scientific reports

Check for updates

OPEN

Intrinsic fluctuations of reinforcement learning promote cooperation

Wolfram Barfuss[1,4] & Janusz M. Meylahn[2,3,4]✉

In this work, we ask for and answer what makes classical temporal-difference reinforcement learning with ϵ-greedy strategies cooperative. Cooperating in social dilemma situations is vital for animals, humans, and machines. While evolutionary theory revealed a range of mechanisms promoting cooperation, the conditions under which agents learn to cooperate are contested. Here, we demonstrate which and how individual elements of the multi-agent learning setting lead to cooperation. We use the iterated Prisoner's dilemma with one-period memory as a testbed. Each of the two learning agents learns a strategy that conditions the following action choices on both agents' action choices of the last round. We find that next to a high caring for future rewards, a low exploration rate, and a small learning rate, it is primarily intrinsic stochastic fluctuations of the reinforcement learning process which double the final rate of cooperation to up to 80%. Thus, inherent noise is not a necessary evil of the iterative learning process. It is a critical asset for the learning of cooperation. However, we also point out the trade-off between a high likelihood of cooperative behavior and achieving this in a reasonable amount of time. Our findings are relevant for purposefully designing cooperative algorithms and regulating undesired collusive effects.

Problems of cooperation are ubiquitous and essential, for biological phenomena, as in the evolution of cooperation under natural selection, for human behavior, such as in cartel pricing or traffic, and increasingly so for intelligent machines with automated trading and self-driving cars[1–3]. In social dilemmas, individual incentives and collective welfare are not aligned. Individuals profit from exploiting others or fear being exploited by others, while at the same time, the collective welfare is maximized if all choose to cooperate[4].

Understanding the conditions under which self-learning agents learn to cooperate spontaneously—without explicit intent to do so—is critical for three reasons: (1) It provides an alternative route to the emergence of (human and animal) cooperation when an evolutionary explanation is unlikely. (2) It guides the design of intelligent self-learning algorithms, which are supposed to be cooperative. (3) It provides policymakers and regulators the necessary background to design novel anti-trust legislation against undesirable collusion, e.g., in algorithmic pricing situations, where not doing so could lead to significant loss of consumer welfare[5].

While evolutionary theory revealed a range of mechanisms that promote cooperative behavior, from direct and indirect reciprocity to spatial and network effects[6–10], the conditions under which individually learning agents learn to cooperate are contested. Some works suggest that independent reinforcement learning agents are capable of spontaneously cooperating without explicit intent to do so[11–20]. Other works argue that the emergence of cooperation from independent learning agents is unlikely[21–25], and therefore specific algorithmic features are required to promote cooperation[26–38]. As such, reinforcement learning variants called *aspiration learning*, which go back to a seminal work in psychology from Bush and Mosteller[39], have been extensively investigated in social dilemmas. Whether two co-players learn to cooperate depends on the (dynamics of the) aspiration level[40–42]. This finding has been confirmed and extended to spatial or networked social dilemmas[43–46]. Aspiration learning has also been found to explain human play in behavioral experiments well[13,14]. However, comparably little is known about whether, when, and how cooperative behavior spontaneously emerges from the reinforcement learning variants called *temporal-difference learning*, which are extensively used in machine learning applications and is the dominant model used to explain neuroscientific experiments[47].

In this work, we ask when and how cooperative behavior spontaneously emerges from temporal-difference learning with ϵ-greedy strategy functions. This question is motivated by recent work on algorithmic collusion[48]

[1]Tübingen AI Center, University of Tübingen, Tübingen, Germany. [2]Department of Applied Mathematics, University of Twente, Enschede, The Netherlands. [3]Dutch Institute of Emergent Phenomena, University of Amsterdam, Amsterdam, The Netherlands. [4]These authors contributed equally: Wolfram Barfuss and Janusz M. Meylahn. ✉email: j.m.meylahn@utwente.nl

and the fact that ϵ-greedy strategies are frequently used in machine learning[49]. The problem is that reinforcement learning is typically highly stochastic and data-inefficient, making it challenging to understand which features are decisive for learning cooperation. We solve this problem by dissecting the reinforcement learning processes into three parts using multiple mathematical techniques.

First, we consider the *stability* of strategies under reinforcement learning. We analytically derive when strategies are stable given how much the agents care for future rewards and explore the environment. Only one out of the three possible, stable strategies supports cooperation robustly.

Second, we consider the *learnability* of this equilibrium. We use deterministic strategy-average learning dynamics to compute the size of the basins-of-attraction given the agents' learning rate. We find a maximum of approx. 40–50% of robust cooperation.

Third and last, we consider the *stochasticity* of the learning process. We simulate a stochastic batch-learning algorithm and find that the cooperative equilibrium steadily increases to 80-100%. Thus, a significant fraction of trials reaching the cooperative equilibrium must be due to the inherent fluctuations of the reinforcement learning process.

Learning algorithm and environment. We consider the generalized and advantageous temporal-difference reinforcement learning algorithm *Expected SARSA*[49–51] with ϵ-greedy exploration. At each time step t, agent $i \in \{1, 2\}$ chooses between two possible actions, $a \in \mathcal{A}^1 = \mathcal{A}^2 = \{\mathsf{c}, \mathsf{d}\}$, which represent a cooperative or a defective act. Given the joint action $\boldsymbol{a} = \{a^1, a^2\}$ and the current state of the environment $s \in \mathcal{S}$, each agent receives a payoff or reward $r^i(s, \boldsymbol{a})$ and the environment transitions to a new state $s' \in \mathcal{S}$ with probability $p(s'|\boldsymbol{a}, s)$. Agent i chooses action a with frequency $x_t^i(a|s)$ which depends on the current environmental state $s \in \mathcal{S}$.

Agents derive these frequencies $x_t^i(a|s)$ from their state-action values $q_t^i(s, a)$ according to the ϵ-greedy exploration scheme. Each agent selects the action with the largest state-action value with probability $1 - \epsilon$, and with probability ϵ, it selects an action uniformly at random. For the two-action case,

$$x_t^i(\mathsf{c}|s) = \begin{cases} 1 - \epsilon/2 & \text{if } q_t^i(\mathsf{c}, a) > q_t^i(\mathsf{d}, a) \\ \epsilon/2 & \text{otherwise} \end{cases}. \tag{1}$$

$x_t^i(\mathsf{d}|s)$ is defined analogously. The parameter ϵ regulates the exploration-exploitation trade-off.

The state-action values are updated after each time step as,

$$q_{t+1}^i(s_t, a_t) = (1 - \alpha) q_t^i(s_t, a_t) + \alpha \left[r_t^i + \delta \sum_a x_t^i(a|s_{t+1}) q_t^i(s_{t+1}, a) \right], \tag{2}$$

where the parameter $\alpha \in [0, 1]$ denotes the learning rate, $r_t^i = r^i(s_t, \boldsymbol{a}_t)$ denotes the rewards agent i receives at time step t and $\delta \in [0, 1)$ denotes the agents' discount factor, regulating how much they care for future rewards. For simplicity, we consider homogeneous and constant parameters α, ϵ, δ during the learning process.

The environment we study is the iterated Prisoner's Dilemma. It is perhaps the most iconic and straightforward model system to investigate the preconditions for cooperative behavior, with an established body of research in fields as diverse as political science and evolutionary biology[52]. Because of its simplicity in carving out the tensions between individual incentives and collective welfare, we use it here as a model system to highlight an effect that is, therefore, likely to exist in other larger systems that retain similar tensions between individual incentives and collective welfare. Specifically, we use reward matrices given by,

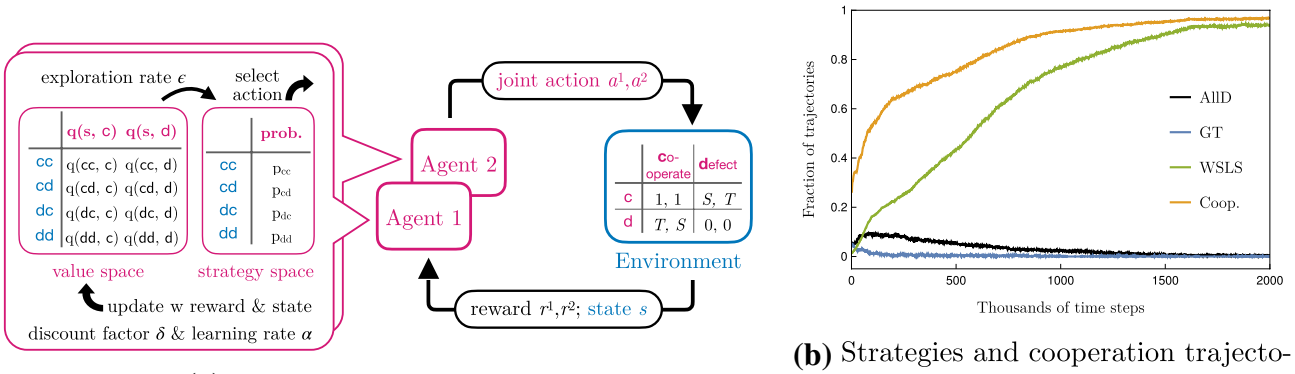
		Agent 2	
		c	d
Agent 1	c	$1, 1$	S, T
	d	T, S	$0, 0$

with $T > 1 > 0 > S$. The rewards for each combination of actions are written in the cells of the matrix. Each cell's first (second) element denotes the payoff for agent 1 (2). With $T > 1$ and $S < 0$, each agent prefers defection over cooperation, regardless of what the other agent is doing. The dilemma is that both agents could achieve a higher reward if both cooperate.

