multiple random seeds, discard burn-in, and estimate the maximal Lyapunov exponent and fractal dimension D_2 by aggregating post-burn-in policy traces. Plotting these quantities as functions of the swept parameter produces Figure 4. It is clear that increasing γ reduces cycling in Matching Pennies drastically as $\gamma \rightarrow 1$. Setting $\epsilon_{End} = 0$, which removes all exploration stochasticity at the end of training, results in convergence to a fixed point, as indicated by $\lambda_{\max}, D_2 \rightarrow 0$. Interestingly, a small increase tends to produce cyclical behaviour, but increasing further reduces this effect which may indicate convergence toward a smaller-radius limit cycle or quasi-cycle compared to the case with $\epsilon_{End} \approx 0.4$. Figure 6 further supports these claims quantitatively: the far left figure shows a clear limit cycle with determinism, the centre left a quasi-cycle as the spread of diagonal lines indicates higher stochasticity, the centre right has hallmarks of the beginnings of chaos, and finally the far right is almost purely stochastic with very faint structure.

These sensitivity experiments reveal not only the *shape* of the MARL phase space, but also *why* it assumes that shape under different hyperparameter settings. This, in turn, enables us to anticipate how learning stability and long-term behaviour evolve as hyperparameters (including sources of stochasticity) vary, thereby providing a principled connection between hyperparameter tuning and dynamical-systems analysis.

5.3 Control

In the previous sections, through the lens of dynamical systems theory we have gained quantitative understanding of the phase space structure and how this changes given differing parameters. So far, our analysis has treated stability and sensitivity as post-hoc diagnostics. A natural next step is to *close the loop*, using these quantities to control the long-run behaviour of a MARL system. For instance, how must we adjust the underlying update functions g in order to ensure a tighter stationary distribution that has no limit or quasi-cycles? In this section we hypothesise about potential avenues rather than explicit algorithm design. We could define control objectives like the following: dampen cycling in Matching Pennies by pushing λ_{\max} negative and/or pushing D_2 to 0.

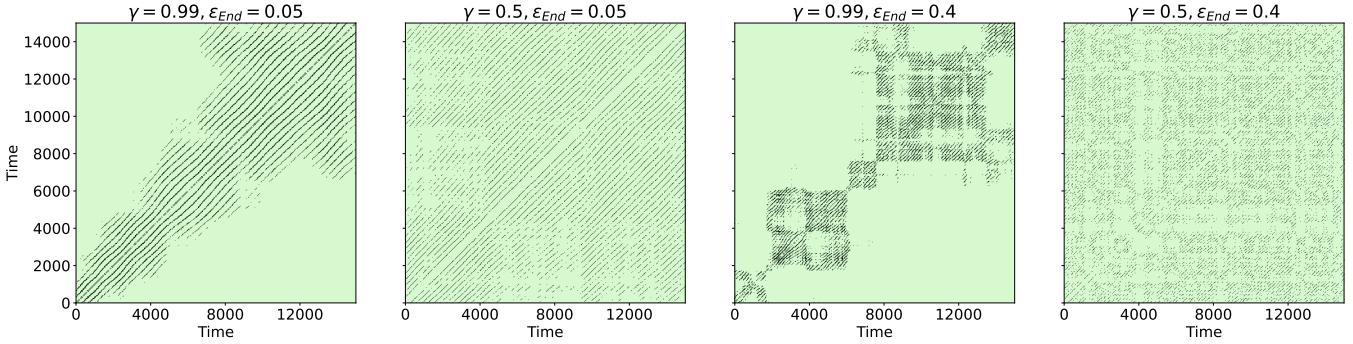


Figure 6: Recurrence plot calculated from a realisation of training two IDQN agents in Matching Pennies. All four combinations of $\gamma = \{0.5, 0.99\}$ and $\epsilon_{End} = \{0.4, 0.0\}$ are presented.

Analytical results on the stability and sensitivity of the system can be leveraged as pseudo-rewards or related quantities to guide agent learning. A straightforward approach is a meta-learning setup in which agent hyperparameters are adapted toward desired stability characteristics driven by dynamical system metrics. This embodies *stability-aware MARL*, where dynamical systems diagnostics inform online adaptation. Such control schemes complement traditional reward-driven learning by ensuring that emergent behaviour operates in regimes that are not only high-reward but also stable, predictable, and interpretable.

