Gradient dynamics in reinforcement learning

Riccardo Fabbricatore and Vladimir V. Palyulin

Phys. Rev. E **106**, 025315 — Published 26 August 2022

DOI: 10.1103/PhysRevE.106.025315

Gradient dynamics in reinforcement learning

Riccardo Fabbricatore[1, *] and Vladimir V. Palyulin[1, †]

[1] *Skolkovo Institite of Science and Technology, 121205, Moscow, Russia*

(Dated: June 21, 2022)

Despite the success achieved by the analysis of supervised learning algorithms in the framework of statistical mechanics, reinforcement learning has remained largely untouched by physicists. Here we move towards closing the gap by analyzing the dynamics of the policy gradient algorithm. For a convex problem, namely the k-armed bandit, we show that the learning dynamics obeys a drift-diffusion motion described by a Langevin equation, which coefficients can be tuned by the learning rate. We explore the striking similarity between our Langevin equation and the Kimura equation, describing genotypes evolution. Furthermore, we propose a mapping between a non-convex reinforcement learning setting describing multiple joints of a robotic arm, and a disordered system, namely a p-spin glass. This novel mapping enables us to show how the learning rate acts as an effective temperature and thus is capable of smoothing rough landscapes, corroborating what is displayed by the drift-diffusive description and paving the way for physics-inspired algorithmic optimization based on annealing procedures in disordered systems.

I. INTRODUCTION

Statistical mechanics is a powerful tool for understanding and constructing optimization algorithms. On one hand, disordered systems, such as spin glasses or polymers, prompted the development of new algorithms (simulated annealing [1], cluster algorithms [2], hysteric optimization [3]). On the other hand, existing optimization algorithms have often been fruitfully analyzed in the statistical physics' framework, yielding knowledge about their behavior, phase transitions and possible improvement [4–8].

In recent years, the vast class of machine learning algorithms [9] has enjoyed a great deal of attention. Neural networks [10, 11] are nowadays used to predict protein folding [12], search for exotic particles in high-energy colliders [13], predict phase transitions in ferromagnetic models [14] as well as properties of liquid crystals with exceptional accuracy [15, 16], and in many other fields [17]. At the same time, reinforcement learning [18, 19] has proven to be a valuable tool for finding optimal jet grooming strategies [20], in the pursue of the conformal bootstrap program [21], or in the engineering of smart active matter [22]. Nonetheless, numerous questions about the algorithms' functioning remain unanswered [23, 24]. Great progress has been made in the study of neural networks, the analogy between their highly non-convex loss function landscapes and the free energy landscape of disordered systems has been extensively studied [25–27]. It has been shown how the stochastic gradient descent algorithm [28, 29] is prone to lead the network's weights towards a needed suboptimal, robust, and well-generalizing region [30, 31]. However, all the results above are applicable to supervised learning problems, which can be mapped to disordered systems by interpreting the loss function as a Hamiltonian.

Despite their late successes, reinforcement learning algorithms have not yet received such analysis. This is perhaps due to the lack of a clear mapping between RL problems and disordered systems. We try to overcome this gap by studying a subset of reinforcement learning algorithms named policy gradients (PG) [32, 33]. PG are the most universal training methods for reward-driven learning, they can be applied without additional knowledge of the agent's surrounding. Their main disadvantage is their tendency to converge to local maxima, thus learning a peculiar behavior, heavily dependent on the initial parameters. Nonetheless, PG-based algorithms were applied with a tremendous success in areas such as robotics [34], natural language processing [35], and games [36]. A proper understanding of the reasons of this success is still an open question. We obtain a description for the learning process in a convex landscape in terms of drift-diffusion dynamics. By mapping a non-convex RL setting to a spin glass at a finite temperature, we are able to explain the effect of hyperparameters on the learning success thanks to a mean-field analysis. As it turns out, the learning rate is coupled to the temperature and, thus, its variation allows one to perform an annealing.

II. THE REINFORCEMENT LEARNING FRAMEWORK

The typical reinforcement learning setting, the so-called *Markov decision process* [37], consists of an agent acting in an environment with the purpose of maximizing a given utility function. The agent bases its decisions on the environmental *state* $s \in \mathcal{S}$, choosing an *action* $a \in \mathcal{A}$, according to its *policy* $\pi(a|s)$. Subsequently, it receives a feedback from the environment in terms of a *reward* $R \in \mathbb{R}$ and the state of the environment changes to a new one $s \to s'$. The reward is generated from a distribution conditioned to the state and the chosen action $q(r|s, a)$ and the transition between states is governed by the probability density $p(s'|s, a)$. From this new state, a