However, when the game is repeated for multiple rounds, agents can condition their frequencies of choosing actions on the actions of past rounds, and mutual defection is no longer inevitable. A famous example is the Tit-for-Tat strategy[6], in which you cooperate if your co-player cooperated, and you defect if your co-player defected in the last round.

We are interested in how two reinforcement learning agents endogenously learn such memory-1 strategies. Therefore, we embed the stateless Prisoner's Dilemma game into an environment where the current environmental state $s_t = a_{t-1}^1 a_{t-1}^2$ signals the actions of the last round. Thus, the state set reads $\mathcal{S} = \{\mathsf{cc}, \mathsf{cd}, \mathsf{dc}, \mathsf{dd}\}$. Fig. 1a illustrates our setting.

Recent work has shown that only three strategy pairs are an equilibrium for ϵ-greedy temporal-difference learning with one-period memory in the iterated Prisoner's Dilemma in the small exploration rate limit[53,54]. Interestingly, the Tit-for-Tat strategy is not an equilibrium. The three strategies are All-Defect (AllD), Grim-Trigger (GT), and Win-Stay, Lose-Shift (WSLS). In AllD, both agents defect regardless of the previous period. In GT, agents play AllD except when both players cooperated in the last period, which is answered by cooperation. And



(a) Agent-Environment Interface

(b) Strategies and cooperation trajectories

Figure 1. Overview. (**a**) Model sketch. (**b**) Fraction of 1000 samples from random initial state-action values in each of the three equilibria (All-Defect (AllD) in black, Grim-Trigger (GT) in blue, and Win-Stay, Lose-Shift (WSLS) in green) as a function of time when using ϵ-greedy temporal-difference learning with $\epsilon = 0.01, \delta = 0.98\, \alpha = 0.1$. The fraction of times that both Q-learners cooperated in the last thousand time periods averaged over 1000 sample trajectories in light gray. Environment parameters are $T = 1.5$ and $S = -0.2$.

in WSLS, agents play GT, except that cooperation also follows a previous period of both players defecting. Only the WSLS equilibrium leads to robust cooperation. Under the GT strategy, both agents keep cooperating, given that they have cooperated in the last round. But erroneous or exploration moves make it more likely to switch from full cooperation to full defection than switching from full defection back to full cooperation.

Figure 1b shows the running average of the fraction of times both players cooperated (yellow) as a function of time, beginning from 1000 random initial state-action values. The trajectories of the fraction of the three stable strategies are likewise shown. In the end, the agents cooperate almost always. This shows that temporal-difference reinforcement learning can spontaneously learn to cooperate. However, it leaves open the question of why the agents learn to cooperate and which features of the learning algorithm are decisive for its ability to cooperate. In particular, the effects of the exploration and learning rates, ϵ and α, and the intrinsic stochasticity of the reinforcement learning process remain unclear.

Dissecting reinforcement learning

To shed light on the questions raised above, we will dissect the reinforcement learning processes into three parts. First, we consider the *stability* of strategy pairs under reinforcement learning, considering agents' discount factor δ and the exploration rate ϵ. Second, we analyze the *learnability* of this equilibrium, taking into account the learning rate α. Third and last, we consider the *stochasticity* of the learning process by introducing a batch-learning variant of our temporal-difference reinforcement algorithm with a batch size parameter K.

Stability. This section shows how the exploration rate ϵ affects the stability landscape. We analytically derive when strategy pairs are stable under the reinforcement learning update outside the small exploration limit. To do so, we refine the mathematical technique of *Mutual Best-Response Networks*[54]. With this method, we construct a directed network where the nodes represent the strategy pairs, and the edges represent a best-response relationship (see "Methods").

We find that AllD is always a solution. The condition for having WSLS as a solution is

$$\delta > \frac{2(T-1) + \epsilon(1-S-T)}{2(1-\epsilon)^2}, \tag{3}$$

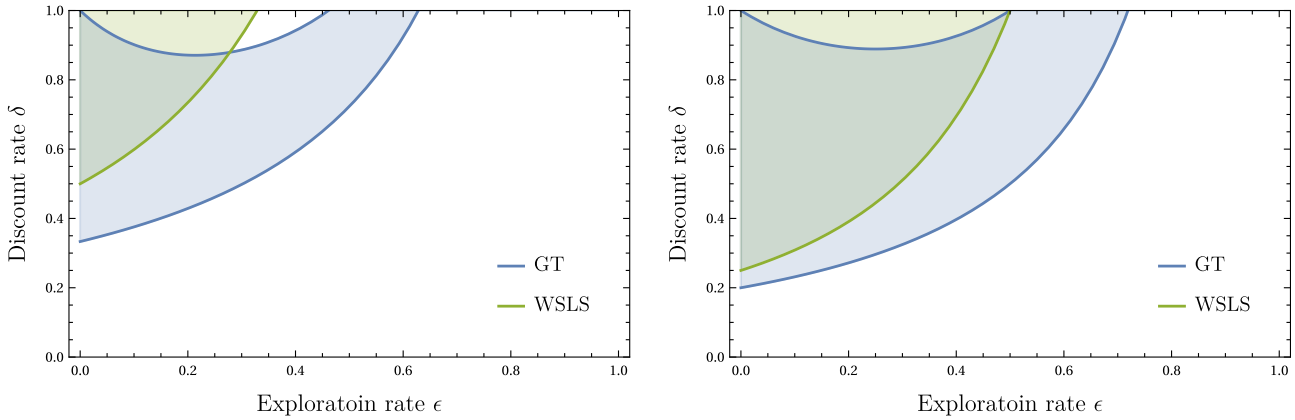
while the condition for having GT as a solution is

$$\frac{2S + \epsilon(1-S-T)}{(1-\epsilon)[(2-\epsilon)S - \epsilon T]} > \delta > \frac{2(T-1) + \epsilon(1-S-T)}{(1-\epsilon)(2T - \epsilon[S+T])}. \tag{4}$$

The condition for WSLS (Eq. 3) is always greater than the lower bound condition for GT (Eq. 4). This means the robustly cooperative WSLS always requires a higher discount factor than the GT strategy equilibrium.

Figure 2 illustrates when the three equilibrium strategy pairs, AllD, GT, and WSLS, are stable, given the discount factor δ and the exploration rate ϵ. The cooperative WSLS strategy pair is stable when δ is high, and ϵ is small. The GT strategy pair also requires a high δ and a small ϵ to become stable, yet, with less extreme parameter values. Interestingly, for large discount factors δ, our theory predicts the GT equilibrium to lose stability for exploration rates ϵ between 0.0 and around 0.4 for the values chosen for T and S in Fig. 2. The AllD equilibrium is always stable.

Learnability. In this section, we show how the learning rate α affects the learnability of the robustly cooperative WSLS equilibrium. With learnability, we mean the likelihood that the learning process reaches an equilibrium, i.e., the size of the state-action-value space from which the WSLS is learned. Following the edges along



(a) Environment parameters $S = -0.2, T = 1.5$ **(b)** Environment parameters $S = -0.25, T = 1.25$

Figure 2. Stability parameter space. Phase diagrams show which strategy equilibrium solutions are possible. The All-Defect (AllD) solution is possible everywhere. The Grim-Trigger (GT) solution is possible in the blue region, and the Win-Stay, Lose-Shift (WSLS) strategy is possible in the green region.

the *Mutual Best-Response Networks* represent the deterministic dynamics of a reinforcement learning algorithm, learning with perfect information and a learning rate of $\alpha = 1$[54]. The maximum learnability of the WSLS equilibrium under these dynamics, as given by its basin of attraction, over all possible parameters (T, S, ϵ and δ) is 0.015625 (see "Methods"). Cooperation is thus not very likely in this case.

To investigate the learning dynamics for a learning rate of $\alpha < 1$, we refine the mathematical technique of deterministic reinforcement learning dynamics[55–57]. This method considers the mean-field of an infinite memory batch to construct idealized learning updates precisely in the direction of the strategy-average temporal-difference error. Previous work investigated the dynamics in strategy space. We formulate the dynamics in state-action-value space to account for ϵ-greedy policies (see "Methods").

To estimate the size of the state-action-value space from which the agents learn WSLS, we let them start from 250 random initial state-action values (Fig. 3). Thus, with deterministic dynamics, the only randomness introduced in this section results from the initial state-action-value conditions.