6 IMPLICATIONS

A central motivation for our work is the need to understand MARL at the level where decisions are actually made: the learning updates of individual agents. Population-based approaches, such as replicator dynamics, offer elegant closed-form descriptions of adaptation but necessarily smooth out the heterogeneity, stochasticity, and algorithmic detail that drive practical MARL behaviour. By adopting an agent-centric dynamical systems perspective, we recover these missing layers of granularity, enabling direct analysis of the coupled dynamics that govern how agents learn, interact, and adapt. From an analytical viewpoint, this perspective provides a principled framework for quantifying stability and sensitivity in the presence of stochasticity and approximation. Techniques from dynamical systems theory, including invariant distributions, Lyapunov exponents, and recurrence analysis, enable us to characterise whether learning updates converge to equilibrium, exhibit cyclical patterns, or enter chaotic regimes. Stable convergence signifies the emergence of a consistent equilibrium policy or joint strategy, whereas oscillatory or chaotic regimes expose conditions under which learning remains non-stationary or unpredictable. By examining how these regimes change under small perturbations to parameters or environment dynamics, we obtain a direct measure of the system’s robustness and sensitivity. Whilst our analysis is conducted in parameter space, these parameters can be mapped into the environment or reward phase space, thereby enabling a practitioner to understand whether parameter convergence corresponds to convergence toward a desirable attractor in reward or environment space. Crucially, this agent-centric framework scales naturally to high-dimensional agents with function approximation

and non-trivial environments, where population-level or mean-field abstractions become intractable. It thus serves as a bridge between theoretical analysis and empirical practice, providing diagnostic tools applicable to both tabular and deep MARL systems. In doing so, it allows practitioners and theorists alike to reason more systematically about when and why multi-agent learning remains stable, how instabilities emerge, and how algorithmic design choices shape long-term dynamical behaviour.

7 CONCLUSION

In this work, we have advanced an agent-centric, dynamical-systems perspective on MARL. By explicitly treating the learning updates of individual agents as coupled stochastic dynamical systems, we have moved beyond deterministic, population-level abstractions in replicator dynamics and related frameworks, advancing toward an individual-level understanding of decision-making in MARL systems. This shift has enabled us to capture sources of instability and stochasticity inherent in practical MARL, such as exploration, gradient noise, and function approximation. Using tools from modern dynamical systems theory we have demonstrated how long-run MARL behaviour can be rigorously characterised even in high-dimensional environments. Sensitivity analyses further linked these dynamical signatures to hyperparameters, providing explanations for abrupt behavioural shifts.

Looking ahead, this work suggests several promising directions, summarised threefold. Firstly, we have explored a small subset of methods from dynamical systems theory to analyse stability and sensitivity; future work could incorporate additional techniques for a more holistic understanding. Secondly, our approach has been empirical. Stochastic invariant distribution could be represented analytically via a Fokker-Planck equation [42], enabling direct calculations of stability and sensitivity, as demonstrated by Leung et al. [34]. Alternatively, it could be modelled empirically using novel partial differential equation methodologies [35]. Thirdly, some forms of stochasticity in MARL diminish over learning; for instance, random dithering typically decays exponentially. What implications might this have for the attractors of such systems? By recognising learning itself as a coupled dynamical process, we establish a principled foundation for analysing, predicting, and controlling complex behaviours that emerge in multi-agent systems.

ACKNOWLEDGMENTS

James Rudd-Jones is supported by grants from the UK EPSRC-DTP (Award 2868483).

