Editors' Suggestion

Deterministic limit of temporal difference reinforcement learning for stochastic games

Wolfram Barfuss,[1,2,*] Jonathan F. Donges,[1,3] and Jürgen Kurths[1,2,4]
[1]*Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany*
[2]*Department of Physics, Humboldt University Berlin, 12489 Berlin, Germany*
[3]*Stockholm Resilience Centre, Stockholm University, 104 05 Stockholm, Sweden*
[4]*Saratov State University, 410012 Saratov, Russia*

(Received 20 September 2018; published 10 April 2019)

Reinforcement learning in multiagent systems has been studied in the fields of economic game theory, artificial intelligence, and statistical physics by developing an analytical understanding of the learning dynamics (often in relation to the replicator dynamics of evolutionary game theory). However, the majority of these analytical studies focuses on repeated normal form games, which only have a single environmental state. Environmental dynamics, i.e., changes in the state of an environment affecting the agents' payoffs has received less attention, lacking a universal method to obtain deterministic equations from established multistate reinforcement learning algorithms. In this work we present a methodological extension, separating the interaction from the adaptation timescale, to derive the deterministic limit of a general class of reinforcement learning algorithms, called temporal difference learning. This form of learning is equipped to function in more realistic multistate environments by using the estimated value of future environmental states to adapt the agent's behavior. We demonstrate the potential of our method with the three well-established learning algorithms Q learning, SARSA learning, and actor-critic learning. Illustrations of their dynamics on two multiagent, multistate environments reveal a wide range of different dynamical regimes, such as convergence to fixed points, limit cycles, and even deterministic chaos.

DOI: 10.1103/PhysRevE.99.043305

I. INTRODUCTION

Individual learning through reinforcements is a central approach in the fields of artificial intelligence [1–3], neuroscience [4,5], learning in games [6], and behavioral game theory [7–10], thereby offering a general purpose principle to either solve complex problems or explain behavior. Also in the fields of complexity economics [11,12] and social science [13], reinforcement learning has been used as a model for human behavior to study social dilemmas.

However, there is a need for improved understanding and better qualitative insight into the characteristic dynamics that different learning algorithms produce. Therefore, reinforcement learning has also been studied from a dynamical systems perspective. In their seminal work, Börgers and Sarin showed that one of the most basic reinforcement learning update schemes, Cross learning [14], converges to the replicator dynamics of evolutionary games theory in the continuous time limit [15]. This has led to at least two, presumably nonoverlapping, research communities, one from statistical physics [16–26] and one from computer science machine learning [27–35]. Thus, Sato and Crutchfield [18] and Tuyls *et al.* [27] independently deduced identical learning equations in 2003.

The statistical physics articles usually consider the deterministic limit of the stochastic learning equations, assuming infinitely many interactions between the agents before an adaptation of behavior occurs. This limit can either be performed in continuous time with differential equations [17–19] or discrete time with difference equations [20–22]. The differences between both variants can be significant [21,23]. Deterministic chaos was found to emerge when learning simple [17] as well as complicated games [25]. Relaxing the assumption of infinitely many interactions between behavior updates revealed that noise can change the attractor of the learning dynamics significantly, e.g., by noise-induced oscillations [20,21].

However, these statistical physics studies so far considered only repeated normal form games. These are games where the payoff depends solely on the set of current actions, typically encoded in the entries of a payoff matrix (for the typical case of two players). Receiving payoff and choosing another set of joint actions is performed repeatedly. This setup lacks the possibility to study dynamically changing environments and their interplay with multiple agents. In those systems, rewards depend not only on the joint action of agents but also on the states of the environment. Environmental state changes may occur probabilistically and depend also on joint actions and the current state. Such a setting is also known as a Markov game or stochastic game [36,37]. Thus, a repeated normal form game is a special case of a stochastic game with only one environmental state. Notably, Akiyama and Kaneko [38,39] did emphasize the importance of a dynamically changing environment; however, they did not utilize a reinforcement learning update scheme.

The computer science machine-learning community dealing with reinforcement learning as a dynamical system (see Ref. [28] for an overview) particularly emphasizes the link between evolutionary game theory and multiagent reinforcement learning as a well grounded theoretical framework for the

[*] barfuss@pik-potsdam.de

latter [28–31]. This dynamical systems perspective is proposed as a way to gain qualitative insights about the variety of multiagent reinforcement learning algorithms (see Ref. [2] for a review). Consequently, this literature developed a focus on the translation of established reinforcement learning algorithms to a dynamical systems description, as well as the development of new algorithms based on insights of a dynamical systems perspective. While there is more work on stateless games (e.g., Q learning [27] and frequency-adjusted multiagent Q learning [32]), multiagent learning dynamics for multistate environments have been developed as well, such as piecewise replicator dynamics [34], state-coupled replicator dynamics [33], or reverse engineering state-coupled replicator dynamics [35].

Both communities, statistical physics and machine learning, share the interest in better qualitative insights into multiagent learning dynamics. While the statistical physics community focuses more on dynamical properties the same set of learning equations can produce, it leaves a research gap of learning equations capable of handling multiple environmental states. The machine-learning community, on the other hand, aims more toward algorithm development, but so far have put their focus less on a dynamical systems understanding. Taken together, there is the challenge of developing a dynamical systems theory of multiagent learning dynamics in varying environmental states.

With this work, we aim to contribute to such a dynamical systems theory of multiagent learning dynamics. We present a methodological extension for obtaining the deterministic limit of multistate temporal difference reinforcement learning. In essence, it consists of formulating the temporal difference error for batch learning, and sending the batch size to infinity. We showcase our approach with the three prominent learning variants of Q learning, SARSA learning, and actor-critic (AC) learning. Illustrations of their learning dynamics reveal multiple different dynamical regimes, such as fixed points, periodic orbits, and deterministic chaos.

In Sec. II we introduce the necessary background and notation. Section III presents our method to obtain the deterministic limit of temporal difference reinforcement learning and demonstrates it for multistate Q learning, SARSA learning, and actor-critic learning. We illustrate their learning dynamics for two previously utilized two-agent two-action two-state environments in Sec. IV. In Sec. V we conclude with a discussion of our work.

II. PRELIMINARIES

We introduce the components (including notation) of our multiagent environment systems (see Fig. 1), followed by a brief introduction of temporal difference reinforcement learning.

A. Multi-agent Markov environments

A multiagent Markov environment (also called stochastic game or Markov game) consists of $N \in \mathbb{N}$ *agents*. The environment can exist in $Z \in \mathbb{N}$ *states* $\mathcal{S} = \{S_1, \dots, S_Z\}$. In each state each agent has $M \in \mathbb{N}$ available *actions* $\mathcal{A}^i = \{A^i_1, \dots, A^i_M\}$, $i = 1, \dots, N$ to choose from. Having an identical number of actions for all states and all agents is notational convenience, no significant restriction. A joint action of all agents is referred to by $\mathbf{a} \in \boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$, the joint action of all agents but agent i is denoted by $\mathbf{a}^{-i} \in \boldsymbol{\mathcal{A}}^{-i} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^{i-1} \times \mathcal{A}^{i+1} \times \cdots \times \mathcal{A}^N$.

Environmental dynamics are given by the probabilities for state changes expressed as a transition tensor $\mathbf{T} \in [0,1]^{Z \times M \times \dots \text{(N times)} \dots \times M \times Z}$. The entry $T_{s\mathbf{a}s'}$ denotes the probability $P(s'|s, \mathbf{a})$ that the environment transitions to state s' given the environment was in state s and the agents have chosen the joint action $\mathbf{a}$. Hence, for all $s, \mathbf{a}$, $\sum_{s'} T_{s\mathbf{a}s'} = 1$ must hold. The assumption that the next state only depends on the current state and joint action makes our system Markovian. We here restrict ourselves to ergodic environments without absorbing states (cf. Ref. [35]).

The rewards receivable by the agents are given by the reward tensor $\mathbf{R} \in \mathbb{R}^{N \times Z \times M \times \dots \text{(N times)} \dots \times M \times Z}$. The entry $R^i_{s\mathbf{a}s'}$ denotes the reward agent i receives when the environment transits from state s to state s' under the joint action $\mathbf{a}$. Rewards are also called payoffs from a game-theoretic perspective.

Agents draw their actions from their behavior profile $\mathbf{X} \in [0,1]^{N \times Z \times M}$. The entry $X^i_{sa} = P(a \,|\, i, s)$ denotes the probability that agent i chooses action a in state s. Thus, for all i and all s, $\sum_a X^i_{sa} = 1$ must hold. We here focus on the case of independent agents, able to fully observe the current state of the environment. With correlated behavior (see, e.g., Ref. [2]) and partially observable environments [40,41], one could extend the multiagent environment systems to be even more general. Note that what we call behavior profile is usually termed policy from a machine-learning perspective or behavioral strategy from a game-theoretic perspective. We chose to introduce our own term because policies and strategies suggest a deliberate choice which we do not want to impose.

B. Averaging out behavior and environment

We define a notational convention that allows a systematic averaging over the current behavior profile $\mathbf{X}$ and the environmental transitions $\mathbf{T}$. It will be used throughout the paper.

Averaging over the whole behavioral profile yields

$$\begin{aligned} \mathbf{x}\langle \circ \rangle &:= \sum_{\mathbf{a}} \mathbf{X}_{s\mathbf{a}} \cdot \circ \\ &:= \sum_{a^1 \in \mathcal{A}^1} \cdots \sum_{a^N \in \mathcal{A}^N} X^1_{sa^1} \cdots X^N_{sa^N} \cdot \circ. \end{aligned} \tag{1}$$

Here $\circ$ serves as a placeholder. If the quantity to be inserted for $\circ$ depends on the summation indices, then those indices will be summed over as well. If the quantity, which is averaged out, is used in tensor form, then it is written in bold. If not, then remaining indices are added after the right angle bracket.

Averaging over the behavioral profile of the other agents, keeping the action of agent i, yields

$$\begin{aligned} \mathbf{x}_{-i}\langle \circ \rangle &:= \sum_{\mathbf{a}^{-i}} \mathbf{X}^{-i}_{s\mathbf{a}^{-i}} \cdot \circ \\ &:= \underbrace{\sum_{a^1 \in \mathcal{A}^1} \cdots \sum_{a^N \in \mathcal{A}^N}}_{\text{excl. } i} \underbrace{X^1_{sa^1} \cdots X^N_{sa^N}}_{\text{excl. } i} \cdot \circ\,. \end{aligned} \tag{2}$$

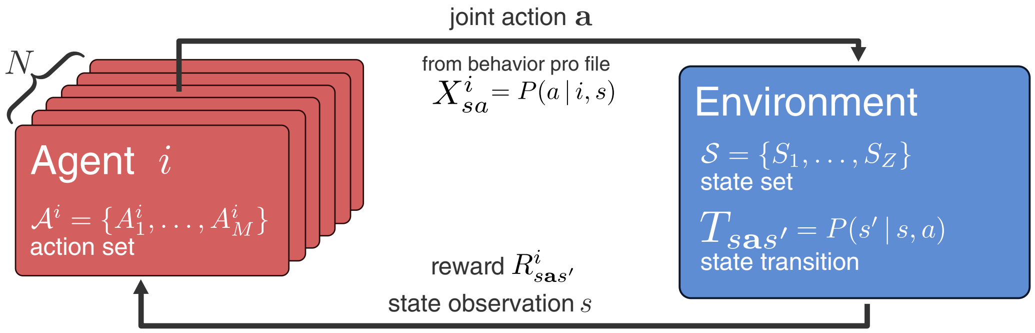
joint action $\mathbf{a}$
from behavior pro file
$X^i_{sa} = P(a \,|\, i, s)$
N
Agent i
$\mathcal{A}^i = \{A^i_1, \ldots, A^i_M\}$
action set
Environment
$\mathcal{S} = \{S_1, \ldots, S_Z\}$
state set
$T_{s\mathbf{a}s'} = P(s' \,|\, s, a)$
state transition
reward $R^i_{s\mathbf{a}s'}$
state observation s

FIG. 1. Multiagent Markov environment (also known as stochastic or Markov game). N agents choose a joint action $\mathbf{a} = (a^1, \ldots, a^N)$ from their action sets $\mathcal{A}^i$, based on the current state of the environment s, according to their behavior profile $X^i_{sa} = P(a|i, s)$. This will change the state of the environment from s to s' with probability $T_{s\mathbf{a}s'}$ and provide each agent with a reward $R^i_{s\mathbf{a}s'}$.