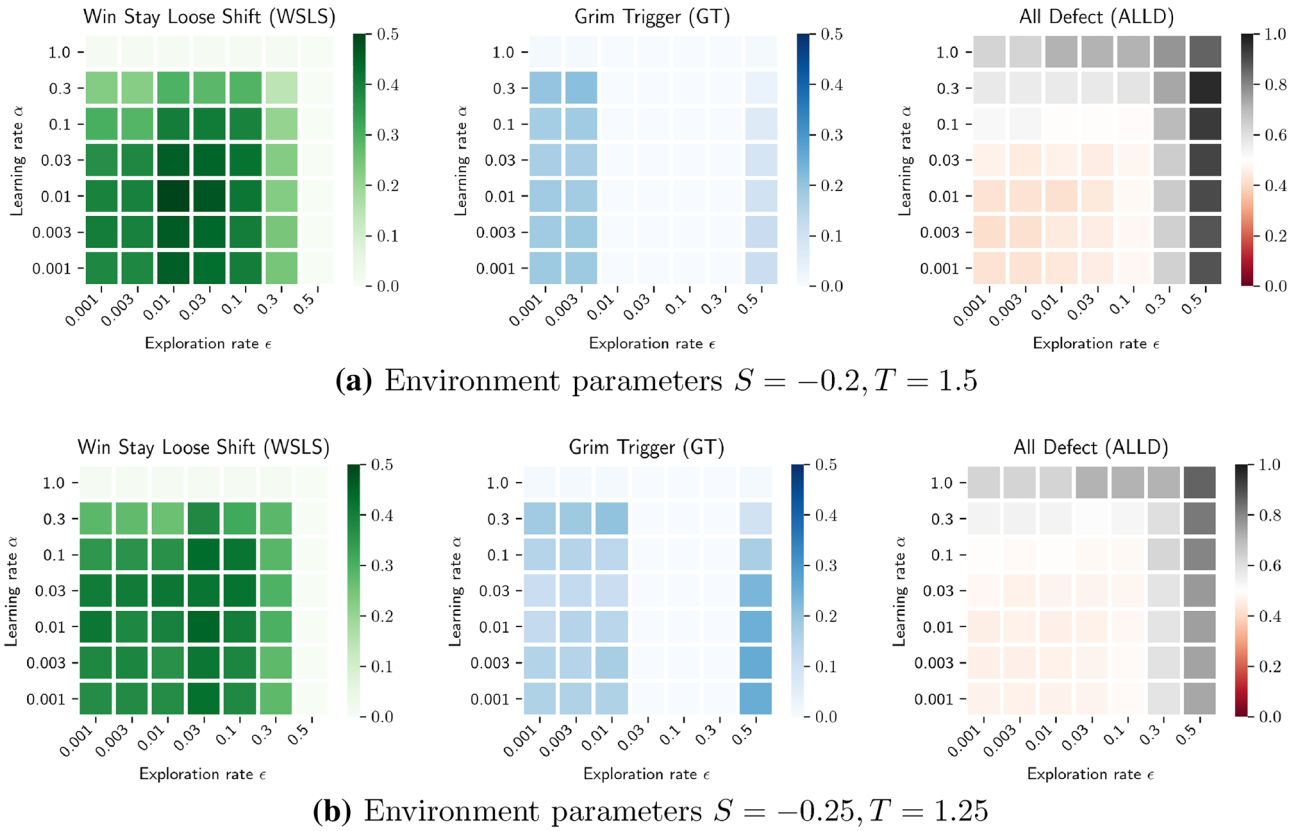
Overall, we find that the robustly cooperative WSLS equilibrium is learned from a maximum of 40–50% of the state-action-value space, given the learning rate α and the exploration rate ϵ are not too large (Fig. 3, left plots in green). Values below 0.1 for each parameter are sufficient, independent of the environmental parameters investigated.

Furthermore, the deterministic learning dynamics confirm the non-trivial predictions of Fig. 2 that the GT strategy is unstable for intermediate values of the exploration rates ϵ and large discount factors δ (Fig. 3, center plots in blue). Fig. 3 also confirms the prediction that at exploration rates ϵ close to zero, the GT stability boundary is steeper for the environment with $S = -0.2$ and $T = 1.5$ than for the environment with $S = -0.25$ and $T = 1.25$ (Fig. 2). Grim Trigger is learned for exploration rate $\epsilon = 0.01$ in the latter environment, but not in the former (Fig. 3).

Lastly, we find that outside the square of learning rate $\alpha = 0.1$ and exploration rate $\epsilon = 0.1$, more than half of the state-action-value space leads to the AllD equilibrium, independent of the environments investigated (Fig. 3, right plots). Inside this square, the AllD equilibrium no longer dominates the state-action-value space. Less than half of it leads to complete defection.

Stochasticity. In this section, we show that intrinsic fluctuations of the typical online reinforcement learning process significantly improve the learnability of the robustly cooperative WSLS equilibrium. To be able to interpolate between fully online learning and deterministic learning, we refine the temporal-difference reinforcement learning algorithm (Eq. 2) with a memory batch of size $K \in \mathbb{N}$. Batch learning is a prominent algorithmic refinement because of its efficient use of collected data and the improved stability of the learning process when used with function approximation[58–60]. The agents store experiences (observed states, rewards, next states) of K time steps inside the memory batch and use their averages to get a more robust learning update of the state-action values (see "Methods"). The batch size K allows us to interpolate between the fully online learning algorithm (Eq. 2) for $K = 1$ and the deterministic learning dynamics for $K = \infty$. We simulate the stochastic batch-learning algorithm for an exemplary set of parameters to showcase the effect intrinsic fluctuations can have on the learning of cooperation. Our goal here is not to optimize this set of parameters, as a thorough theoretical treatment of the resulting stochastic process is beyond the scope of this work.

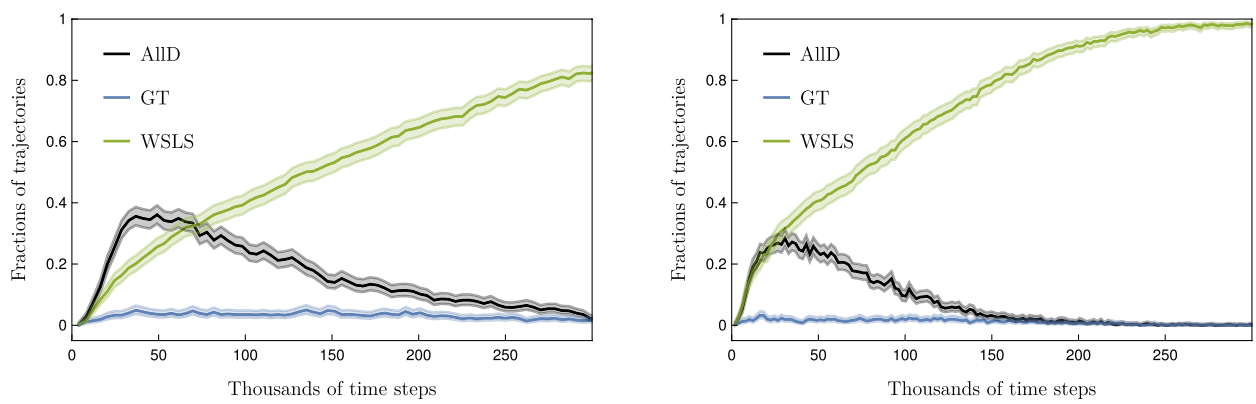
We find that intrinsic fluctuations significantly increase the level of the robustly cooperative WSLS equilibrium compared to the deterministic learning dynamics. At the same time, the batch learning agents require an order of magnitude fewer time steps to reach such high levels of cooperation than the batch-less online algorithm (Fig. 4). We observe that the fraction of the cooperative equilibrium steadily increases to over 80%, about twice the level reached with the deterministic learning dynamics in Fig 3. We hypothesize that the stability of the equilibria under noisy dynamics is a crucial factor. From Fig. 4, we see that the percentage of trajectories in the



(a) Environment parameters $S = -0.2, T = 1.5$

(b) Environment parameters $S = -0.25, T = 1.25$

Figure 3. Learnability parameter spaces. Colors indicate which fraction of 250 random initial state-action values converges to the respective equilibria (the robustly cooperative WSLS in green on the left, GT in blue in the center, and AllD on the right) for a distinct parameter combination. Each plot portrays the parameter space spanned by the learning rate α versus the exploration rate ϵ. The discount factor $\delta = 0.99$.



(a) Environment parameters $S = -0.2, T = 1.5$ **(b)** Environment parameters $S = -0.25, T = 1.25$

Figure 4. Stochasticity of batch learning. Fraction of 1000 from random initial state-action values in each of the three equilibria versus time steps for the sample-batch learning algorithm with $\epsilon = 0.1, \delta = 0.99, \alpha = 0.3$ and $K = 4096$ for (**a**) and $K = 2048$ for (**b**). The shaded region shows a 95% confidence interval calculated using the Wilson Score[62].

WSLS state increases. In contrast, for the two other strategies, the percentage first increases and then decreases (with occasional upward fluctuations). This suggests that the WSLS strategy pair is more stable than the other two, given this choice of environment and algorithm parameters.

Interestingly, Fig. 4 also shows that the high level of robust cooperation is reached on a time scale that is an order of magnitude shorter than that of the purely online algorithm (Eq. 2). Whereas the online algorithm takes in the order of 10^6 time steps, the batch learning algorithm only requires the order of 10^5 time steps to reach high cooperation levels. This is remarkable because, in the batch-learning simulation, we purposefully restrict the agents to update their strategies only after completing an entire batch. In practice, learning a strategy using

a memory batch and learning a model of the environment will be more intertwined[61], offering additional efficiency gains. This suggests the existence of a sweet spot between high levels of final cooperation and the time agents require to learn them.

To check the robustness of our results, we repeat the simulations of Fig. 4 for different combinations of the algorithm parameters K, α and ϵ. The values we investigate are an increasing batch size $K \in \{1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000\}$, for learning and exploration rate around the critical values 0.1 which are decisive for high levels of robust cooperation in the deterministic approximation (Fig. 3): learning rate $\alpha \in \{0.003, 0.006, 0.1, 0.2, 0.3\}$ and exploration rate $\epsilon \in \{0.003, 0.006, 0.1, 0.2, 0.3\}$. We record the fraction of trajectories (based on 1000 samples) in the WSLS strategy pair at time 2×10^6. To get an indication of the speed at which the WSLS strategy pair is learned, we record the time at which the fraction of trajectories for a given set of parameters reaches 0.4.