REFERENCES

- [1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems (NeurIPS'21)* (2021).
- [2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).
- [3] Nicolas Anastassacos, Julian Garcia, Stephen Hailes, and Mirco Musolesi. 2021. Cooperation and Reputation Dynamics with Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'21)*.
- [4] Nicolas Anastassacos, Stephen Hailes, and Mirco Musolesi. 2020. Partner Selection for the Emergence of Cooperation in Multi-Agent Systems using Reinforcement Learning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*.
- [5] Robert Axelrod and William D Hamilton. 1981. The evolution of cooperation. *Science* 211, 4489 (1981), 1390–1396.
- [6] Wolfram Barfuss, Jonathan F Donges, and Jürgen Kurths. 2019. Deterministic limit of temporal difference reinforcement learning for stochastic games. *Physical Review E* 99, 4 (2019), 043305.
- [7] Randall D Beer. 1995. A Dynamical Systems Perspective on Agent-Environment Interaction. *Artificial Intelligence* 72, 1-2 (1995), 173–215.
- [8] Martino Bernasconi, Federico Cacciama, Simone Fioravanti, Nicola Gatti, and Francesco Trovò. 2025. The evolutionary dynamics of soft-max policy gradient in multi-agent settings. *Theoretical Computer Science* 1027 (2025), 115011.
- [9] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).
- [10] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary Dynamics of Multi-Agent Learning: A Survey. *Journal of Artificial Intelligence Research* 53 (2015).
- [11] Tilman Börgers and Rajiv Sarin. 1997. Learning through Reinforcement and Replicator Dynamics. *Journal of Economic Theory* 77, 1 (1997), 1–14.
- [12] Albrecht Böttcher and David Wenzel. 2008. The Frobenius norm and the commutator. *Linear algebra and its Applications* 429, 8-9 (2008), 1864–1885.
- [13] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems (NeurIPS'19)*.
- [14] John G Cross. 1973. A Stochastic Learning Model of Economic Behavior. *The Quarterly Journal of Economics* 87, 2 (1973), 239–266.
- [15] Nayana Dasgupta and Mirco Musolesi. 2025. Investigating the impact of direct punishment on the emergence of cooperation in multi-agent reinforcement learning systems. *Autonomous Agents and Multi-Agent Systems* 39, 1 (2025), 1–37.
- [16] Jean-Pierre Eckmann Eckmann, S Ollifson Kamphorst, and David Ruelle. 1995. Recurrence plots of dynamical systems. In *Turbulence, Strange Attractors and Chaos*. World Scientific, 441–445.
- [17] Andrea Facchini and Holger Kantz. 2007. Curved structures in recurrence plots: The role of the sampling time. *Physical Review E* 75, 3 (2007), 036215.
- [18] J Doyne Farmer. 1982. Information dimension and the probabilistic structure of chaos. *Zeitschrift für Naturforschung A* 37, 11 (1982), 1304–1326.
- [19] Luis H Favela. 2020. Dynamical Systems Theory in Cognitive Science and Neuroscience. *Philosophy Compass* 15, 8 (2020), e12695.
- [20] Tobias Galla and J Doyne Farmer. 2013. Complex dynamics in learning complicated games. *Proceedings of the National Academy of Sciences* 110, 4 (2013), 1232–1236.
- [21] Aram Galstyan. 2013. Continuous Strategy Replicator Dynamics for Multi-agent Q-learning. *Autonomous Agents and Multi-agent Systems* 26, 1 (2013), 37–53.
- [22] Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.
- [23] David Goll, Jobst Heitzig, and Wolfram Barfuss. 2024. Deterministic Model of Incremental Multi-Agent Boltzmann Q-Learning: Transient Cooperation, Metastability, and Oscillations. *arXiv preprint arXiv:2501.00160* (2024).
- [24] Brain Team Google Research. 2023. JAX: Autograd and XLA for high-performance machine learning research. <https://github.com/google/jax>
- [25] Peter Grassberger and Itamar Procaccia. 1984. Dimensions and entropies of strange attractors from a fluctuating dynamics approach. *Physica D: Nonlinear Phenomena* 13, 1-2 (1984), 34–54.
- [26] Charles R. Harris, K. Jarrod Millman, Stefan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.
- [27] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*.
- [28] Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duéñez-Guzmán, et al. 2020. Neural Replicator Dynamics: Multiagent Learning via Hedging Policy Gradients. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'20)*.
- [29] Daniel Hennes, Karl Tuyls, and Matthias Rautenberg. 2009. State-Coupled Replicator Dynamics. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*.
- [30] Josef Hofbauer and Karl Sigmund. 1998. *Evolutionary Games and Population Dynamics*. Cambridge University Press.
- [31] Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24, 6 (1933), 417.
- [32] Aamal Hussain, Francesco Belardinelli, and Georgios Piliouras. 2023. Beyond Strict Competition: Approximate Convergence of Multi Agent Q-Learning Dynamics. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI'23)*.
- [33] Joel Z. Leibo, Vincenzo Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-Agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'17)*.
- [34] Chin-wing Leung, Shuyu Hu, and Ho-fung Leung. 2023. The stochastic evolutionary dynamics of softmax policy gradient in games. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS'24)*.
- [35] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2021. Fourier Neural Operator for Parametric Partial Differential Equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*.
- [36] Aleksandr Mikhailovich Lyapunov. 1992. The general problem of the stability of motion. *Internat. J. Control* 55, 3 (1992), 531–534.
- [37] Norbert Marwan, M Carmen Romano, Marco Thiel, and Jürgen Kurths. 2007. Recurrence plots for the analysis of complex systems. *Physics Reports* 438, 5-6 (2007), 237–329.
- [38] John Maynard Smith. 1982. *Evolution and the Theory of Games*. Cambridge University Press.
- [39] Eric Mazumdar, Lillian J. Ratliff, Michael I. Jordan, and S. Shankar Sastry. 2020. Policy-Gradient Algorithms Have No Guarantees of Convergence in Linear Quadratic Games. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'20)*.
- [40] Guido Novati, Hugues Lascombes de Laroussilhe, and Petros Koumoutsakos. 2021. Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence* 3, 1 (2021), 87–96.
- [41] Martin A Nowak. 2006. Five Rules for the Evolution of Cooperation. *Science* 314, 5805 (2006), 1560–1563.
- [42] Hannes Risken. 1989. Fokker-Planck equation. In *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer, 63–95.
- [43] James Rudd-Jones, Mirco Musolesi, and María Pérez-Ortiz. 2025. Multi-Agent Reinforcement Learning Simulation for Environmental Policy Synthesis. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'25)*.
- [44] James Rudd-Jones, Fiona Thendean, and María Pérez-Ortiz. 2025. Crafting desirable climate trajectories with reinforcement learning explored socio-environmental simulations. *Environmental Data Science* 4 (2025), e41.
- [45] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson Juto, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schroeder de Witt, et al. 2024. JaxMARL: Multi-Agent Reinforcement Learning Environments and Algorithms in JAX. *Advances in Neural Information Processing Systems (NeurIPS'24)* (2024).
- [46] James BT Sanders, J Doyne Farmer, and Tobias Galla. 2018. The prevalence of chaotic dynamics in games with many players. *Scientific Reports* 8, 1 (2018), 4902.
- [47] Yuzuru Sato, Eizo Akiyama, and J Doyne Farmer. 2002. Chaos in learning a simple two-person game. *Proceedings of the National Academy of Sciences* 99, 7 (2002), 4748–4751.
- [48] Jeffrey D Scargle. 1989. An introduction to chaotic and random time series analysis. *International Journal of Imaging Systems and Technology* 1, 2 (1989), 243–253.
- [49] Heinz Georg Schuster and Wolfram Just. 2005. *Deterministic Chaos: An Introduction*. Wiley.