Last, averaging over the subsequent state s' yields

$${}_{\mathbf{T}}\langle \circ \rangle := \sum_{s'} T_{s\mathbf{a}s'} \cdot \circ := \sum_{s' \in \mathcal{S}} T_{sa^1 \ldots a^N s'} \cdot \circ. \qquad (3)$$

Of course, these operations may also be combined as ${}_{\mathbf{TX}}\langle \circ \rangle$ and ${}_{\mathbf{TX}^{-i}}\langle \circ \rangle$ by multiplying both summations.

For example, given a behavior profile $\mathbf{X}$, the resulting effective Markov Chain transition matrix reads ${}_{\mathbf{X}}\langle T \rangle_{ss'}$, which encodes the transition probabilities from state s to s'. From ${}_{\mathbf{X}}\langle T \rangle_{ss'}$ the stationary distribution of environmental states $\boldsymbol{\sigma}(\mathbf{X})$ can be computed. $\boldsymbol{\sigma}(\mathbf{X})$ is the eigenvector corresponding to the eigenvalue 1 of ${}_{\mathbf{X}}\langle T \rangle_{ss'}$. Its entries encode the ratios of the average durations the agents find themselves in the respective environmental states.

The average reward agent i receives from state s under action a, given all other agents follow the behavior profile $\mathbf{X}$ reads ${}_{\mathbf{TX}^{-i}}\langle R \rangle^i_{sa}$. Including agent i's behavior profile gives the average reward it receives from state s: ${}_{\mathbf{TX}}\langle R \rangle^i_s$. Hence, ${}_{\mathbf{TX}}\langle R \rangle^i_s = \sum_a X^i_{sa} \cdot {}_{\mathbf{TX}^{-i}}\langle R \rangle^i_{sa}$ holds.

C. Agent's preferences and values

Typically, agents are assumed to maximize their exponentially discounted sum of future rewards, called return $G^i(t) = (1 - \gamma^i) \sum_{k=0}^{\infty} (\gamma^i)^k r^i(t + k)$, where $\gamma^i \in [0, 1)$ is the discount factor of agent i and $r^i(t + k)$ denotes the reward received by agent i at time step $t + k$. Exponential discounting is most commonly used for its mathematical convenience and because it ensures consistent preferences over time. Other formulations of a return use, e.g., finite-time horizons, average reward settings, as well as other ways of discounting, such as hyperbolic discounting. Those other forms require their own form of reinforcement learning.

Given a behavior profile $\mathbf{X}$, the expected return defines the *state-value function* $V^i_s(\mathbf{X}) := {}_{\mathbf{TX}}\langle G^i(t) \mid s(t) = s \rangle^i_s$, which is independent of time t. The operation ${}_{\mathbf{TX}}\langle \ldots \mid s(t) = s \rangle$ denotes the behavioral and environmental average as defined in Eqs. (1) and (3) given that in the current time step t the environment is in state s. Inserting the return yields the *Bellman equation* [42],

$$V^i_s(\mathbf{X}) = {}_{\mathbf{TX}}\big\langle (1 - \gamma^i) r^i(t) + \gamma^i V^i_{s(t+1)}(\mathbf{X}) \big| s(t) = s \big\rangle^i_s. \qquad (4)$$

This recursive relationship between state values declares that the value of a state s is the discounted value of the subsequent state $s(t + 1)$ plus $(1 - \gamma^i)$ times the reward received along the way. Evaluating the behavioral and environmental average ${}_{\mathbf{TX}}\langle\ \rangle$ and writing in matrix form we get:

$$\mathbf{V}^i(\mathbf{X}) = (1 - \gamma^i) \cdot {}_{\mathbf{TX}}\langle \mathbf{R} \rangle^i + \gamma^i \cdot {}_{\mathbf{X}}\langle \mathbf{T} \rangle \cdot \mathbf{V}^i(\mathbf{X}). \qquad (5)$$

The reward $r^i(t)$ received at time step t is evaluated to reward ${}_{\mathbf{TX}}\langle R \rangle^i_s$ for state s, since the behavioral and environmental average was conditioned on starting in state $s(t) = s$. The average subsequent state value $V^i_{s(t+1)}(\mathbf{X})$ from the current state s can be expressed as a matrix multiplication of the effective Markov transition matrix and the vector of state values: $\sum_{s'} {}_{\mathbf{X}}\langle T \rangle_{ss'} \cdot \mathbf{V}^i_{s'}(\mathbf{X})$.

A solution of the state values $\mathbf{V}^i(\mathbf{X})$ can be obtained using matrix inversion

$$\mathbf{V}^i(\mathbf{X}) = (1 - \gamma^i)(\mathbb{1}_Z - \gamma^i {}_{\mathbf{X}}\langle \mathbf{T} \rangle)^{-1} {}_{\mathbf{TX}}\langle \mathbf{R} \rangle^i. \qquad (6)$$

The computational complexity of matrix inversion makes this solution strategy infeasible for large systems. Therefore many iterative solution methods exist [3].

Equivalently, *state-action-value functions* Q^i_{sa} are defined as the expected return, given agent i applied action a in state s and then followed $\mathbf{X}$ accordingly: $Q^i_{sa}(\mathbf{X}) := {}_{\mathbf{TX}}\langle G^i(t) \mid s(t) = s, a(t) = a \rangle^i_{sa}$. Even though this is the behavioral average over the whole behavioral profile, the resulting object carries an action index because the operation is conditioned on the current action to be $a(t) = a$. They can be computed via

$$Q^i_{sa}(\mathbf{X}) = (1 - \gamma^i) {}_{\mathbf{TX}^{-i}}\langle R \rangle^i_{sa} + \gamma^i \sum_{s'} {}_{\mathbf{X}}\langle T \rangle_{ss'} \cdot V^i_{s'}(\mathbf{X}). \qquad (7)$$

One can show that $V^i_s(\mathbf{X}) = \sum_a X^i_{sa} Q^i_{sa}(\mathbf{X})$ holds for the inverse relation of state-action and state values.

D. Learning through reinforcement

In contrast to the typical game-theoretic assumption of perfect information, we assume that agents know nothing about the game in advance. They can only gain information about the environment and other agents through interactions. They do not know the true reward tensor $\mathbf{R}$ or the true transition probabilities $T_{s\mathbf{a}s'}$. They experience only reinforcements (i.e., particular rewards $R^i_{s\mathbf{a}s'}$), while observing the current true Markov state of the environment.

In essence, reinforcement learning consists of iterative behavior changes toward a behavior profile with maximum state values. However, due to the agents' limited information about the environment, they generally cannot compute a behavior profile's true state and state-action values, $V^i_s(\mathbf{X})$ and $Q^i_{sa}(\mathbf{X})$, as defined in the previous section. Therefore, agents use time-dependent *state-value* and *state-action-value approximations*, $\tilde{V}^i_s(t)$ and $\tilde{Q}^i_{sa}(t)$, during the reinforcement learning process.

1. Temporal difference learning

Basically, state-action-value approximations $\tilde{Q}^i_{sa}$ get iteratively updated by a temporal difference error $D^i_{sa}(t)$:

$$\tilde{Q}^i_{sa}(t+1) = \tilde{Q}^i_{sa}(t) + \alpha^i D^i_{sa}(t), \tag{8}$$

with $\alpha^i \in (0, 1)$ being the *learning rate* of agent i. These state-action propensities $\tilde{Q}^i_{sa}$ can be interpreted as estimates of the state-action values Q^i_{sa}.

The temporal difference error expresses a difference in the estimation of state-action values. New experience is used to compute a new estimate of the current state-action value and corrected by the old estimate. The estimate from the new experience uses exactly the recursive relation of value functions from the Bellmann equation [Eq. (4)],

$$\begin{aligned} D^i_{sa}(t) = {} & \delta_{ss(t)}\delta_{aa(t)} \\ & \cdot \Big[\underbrace{(1-\gamma^i) R^i_{s(t)a(t)\mathbf{a}^{-i}(t)s(t+1)} + \gamma^i \, \Upsilon^i_{s(t+1)}\,(t)}_{\text{estimate from new experience}} \\ & - \underbrace{\Upsilon^i_{s(t)}(t)}_{\text{old estimate}} \Big]. \end{aligned} \tag{9}$$

Here s and a denote the state-action pair whose temporal difference error is calculated. With $s(t)$, $a(t)$, etc., we refer to the state, action, etc., that occurred at time step t. Thus, the notation $R^i_{s(t)a(t)\mathbf{a}^{-i}(t)s(t+1)}$ refers to the entry of the reward tensor $R^i_{sa\mathbf{a}^{-i}s'}$ when at time step t the environmental state was s [$s(t) = s$], agent i chose action a [$a(t) = a$], the other agents chose the joint action $\mathbf{a}^{-i}$ [$\mathbf{a}^{-i}(t) = \mathbf{a}^{-i}$] and the next environmental state was s' [$s(t+1) = s'$]. The $\Upsilon^i_{s(t+1)}(t)$ indicates the state-value estimate at time step t of the state visited at the next time step $s(t+1)$. $\Upsilon^i_{s(t)}(t)$ denotes the state-value estimate at time step t of the current state $s(t)$. Different choices for these estimations are possible, leading to different learning variants (see below).

The Kronecker deltas $\delta_{ss(t)}$, $\delta_{aa(t)}$ indicate that the temporal difference error for state-action pair (s, a) is only nonzero when (s, a) was actually visited in time step t. This denotes and emphasizes that agents can only learn from experience. In contrast, e.g., experience-weighted-attraction learning [9] assumes that action propensities can be updated with hypothetical rewards an agent would have received if she had played a different action than the current action. These two cases have been referred to as *full vs. partial information* [16]. Thus, the Kronecker deltas in Eq. (9) indicate a partial information update. The agents use only information experienced through interaction.

The state-action-value approximations $\tilde{Q}^i_{sa}$ are translated to a behavior profile according to the Gibbs-Boltzmann distribution [1] (also called softmax),

$$X^i_{sa}(t) = \frac{\exp\left[\beta^i \tilde{Q}^i_{sa}(t)\right]}{\sum_b \exp\left[\beta^i \tilde{Q}^i_{sb}(t)\right]}. \tag{10}$$

The behavior profile $\mathbf{X}$ becomes a dynamic variable as well. The parameter β^i controls the *intensity of choice* or the *exploitation level* of agent i controlling the *exploration-exploitation trade-off*. In analogy to statistical physics, β^i is the inverse temperature. For high β^i, agents tend to exploit their learned knowledge about the environment, leaning toward actions with high estimated state-action value. For low β^i, agents are more likely to deviate from these high-value actions in order to explore the environment further with the chance of finding actions, which eventually lead to even higher values. Other behavior profile translations exist as well (e.g., ϵ-greedy [1]).

2. Three learning variants

The specific choices of the value estimates Υ in the temporal difference error result in different reinforcement learning variants.

a. Q learning. For the Q learning algorithm [1,3], $\Upsilon^i_{s(t+1)}(t) = \max_b \tilde{Q}^i_{s(t+1)b}(t)$ and $\Upsilon^i_{s(t)}(t) = \tilde{Q}^i_{s(t)a(t)}(t)$. Thus, the Q learning update takes the maximum of the next state-action-value approximations as an estimate for the next state value, regardless of the actual next action the agent plays. This is reasonable because the maximum is the highest value achievable given the current knowledge. For the state-value estimate of the current state, the Q learner takes the current state-action-value approximation $Q^i_{s(t)a(t)}(t)$. This is reasonable because it is exactly the quantity that gets updated by Eq. (8).

b. SARSA learning. For SARSA learning [1,3], $\Upsilon^i_{s(t+1)}(t) = \tilde{Q}^i_{s(t+1)a(t+1)}(t)$ and $\Upsilon^i_{s(t)}(t) = \tilde{Q}^i_{s(t)a(t)}(t)$, where $a(t+1)$ denotes the action taken by agent i at the next time step. Thus, the SARSA algorithm uses the five ingredients of an update sequence of state, action, reward, next state, and next action to perform one update. In practice, the SARSA sequence has to be shifted one time step backward to know what the actual "next" action of the agent was.

c. Actor-critic learning. For AC learning [1,3], $\Upsilon^i_{s(t+1)}(t) = \tilde{V}^i_{s(t+1)}(t)$ and $\Upsilon^i_{s(t)}(t) = \tilde{V}^i_{s(t)}(t)$. Compared to Q and SARSA learners, it has an additional data structure of state-value approximations which get separately updated according to $\tilde{V}^i_s(t+1) = \tilde{V}^i_s(t) + \alpha^i \cdot D^i_{sa}(t)$. The state-action-value approximations $\tilde{Q}^i_{sa}$ serve as the actor which gets criticized by the state-value approximations $\tilde{V}^i_s$.