In Fig. 5, we show the results for both environments around the critical parameter space point $(\epsilon, \alpha) = (0.1, 0.1)$. The rest of the results are presented in the *Supplementary Information*. We find that the fraction of trajectories in the WSLS strategy pair reaches values close to one for a large proportion of the investigated parameters. The results are thus robust to changes in the algorithm parameters.

Our robustness analysis also shows that intrinsic fluctuations do not make the other parameters irrelevant. We are able to draw some elementary conclusions regarding the combinations of parameters that lead to high levels of cooperation: (1) agents must not explore too much. Using an $\epsilon = 0.3$ leads to low levels of cooperation. (2) agents must not explore too little. We see that an $\epsilon = 0.03$ consistently leads to slower learning speeds than using intermediate values of the exploration rate. (3) larger learning rates lead to quicker learning speeds in the range of values we tested. In some cases, however, increasing the learning rate leads to lower levels of cooperation. The effect of changes in the batch size does not reveal a consistent pattern across the parameter ranges we tested. But if we restrict ourselves to the parameter values for the learning and exploration rates suggested by points (1)–(3), for example, $\alpha, \epsilon \in \{0.1, 0.2\}$, we see that an intermediate batch size of $K \in \{3000, 4000, 5000\}$ gives high levels of cooperation and achieves these quickly. Clearly, the interaction between these three parameters in how they influence the level of cooperation and the learning speed is complex. We leave a more detailed (theoretical) analysis of this interaction for future work.

Discussion

Contributions. In this article, we have shown that learning with imperfect, inherently noisy information is critical for the emergence of cooperation. We have done so by dissecting the widely used temporal-difference reinforcement learning process into three components.

First, cooperation can only be learned if a *stable* equilibrium supports it. We have shown how the existence of all possible equilibria depends on the combination of environmental parameters, T, S, the agents' exploration rate, ϵ, and how much they care for future rewards, δ; under the assumption that the reinforcement learning update takes into account perfect information about the environment and the other agent's current strategy. The robustly cooperative Win-Stay, Lose-Shift (WSLS) equilibrium requires a small exploration rate, ϵ, and a large discount factor δ to be stable (Fig. 2), but it is not the only stable equilibrium.

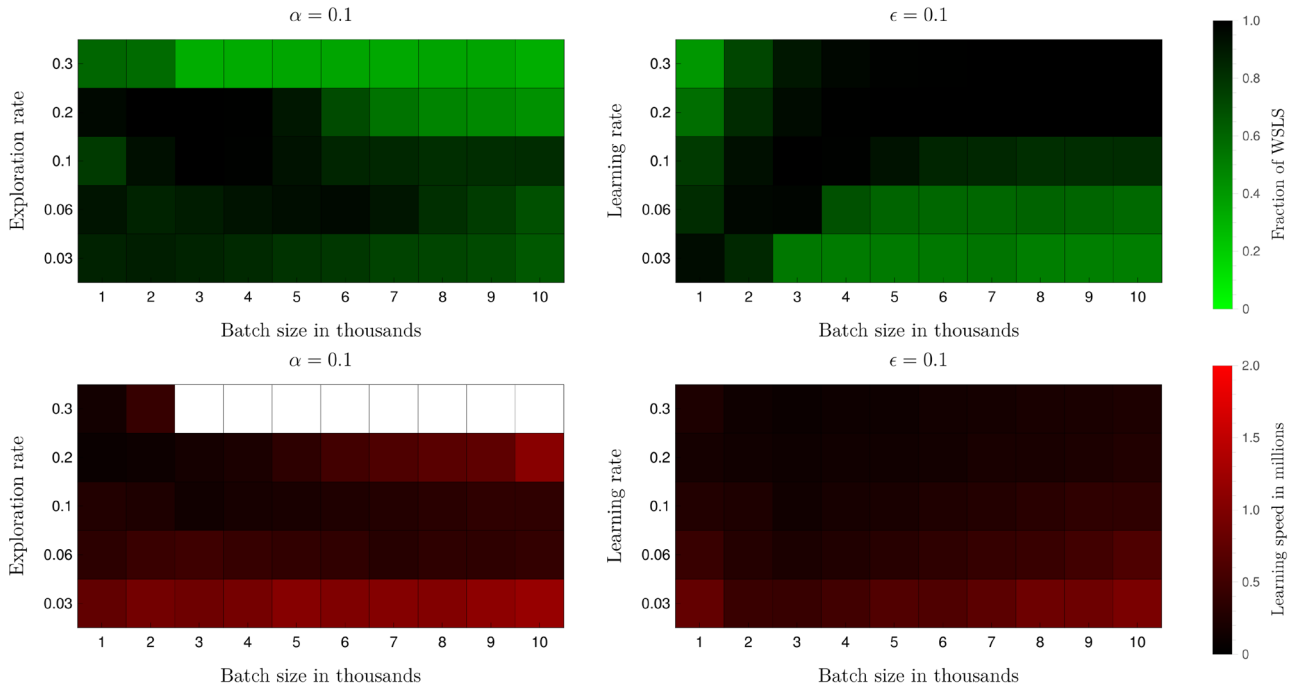
Second, cooperation will only be *learned* if the WSLS equilibrium gets selected. This is more likely, the greater the size of the region of attraction leading to the WSLS equilibrium under the learning process. We have shown for a large discount factor δ how the likelihood of learning all possible equilibria depends on the combination of the agents' learning and exploration rates α and ϵ; as well as under the assumption that the reinforcement learning update takes into account perfect information about the environment and the other agent's current strategy. The robustly cooperative Win-Stay, Lose-Shift (WSLS) equilibrium requires a small $\alpha < 0.1$ and a small $\epsilon < 0.1$ to achieve cooperation levels of 40–50% (Fig. 3). It is already interesting to observe that even though we give the reinforcement learners perfect information about the environment and the other agent's strategy, not using all of it for a learning update ($\alpha < 1$) is required to achieve cooperation levels of 40–50%.

Third, we have shown that the internal *stochasticity* of the learning process significantly improves the learnability of the robustly cooperative WSLS equilibrium. We have done so by simulating a sample batch version of the algorithm. Surprisingly, this algorithm learns to cooperate on a significantly shorter time scale than the online algorithm (Fig. 4). This highlights an essential trade-off between the cooperative learning outcome and the time it takes the agents to learn this outcome. For example, our finding suggests that in the seminal work of Sandholm and Crites[21] the number of iterations and the amount of exploration for each trial was set too small to observe a high cooperation rate between two learning agents.

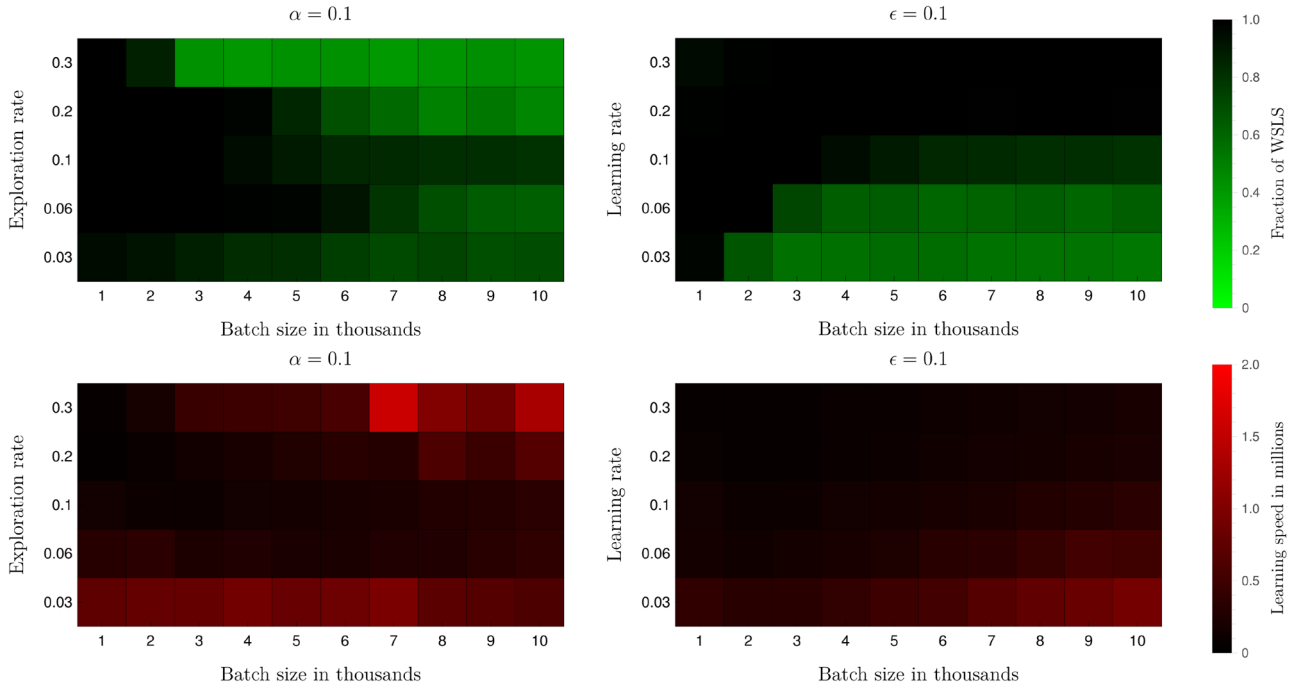
The fact that intrinsic fluctuations of reinforcement learning promote cooperation is remarkable if we consider learning as a necessary tool to approximate an optimal solution when we don't have all information about the environment available. Indeed, temporal-difference learning will always converge to an optimal solution, given a decreasing learning rate and sufficient exploration[49]. However, this is only true in a single-agent environment. There, learning serves as a means to overcome a lack of information for optimal decision-making. More information could only improve learning and decision-making.