[*] ri.fabbricatore@gmail.com
[†] v.palyulin@gmail.com

new action can be taken, generating again a new reward and a new state-transition. The sequence of rewards obtained through this iteration is the agent's maximization goal. The central evaluated quantity is the *return*: $G = \sum_{t=0}^{\infty} R_t \gamma^t$, i.e. the sum of the obtained reward sequence discounted by a factor γ, $0 \leq \gamma < 1$, which tunes the importance of memory. Note that we used capital letters for R and G because they are, in general, stochastic variables. The utility function of the agent is the average return: $Q_\pi(s,a) = E_{\pi,p,q}[G|S_0 = s, A_0 = a]$. Denoting the distribution of initial states $\rho_0(s)$, the expected return of the policy π reads:

$$J_\pi = \sum_s \rho_0(s) \sum_a \pi(a|s) Q_\pi(s,a). \tag{1}$$

Reinforcement learning aims to efficiently find a policy π that maximizes J_π. In general, the agent does not know the rules that govern the environment (e.g. p and q), and it must build its strategy based on the information that it acquires while learning.

Here we analyze *policy gradient* algorithm [18]. It exploits the well-known idea of gradient ascent to find the maximum of the return function (1). In this case the policy $\pi(a|s,\boldsymbol{\theta})$ is parametrized with a d-dimensional set of numbers $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_d\}$. The gradient ascent consists in updating these parameters in the direction of the steepest ascent of the average return (1). At state s and for action a it can be proven to be $\partial_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = E_{\pi,q,p}\left[Q_\pi(s,a)\partial_{\boldsymbol{\theta}} \log \pi(a|s,\boldsymbol{\theta})\right]$. However, since the agent does not know how to compute this average (it does not know p and q, as well as the utility function), it has to rely on an estimate of this gradient. One solution is to use the quantity $(G(s,a) - h(s))\partial_{\boldsymbol{\theta}} \log \pi(a|s,\boldsymbol{\theta})$, where $G(s,a)$ is an estimate of the quality function, and $h(s)$ is an arbitrary action-independent function called *baseline*. At each time step t, the new parameters $\boldsymbol{\theta}_{(t+1)}$ will be derived from the current ones $\boldsymbol{\theta}_{(t)}$ by adding the gradient, multiplied by a coefficient α, called *learning rate*. To render the procedure invariant from the policy parametrization, one can fix the Kullback-Leibler divergence $D(\pi_{t+1}||\pi_t)$ at all steps, therefore obtaining the so-called *natural policy gradient* [38, 39]:

$$\begin{aligned}\boldsymbol{\theta}_{(t+1)} = \boldsymbol{\theta}_{(t)} + \alpha\, F^{-1}_{(t)}\ &\left(G(s_{(t)},a_{(t)}) - h(s_{(t)})\right) \\ &\times \partial_{\boldsymbol{\theta}} \log \pi(a_{(t)}|s_{(t)},\boldsymbol{\theta}_{(t)}),\end{aligned} \tag{2}$$

where

$$(F)_{ij} = E_\pi\left[\partial_{\theta_i} \log \pi(a|s,\boldsymbol{\theta}) \partial_{\theta_j} \log \pi(a|s,\boldsymbol{\theta})\right]. \tag{3}$$

The matrix F is the *Fisher information metric* of the policy for the parameters $\boldsymbol{\theta}$ [40]. There are several ways to choose $G(s,a)$, defining different types of policy gradient algorithms. One straightforward possibility is to compute the future return by sampling the rewards for the next step of the process at fixed policy. This procedure is called *reinforce policy gradient* [32].

III. DIFFUSION APPROXIMATION FOR ONE-DIMENSIONAL K-ARMED BANDIT

We will begin our analysis by studying a case in which a single agent can use k actions in an environment composed of only one state. Such a problem is known in literature as *k-armed bandit* [41] since it is analogous to a slot machine with k arms, for which the player must infer which arms give better rewards, whilst trying to maximize his win. We will start with a scenario with only two possible actions: $\mathcal{A} = \{1, 2\}$. Since the gradient is not affected by the particular parametrization choice, we will use the convenient softmax function:

$$\pi(1|\theta) = x(\theta) = \frac{1}{1+e^{-\theta}}, \quad \pi(2|\theta) = 1 - x(\theta). \tag{4}$$

At every step t, the agent will choose actions 1 and 2 with probabilities $x(t) \equiv x(\theta(t))$ and $1 - x(t)$, respectively. This will yield the total average return (1) for $\gamma = 0$:

$$J(\theta(t)) = x(t)R_1 + (1 - x(t))R_2, \tag{5}$$

where R_a represent the stochastic reward extracted from its corresponding distribution $R_a \sim q_a = \mathcal{N}(r_a, \sigma_a)$. The bandit setting allows us to choose a zero discount factor $\gamma = 0$ without losing generality since the best policy is independent of it and we will keep this through the rest of the paper.