Table I summarizes the values estimates Υ for these three learning variants. Q and SARSA learning are structurally more similar compared to the actor-critic learner, which uses an additional data structure of state-value approximations $\tilde{V}^i_s$.

III. DETERMINISTIC LIMIT

So far we gave a brief introduction to temporal difference reinforcement learning. A more comprehensive presentation can be found in Ref. [1]. In this section we will present an extension to the methodology of interaction-adaptation timescales separation to the general class of temporal

TABLE I. Overview of the three reinforcement learning variants. Shown in the columns are the value estimates for the next state $\Upsilon^i_{s(t+1)}(t)$ and the current state $\Upsilon^i_{s(t)}(t)$ for both ends of the batch size spectrum: $K = 1$ and $K = \infty$.

	(a) $K = 1$		(b) $K = \infty$	
	$\Upsilon^i_{s(t+1)}(t)$	$\Upsilon^i_{s(t)}(t)$	$\Upsilon^i_{s(t+1)}(t)$	$\Upsilon^i_{s(t)}(t)$
Q learning	$\max_b \tilde{Q}^i_{s(t+1)b}(t)$	$\tilde{Q}^i_{s(t)a(t)}(t)$	${}^{\max}\mathcal{Q}^i_{sa}(\mathbf{X})$	$\frac{1}{\beta^i}\log X^i_{sa}(t)$
SARSA learning	$\tilde{Q}^i_{s(t+1)a(t+1)}(t)$	$\tilde{Q}^i_{s(t)a(t)}(t)$	${}^{\text{next}}\mathcal{V}^i_{sa}(\mathbf{X})$	$\frac{1}{\beta^i}\log X^i_{sa}(t)$
AC learning	$\tilde{V}^i_{s(t+1)}(t)$	$\tilde{V}^i_{s(t)}(t)$	${}^{\text{next}}\mathcal{V}^i_{sa}(\mathbf{X})$	/

difference reinforcement learning. In summary, we (i) give a batch formulation of the temporal difference error, (ii) separate the timescales of interaction and adaptation by sending the batch size to infinity, and (iii) present a resulting deterministic limit conversion rule for discrete time updates. We showcase our method in the three learning variants of Q, SARSA, and actor-critic learning. For the statistical physics community, we present learning equations, capable of handling environmental state transitions. For the machine-learning community, we present the systematic methodology we use to obtain the deterministic learning equations. Note that these deterministic learning equations will not depend on the state-value or state-action-value approximations anymore, being iterated maps of the behavior profile alone.

Following, e.g., Refs. [18,19,22], we first combine Eqs. (8) and (10) and obtain

$$X^i_{sa}(t+1) = \frac{X^i_{sa}(t)\exp\left[\alpha^i\beta^i D^i_{sa}(t)\right]}{\sum_b X^i_{sb}(t)\exp\left[\alpha^i\beta^i D^i_{sb}(t)\right]}. \tag{11}$$

Although it appears that only the product $\alpha^i\beta^i$ matters for a behavior profile update, the temporal difference error D^i_{sa} may depend only on the exploitation level β^i, as we will show below.

Next, we formulate the temporal difference error for batch learning.

A. Batch learning

With batch learning we mean that several time steps of interaction with the environment and the other agents take place before an update of the state-action-value approximations and the behavior profile occurs. It has also been interpreted as a form of history replay [43] which is essential to stabilize the learning process when function approximation (e.g., by deep neural networks) is used [44]. History (i.e., already experienced state, action, next state triples) is used again for an update of the state-action-value approximations.

Imagine that the information from these interactions are stored inside a batch of size $K \in \mathbb{N}$. We introduce the corresponding temporal difference error of batch size K:

$$\begin{aligned} D^i_{sa}(t;K) := \frac{1}{K(s,a)}\sum_{k=0}^{K-1}\big\{&\delta_{ss(t+k)}\delta_{aa(t+k)} \\ &\times\left[(1-\gamma^i)R^i_{s(t+k)a(t+k)\mathbf{a}^{-i}(t+k)s(t+k+1)}\right. \\ &\left.+\gamma^i\,\Upsilon^i_{s(t+k+1)}(t) - \Upsilon^i_{s(t)}(t)\right]\big\}, \end{aligned} \tag{12}$$

where $K(s,a) = \max[1, \sum_{k=0}^{K-1}\delta_{ss(t+k)}\delta_{aa(t+k)}]$ denotes the number of times the state-action pair (s,a) was visited. If the state-action pair (s,a) was never visited, then $K(s,a) = 1$. The agents interact K times under the same behavior profile and use the sample average to summarize the new experience in order to update the state-action-value approximations:

$$\tilde{Q}^i_{sa}(t+K) = \tilde{Q}^i_{sa}(t) + \alpha^i D^i_{sa}(t;K). \tag{13}$$

The notation $D^i_{sa}(t)$ denotes a batch update of batch size 1: $D^i_{sa}(t) = D^i_{sa}(t;1)$.

B. Separation of timescales

We obtain the deterministic limit of the temporal difference learning dynamics by sending the batch size to infinity, $K \to \infty$. Equivalently, this can be regarded as a separation of timescales. Two processes can be distinguished during an update of the state-action-value approximations $\Delta\tilde{Q}^i_{sa}(t) := \tilde{Q}^i_{sa}(t+1) - \tilde{Q}^i_{sa}(t)$: adaptation and interaction,

$$\begin{aligned} \Delta\tilde{Q}^i_{sa}(t) = \alpha^i\delta_{ss(t)}\delta_{aa(t)}\cdot& \\ \overbrace{\left[\underbrace{(1-\gamma^i)R^i_{s(t)a(t)\mathbf{a}^{-i}(t)s(t+1)} + \gamma^i\,\Upsilon^i_{s(t+1)}(t)}_{\text{interaction}} - \Upsilon^i_{s(t)}(t)\right]}^{\text{adaptation}}&. \end{aligned} \tag{14}$$

By separating the timescales of both processes, we assume that (infinitely) many interactions happen before one step of behavior profile adaptation occurs.

Under this assumption and because of the assumed ergodicity one can replace the sample average, i.e., the sum over sequences of states and actions with the behavior profile average, i.e., the sum over state-action behavior and transition probabilities according to

$$\boxed{\frac{1}{K(s,a)}\sum_{k=0}^{K-1}\delta_{ss(t+k)}\delta_{aa(t+k)} \to \sum_{s'}\sum_{\mathbf{a}^{-i}}\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}T_{s a \mathbf{a}^{-i} s'}.} \tag{15}$$

For example, the immediate reward $R^i_{s(t)a(t)\mathbf{a}^{-\mathbf{i}}(t)s(t+1)}$ in the temporal difference error becomes ${}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}$. The time t gets resealed accordingly, as well.

Taking the limit $K \to \infty$ in this way, we choose to stay in discrete time, leaving the continuous time limit following Refs. [18,19,25] for future work.

C. Three learning variants

Next we present the deterministic limit of the temporal difference error of the three learning variants of Q, SARSA, and actor-critic learning. Inserting them into Eq. (11) yields the complete description of the behavior profile update in the deterministic limit. Table I presents an overview of the resulting equations and a comparison to their batch size $K = 1$ versions.

1. Q learning

The temporal difference error of Q learning consists of three terms: (i) $R^i_{s(t)a(t)\mathbf{a}^{-\mathbf{i}}(t)s(t+1)}$, (ii) $\max_b \tilde{Q}^i_{s(t+1)b}(t)$, and (iii) $\tilde{Q}^i_{s(t)a(t)}(t)$. As already stated, $R^i_{s(t)a(t)\mathbf{a}^{-\mathbf{i}}(t)s(t+1)} \rightarrow {}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}$ under $K \rightarrow \infty$. $\max_b \tilde{Q}^i_{s(t+1)b}(t) \rightarrow {}^{\max}\mathcal{Q}^i_{sa}(\mathbf{X})$, which is defined as

$$ {}^{\max}\mathcal{Q}^i_{sa}(\mathbf{X}) := \sum_{s'} \sum_{\mathbf{a}^{-i}} \mathbf{X}^{-i}_{s\mathbf{a}^{-i}} T_{sa\mathbf{a}^{-i}s'} \max_b Q^i_{s'b}(\mathbf{X}) \tag{16} $$

using the deterministic limit conversion rule [Eq. (15)]. Because of the assumption of infinite interactions, we can here replace the state-action-value approximations $\tilde{Q}^i_{s(t+1)b}$ with the true state-action values $Q^i_{s'b}$ as defined by Eq. (7).

For the third term, we invert Eq. (10), yielding $\tilde{Q}^i_{sa}(t) = (\beta^i)^{-1} \log X^i_{sa}(t) + \text{const}^i_s$, where const^i_s is constant in actions but may vary for each agent and state. Now, one can show that the dynamics induced by Eq. (11) are invariant against additive transformations in the temporal difference error $D^i_{sa}(t, \infty) \rightarrow D^i_{sa}(t, \infty) + \text{const}^i_s$. Thus, the third term can be converted according to $\tilde{Q}^i_{s(t)a(t)}(t) \rightarrow (\beta^i)^{-1} \log X^i_{sa}(t)$.

All together, the temporal difference error for Q learning in the deterministic limit reads

$$ \begin{aligned} {}^{q}D^i_{sa}(t, \infty) = {}& (1 - \gamma^i)_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} \\ & + \gamma^{i\,\max}\mathcal{Q}^i_{sa}(\mathbf{X}) - \frac{1}{\beta^i} \log X^i_{sa}(t). \end{aligned} \tag{17} $$

2. SARSA learning

Two of the three terms of the SARSA temporal difference error are identical to the one of Q learning, leaving $\tilde{Q}^i_{s(t+1)a(t+1)}(t)$, which we replace by

$$ {}^{\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X}) := \sum_{s'} \sum_{\mathbf{a}^{-i}} \mathbf{X}^{-i}_{s\mathbf{a}^{-i}} T_{sa\mathbf{a}^{-i}s'} \sum_b X^i_{s'b} Q^i_{s'b}(\mathbf{X}) \tag{18} $$

using again the deterministic limit conversion rule [Eq. (15)] and the state-action value $Q^i_{s'b}(\mathbf{X})$ of the behavior profile $\mathbf{X}$ according to Eq. (7).

Thus, the temporal difference error for the SARSA learning update in the deterministic limit reads

$$ \begin{aligned} {}^{\text{sarsa}}D^i_{sa}(t; \infty) = {}& (1 - \gamma^i)_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} \\ & + \gamma^{i\,\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X}) - \frac{1}{\beta^i} \log X^i_{sa}(t). \end{aligned} \tag{19} $$

3. Actor-critic learning

For the temporal difference error for AC learning we have to find replacements for (i) $\tilde{V}^i_{s(t+1)}(t)$ and (ii) $\tilde{V}^i_{s(t)}(t)$. Applying again Eq. (15) yields $\tilde{V}^i_{s(t+1)}(t) \rightarrow {}^{\text{next}}\mathcal{V}^i_{sa}$, defined as

$$ {}^{\text{next}}\mathcal{V}^i_{sa} := \sum_{s'} \sum_{\mathbf{a}^{-i}} \mathbf{X}^{-i}_{s\mathbf{a}^{-i}} T_{sa\mathbf{a}^{-i}s'} V^i_{s'}(\mathbf{X}), \tag{20} $$

using Eq. (6) for the state value $V^i_{s'}(\mathbf{X})$. This is the average value of the next state given that in the current state the agent took action a. One can show that ${}^{\text{next}}\mathcal{V}^i_{sa}(\mathbf{X}) = {}^{\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X})$ from the SARSA update.

The second remaining term belongs to the slower adaptation timescale or, in other words, occurs outside the batch. Thus, our deterministic limit conversion rule [Eq. (15)] does not apply. We could think of a conversion $\tilde{V}^i_{s(t)}(t) := \sum_a X^i_{sa} \tilde{Q}^i_{s(t)a(t)}(t) \rightarrow (\beta^i)^{-1} \sum_a X^i_{sa}(t) \log X^i_{sa}(t)$. However, the remaining term is constant in action, and therefore irrelevant for the dynamics, as we have argued above. Thus, we can simply put $\tilde{V}^i_{s(t)}(t) \rightarrow 0$.