For multi-agent learning, the situation is radically different. We have shown that learning with imperfect information is not a necessary evil to overcome a lack of knowledge about the environment. Intrinsic fluctuations in the learning process are a crucial asset to learning collectively high-rewarding, cooperative solutions.

Methodologically, obtaining our result was possible by two complementary tools for studying strategy-average reinforcement learning dynamics in stylized games. We introduced mutual best-response networks for describing the dynamics in the strategy space and strategy-average learning dynamics for describing the learning in the value space. These methods are not tailored to investigate the iterated Prisoner's Dilemma. They are likewise applicable to derive insights from many other possible learning environments.



(a) Environment parameters $S = -0.2, T = 1.5$



(b) Environment parameters $S = -0.25, T = 1.25$

Figure 5. Robustness analysis for stochastic learning. The green plots show the fraction of trajectories (1000 samples) that end in the WSLS strategy pair at time 2×10^6, and the red plots show the time it takes for the fraction of trajectories in the WSLS strategy pair to reach 0.4 in millions of time steps (we use white to represent trajectories that never reached 0.4). The x-axis always represents the batch size in thousands, and the y-axis represents either the learning rate α or the exploration rate ϵ. In all cases, we set $\delta = 0.99$.

Related work. Our main result, that intrinsic fluctuations in temporal-difference reinforcement learning promote cooperation, is in general agreement with the result that noise in biological systems is not negligible[63]. With respect to evolutionary and learning dynamics, it is important to distinguish different noise concepts.

Firstly, there is noise arising from suboptimal decision-making. In evolutionary game theory, such noise models the irrational or erroneous decision of players when adopting a less promising or rejecting a more

promising strategy of another player. Such noise can be beneficial for cooperation[10,64,65]. For individual learning, the analogous noise concept arises from the need to deviate from the currently optimal course of action to further explore the environment and improve the current strategy. Thus, it is not necessarily irrational or erroneous to do so, but required for an individual learner. Analogous to evolutionary dynamics[64], this exploration parameter can cause bifurcations towards highly desirable equilibria[66]. In our setting, the exploration rate, ϵ, regulates this exploration-exploitation trade-off, and we show analytically that a small ϵ is required for robust cooperation (Fig. 2).

Secondly, external noise affecting the payoffs or rewards the agents receive can enhance cooperation in evolutionary dynamics[67,68]. Similarly, it was recently shown that external Lévy noise promotes cooperation[69] in reinforcement learning.

Thirdly, noise in the perceptions of agents can affect cooperation in learning and evolutionary dynamics[70]. For example, inaccurate observations can lead to better learning outcomes in faster learning time, the stabilization of an otherwise chaotic learning process, and the mitigation of social dilemmas[71] (Fig. SI 1).

Fourthly, there is the intrinsic noise of the evolutionary or learning process itself. In evolutionary game theory, such intrinsic noise arises because of finite populations, which can be highly beneficial for the evolution of cooperation[72]. With respect to learning dynamics, such intrinsic noise has been found to lead to noise-sustained cycling between cooperation and defection[73–75]. This is the noise concept we are referring to when we speak about intrinsic fluctuations, and we have shown empirically how these fluctuations can be highly beneficial for the learning of cooperation.

Limitations and future work. Our results show that understanding the effects of intrinsic fluctuations in reinforcement learning is crucial in multi-agent systems. A formal treatment of these fluctuations is currently lacking and is an important avenue for future work.

The time scale on which the agents learn cooperation in our simulation with the sample batch algorithm is an order of magnitude faster than the online algorithm. Tuning the sample-batch-algorithm parameters, refining the algorithm with techniques such as optimism and leniency[37,76], and using more refined model-based variants[61] may further improve the learning speed.

Our work has focused on ϵ-greedy learning policies, which differ significantly from softmax exploration. Studying the learning dynamics under such policies will determine whether the results are a feature of exploration in general or are specific to ϵ-greedy exploration.

The environment of the iterated prisoner's dilemma is paradigmatic, but certainly not the only environment for studying cooperation. Our methods lend themselves to be applied in a variety of settings, such as a pricing duopoly with a discrete price space[48], public goods games or common-pool resource harvesting with more than two learning agents[77,78], and social dilemma situations with changing external environments[79,80].

Practical implications. Our results highlight that both designers of cooperative algorithms and regulators of algorithmic collusion must not focus solely on the learning outcome, but also on the learning efficiency. The existence of (online) algorithms that learn to cooperate under self-play is not sufficient for them to be applied in practice unless cooperation occurs on reasonable time scales, and they can learn reasonable strategies against a large class of algorithms currently employed in practice[81].

Overall, when designing sample batch algorithms, cooperation can be optimized, given the environment (T and S), by choosing δ, ϵ, α, and K following three guiding criteria: (1) the cooperative equilibrium exists and has a relatively large basin of attraction, (2) the difference in stability between the cooperative equilibrium and the other equilibria is maximized in favor of the cooperative equilibrium, and that (3) the time scale on which cooperation is achieved is minimized.

Methods

Mutual best-response networks (MBRN). An ϵ-greedy strategy can be characterized by a pure strategy, determined using the ordering of state-action values, together with exploration. If ϵ is fixed, all possible ϵ-greedy strategies can be enumerated and represented using a four-dimensional vector. Given that the opponent plays a fixed ϵ-greedy strategy, we can solve the Bellman equations to obtain the ϵ-greedy strategy that is a best response.

The state the system is in at any time, given two agents using an ϵ-greedy strategy, is similarly characterized by an eight-dimensional vector representing both strategies. We refer to this eight-dimensional vector as a *strategy pair*. A mutual best-response to a strategy pair is a strategy pair in which both agents play a strategy that is an ϵ-greedy best response to the opponent's previous strategy. In this way, we construct a directed network (of 256 strategy pairs) with edges representing mutual best responses.

By considering all possible strategy pairs, we can tabulate which edges are possible in the resulting MBRN, as well as the conditions for their existence. Taking the intersection of all possible combinations of edge conditions splits the parameter space into regions, so each region corresponds to a different MBRN (similar to the phase diagrams in[54]). As a result, we can calculate maxima and minima over the entire parameter space by considering all possible MBRNs.

The structure of this network depends on the reward parameters (T and S), ϵ, and δ. Strategy pairs with self-loops are an equilibrium under mutual best-response dynamics. By solving the Bellman equations self-consistently, we can determine the critical conditions at which strategy pairs become equilibria.

We define the fraction of strategy pairs that lead to an equilibrium under the mutual best-response dynamics as its basin of attraction. Given an initialization that selects an initial strategy pair uniformly at random from all possible strategy pairs, the basin of attraction of an equilibrium strategy pair represents the probability of ending in that strategy pair under the mutual best-response dynamics.

Learning dynamics. In essence, deterministic temporal-difference learning dynamics use strategy averages instead of individual samples of obtained rewards and estimated next-state values. They model the idealized learning behavior of agents with an infinite memory batch[56] or with separated time scales between the process of interaction and adaptation[57]. Existing learning dynamic equations with ϵ-greedy strategies were derived only for stateless interactions[82]. State-full learning dynamics employed only softmax strategies[55]. In the following, we present the deterministic Expected SARSA equations for state-full environments with ϵ-greedy strategies in discrete time.

These dynamics operate in the joint state-action-value space $\boldsymbol{q} = \bigotimes_{i,s,a} q^i(s,a)$. In order to formulate the strategy-average update of $\boldsymbol{q}$ we define the joint strategy $\boldsymbol{x} = \bigotimes_{i,s,a} x^i(a|s)$ with $x^i(a|s)$ as the probability that agents i will take action a in state s. For ϵ-greedy strategies, $\boldsymbol{x}$ is uniquely determined by $\boldsymbol{q}$ and ϵ. To obtain deterministic dynamics, we need to derive the strategy-average version of the state-action update (Eq. 2),

$$q^i_{t+1}(s,a) = q^i_t(s,a) + \alpha\left[r^i_{\boldsymbol{x}_t}(a|s) + \delta \cdot {}^{\text{next}}q^i_{\boldsymbol{x}_t}(s,a) - q^i_t(s,a)\right], \tag{5}$$

where $r^i_{\boldsymbol{x}_t}(s,a)$ is the strategy-average version of the current reward and ${}^{\text{next}}q^i_{\boldsymbol{x}_t}(s,a)$ the strategy-average version of the expected value of the next state.