- [50] Steven H Strogatz. 2024. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*. Chapman and Hall/CRC.
- [51] Pol Suárez, Francisco Alcántara-Ávila, Arnau Miró, Jean Rabault, Bernat Font, Oriol Lehmkühl, and Ricardo Vinuesa. 2025. Active Flow Control for Drag Reduction Through Multi-agent Reinforcement Learning on a Turbulent Cylinder at $Re_D = 3900$. *Flow, Turbulence and Combustion* 114, 3 (2025), 1–25.
- [52] George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. 2012. Detecting causality in complex ecosystems. *Science* 338, 6106 (2012), 496–500.
- [53] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLOS One* 12, 4 (2017), e0172395.
- [54] James Theiler. 1990. Estimating fractal dimension. *Journal of the Optical Society of America A* 7, 6 (1990), 1055–1073.
- [55] Karl Tuyls, Pieter Jan’T Hoorn, and Bram Vanschoenwinkel. 2006. An Evolutionary Dynamical Analysis of Multi-Agent Learning in Iterated Games. *Autonomous Agents and Multi-Agent Systems* 12, 1 (2006), 115–153.
- [56] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.
- [57] Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards*. PhD Thesis. King’s College, University of Cambridge, Cambridge, United Kingdom.
- [58] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML’18)*, Vol. 80. 5571–5580.
- [59] Yaodong Yang and Jun Wang. 2020. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583* (2020).
- [60] Tianyu Zhang, Andrew Williams, Phillip Wozny, Kai-Hendrik Cohrs, Koen Ponse, Marco Jiralerspong, Soham Phade, Sunil Srinivasa, Lu Li, Yang Zhang, Prateek Gupta, Erman Acar, Irina Rish, Yoshua Bengio, and Stephan Zheng. 2025. AI for Global Climate Cooperation: Modeling Global Climate Negotiations, Agreements, and Long-Term Cooperation in RICE-N. In *Proceedings of the 42nd International Conference on Machine Learning (ICML’25)*.