All together, the temporal difference error of the actor-critic learner in the deterministic limit reads

$$ {}^{\text{ac}}D^i_{sa}(t, \infty) = (1 - \gamma^i)_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} + \gamma^{i\,\text{next}}\mathcal{V}^i_{sa}(\mathbf{X}). \tag{21} $$

IV. APPLICATION TO EXAMPLE ENVIRONMENTS

In the following we apply the derived deterministic learning equations in two different environments. Specifically, we compare the three well-established temporal difference learning variants (Q learning, SARSA learning, and AC learning) in two different two-agent ($N = 2$), two-action ($M = 2$), and two-state ($Z = 2$) environments: a two-state matching pennies game and a two-state prisoner's dilemma. Since the main contribution of this paper is the derivation of the deterministic temporal difference learning equations, we are not trying to make a case with our example environments beyond a systematic comparison of our learners. Therefore, we chose environments that have been used previously in related literature [33–35,45]. Note also that we leave a comparison between the deterministic limit and the stochastic equations to future work, which would add a noise term to our equations following the example of Ref. [20].

To measure the performance of an agent's behavior profile in a single scalar, we use the dot product between the stationary state distribution $\boldsymbol{\sigma}(\mathbf{X})$ of the effective Markov Chain with the transition matrix ${}_{\mathbf{X}}\langle\mathbf{T}\rangle$ and the behavior average reward ${}_{\mathbf{TX}}\langle\mathbf{R}\rangle^i$. Interestingly, we find this relation to be identical to the dot product of the stationary distribution and the state value $\mathbf{V}^i(\mathbf{X})$:

$$ \boldsymbol{\sigma}(\mathbf{X}) \cdot {}_{\mathbf{TX}}\langle\mathbf{R}\rangle^i = \boldsymbol{\sigma}(\mathbf{X}) \cdot \mathbf{V}^i(\mathbf{X}). \tag{22} $$

This relation can be shown by using Eq. (6) and the fact that $\boldsymbol{\sigma}(\mathbf{X})$ is an eigenvector of ${}_{\mathbf{X}}\langle\mathbf{T}\rangle$.

In the following examples we will only investigate homogeneous agents, i.e., agents whose parameters will not differ from each other. We will therefore drop the agent indices from α^i, β^i, and γ^i. The heterogeneous agent case is to be explored in future work.

A. Two-state matching pennies

The single-state matching pennies game is a paradigmatic two-agent, two-action game. Imagine the situation of soccer

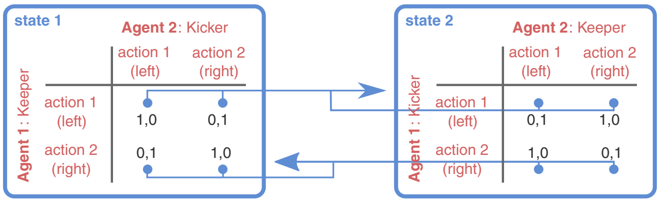

FIG. 2. Two-state matching pennies. Rewards are given in black type in the payoff tables for each state. State-transition probabilities are indicated by (blue) arrows.

penalty kicks. The keeper (agent 1) can choose to jump either to the left or right side of the goal, and the kicker (agent 2) can choose to kick the ball also either to the left or the right. If both agents choose the identical side, then the keeper agent wins; otherwise, the kicker wins.

In the two-state version of the game, according to Ref. [35], the rules are extended as follows: In state 1 the situation is as described in the single-state version. Whenever agent 1 (the keeper) decides to jump to the left, the environment transitions to state 2, in which the agents switch roles: Agent 1 now plays the kicker and agent 2 the keeper. From here, whenever agent 1 (now the kicker) decides to kick to the right side the environment transition again to state 1 and both agents switch their roles again.

Figure 2 illustrates this two-state matching pennies games. Formally, the payoff matrices are given by

$$\begin{pmatrix} R^1_{111s'}, R^2_{111s'} & R^1_{112s'}, R^2_{112s'} \\ R^1_{121s'}, R^2_{121s'} & R^1_{122s'}, R^2_{122s'} \end{pmatrix} = \begin{pmatrix} 1, 0 & 0, 1 \\ 0, 1 & 1, 0 \end{pmatrix}$$

in state 1 and

$$\begin{pmatrix} R^1_{211s'}, R^2_{211s'} & R^1_{212s'}, R^2_{212s'} \\ R^1_{221s'}, R^2_{221s'} & R^1_{222s'}, R^2_{222s'} \end{pmatrix} = \begin{pmatrix} 0, 1 & 1, 0 \\ 1, 0 & 0, 1 \end{pmatrix}$$

in state 2 for $s' \in \{1, 2\}$. State transitions are governed by

$$\begin{pmatrix} T_{1112} & T_{1122} \\ T_{1212} & T_{1222} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} T_{2111} & T_{2121} \\ T_{2211} & T_{2221} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

Thus, by construction, the probability of transitioning to the other state is independent of agent 2's action. Only agent 1 has agency over the state transitions. By playing a uniformly random behavior profile $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$, both agents would obtain an average reward of 0.5 per time step.

With Fig. 3 we compare the temporal difference error in the behavior space sections for each environmental state at a comparable low discount factor $\gamma \in [0, 1)$ of $\gamma = 0.1$, as well as learning trajectories for an exemplary initial condition for two learning rates $\alpha \in (0, 1)$, a low one ($\alpha = 0.02$) and a high one ($\alpha = 0.8$). Overall, we observe a variety of qualitatively different dynamical regimes, such as fixed points, periodic orbits and chaotic motion.

Specifically, we see that Q learners and SARSA learners behave qualitatively similarly in contrast to the AC learners for both learning rates α. For the low learning rate $\alpha = 0.02$, Q and SARSA learners reach a fixed point of playing both actions with equal probability in both states, yielding a reward of 0.5. Due to the low α, this takes approximately 600 time steps. In contrast, the reward trajectory of the AC learners appears to be chaotic. Figure 5 confirms this observation, which we will discuss in more detail below.

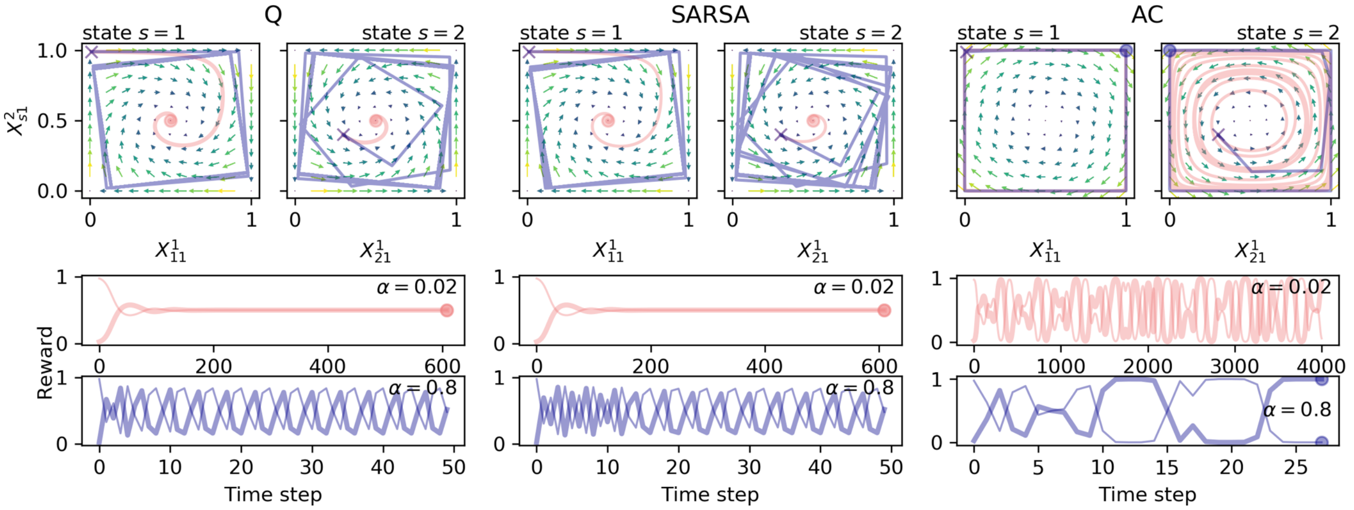

FIG. 3. Three learners in two-state matching pennies environment for low discount factor $\gamma = 0.1$; intensity of choice $\beta = 5.0$. At the top, the temporal difference errors for the Q learners [Eq. (17)], SARSA learners [Eq. (19)], and AC learners [Eq. (21)] are shown in two behavior phase-space sections, one for each state. The arrows indicate the average direction the temporal difference errors drive the learner toward, averaged over all phase-space points of the other state. Arrow colors (and shadings) additionally encode their lengths. Selected trajectories are shown in the phase-space sections, as well as by reward trajectories, plotting the average reward value [Eq. (22)] over time steps. Crosses in the phase-space subsections indicate the initial behavior $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.01, 0.99, 0.3, 0.4)$. Circles signal the arrival at a fixed point, determined by the absolute difference of behavior profiles between two subsequent time steps being below $\epsilon = 10^{-6}$. Trajectories are shown for two different learning rates $\alpha = 0.02$ (light red) and $\alpha = 0.8$ (dark blue). The bold reward trajectory belongs to agent 1 and the thin one to agent 2. Note that the temporal difference error is independent from the learning rate α. A variety of qualitatively different dynamical regimes can be observed.

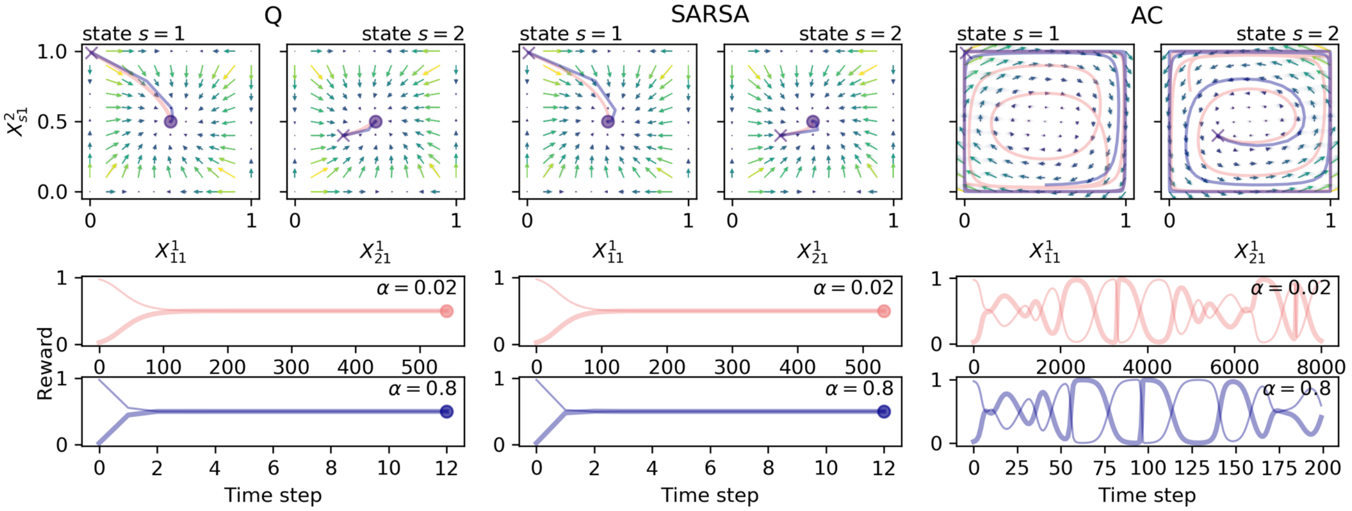

FIG. 4. Two-state matching pennies environment for high discount factor $\gamma = 0.9$; otherwise, identical to Fig. 3.

For the high learning rate $\alpha = 0.8$, both Q and SARSA learners enter a periodic limit cycle. Differences in the trajectories of Q and SARSA learners are clearly visible. The time average reward of this periodic orbit appears to be approximately 0.5 for each agent, identical to the reward of the fixed point at lower α. The AC learners, however, converge to a fixed point after oscillating near the edges of the phase space. At this fixed point agent 1 plays action 1 in state 1 with probability 1. Thus, it has trapped the system into state 2. In state 2, agent 1 plays action 2 and agent 2 plays action 1 with probability 1 and, consequently, agent 1 receives a reward of 1, whereas agent 2 receives 0 reward. One might ask, Why does agent 2 not decrease her probability for playing action 1, thereby increasing her own reward? And, indeed, the arrows of the temporal difference error suggest this change of behavior profile. However, agent 2 cannot follow because her behavior is trapped on the simplex of nonzero action probabilities X^2_{2a}. For only $M = 2$ actions, $X^2_{21} = 1$ thus can no longer change, regardless of the temporal difference error.