The strategy-average version of the current reward is obtained as

$$r^i_{\boldsymbol{x}_t}(s,a) = \sum_{s'}\sum_{j\neq i}\prod_{a_j} x^j(a^j|s)\cdot p(s'|\boldsymbol{a},s)\cdot r^i(s'). \tag{6}$$

For each agent i, taking action a in state s when all other agents j act according to their policies $x^j(a^j|s)$ causes the next state s' via the *transition probability* $p(s'|\boldsymbol{a},s)$ at which agent i obtains the reward, $r^i(s')$.

Second, the strategy-average version of the expected value of the next state is likewise computed by averaging over all actions of the other agents and next states. For each agent i and state s, all other agents $j \neq i$ choose their action a^j with probability $x^j(s,a^j)$. Consequently, the environment transitions to the next state s' with probability $p(s'|\boldsymbol{a},s)$. At s', the agent estimates the quality to be the average of $q^i_{\boldsymbol{x}_t}(s',b)$ with respect to its own strategy. Mathematically, we write

$${}^{\text{next}}q^i_{\boldsymbol{x}_t}(s,a) := \sum_{a^j}\sum_{s'}\prod_{j\neq i} x^j(a^j|s)p(s'|\boldsymbol{a},s) \times \sum_b x^i(b|s')q^i_{\boldsymbol{x}_t}(s',b). \tag{7}$$

Here, we replace the quality estimates $q^i_t(s,a)$, which evolve in time t (Eq. 2), with the strategy-average state-action quality, $q^i_{\boldsymbol{x}_t}(s,a)$, which is the expected discounted sum of future rewards from executing action a in state s and then following along the joint strategy $\boldsymbol{x}$. It is obtained by adding the current strategy-average reward $r^i_{\boldsymbol{x}_t}(s,a)$ to the discounted strategy-average state quality of the next state $v^i_{\boldsymbol{x}_t}(s')$,

$$q^i_{\boldsymbol{x}_t}(s,a) = r^i_{\boldsymbol{x}_t}(s,a) + \delta\sum_{s'} p^i_{\boldsymbol{x}_t}(s'|a^i,s)\cdot v^i_{\boldsymbol{x}_t}(s'). \tag{8}$$

Here, $p^i_{\boldsymbol{x}_t}(s'|a^i,s)$ is agent i's strategy-average transition probability to state s' from state s^i under action a^i. It is computed by averaging over all actions of the other agents. For each agent i at state s, selecting action a^i, all other agents $j \neq i$ select action a^j with probability $x^j(a^j|s)$. Consequently, the environment will transition to the state s' with probability $p(s'|\boldsymbol{a},s)$. Mathematically, we write

$$p^i_{\boldsymbol{x}_t}(s'|a^i,s) = \sum_{a^j}\prod_{j\neq i} x^j(a^j|s)\cdot p(s'|\boldsymbol{a},s). \tag{9}$$

Further, at Eq. (8), $v^i_{\boldsymbol{x}_t}(s)$ is the strategy-average state quality, i.e., the expected discounted sum of future rewards from state s and then following along the joint strategy $\boldsymbol{x}$. They are computed via matrix inversion according to

$$\boldsymbol{v}^i_{\boldsymbol{x}_t} = [\mathbb{1}_{|\mathcal{S}|} - \delta\boldsymbol{p}_{\boldsymbol{x}_t}]^{-1}\boldsymbol{r}^i_{\boldsymbol{x}_t}, \tag{10}$$

where $\boldsymbol{v}^i_{\boldsymbol{x}_t}$ denotes the $|\mathcal{S}|$-dimensional vector containing $v^i_{\boldsymbol{x}_t}(s)$ in entry s, $\boldsymbol{r}^i_{\boldsymbol{x}_t}$ is defined analogously and $\boldsymbol{p}_{\boldsymbol{x}_t}$ is a $|\mathcal{S}| \times |\mathcal{S}|$ matrix containing $p_{\boldsymbol{x}_t}(s,s')$ (defined in Eq. (11) below) at entry (s,s'). Eq. (10) is a direct conversion of the Bellman equation $v^i_{\boldsymbol{x}_t}(s) = r^i_{\boldsymbol{x}_t}(s) + \delta\sum_{s'} p_{\boldsymbol{x}_t}(s,s')v^i_{\boldsymbol{x}_t}(s')$, which expresses that the value of the current observation is the discount factor weighted average of the current payoff and the value of the next state. Bold symbols indicate that the corresponding object is a vector or matrix, and $\mathbb{1}_Z$ is the Z-by-Z identity matrix.

The strategy-averaged transition matrix is denoted by $\boldsymbol{p}_{\boldsymbol{x}_t}$. The entry $p_{\boldsymbol{x}_t}(s,s')$ indicates the probability that the environment will transition to state s' after being in state s, given all agents follow the joint strategy $\boldsymbol{x}$. We compute them by averaging over all actions from all agents,

$$p_{\boldsymbol{x}_t}(s,s') = \sum_{a^j}\prod_j x^j(a|s)\cdot p(s'|\boldsymbol{a},s). \tag{11}$$

Further, in Eq. (10), $r^i_{\boldsymbol{x}_t}(s)$ denotes the strategy-average reward agent i obtains at state s. We compute them by averaging all actions from all agents and all next states. For each i at state s, all agents j choose action a^j with probability $x^j(a^j|s)$. Hence, the environment transitions to the next state s with probability $p(s'|\boldsymbol{a},s)$ and agent i receives the reward $r(s')$,

$$r^i_{\boldsymbol{x}_t}(s) := \sum_{a^j} \prod_j x^j(a^j|s) p(s'|\boldsymbol{a}, s) r(s'). \tag{12}$$

Note that the quality ${}^{\text{next}}q^i_{\boldsymbol{x}_t}(s,a)$ depends on s and a although it is the strategy-averaged expected value of the next state.

We finally obtained all necessary terms of state-full temporal-difference learning with ϵ-greedy strategies in value space $\boldsymbol{q}$. Using an efficient python implementation, we can apply those learning equations for simulation studies to investigate multi-agent learning phenomena in a fast and deterministically reproducible way.

Batch learning. The batch reinforcement learning problem was originally defined as learning the best strategy from a fixed set of a-priori-known transition samples[58]. However, our goal is to construct an algorithm able to interpolate between the fully online and fully deterministic version of the temporal-difference reinforcement learning process. The learning process is divided into two phases, an interaction phase, and an adaptation phase. During the interaction phase, the agent keeps its strategy fixed while interacting with its environment for K timesteps, collecting state, action, and reward information. During the adaptation phase, the agent uses the collected information to update its strategy. Key is the use of two state-action-value tables, one for acting (q_{act}), the other for improved value estimation (q_{val}). While q_{act} is kept constant during the interaction phase, q_{val} is iteratively updated[56,57].

Furthermore, we use an auxiliary, time-dependent learning rate $\alpha(s, a, t_{s,a})$ for q_{val} and a global learning rate α for q_{act}. Here $t_{s,a}$ is the local time of the state-action pair (s, a), which is given by the number of times the state-action value $q_{\text{val}}(s, a)$ has been updated during the batch. Since the environment is kept fixed for the duration of the batch, each sample in the batch should be valued equally. This can be achieved by using a state, action, and time-dependent learning rate $\alpha(s, a, t) = \frac{1}{t+1}$ (Algorithm 1).

Algorithm 1 Sample-Batch Temporal-Difference Learning

Given learning rate α, exploration rate ϵ, discount factor δ

```
begin
    Initialize q_act(s,a) = q_val(s,a) randomly.
    Initialize p(s'|a,s), n(s,a) and r(s,a) to zero.
    Set x(a|s) as ε-greedy strategy from q_act(s,a).
    Observe current state s.
    repeat
        for k = 1 to K do                                   ▷ Interaction phase
            Execute action from x(a|s);
            Observe reward r and next state s';
            Set n(s,a) ← n(s,a) + 1;
            Set p(s'|a,s) ← p(s'|a,s) + 1;
            Set r(s,a) ← r(s,a) + r;
            Set α̃ ← 1/(n(s,a)+1);
            Set q_val(s,a) ← (1 − α̃) q_val(s,a) + α̃ [ r + δ Σ_b x(b|s') q_val(s',b) ];
            Set s ← s';
        end
        foreach ŝ, â, ŝ' do                                 ▷ Adaption phase
            Set r̃ ← r(ŝ,â) / max{1, n(ŝ,â)};
            Set ṽ ← Σ_{b,z} p(z|ŝ,â) / max{1, n(ŝ,â)} x(b|z) q_val(z,b);
            Set q_act(ŝ,â) ← (1 − α) q_act(ŝ,â) + α [ r̃ + δ ṽ ];
            Set x(â|ŝ) as ε-greedy strategy ( q_act(ŝ,â) );
            Set q_val(ŝ,â) ← q_act(ŝ,â);
            Set p(ŝ'|â,ŝ), n(ŝ,â), and r(ŝ,â) to zero;
        end
    until done;
end
```

Data availability

Code to reproduce all results is available at: https://github.com/wbarfuss/intrinsic-fluctuations-cooperation and is archived at: https://doi.org/10.5281/zenodo.7303593.