A ENVIRONMENTS

We first consider a set of four stateless matrix games, which serve as canonical benchmarks for analysing fundamental multi-agent interaction dynamics. These environments consist of single-step interactions in which agents select actions simultaneously and receive rewards according to fixed payoff matrices. As there is no temporal component or state transition, they provide a controlled setting for examining coordination, competition, and equilibrium behaviour under simplified yet illustrative conditions. Exact implementations of the four environment’s payoff matrices are in the tables below.

Table 1: Prisoner’s Dilemma.

	C	D
C	1,1	5,0
D	0,5	3,3

Table 2: Matching Pennies.

	H	T
H	1,-1	-1,1
T	-1,1	1,-1

Table 3: Stag Hunt.

	S	H
S	4,4	0,0
H	0,0	3,3

Table 4: Chicken.

	A	B
A	-1,-1	4,0
B	0,4	2,2

We also use the Overcooked environment from the JaxMARL benchmark suite [45], a cooperative multi-agent cooking task adapted from the original Overcooked-AI environment [13]. In this environment, two agents coordinate in a shared kitchen to prepare and deliver soups, requiring precise spatial and temporal coordination.

B DYNAMICAL SYSTEMS ANALYSIS TECHNIQUES: ADDITIONAL DETAILS

This appendix details the numerical and statistical methods used to analyse the emergent dynamics of the interacting agents. All calculations are implemented in JAX [24] or NumPy [26] using vectorised operations for reproducibility and efficiency.

B.1 Stationary Distribution Estimation

The stationary distributions of joint policy trajectories that represent the coupled stochastic dynamical system are estimated empirically from long-run samples of the agents’ learning dynamics. For each game (e.g., Prisoner’s Dilemma, Matching Pennies), the coupled learning system is simulated for n_{steps} iterations over n_{runs} random seeds. A burn-in phase of n_{burn} steps is discarded to eliminate transient behaviour.

Denoting by $\theta_h = (\theta_h^1, \theta_h^2)$ the parameter vector for two agents at update time h , the stationary samples are obtained as

$$\mathcal{S} = \bigcup_{i=1}^{n_{\text{runs}}} \{\theta_h^{(i)} : h > n_{\text{burn}}\}. \quad (1)$$

If θ^1, θ^2 are scalars, and there are two agents, then they are combined into a two-dimensional array and used to form a normalised empirical density $\hat{p}(\theta^1, \theta^2)$ visualised as a 2D histogram. If θ^1, θ^2 are vectors, or there are more than two agents, then dimensions must be reduced to enable a visualisation. There are many techniques for doing this each with their pros and cons, and with inherent information loss. This serves as an approximation of the invariant measure of the stochastic learning process.