Increasing the discount factor to $\gamma = 0.9$, we observe the learning rate α to set the timescale of learning (Fig. 4). The intensity of choice remained $\beta = 5.0$. A high learning rate $\alpha = 0.8$ corresponds to faster learning in contrast to a low learning rate $\alpha = 0.02$. Also, the ratio of learning timescales is comparable to the inverse ratio of learning rates. For both α, Q and SARSA learners reach a fixed point, whereas the AC learners seem to move chaotically (details to be investigated below). Comparing the trajectories between the learning rates α, we observe a similar shape for each pair of learners. However, the similarity of the AC trajectories decreases at larger time steps.

So far, we varied two parameters: the discount factor $\gamma \in [0, 1)$ and the learning rate $\alpha \in (0, 1)$. Combining Figs. 3 and 4, we investigated all four combinations of a low and a high γ with a low and a high α. We can summarize that Q and SARSA learners converge to a fixed point for all combinations of discount factor γ and learning rate α, except when γ is low and α simultaneously high. Actor-critic dynamics seem chaotic for all combinations of α and γ.

To investigate the relationship between the parameters more thoroughly, Fig. 5 shows bifurcation diagrams with the bifurcation parameters α and γ. Additionally, it also gives the largest Lyapunov exponents for each learner and each parameter combination. A largest Lyapunov exponent greater than zero is a key characteristic of chaotic motion. We computed the Lyapunov exponent from the analytically derived Jacobian matrix, iteratively used in a QR decomposition according to Ref. [46]. See Appendix for details.

The largest Lyapunov exponent for Q and SARSA learners align almost perfectly with each other, whereas the largest Lyapunov exponent of the AC learners behaves qualitatively different. We first describe the behavior of the Q and SARSA learner: For high learning rates α and low farsightedness γ, Fig. 5 shows a periodic orbit with few (four) points in phase space. Largest Lyapunov exponents are distinctly below 0 at those regimes. Increasing the farsightedness γ both learners enter a regime of visiting many points in phase space around the stable fixed point $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$. The largest Lyapuonv exponents are close to zero. With increasing γ the distance around this fixed-point solution decreases until the dynamics converge from a farsightedness γ slightly greater than 0.5 onward. From there the largest Lyapunov exponent decreases again for further increasing γ. The same observations can be made along a decreasing bifurcation parameter α, except that at the end, for low α, the largest Lyapunov exponents do not decrease as distinctly as for high γ.

The behavior of the actor-critic dynamics is qualitatively different from the one of Q and SARSA. The placement of the fixed points on the natural numbers grid suggests that the AC learners get confined on one of the 16 (M^{NZ}) corners of the behavior phase space. No regularity to which fixed point the AC learners converge can be deduced. The largest Lyapunov exponent is always above zero and experiences an overall decreasing behavior. Similarly, for a decreasing bifurcation parameter α, the largest Lyapunov exponent tends to decrease as well. Different from the bifurcation diagram along γ, for low α the system might enter a periodic motion but only

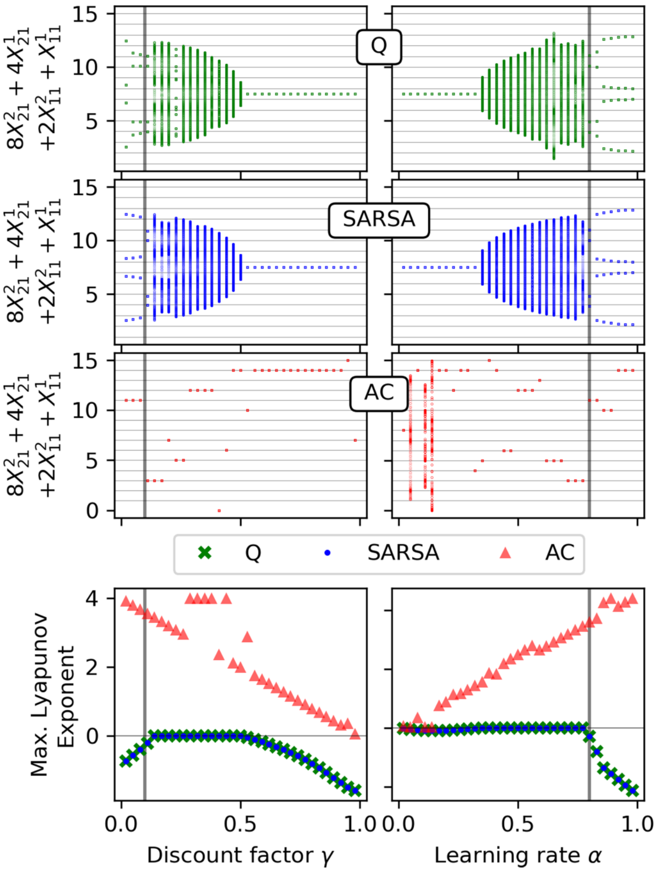

FIG. 5. Varying discount factor γ and learning rate α in two-state matching pennies environment for intensity of choice $\beta = 5.0$ for the Q learners (green crosses), the SARSA learners (blue dots), and the AC learners (red triangles). On the left, the discount factor γ is varied with learning rate $\alpha = 0.8$, as indicated by the gray vertical lines on the right. On the right, the learning rate α is varied with discount factor $\gamma = 0.1$ as indicated by the gray vertical lines on the left. The three top panels show the visited behavior points during 1000 iterations after a transient period of 100 000 time steps from initial behavior $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.01, 0.99, 0.3, 0.4)$. Visited points are mapped to the function $8X^2_{21} + 4X^1_{21} + 2X^2_{11} + X^1_{11}$ on the vertical axes to give a fuller image of the visited behavior profiles. The bottom panel shows the corresponding largest Lyapunov exponents for the three learners. Overall, Q and SARSA learners behave qualitatively more similarly than the actor-critic learners.

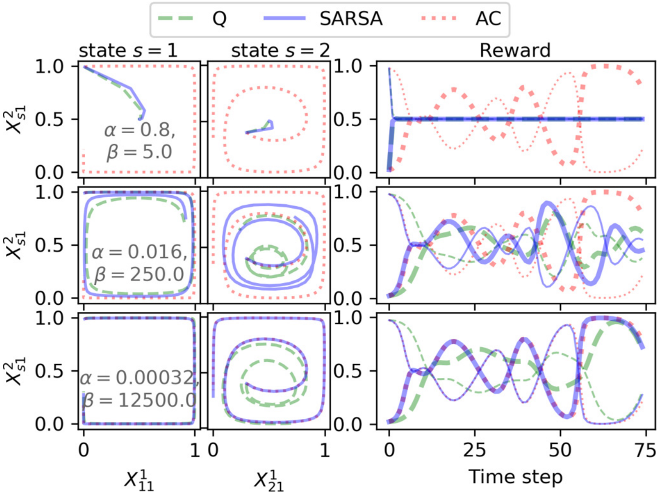

FIG. 6. Varying intensity of choice β under constant $\alpha\beta$ in a two-state matching pennies environment for discount factor $\gamma = 0.9$. On the left trajectories of the three learners [Q, green dashed; SARSA, blue straight; AC, red dotted] are shown in the two behavior space sections, one for each state. On the right, the corresponding reward trajectories are shown. The initial behavior was $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.01, 0.99, 0.3, 0.4)$. The bold reward trajectory belongs to agent 1 and the thin one to agent 2. One observes the deterministic limit of actor-critic learning to be invariant under constant $\alpha\beta$ and SARSA learning to converge to AC learning under $\beta \to \infty$.

for some parameters α. No regularity can be determined at which parameters α the AC learners enter a periodic motion. A more thorough investigation of the nonlinear dynamics, especially those of the actor-critic learner, seems of great interest but is, however, beyond the scope of this article and leaves promising paths for future work.

Concerning the parameter β, the intensity of choice, one can infer from the update equations [Eq. (11) combined with Eq. (19) and Eq. (21)] that the dynamics for the AC learners are invariant for a constant product $\alpha\beta$. This is because the temporal difference error of the actor-critic learners in the deterministic limit is independent of β. Further, the dynamics of the SARSA learners will converge to the dynamics of the AC learners under $\beta \to \infty$. Figure 6 nicely confirms these two observations. Observing Table I is another way to see this. Since the value estimate of the future state is identical for SARSA and AC learning, letting the value estimate of the current state vanish by sending $\beta \to \infty$ makes the SARSA learners approximate the AC learners.

As mentioned before, β controls the exploration-exploitation trade-off. In the temporal difference errors of Q and SARSA learning it appears in the term indicating the value estimate of the current state $-1/\beta^i \log(X^i_{sa})$. If this term dominates the temporal difference error (i.e., if β is small), then the learners tend toward the center of behavior space, i.e., $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$, forgetting what they have learned about the obtainable reward. This characteristic happens to be favorable in our two-state matching pennies environment, which is why Q and SARSA learners perform better in finding the $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$ solution. On the other hand, if β is large, then the temporal difference error is dominated by the current reward and future value estimate. Not being able to forget, the learners might get trapped in unfavorable behavior, as we can see observing the actor-critic learners. To calibrate β it is useful to make oneself clear that it must come in units of [log behavior]/[reward].

B. Two-state prisoner's dilemma

The single-state prisoner's dilemma is another paradigmatic two-agent, two-action game. It has been used to model social dilemmas and study the emergence of cooperation. It describes a situation in which two prisoners are separately interrogated, leaving them with the choice to either cooperate with each other by not speaking to the police or defecting by testifying.

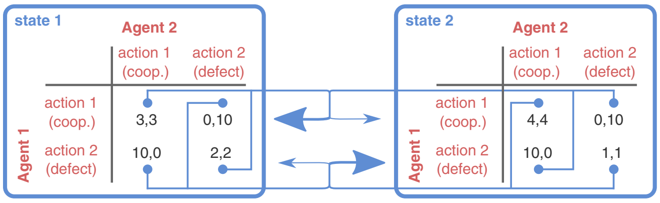

FIG. 7. Two-state prisoner's dilemma. Rewards are given in black type in the payoff tables for each state. State-transition probabilities are indicated by (blue) arrows.

The two-state version, which has been used as a test environment also in Refs. [33–35], extends this situation somewhat artificially by playing a prisoner's dilemma in each of the two states with a transition probability of 10% from one state to the other if both agents chose the same action and a transition probability of 90% if both agents chose opposite actions.

Figure 7 illustrates these game dynamics. Formally, the payoff matrices are given by

$$\begin{pmatrix} R^1_{111s'}, R^2_{111s'} & R^1_{112s'}, R^2_{112s'} \\ R^1_{121s'}, R^2_{121s'} & R^1_{122s'}, R^2_{122s'} \end{pmatrix} = \begin{pmatrix} 3,3 & 0,10 \\ 10,0 & 2,2 \end{pmatrix}$$

in state 1 and

$$\begin{pmatrix} R^1_{211s'}, R^2_{211s'} & R^1_{212s'}, R^2_{212s'} \\ R^1_{221s'}, R^2_{221s'} & R^1_{222s'}, R^2_{222s'} \end{pmatrix} = \begin{pmatrix} 4,4 & 0,10 \\ 10,0 & 1,1 \end{pmatrix}$$

in state 2 for $s' \in \{1, 2\}$, respectively. The corresponding state transition probabilities are given by

$$\begin{pmatrix} T_{1112} & T_{1122} \\ T_{1212} & T_{1222} \end{pmatrix} = \begin{pmatrix} T_{2111} & T_{2121} \\ T_{2211} & T_{2221} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix}.$$

To be precise, the rewards in each state do not resemble a classical social dilemma situation. This is because if both agents would alternately cooperate and defect, both could receive a larger reward per time step compared to always cooperating. Hence, this stochastic game, as it was used in Refs. [33–35], presents more a coordination than a cooperation challenge to the agents. The multistate environment can here function as a coordination device.

A behavior profile in which one agent exploits the other in one state, while being exploited in the other state, would result in an average reward per time step of 5 for each agent, e.g., $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0, 1, 1, 0)$.

However, for all three learning types with a midranged farsightedness ($\gamma = 0.45$) and an intensity of choice $\beta = 5.0$, the temporal difference error arrows are pointing on average toward the lower-left defection-defection point for each state in behavior phase space (Fig. 8). To see whether the three learning types may converge to the described defect-cooperate–cooperate-defect equilibrium, individual trajectories from two exemplary initial conditions and for two learning rates α are shown, a small one ($\alpha = 0.02$) and a high one ($\alpha = 0.8$).