Received: 26 October 2022; Accepted: 5 January 2023
Published online: 24 January 2023

References

1. Dafoe, A. *et al.* Cooperative AI: Machines must learn to find common ground. *Nature* **593**(7857), 33–36. https://doi.org/10.1038/d41586-021-01170-0 (2021).
2. Bertino, E., Doshi-Velez, F., Gini, M., Lopresti, D. & Parkes, D. Artificial Intelligence & Cooperation. Technical report. https://cra.org/ccc/resources/ccc-led-whitepapers/#2020-quadrennial-papers (2020).
3. Levin, S. A. Collective cooperation: from ecological communities to global governance and back. In *Collective Cooperation: From Ecological Communities to Global Governance and Back* 311–317 (Princeton University Press, 2020) ISBN 978-0-691-19532-2. https://doi.org/10.1515/9780691195322-025. https://www.degruyter.com/document/doi/10.1515/9780691195322-025/html.
4. Dawes, R. M. Social dilemmas. *Annu. Rev. Psychol.* **31**(1), 169–193. https://doi.org/10.1146/annurev.ps.31.020180.001125 (1980).
5. Harrington, J. E. Developing competition law for collusion by autonomous artificial agents. *J. Compet. Law Econ.* **14**(3), 331–363. https://doi.org/10.1093/joclec/nhz001 (2018).
6. Axelrod, R. & Hamilton, W. D. The evolution of cooperation. *Science* **211**(4489), 1390–1396. https://doi.org/10.1126/science.7466396 (1981).
7. Nowak, M. & Sigmund, K. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature* **364**(6432), 56–58. https://doi.org/10.1038/364056a0 (1993) (**ISSN 1476-4687**).
8. Nowak, M. A. Five rules for the evolution of cooperation. *Science* **314**(5805), 1560–1563. https://doi.org/10.1126/science.1133755 (2006).
9. Perc, M., Gómez-Gardenes, J., Szolnoki, A., Floría, L. M. & Moreno, Y. Evolutionary dynamics of group interactions on structured populations: A review. *J. R. Soc. Interface* **10**(80), 20120997. https://doi.org/10.1098/rsif.2012.0997 (2013).
10. Perc, M. *et al.* Statistical physics of human cooperation. *Phys. Rep.* **687**, 1–51. https://doi.org/10.1016/j.physrep.2017.05.004 (2017) (**ISSN 0370-1573**).
11. Masuda, N. & Ohtsuki, H. A theoretical analysis of temporal difference learning in the iterated Prisoner's Dilemma game. *Bull. Math. Biol.* **71**(8), 1818–1850. https://doi.org/10.1007/s11538-009-9424-8 (2009) (**ISSN 1522-9602**).
12. Ezrachi, A. & Stucke, M. E. *Virtual Competition: The Promise and Perils of the Algorithm-Driven Economy* (Harvard University Press, Cambridge, 2016).
13. Cimini, G. & Sánchez, A. Learning dynamics explains human behaviour in Prisoner's Dilemma on networks. *J. R. Soc. Interface* **11**(94), 20131186. https://doi.org/10.1098/rsif.2013.1186 (2014).
14. Ezaki, T., Horita, Y., Takezawa, M. & Masuda, N. Reinforcement learning explains conditional cooperation and its moody cousin. *PLOS Comput. Biol.* **12**(7), e1005034. https://doi.org/10.1371/journal.pcbi.1005034 (2016) (**ISSN 1553-7358**).
15. Ezrachi, A. & Stucke, M. E. Artificial intelligence & collusion: When computers inhibit competition. *U. Ill. L. Rev.* 1775 (2017).
16. Perolat, Julien & L., Joel Z., Zambaldi, V., Beattie, C., Tuyls, K., & Graepel, T., A multi-agent reinforcement learning model of common-pool resource appropriation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17* 3646–3655 (Red Hook, NY, USA. Curran Associates Inc, 2017) ISBN 978-1-5108-6096-4. https://proceedings.neurips.cc/paper/2017/file/2b0f658cbffd284984fb11d90254081f-Paper.pdf.
17. Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J. & Graepel, T. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17* 464–473 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2017).
18. Kühn, K.-U., Tadelis, S. Algorithmic collusion. In *Presentation Prepared for CRESSE* (2017).
19. Calvano, E., Calzolari, G., Denicolò, V. & Pastorello, S. Algorithmic pricing what implications for competition policy?. *Rev. Ind. Organ.* **55**(1), 155–171. https://doi.org/10.1007/s11151-019-09689-3 (2019).
20. Barbosa, J. V., Costa, A. H. R., Melo, F. S., Sichman, J. S. & Santos, F. C. Emergence of cooperation in N-person dilemmas through actor-critic reinforcement learning. In *Proc. of the Adaptive and Learning Agents Workshop (ALA 2020)* 9 (2020).
21. Sandholm, T. W. & Crites, R. H. Multiagent reinforcement learning in the Iterated Prisoner's Dilemma. *Biosystems* **37**(1), 147–166. https://doi.org/10.1016/0303-2647(95)01551-5 (1996) (**ISSN 0303-2647**).
22. Schrepel, T. Here's why algorithms are NOT (really) a thing. *Concurrentialiste*, May 2017 (online), (2017).
23. Schwalbe, U. Algorithms, machine learning, and collusion. *J. Compet. Law Econ.* **14**(4), 568–607. https://doi.org/10.1093/joclec/nhz004 (2018).
24. Peysakhovich, A. & Lerer, A. Towards AI that can solve social dilemmas. In *AAAI Spring Symposium Series* 7 (2018).
25. Dafoe, A., Hughes, E., Bachrach, Y., Collins, T., McKee, K. R., Leibo, J. Z., Larson, K. & Graepel, T. Open problems in cooperative AI. arXiv preprint arXiv:2012.08630v1 (2020).
26. Peysakhovich, A. & Lerer, A. Consequentialist conditional cooperation in social dilemmas with imperfect information. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BkabRiQpb (2018).
27. Lerer, Adam, Peysakhovich, Alexander. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. arXiv:1707.01068v4 (2018).
28. Foerster, J. *et al.* Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18* 122–130 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2018).
29. Hughes, E. *et al.* Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in Neural Information Processing Systems* Vol. 31 (Curran Associates Inc., 2018) https://proceedings.neurips.cc/paper/2018/hash/7fea637fd6d02b8f0adf6f7dc36aed93-Abstract.html.
30. Eccles, T., Hughes, E., Kramár, J., Wheelwright, S. & Leibo, J. Z. The imitation game: Learned reciprocity in Markov games. In *AAMAS '19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* 3 (2019).
31. Baker, B. Emergent reciprocity and team formation from randomized uncertain social preferences. In *Advances in Neural Information Processing Systems* Vol. 33 (eds Larochelle, H. *et al.*) 15786–15799 (Curran Associates Inc., 2020) https://proceedings.neurips.cc/paper/2020/file/b63c87b0a41016ad29313f0d7393cee8-Paper.pdf.
32. Wang, J. X. *et al.* Evolving intrinsic motivations for altruistic behavior. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19* 683–692 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2019) ISBN 9781450363099 .
33. Hughes, E., Anthony, T. W., Eccles, T., Leibo, J. Z., Balduzzi, D. & Bachrach, Y. Learning to resolve alliance dilemmas in many-player zero-sum games. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20* 538–547 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2020) ISBN 9781450375184.