B.2 Covariance Analysis and Frobenius Norm

For a multivariate trajectory $\theta = (\theta^1, \theta^2, \dots, \theta^n)$, the covariance matrix

$$\Sigma = \frac{1}{H-1} (\theta - \bar{\theta})^\top (\theta - \bar{\theta}) \quad (2)$$

captures the linear dependencies among parameters, where H is the total number of update steps. To reduce this to a one-dimensional

value for paper results we can take any of the following approaches: total variance (i.e. the Trace), average variance (of the Trace), or the Frobenius norm [12], amongst many other options. We focus on the last option, the Frobenius norm $\|\Sigma\|_F$ computed as:

$$\|\Sigma\|_F = \sqrt{\sum_{i,j} \Sigma_{ij}^2}. \quad (3)$$

This provides a scalar summary of the overall magnitude of variability and coupling strength among all components. Unlike the other two options, the Frobenius norm accounts for covariance (or cross) terms rather than just the diagonal terms with the Trace. This norm is a compact descriptor of the complexity of the stationary fluctuations [12].

B.3 (Maximum) Lyapunov Exponent Estimation

To quantify local sensitivity to initial conditions, the maximal Lyapunov exponent λ_{\max} is estimated using a nearest-neighbour divergence method applied to a multivariate trajectory $\theta = [\theta_h]_{h=1}^H \in \mathbb{R}^{H \times \Theta}$, where Θ is the joint dimensionality of agent parameters. For each point θ_i , its nearest neighbour $\theta_{j(i)}$ is identified subject to a Theiler [54] window (i.e. ignoring temporal neighbours within $\pm w$ indices). The Euclidean distance between their forward evolutions at lag z is tracked as:

$$d_i(z) = \|\theta_{i+z} - \theta_{j(i)+z}\|_2, \quad (4)$$

and the average logarithmic divergence ($\log d(z)$) is fitted linearly over a time window $z \in [z_{\min}, z_{\max}]$. The slope of this line provides an estimate of λ_{\max} :

$$\lambda_{\max} \approx \frac{d}{dt} \langle \log d(z) \rangle. \quad (5)$$

A positive value indicates exponential divergence and thus chaotic behaviour in the coupled learning dynamics.

B.4 Recurrence and Correlation Plots

Recurrence plots visualise the structure of revisitations in phase space by thresholding pairwise distances between states:

$$R_{ij} = \mathbf{1}\{\|\theta_i - \theta_j\|_2 \leq \varepsilon\}. \quad (6)$$

The threshold ε is chosen such that a desired recurrence rate (e.g., 8%) is achieved. The resulting binary matrix R encodes temporal proximity patterns; diagonal lines correspond to epochs of predictability, whereas scattered points indicate stochastic or chaotic transitions. For more details on reading recurrence plots please refer to the following resources [16, 17, 37].

B.5 Correlation Dimension (Fractal Dimension)

The correlation (fractal) dimension D_2 is computed using the Grassberger–Procaccia algorithm [25]. For a scalar or low-dimensional time series θ_h , delay embedding with dimension m and lag τ constructs vectors

$$\Theta_h = [\theta_h, \theta_{h+\tau}, \dots, \theta_{h+(m-1)\tau}] \in \mathbb{R}^m. \quad (7)$$

Pairwise distances $d_{ij} = \|\Theta_i - \Theta_j\|_2$ are evaluated, and the correlation sum

$$C(r) = \frac{2}{N(N-1)} \sum_{i < j} \mathbf{1}\{d_{ij} < r\} \quad (8)$$

is estimated over a logarithmic range of radii r , where N is the effective number of reconstructed state vectors available after embedding. In the scaling region where $C(r) \propto r^{D_2}$, a linear regression of $\log C(r)$ vs. $\log r$ yields the correlation dimension D_2 . This measures the fractal geometry of the attractor underlying the agents' stationary dynamics.

B.6 Summary of Computed Metrics

In summary, for each analysed trajectory the following quantities are reported:

- Stationary distribution $\hat{p}(\theta)$: empirical invariant measure.
- $\|\Sigma\|_F$: Frobenius norm of the covariance, capturing overall variability.
- λ_{\max} : largest Lyapunov exponent, measuring chaotic divergence.
- Recurrence plots: graphical tools used to visualize the times at which a dynamical system returns to states similar to previous ones, as defined by a chosen error tolerance.
- D_2 : correlation (fractal) dimension of the reconstructed attractor.

Together, these diagnostics provide a comprehensive characterisation of the dynamical complexity, stationarity, and stability properties of the interacting reinforcement learning agents.