We observe qualitatively different behavior across all three learners. The Q learners converge to equilibria with average rewards distinctly below 5, and the SARSA learners converge to equilibira with average rewards of almost 5 for both learning rates α and both exemplary initial conditions. Both Q and SARSA learners converge to solutions of proper probabilistic behavior, i.e., choosing action cooperate and action defect with nonvanishing chance. The actor-critic learners, on the other hand, converge to the deterministic defect-cooperate–cooperate-defect behavior described above for the initial condition shown with the nondashed lines in Fig. 8 for both learning rates α (shown in light red and dark blue). For the other exemplary initial condition, shown with the dashed lines, it converges to an all-defection solution in both states for both α.

Interestingly, for all learners, all combinations of initial conditions and learning rates converge to a fixed point solution, except for the Q learners with a comparably high learning rate $\alpha = 0.8$, which enter a periodic behavior solution for the initial condition with the nondashed line. The same phenomenon occurred also in the matching pennies environment for low farsightedness $\gamma = 0.1$, however, there for both Q and SARSA learners. It seems to be caused by the comparably high learning rate. A high learning rate overshoots the behavior update, resulting in a circling behavior

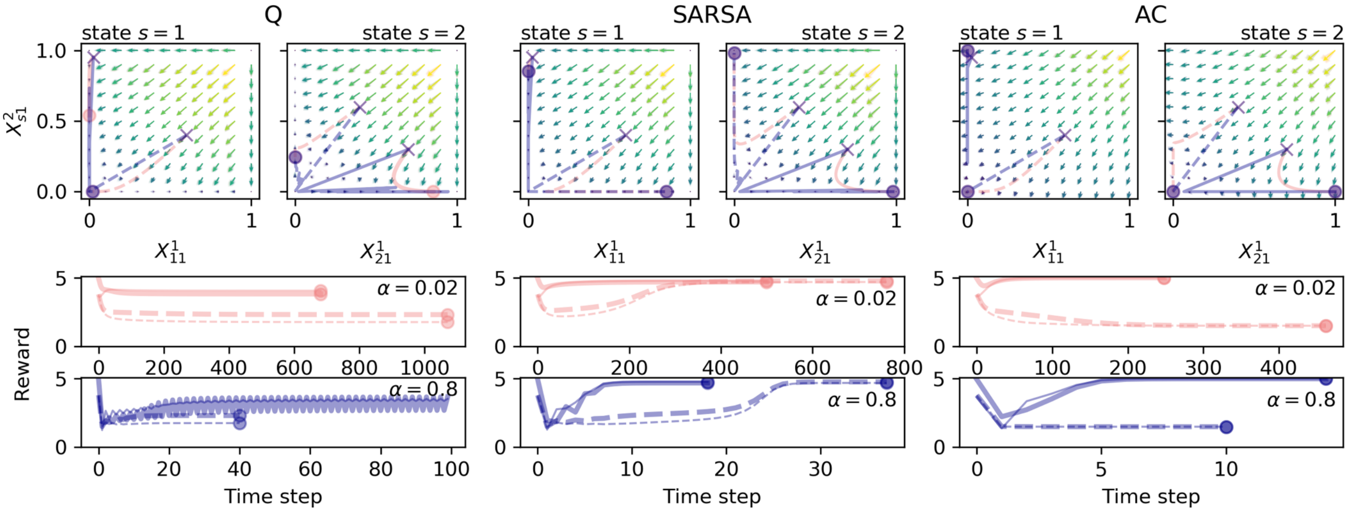

FIG. 8. Two-state prisoner's dilemma environment for discount factor $\gamma = 0.45$; otherwise, identical to Fig. 3.

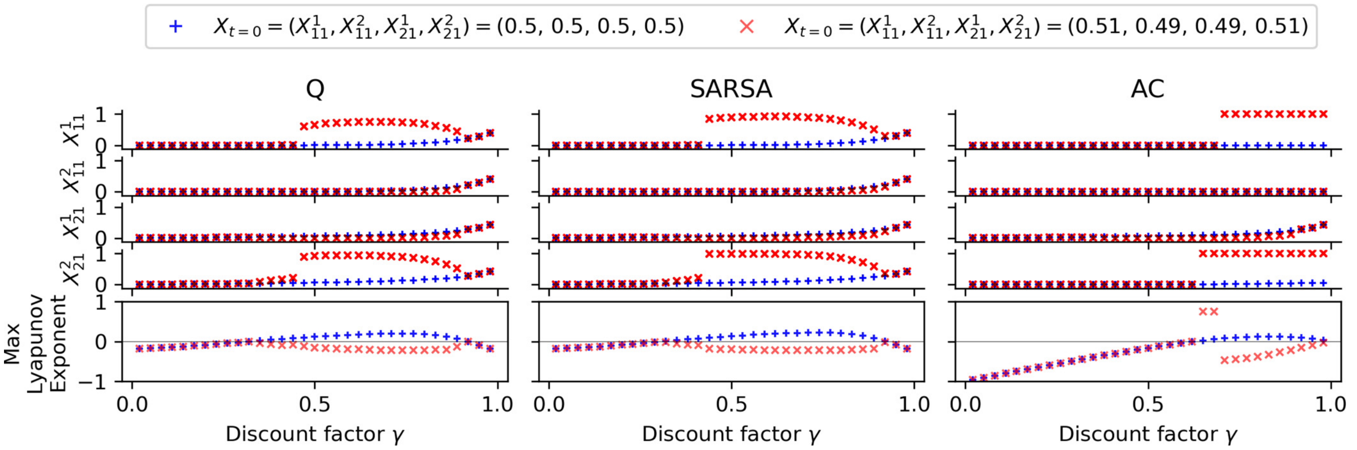

FIG. 9. Varying discount factor γ in two-state prisoner's dilemma environment for learning rate $\alpha = 0.2$ and intensity of choice $\beta = 5.0$ for the Q learners on the left, the SARSA learners in the middle, and the actor-critic learners on the right. The four top panels for each learner show the visited behavior points $X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}$ during 1000 iterations after a transient period of 5000 time steps from initial behavior $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$ in blue pluses and from initial behavior $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.51, 0.49, 0.49, 0.51)$ in red crosses. The bottom panels show the corresponding largest Lyapunov exponents for the two initial conditions. Above a critical discount factor γ all learners find the high rewarding solution from the red crosses initial condition but do not do so from the blue pluses initial condition.

around the fixed point. As in Fig. 3, the time average reward of the periodic orbit seems to be comparable to the reward of the corresponding fixed point at lower α. Furthermore, we observe the same time rescaling effect of the learning rate α in Fig. 8 as in Fig. 4.

To visualize the influence of the discount factor γ on the converged behavior, Fig. 9 shows a bifurcation diagram along the bifurcation parameter γ for two initial conditions. Pluses in blue result from a uniformly random behavior profile of $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.5, 0.5, 0.5, 0.5)$, whereas the crosses in red initially started from the behavior profile $(X^1_{11}, X^2_{11}, X^1_{21}, X^2_{21}) = (0.51, 0.49, 0.49, 0.51)$.

Across all learners, lower discount factors γ correspond to all-defect solutions, whereas for higher γ the solutions from the initial condition shown with red crosses tend toward the cooperate-defect–defect-cooperate solution. For low γ, the agents are less aware of the presence of other states and find the all-defect equilibrium solution of the iterated normal form prisoner's dilemma. The state transition probabilities have less effect on the learning dynamics. Only above a certain farsightedness do the agents find the more rewarding cooperate-defect–defect-cooperate solution.

The observation from Fig. 8 is confirmed that the probability to cooperate (i.e., here X^1_{11} and X^2_{21}) is lowest for the Q learners, midrange for the SARSA learners, and 1 for the actor-critic learners. One reason for this observation can be found in the intensity of choice parameter β. It balances the reward obtainable in the current behavior space segment with the forgetting of current knowledge to be open to new solutions. Such forgetting expresses itself by temporal difference error components pointing toward the center of behavior space. Thus, a relatively small $\beta = 5.0$ can explain why solutions at the edge of the behavior space cannot be reached by Q and SARSA learners. The AC learners mis this forgetting term in the deterministic limit and can therefore easily enter behavior profiles at the edge of the behavior space.

Q and SARSA learners have a critical discount factor γ above which the cooperate-defect–defect-cooperate high reward solution is obtained and below which the all-defect low reward solution gets selected. However, for increasing discount factors γ up to 1, Q and SARSA learners experience a drop in playing the cooperative action probability.

The actor-critic learners approach the cooperate-defect–defect-cooperate solution in two steps. For increasing γ, first the cooperation probability of agent 2 in state 2 (X^2_{21}) jumps from zero to 1 while agent 1 still defects in state 1. Only after a slight increase of γ does agent 1 then also cooperate in state 1 (X^1_{11}).

Interestingly, for the uniformly random initial behavior condition shown with blue pluses, there is no critical discount factor γ and no learners come close to the cooperate-defect–defect-cooperate solution. Here, only for γ close to 1 do all cooperation probabilities X^i_{s1} gradually increase. Furthermore, exactly at those γ, where the cooperate-defect–defect-cooperate solution is obtained from the initial behavior condition shown with red crosses, the solutions from the uniformly random initial behavior condition (blue pluses) have a largest Lyapuonv exponent greater than 0. At other values of γ, the largest Lyapunov exponents for the two initial conditions overlap. This suggests that the largest Lyapunov exponents greater than zero may point to the fact that other, perhaps more rewarding, solutions may exist in phase space. A more thorough investigation regarding this multistability is an open point for future research.

As we have argued above, the two-state prisoner's dilemma as it was used in Refs. [33–35] presents rather a coordination than a cooperation challenge to the agents. Figure 10 demonstrates that our learning dynamics are also capable of solving a cooperation challenge in a stochastic game setting, for which we adapt a two-state prisoner's dilemma in analogy to Ref. [45]. Figure 10 confirms previous findings that cooperation emerges only in the stochastic game, compared to

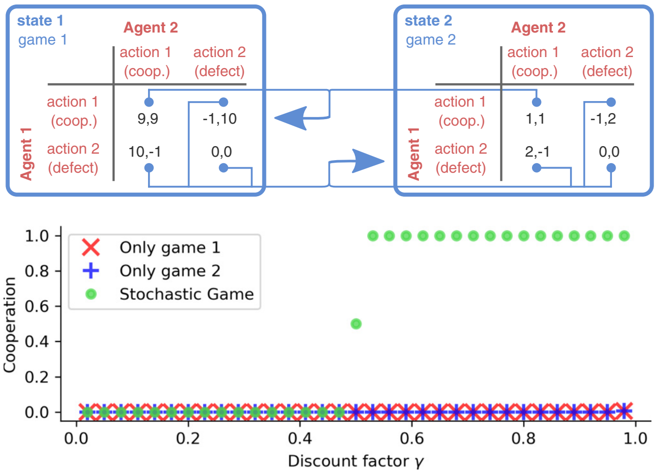

FIG. 10. Cooperation challenge in a two-state prisoner's dilemma. Top panel shows a two-state prisoner's dilemma game, whose state games individually favor defection. Bottom panel shows the level of cooperation SARSA learners with $\alpha = 0.016$, $\beta = 250$ play after reaching a fixed point from the center of behavior space ($X^i_{sa} = 0.5$ for all i, s, a) for varying discount factors γ. Results for Q and AC learners are similar. Cooperation levels are shown for the full stochastic game as well as for each individual state game played repeatedly. For sufficiently large farsightedness, cooperation can emerge in the stochastic game, in contrast to the individual repeated games.

playing each prisoner's dilemma repeatedly [45]. Further, cooperation only emerges for sufficiently large farsightedness γ.

V. DISCUSSION

The main contribution of this paper is the development of a technique to obtain the deterministic limit of temporal difference reinforcement learning. Through our work we have combined the literature on learning dynamics from statistical physics with the evolutionary game theory-inspired learning dynamics literature from machine learning. For the statistical physics community, we present learning equations capable of handling environmental state transitions. For the machine-learning community, we present the systematic methodology we have used to obtain the deterministic learning equations.

We have demonstrated our approach with the three prominent reinforcement learning algorithms from computer science: Q learning, SARSA learning, and actor-critic learning. A comparison of their dynamics in previously used two-agent, two-action, two-state environments has revealed the existence of a variety of qualitatively different dynamical regimes, such as convergence to fixed points, periodic orbits, and deterministic chaos.