34. Meylahn, J. M. & den Boer, A. V. Learning to collude in a pricing duopoly. *Manuf. Serv. Oper. Manag.*https://doi.org/10.1287/msom.2021.1074 *(2022).*
35. Bowling, M. & Veloso, M. Multiagent learning using a variable learning rate. *Artif. Intell.* **136**(2), 215–250. https://doi.org/10.1016/S0004-3702(02)00121-2 (2002) (**ISSN 00043702**).
36. de Cote, E. M., Lazaric, A. & Restelli, M. Learning to cooperate in multi-agent social dilemmas. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '06* 783–785 (Association for Computing Machinery, New York, NY, USA, 2006). ISBN 978-1-59593-303-4 https://doi.org/10.1145/1160633.1160770.
37. Panait, L., Sullivan, K. & Luke, S. Lenient learners in cooperative multiagent systems. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* 801–803 (2006).
38. Stimpson, J. L. & Goodrich, M. A. Learning to cooperate in a social dilemma: A satisficing approach to bargaining. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03* 728–735 (AAAI Press, Washington, DC, USA, 2003). ISBN 978-1-57735-189-4.
39. Bush, R. R. & Mosteller, F. A mathematical model for simple learning. *Psychol. Rev.* **58**, 313–323. https://doi.org/10.1037/h0054388 (1951) (**ISSN 1939-1471**).
40. Macy, M. W. & Flache, A. Learning dynamics in social dilemmas. *Proc. Natl. Acad. Sci.* **99**(suppl–3), 7229–7236. https://doi.org/10.1073/pnas.092080099 (2002).
41. Izquierdo, S. S., Izquierdo, L. R. & Gotts, N. M. Reinforcement learning dynamics in social dilemmas. *J. Artif. Soc. Soc. Simul.* **11**(2), 1 (2008).
42. Masuda, N. & Nakamura, M. Numerical analysis of a reinforcement learning model with the dynamic aspiration level in the iterated Prisoner's dilemma. *J. Theor. Biol.* **278**(1), 55–62. https://doi.org/10.1016/j.jtbi.2011.03.005 (2011) (**ISSN 0022-5193**).
43. Zhang, H.-F., Wu, Z.-X. & Wang, B.-H. Universal effect of dynamical reinforcement learning mechanism in spatial evolutionary games. *J. Stat. Mech. Theory Exp.* **2012**(06), P06005. https://doi.org/10.1088/1742-5468/2012/06/P06005 (2012) (**ISSN 1742-5468**).
44. Jia, N. & Ma, S. Evolution of cooperation in the snowdrift game among mobile players with random-pairing and reinforcement learning. *Phys. A Stat. Mech. Appl.* **392**(22), 5700–5710. https://doi.org/10.1016/j.physa.2013.07.049 (2013) (**ISSN 0378-4371**).
45. Jia, D. *et al.* Local and global stimuli in reinforcement learning. *New J. Phys.* **23**(8), 083020. https://doi.org/10.1088/1367-2630/ac170a (2021) (**ISSN 1367-2630**).
46. Song, Z. *et al.* Reinforcement learning facilitates an optimal interaction intensity for cooperation. *Neurocomputing* **513**, 104–113. https://doi.org/10.1016/j.neucom.2022.09.109 (2022) (**ISSN 0925-2312**).
47. Botvinick, M., Wang, J. X., Dabney, W., Miller, K. J. & Kurth-Nelson, Z. Deep reinforcement learning and its neuroscientific implications. *Neuron* **107**(4), 603–616. https://doi.org/10.1016/j.neuron.2020.06.014 (2020) (**ISSN 08966273**).
48. Calvano, E., Calzolari, G., Denicolò, V., Harrington, J. E. & Pastorello, S. Protecting consumers from collusive prices due to AI. *Science* **370**(6520), 1040–1042. https://doi.org/10.1126/science.abe3796 (2020).
49. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT press, 2018).
50. Rummery, G. A. & Niranjan, M. *On-Line Q-Learning Using Connectionist Systems* Vol. 37 (Citeseer, 1994).
51. Sutton, R. S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Adv. Neural Inf. Process. Syst.* **8** (1995).
52. Press, W. H. & Dyson, F. J. Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent. *Proc. Natl. Acad. Sci.* **109**(26), 10409–10413. https://doi.org/10.1073/pnas.1206569109 (2012) (**ISSN 0027-8424, 1091-6490**).
53. Usui, Y. & Ueda, M. Symmetric equilibrium of multi-agent reinforcement learning in repeated prisoner's dilemma. *Appl. Math. Comput.* **409**, 126370. https://doi.org/10.1016/j.amc.2021.126370 (2021) (**ISSN 0096-3003.**).
54. Meylahn, J. M. *et al.* Limiting dynamics for Q-learning with memory one in symmetric two-player, two-action games. *Complexity*https://doi.org/10.1155/2022/4830491 *(2022).*
55. Barfuss, W., Donges, J. F. & Kurths, J. Deterministic limit of temporal difference reinforcement learning for stochastic games. *Phys. Rev. E* **99**, 043305. https://doi.org/10.1103/PhysRevE.99.043305 (2019).
56. Barfuss, W. Reinforcement learning dynamics in the infinite memory limit. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* 1768–1770 (2020).
57. Barfuss, W. Dynamical systems as a level of cognitive analysis of multi-agent learning. *Neural Comput. Appl.* **34**(3), 1653–1671. https://doi.org/10.1007/s00521-021-06117-0 (2022).
58. Lange, S., Gabel, T. & Riedmiller, M. Batch reinforcement learning. In *Reinforcement Learning* 45–73 (Springer, 2012). https://doi.org/10.1007/978-3-642-27645-3_2.
59. Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* **8**(3), 293–321. https://doi.org/10.1007/BF00992699 (1992) (**ISSN 1573-0565**).
60. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533. https://doi.org/10.1038/nature14236 (2015) (**ISSN 1476-4687**).
61. Van S., Harm, S. & Richard S. A. deeper look at planning as learning from replay. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML'15*, Vol. 37, 2314–2322. JMLR.org, (2015).
62. Wilson, E. B. Probable inference, the law of succession, and statistical inference. *J. Am. Stat. Assoc.* **22**(158), 209–212 (1927).
63. Bialek, W. S. *Biophysics: Searching for Principles* (Princeton University Press, Princeton, 2012) (**ISBN 978-0-691-13891-6**).
64. Vukov, J., Szabó, G. & Szolnoki, A. Cooperation in the noisy case: Prisoner's dilemma game on two types of regular random graphs. *Phys. Rev. E* **73**(6), 067103. https://doi.org/10.1103/PhysRevE.73.067103 (2006).
65. Szolnoki, A., Vukov, J. & Szabó, G. Selection of noise level in strategy adoption for spatial social dilemmas. *Phys. Rev. E* **80**(5), 056112. https://doi.org/10.1103/PhysRevE.80.056112 (2009).
66. Leonardos, S. & Piliouras, G. Exploration-exploitation in multi-agent learning: Catastrophe theory meets game theory. *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(13), 11263–11271. https://doi.org/10.1609/aaai.v35i13.17343 ISSN 2374-3468. https://ojs.aaai.org/index.php/AAAI/article/view/17343 (2021).
67. Jia, C.-X., Liu, R.-R., Yang, H.-X. & Wang, B.-H. Effects of fluctuations on the evolution of cooperation in the prisoner's dilemma game. *Europhys. Lett.* **90**(3), 30001. https://doi.org/10.1209/0295-5075/90/30001 (2010) (**ISSN 0295-5075**).
68. Assaf, M., Mobilia, M. & Roberts, E. Cooperation dilemma in finite populations under fluctuating environments. *Phys. Rev. Lett.* **111**(23), 238101. https://doi.org/10.1103/PhysRevLett.111.238101 (2013).
69. Wang, L. *et al.* Lévy noise promotes cooperation in the prisoner's dilemma game with reinforcement learning. *Nonlinear Dyn.*https://doi.org/10.1007/s11071-022-07289-7 *(2022)* (**ISSN 1573-269X**).
70. Santos, F. P., Levin, S. A. & Vasconcelos, V. V. Biased perceptions explain collective action deadlocks and suggest new mechanisms to prompt cooperation. *iScience* **24**(4), 102375. https://doi.org/10.1016/j.isci.2021.102375 (2021) (**ISSN 2589-0042**).
71. Barfuss, W. & Mann, R. P. Modeling the effects of environmental and perceptual uncertainty using deterministic reinforcement learning dynamics with partial observability. *Phys. Rev. E* **105**(3), 034409. https://doi.org/10.1103/PhysRevE.105.034409 (2022).
72. Nowak, M. A., Sasaki, A., Taylor, C. & Fudenberg, D. Emergence of cooperation and evolutionary stability in finite populations. *Nature* **428**(6983), 646–650. https://doi.org/10.1038/nature02414 (2004) (**ISSN 1476-4687**).
73. Galla, T. Intrinsic noise in game dynamical learning. *Phys. Rev. Lett.* **103**(19), 198702. https://doi.org/10.1103/PhysRevLett.103.198702 (2009) (**ISSN 0031-9007, 1079-7114**).
74. Galla, T. Cycles of cooperation and defection in imperfect learning. *J. Stat. Mech. Theory Exp.* **2011**(08), N08001. https://doi.org/10.1088/1742-5468/2011/08/N08001 (2011) (**ISSN 1742-5468**).

75. Bladon, A. J. & Galla, T. Learning dynamics in public goods games. *Phys. Rev. E* **84**(4), 041132. https://doi.org/10.1103/PhysRevE.84.041132 (2011).
76. Panait, L., Tuyls, K. & Luke, S. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *J. Mach. Learn. Res.* **9**, 423–457 (2008).
77. Barfuss, W., Donges, J. F., Wiedermann, M. & Lucht, W. Sustainable use of renewable resources in a stylized social-ecological network model under heterogeneous resource distribution. *Earth Syst. Dyn.* **8**(2), 255–264. https://doi.org/10.5194/esd-8-255-2017 (2017).
78. Geier, F., Barfuss, W., Wiedermann, M., Kurths, J. & Donges, J. F. The physics of governance networks: Critical transitions in contagion dynamics on multilayer adaptive networks with application to the sustainable use of renewable resources. *Eur. Phys. J. Spec. Top.* **228**(11), 2357–2369. https://doi.org/10.1140/epjst/e2019-900120-4 (2019) (**ISSN 1951-6401**).
79. Barfuss, W., Donges, J. F., Lade, S. J. & Kurths, J. When optimization for governing human-environment tipping elements is neither sustainable nor safe. *Nat. Commun.* **9**(1), 2354. https://doi.org/10.1038/s41467-018-04738-z (2018) (**ISSN 2041-1723**).
80. Barfuss, W., Donges, J. F., Vasconcelos, V. V., Kurths, J. & Levin, S. A. Caring for the future can turn tragedy into comedy for long-term collective action under risk of collapse. *Proc. Natl. Acad. Sci.* **117**(23), 12915–12922. https://doi.org/10.1073/pnas.1916545117 (2020).
81. den Boer, A. V., Meylahn, J. M. & Schinkel, M. Pieter. Artificial collusion: Examining supra-competitive pricing by autonomous Q-learning algorithms. Available at SSRN. https://ssrn.com/abstract=4213600 (2022).
82. Wunder, M., Littman, M. L. & Babes, M. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *ICML* 1167–1174. https://icml.cc/Conferences/2010/papers/191.pdf (2010).

Acknowledgements
This work was supported by the German Federal Ministry of Education and Research (BMBF): Tuebingen AI Center, FKZ: 01IS18039A, and the Dutch Institute for Emergent Phenomena (DIEP) cluster at the University of Amsterdam.

Author contributions
Both authors contributed equally to this work.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-27672-7.

Correspondence and requests for materials should be addressed to J.M.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2023