We have found that Q and SARSA learners tend to behave qualitatively more similar in comparison to the actor-critic learning dynamics. This characteristic results at least partly from our relatively low intensity of choice parameter β, controlling the exploration-exploitation trade-off via a forgetting term in the temporal difference errors. Sending $\beta \to \infty$, the SARSA learning dynamics approach the actor-critic learning dynamics, as we have shown. Overall the actor-critic learners have a tendency to enter confining behavior profiles, due to their nonexisting forgetting term. This characteristic leaves them trapped at the edges of the behavior space. In contrast, Q and SARSA learner do not show such learning behavior. Interestingly, this characteristic of the AC learners turns out to be favorable in the two-state prisoner's dilemma environment, where they find the most rewarding solution in more cases compared to Q and SARSA but hinders the convergence to the fixed point solution in the two-state matching pennies environment. Thus, the most favorable level of forgetting depends on the environment. In order to tune the respective parameter β, our consideration that it must come in the unit of [log behavior]/[reward] may be helpful.

We have demonstrated the effect of the learning rate α adjusting the speed of learning by controlling the amount of new information used in a behavior profile update. Thereby, within limits, α functions as a time rescaling. However, a comparably large learning rate α might cause an overshooting phenomenon, hindering the convergence to a fixed point. Instead, the learners enter a limit cycle around that point. Nevertheless, the average reward of the limit-cycling behavior was approximately equal to the one of the fixed point obtained at lower α but took fewer time steps to reach. Thus, perhaps other dynamical regimes than fixed points, such as limit cycles or strange attractors, could be of interest in some applications of reinforcement learning.

We have also shown the effect of the discount factor γ adjusting the farsightedness of the agents. At low γ the state transition probabilities have less effect on the learning dynamics compared to high discount factors.

To summarize the three parameters α, β, and γ: The level of exploitation β and the farsightedness γ control *where* the learners adapt toward in behavior space, weighting current reward, expected future reward, and the level of forgetting. The learning rate α controls *how fast* the learners adapt along these directions.

We hope that our work might turn out useful for the application of reinforcement learning in various domains, with respect to parameter tuning, the design of new algorithms, and the analysis of complex strategic interactions using meta strategies, as Bloembergen *et al.* [28] have pointed out. In this regard, future work could extend the presented methodology to partial observability of the Markov states of the environment [40,41], behavior profiles with history, and other-regarding agent (i.e., joint-action) learners (cf. Ref. [2] for an overview of other-regarding agent learning algorithms). Also, the combination of individual reinforcement learning and social learning through imitation [47–50] seems promising. Such endeavors would naturally lead to the exploration of network effects. It is important to note that only a few dynamical systems reinforcement learning studies have begun to incorporate network structures between agents [22,23].

Apart from these more technical extensions, we hope that our learning equations will prove themselves useful when studying the evolution of cooperation in stochastic games [45]. With stochastic games one is able to explicitly account for a changing environment. Therefore, such studies are likely to contribute to the advancement of theoretical research on the sustainability of interlinked social-ecological systems [51,52]. Interactions, synergies, and trade-offs between social [13,53] and ecological [54] dilemmas can be explored using the framework of stochastic games. More realistic environments, modeling, e.g., the harvesting of common-pool renewable resources [55,56] or the prevention of dangerous climate change [57,58], for our learning dynamics are likely to prove themselves useful. Here, it may be of interest to evaluate the learning process not only in terms of efficiency but also how close it came to the optimal behavior. Other paradigms than value optimization may also be important [59], such as sustainability or resilience [60].

Python code for reproduction of the figures of this article is available online at [61].

ACKNOWLEDGMENTS

This work was developed in the context of the COPAN project on Coevolutionary Pathways in the Earth System at the Potsdam Institute for Climate Impact Research (PIK). We are grateful for financial support by the Heinrich Böll Foundation, the Stordalen Foundation (via the Planetary Boundaries Research Network PB.net), the Earth League's EarthDoc program, and the Leibniz Association (project DOMINOES). We thank Jobst Heitzig for discussions and comments on the manuscript.

APPENDIX: COMPUTATION OF LYAPUNOV EXPONENTS

We compute the Lyapunov exponents using an iterative QR decomposition of the Jacobian matrix according to Sandri [46]. In the following we present the derivation of the Jacobian matrix.

Equation (11) constitutes a map f, which iteratively updates the behavior profile $\mathbf{X} \in \mathbb{R}^{N \times M \times Z}$. Consequently, we can represent its derivative as a Jacobian tensor $f'(\mathbf{X}) \in \mathbb{R}^{N \times M \times Z \times N \times M \times Z}$.

Let $A^i_{sa} := X^i_{sa} \exp[\alpha^i \beta^i D^i_{sa}(\mathbf{X})]$ be the numerator of Eq. (11) and $B^i_s := \sum_b A^i_{sb}$ its denominator, i.e., $f =: A/B$. Hence,

$$f'(\mathbf{X}) = \frac{A'B - B'A}{B^2} \tag{A1}$$

or, more precisely, in components,

$$\frac{df^i_{sa}(\mathbf{X})}{dX^j_{rb}} = \frac{\frac{dA^i_{sa}(\mathbf{X})}{dX^j_{rb}} B^i_s(\mathbf{X}) - \frac{dB^i_s}{dX^j_{rb}}(\mathbf{X}) A^i_{sa}(\mathbf{X})}{[B^i_s(\mathbf{X})]^2}. \tag{A2}$$

A and B are known, and if A' is known, then B' is easily obtained by $\frac{dB^i_s(\mathbf{X})}{dX^j_{rb}} = \sum_c \frac{dA^i_{sc}(\mathbf{X})}{dX^j_{rb}}$. Therefore we need to compute A' for the three learner types Q, SARSA, and actor-critic learning.

1. Q learning

Let us rewrite A^i_{sa} for the Q learner according to

$$A^i_{sa} := \left(X^i_{sa}\right)^{(1-\alpha^i)} \exp\left[\alpha^i \beta^i \hat{D}^i_{sa}(\mathbf{X})\right], \tag{A3}$$

where we removed the estimate of the current value from the temporal difference error, leaving the truncated temporal difference error as

$$\hat{D}^i_{sa}(\mathbf{X}) := (1 - \gamma^i)\, {}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} + \gamma^{i\,\max} Q^i_{sa}(\mathbf{X}). \tag{A4}$$

Hence, we can write the derivative of A as

$$\frac{dA^i_{sa}(\mathbf{X})}{dX^j_{rb}} = \exp\left[\alpha^i \beta^i \hat{D}^i_{sa}(\mathbf{X})\right]\left[(1 - \alpha^i)\left(X^i_{sa}\right)^{-\alpha^i} \frac{dX^i_{sa}}{dX^j_{rb}} + \alpha^i \beta^i \left(X^i_{sa}\right)^{(1-\alpha^i)} \frac{d\hat{D}^i_{sa}(\mathbf{X})}{dX^j_{rb}}\right]. \tag{A5}$$

Since $\sum_c X^i_{sc} = 1$, dX^i_{sa}/dX^j_{rb} can be expressed as

$$\frac{dX^i_{sa}}{dX^j_{rb}} = \delta_{ij}\delta_{sr}(2\delta_{ab} - 1). \tag{A6}$$

The derivative of the truncated temporal difference error reads

$$\frac{d\hat{D}^i_{sa}(\mathbf{X})}{dX^j_{rb}} = (1 - \gamma^i)\frac{d\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}}{dX^j_{rb}} + \gamma^i \frac{d^{\max} Q^i_{sa}(\mathbf{X})}{dX^j_{rb}}. \tag{A7}$$

Let us write the derivative of the reward as

$$\frac{d\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}}{dX^j_{rb}} = \sum_{s'} \sum_{\mathbf{a}^{-i}} \frac{d\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}}{dX^j_{rb}} T_{sa\mathbf{a}^{-i}s'} R^i_{sa\mathbf{a}^{-i}s'} \tag{A8}$$

using Eq. (2) and Eq. (3), where the derivatives $d\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}/dX^j_{rb}$ need to be executed according to Eq. (A6).

For the derivative of the maximum next value we write accordingly

$$\frac{d^{\max} Q^i_{sa}(\mathbf{X})}{dX^j_{rb}} = \sum_{s'} \sum_{\mathbf{a}^{-i}} \frac{d\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}}{dX^j_{rb}} T_{sa\mathbf{a}^{-i}s'} \max_c Q^i_{s'c}(\mathbf{X}) + \sum_{s'} \sum_{\mathbf{a}^{-i}} \mathbf{X}^{-i}_{s\mathbf{a}^{-i}} T_{sa\mathbf{a}^{-i}s'} \frac{d \max_c Q^i_{s'c}(\mathbf{X})}{dX^j_{rb}}. \tag{A9}$$

Let $a^m := \arg\max_a Q^i_{sa}(\mathbf{X})$, then

$$\frac{d \max_c Q^i_{sc}(\mathbf{X})}{dX^j_{rb}} = \delta_{aa^m} \frac{dQ^i_{sa}(\mathbf{X})}{dX^j_{rb}} \tag{A10}$$

and

$$\frac{dQ^i_{sa}(\mathbf{X})}{dX^j_{rb}} = (1 - \gamma^i)\frac{d\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}}{dX^j_{rb}} + \gamma^i \sum_{s'} \frac{d\,{}_{\mathbf{X}}\langle T\rangle_{ss'}}{dX^j_{rb}} V^i_{s'}(\mathbf{X}) + {}_{\mathbf{X}}\langle T\rangle_{ss'} \frac{dV^i_{s'}(\mathbf{X})}{dX^j_{rb}}. \tag{A11}$$

For the derivative of the effective Markov Chain transition tensor we can write

$$\frac{d\,{}_{\mathbf{X}}\langle T\rangle_{ss'}}{dX^j_{rb}} = \sum_{\mathbf{a}} \frac{d\mathbf{X}_{s\mathbf{a}}}{dX^j_{rb}} T_{sa\mathbf{a}^{-i}s'}, \tag{A12}$$

using Eqs. (2) and (3), where again the derivatives $d\mathbf{X}_{s\mathbf{a}}/dX^j_{rb}$ need to be executed according to Eq. (A6).

For the derivative of the state value let us rewrite Eq. (6) as $V^i_s = (1-\gamma^i)\sum_{s'} M^{-1}_{ss'}\,{}_{\mathbf{TX}}\langle R\rangle^i_{s'}$ with $M := (\mathbb{1}_Z - \gamma^i\,{}_{\mathbf{X}}\langle T\rangle)$. Thus,

$$\frac{dV^i_s(\mathbf{X})}{dX^j_{rb}} = (1-\gamma^i)\sum_{s''}\frac{d\left(M^{-1}_{ss''}\right)}{dX^j_{rb}}\,{}_{\mathbf{TX}}\langle R\rangle^i_{s''} + M^{-1}_{ss''}\frac{d\,{}_{\mathbf{TX}}\langle R\rangle^i_{s''}}{dX^j_{rb}}. \tag{A13}$$

To obtain the derivative of the inverse matrix M^{-1} we use $(M^{-1}M)' = 0 = (M^{-1})'M + M^{-1}M'$ and therefore $(M^{-1})' = -M^{-1}M'M^{-1}$. For M' we write

$$\frac{dM_{ss'}}{dX^j_{rb}} = -\gamma^i\frac{d\,{}_{\mathbf{X}}\langle T\rangle_{ss'}}{dX^j_{rb}}. \tag{A14}$$

We obtain the derivative of the reward according to

$$\frac{d\,{}_{\mathbf{TX}}\langle R\rangle^i_s}{dX^j_{rb}} = \sum_{s'}\sum_{\mathbf{a}}\frac{d\mathbf{X}_{s\mathbf{a}}}{dX^j_{rb}}T_{s\mathbf{a}s'}R^i_{s\mathbf{a}s'}, \tag{A15}$$

using Eq. (1) and Eq. (3), where the derivatives dX^i_{sa}/dX^j_{rb} need to be executed according to Eq. (A6).

Now we can compute the Jacobian matrix for the Q learning dynamics in their deterministic limit.

2. SARSA learning

The computation of the Jacobian matrix for the SARSA learning update in its deterministic limit is similar, except the truncated temporal difference error reads

$$\hat{D}^i_{sa}(\mathbf{X}) := (1-\gamma^i)\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} + \gamma^i\,{}^{\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X}) \tag{A16}$$

instead of Eq. (A4). Hence,

$$\frac{d\hat{D}^i_{sa}(\mathbf{X})}{dX^j_{rb}} = (1-\gamma^i)\frac{d\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa}}{dX^j_{rb}} + \gamma^i\frac{d\,{}^{\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X})}{dX^j_{rb}} \tag{A17}$$

and

$$\begin{aligned}\frac{d\,{}^{\text{next}}\mathcal{Q}^i_{sa}(\mathbf{X})}{dX^j_{rb}} &= \sum_{s'}\sum_{\mathbf{a}^{-i}}\frac{d\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}}{dX^j_{rb}}T_{sa\mathbf{a}^{-i}s'}\sum_c X^i_{s'c}\mathcal{Q}^i_{s'c}(\mathbf{X})\\ &\quad + \sum_{s'}\sum_{\mathbf{a}^{-i}}\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}T_{sa\mathbf{a}^{-i}s'}\frac{d\left[\sum_c X^i_{s'c}\mathcal{Q}^i_{s'c}(\mathbf{X})\right]}{dX^j_{rb}}.\end{aligned} \tag{A18}$$

The derivative of $\sum_c X^i_{s'c}\mathcal{Q}^i_{s'c}(\mathbf{X})$ reads

$$\frac{d\left[\sum_c X^i_{s'c}\mathcal{Q}^i_{s'c}(\mathbf{X})\right]}{dX^j_{rb}} = \sum_c\left[\frac{dX^i_{s'c}}{dX^j_{rb}}\mathcal{Q}^i_{s'c}(\mathbf{X}) + X^i_{s'c}\frac{d\mathcal{Q}^i_{s'c}}{dX^j_{rb}}\right]. \tag{A19}$$

All remaining terms have already been given in the previous section for the Q learning Jacobian matrix.

3. Actor-critic learning

For the actor-critic learning update, Eq. (A3) reads

$$A^i_{sa} := X^i_{sa}\exp\left[\alpha^i\beta^i\hat{D}^i_{sa}(\mathbf{X})\right], \tag{A20}$$

with the truncated temporal difference error

$$\hat{D}^i_{sa}(\mathbf{X}) := (1-\gamma^i)\,{}_{\mathbf{TX}^{-i}}\langle R\rangle^i_{sa} + \gamma^i\,{}^{\text{next}}\mathcal{V}^i_{sa}(\mathbf{X}). \tag{A21}$$

The derivative of the next value estimate is obtained by

$$\begin{aligned}\frac{d\,{}^{\text{next}}\mathcal{V}^i_{sa}(\mathbf{X})}{dX^j_{rb}} &= \sum_{s'}\sum_{\mathbf{a}^{-i}}\frac{d\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}}{dX^j_{rb}}T_{sa\mathbf{a}^{-i}s'}V^i_{s'}(\mathbf{X})\\ &\quad + \sum_{s'}\sum_{\mathbf{a}^{-i}}\mathbf{X}^{-i}_{s\mathbf{a}^{-i}}T_{sa\mathbf{a}^{-i}s'}\frac{dV^i_{s'}(\mathbf{X})}{dX^j_{rb}}.\end{aligned} \tag{A22}$$

The derivative of the next value $V^i_{s'}$ is given by Eq. (A13). These are all terms necessary to compute the Jacobian matrix for the actor-critic learning update.

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 1998).

[2] L. Busoniu, R. Babuska, and B. De Schutter, A comprehensive survey of multiagent reinforcement learning, IEEE Trans. Syst. Man Cybernet. C **38**, 156 (2008).

[3] M. Wiering and M. van Otterlo, *Reinforcement Learning: State-of-the-Art* (Springer Verlag, Berlin, 2012).

[4] A. Shah, Psychological and neuroscientific connections with reinforcement learning, in *Reinforcement Learning* (Springer, Berlin, 2012), pp. 507–537.

[5] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, Neuroscience-inspired artificial intelligence, Neuron **95**, 245 (2017).

[6] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, Vol. 2 (MIT Press, Cambridge, MA, 1998).

[7] A. E. Roth and I. Erev, Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term, Games Econ. Behav. **8**, 164 (1995).

[8] I. Erev and A. E. Roth, Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria, Am. Econ. Rev. **88**, 848 (1998).

[9] C. Camerer and T. Hua Ho, Experienced-weighted attraction learning in normal form games, Econometrica **67**, 827 (1999).

[10] C. Camerer, *Behavioral Game Theory: Experiments in Strategic Interaction* (Princeton University Press, Princeton, NJ, 2003).

[11] W. B. Arthur, Designing economic agents that act like human agents: A behavioral approach to bounded rationality, Am. Econ. Rev. **81**, 353 (1991).

[12] W. B. Arthur, Complexity and the economy, Science **284**, 107 (1999).

[13] M. W. Macy and A. Flache, Learning dynamics in social dilemmas, Proc. Natl. Acad. Sci. USA **99**, 7229 (2002).

[14] J. G. Cross, A stochastic learning model of economic behavior, Quart. J. Econ. **87**, 239 (1973).

[15] T. Börgers and R. Sarin, Learning through reinforcement and replicator dynamics, J. Econ. Theory **77**, 1 (1997).

[16] M. Marsili, D. Challet, and R. Zecchina, Exact solution of a modified el farol's bar problem: Efficiency and the role of market impact, Physica A **280**, 522 (2000).

[17] Y. Sato, E. Akiyama, and J. D. Farmer, Chaos in learning a simple two-person game, in Ref. [18], pp. 4748–4751.

[18] Y. Sato and J. P. Crutchfield, Coupled replicator equations for the dynamics of learning in multiagent systems, in Ref. [15], p. 015206.

[19] Y. Sato, E. Akiyama, and J. P Crutchfield, Stability and diversity in collective adaptation, Physica D **210**, 21 (2005).

[20] T. Galla, Intrinsic Noise in Game Dynamical Learning, Phys. Rev. Lett. **103**, 198702 (2009).

[21] T. Galla, Cycles of cooperation and defection in imperfect learning, J. Stat. Mech.: Theory Exp. (2011) P08007.

[22] A. J. Bladon and T. Galla, Learning dynamics in public goods games, Phys. Rev. E **84**, 041132 (2011).

[23] J. Realpe-Gomez, B. Szczesny, L. Dall'Asta, and T. Galla, Fixation and escape times in stochastic game learning, J. Stat. Mech.: Theory Exp. (2012) P10022.

[24] J. B. T. Sanders, T. Galla, and J. L. Shapiro, Effects of noise on convergent game-learning dynamics, J. Phys. A: Math. Theor. **45**, 105001 (2012).

[25] T. Galla and J. D. Farmer, Complex dynamics in learning complicated games, Proc. Natl. Acad. Sci. USA **110**, 1232 (2013).

[26] A. Aloric, P. Sollich, P. McBurney, and T. Galla, Emergence of cooperative long-term market loyalty in double auction markets, PloS one **11**, e0154606 (2016).

[27] K. Tuyls, K. Verbeeck, and T. Lenaerts, A selection-mutation model for q-learning in multi-agent systems, in *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems* (ACM, New York, 2003), pp. 693–700.

[28] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, Evolutionary dynamics of multi-agent learning: A survey, J. Artif. Intell. Res. **53**, 659 (2015).

[29] K. Tuyls and A. Nowé, Evolutionary game theory and multi-agent reinforcement learning, Knowl. Eng. Rev. **20**, 63 (2005).

[30] K. Tuyls, P. Jan'T Hoen, and B. Vanschoenwinkel, An evolutionary dynamical analysis of multi-agent learning in iterated games, Auton. Agents Multi-Agent Syst. **12**, 115 (2006).

[31] K. Tuyls and S. Parsons, What evolutionary game theory tells us about multiagent learning, Artif. Intell. **171**, 406 (2007).

[32] M. Kaisers and K. Tuyls, Frequency adjusted multi-agent q-learning, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 1 (International Foundation for Autonomous Agents and Multiagent Systems, 2010), pp. 309–316.

[33] D. Hennes, K. Tuyls, and M. Rauterberg, State-coupled replicator dynamics, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 2 (International Foundation for Autonomous Agents and Multiagent Systems, 2009), pp. 789–796.

[34] P. Vrancx, K. Tuyls, and R. Westra, Switching dynamics of multi-agent learning, in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Volume 1 (International Foundation for Autonomous Agents and Multiagent Systems, 2008), pp. 307–313.

[35] D. Hennes, M. Kaisers, and K. Tuyls, Resq-learning in stochastic games, in *Proceedings of the AAMAS Workshop on Adaptive and Learning Agents*, May 2010, Toronto, Canada (2010), p. 8.

[36] L. S. Shapley, Stochastic games, Proc. Natl. Acad. Sci. USA **39**, 1095 (1953).

[37] J.-F. Mertens and A. Neyman, Stochastic games, Int. J. Game Theory **10**, 53 (1981).

[38] E. Akiyama and K. Kaneko, Dynamical systems game theory and dynamics of games, Physica D **147**, 221 (2000).

[39] E. Akiyama and K. Kaneko, Dynamical systems game theory II: A new approach to the problem of the social dilemma, Physica D **167**, 36 (2002).

[40] M. T. J. Spaan, Partially observable markov decision processes, in *Reinforcement Learning* (Springer, Berlin, 2012), pp. 387–414.

[41] F. A. Oliehoek, Decentralized pomdps, in *Reinforcement Learning* (Springer, Berlin, 2012), pp. 471–503.

[42] R. Bellman, A Markovian decision process, Ind. Univ. Math. J. **6**, 679 (1957).

[43] S. Lange, T. Gabel, and M. Riedmiller, Batch reinforcement learning, in *Reinforcement Learning* (Springer, Berlin, 2012), pp. 45–73.

[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, Human-level control through deep reinforcement learning, Nature **518**, 529 (2015).

[45] C. Hilbe, Š. Šimsa, K. Chatterjee, and M. A. Nowak, Evolution of cooperation in stochastic games, Nature **559**, 246 (2018).

[46] M. Sandri, Numerical calculation of Lyapunov exponents, Math. J. **6**, 78 (1996).

[47] A. Bandura, *Social Learning Theory* (Prentice-Hall, Upper Saddle River, NJ, 1977), pp. 15–55.

[48] M. Smolla, R. T. Gilman, T. Galla, and S. Shultz, Competition for resources can explain patterns of social and individual learning in nature, Proc. Roy. Soc. Lond. B **282** 20151405 (2015).

[49] W. Barfuss, J. F. Donges, M. Wiedermann, and W. Lucht, Sustainable use of renewable resources in a stylized social-ecological network model under heterogeneous resource distribution, Earth Syst. Dynam. **8**, 255 (2017).

[50] S. Banisch and E. Olbrich, Opinion polarization by learning from social feedback, J. Math. Soc. **43**, 76 (2019).

[51] S. A. Levin, The mathematics of sustainability, Not. Am. Math. Soc. **60**, 392 (2013).

[52] J. F. Donges, R. Winkelmann, W. Lucht, S. E. Cornell, J. G. Dyke, J. Rockström, J. Heitzig, and H. J. Schellnhuber, Closing the loop: Reconnecting human dynamics to Earth System science, Anthrop. Rev. **4**, 151 (2017).

[53] R. M. Dawes, Social dilemmas, Annu. Rev. Psychol. **31**, 169 (1980).

[54] J. Heitzig, T. Kittel, J. F. Donges, and N. Molkenthin, Topology of sustainable management of dynamical systems with desirable states: From defining planetary boundaries to safe operating spaces in the Earth system, Earth Syst. Dynam. **7**, 21 (2016).

[55] E. Lindkvist and J. Norberg, Modeling experiential learning: The challenges posed by threshold dynamics for sustainable renewable resource management, Ecol. Econ. **104**, 107 (2014).

[56] C. Schill, T. Lindahl, and A.-S. Crépin, Collective action and the risk of ecosystem regime shifts: Insights from a laboratory experiment, Ecol. Soc. **20**, 48 (2015).

[57] M. Milinski, R. D. Sommerfeld, H.-J. Krambeck, F. A. Reed, and J. Marotzke, The collective-risk social dilemma and the

prevention of simulated dangerous climate change, Proc. Natl. Acad. Sci. USA **105**, 2291 (2008).

[58] S. Barrett and A. Dannenberg, Climate negotiations under scientific uncertainty, Proc. Natl. Acad. Sci. USA **109**, 17372 (2012).

[59] W. Barfuss, J. F. Donges, S. J. Lade, and J. Kurths, When optimization for governing human-environment tipping elements is neither sustainable nor safe, Nat. Commun. **9**, 2354 (2018).

[60] J. F. Donges and W. Barfuss, From math to metaphors and back again: Social-ecological resilience from a multi-agent-environment perspective, GAIA **26**, 182 (2017).

[61] W. Barfuss, wbarfuss/DetRL: Release for reference in publication, Zenodo (2018), http://doi.org/10.5281/zenodo.1495091.