Our aim is to obtain an effective stochastic description of the temporal evolution of the learning process, i.e. of the trajectory of the policy $x(t)$. In supervised learning, the effective noise of stochastic gradient descent is often modeled by heavy-tailed distributions [42, 43]. In our case, since the stochasticity is induced by uncorrelated Gaussian fluctuations in the rewards, we can describe the process in terms of a Langevin equation:

$$\frac{dx}{dt} = u(x) + \sqrt{2D(x)} \cdot \eta_t, \tag{6}$$

where η_t is white Gaussian noise with zero mean and correlation $E_t[\eta_\tau \eta_{\tau'}] = \delta(\tau - \tau')$. To this end, we expand the policy for small α by Taylor series:

$$dx(t) = \left.\frac{dx}{d\theta}\right|_{\theta=\theta_{(t)}} d\theta_{(t)} + \frac{1}{2}\left.\frac{d^2x}{d\theta^2}\right|_{\theta=\theta_{(t)}} d\theta^2_{(t)} + o(\alpha^2). \tag{7}$$

Substituting the parameter update (2) in this expression, and computing the derivatives of (4), we obtain the policy increments. The drift and the diffusion terms are given by the average and the variance of these increments, $u(x) = E_t[\dot{x}(t)|x(t)]$, and $D(x) = \mathrm{Var}_t[\dot{x}(t)|x(t)]/2$. We refer the reader to the Supplemental Material for a thorough derivation of these terms, while reporting here only their final form obtained by expanding up to the second order in α:

$$
\begin{aligned}
u(x) &= \underbrace{\alpha x(1-x)(r_1-r_2)}_{\text{Selection}} + \underbrace{\frac{\alpha^2}{2}(1-2x)\; m}_{\text{Mutations}}, \\
D(x) &= \underbrace{\frac{\alpha^2}{2}x(1-x)d_1 + \frac{\alpha^4}{4}(1-2x)^2 d_2}_{\text{Random genetic drift}}.
\end{aligned}
\tag{8}
$$

The three coefficients m, d_1 and d_2 are positive and depend on the reward variances as well as the policy, the average rewards, and the baseline:

$$
\begin{aligned}
m &= (1-x)\left(\sigma_{R1}^2 + l_1^2\right) + x\left(\sigma_{R2}^2 + l_2^2\right), \\
d_1 &= (1-x)\sigma_{R1}^2 + x\sigma_{R2}^2 + \left[(1-x)l_1 + xl_2\right]^2, \\
d_2 &= (1-x)^2\frac{(3-x)c_1^2 - 2l_1^4}{x} + x^2\frac{(2+x)c_2^2 - 2l_2^4}{1-x},
\end{aligned}
\tag{9}
$$

where $c_a = \sigma_a^2 + l_a^2$ and $l_a = r_a - h$.

It is interesting to highlight the similarity with an evolving population of competing species/genotypes, described by the Kimura equation [44, 45]:

$$
\begin{aligned}
u_K(x) &= \underbrace{x(1-x)(f_1-f_2)}_{\text{Selection}}\underbrace{-\mu_{12}x + \mu_{21}(1-x)}_{\text{Mutations}}, \\
D_K(x) &= \underbrace{\frac{1}{2N}x(1-x)}_{\text{Random genetic drift}} \quad ,
\end{aligned}
\tag{10}
$$

where f_i is the fitness of the genotype i, μ_{ij} is the mutation rate from genotype i to j, and N is the population size. The mapping can be done by identifying genotypes with the actions and the policy of each action with the genotype frequency. In contrast to our expansion, the Kimura equation is obtained by manually adding the evolutionary forces: *selection*, *mutation* and *random genetic drift*. Our derivation can perhaps be considered more natural and clearly shows the symmetry between the deterministic and stochastic forces, adding a term proportional to $(1-2x)$ in the diffusion coefficient.

It is easy now to grasp how this dynamics evolves and how it is affected by the algorithm's parameters. Figure 1 shows the overall gradient dynamics, as well as the individual effects of the drift coefficients on it. The term corresponding to natural selection attracts the policy towards the best action, while the mutation term pushes it away from pure strategies, i.e. $x \sim 0$ and $x \sim 1$. The intrinsic stochasticity of the algorithm appears in the diffusion coefficient (8): small learning rates confine stochasticity to the mixed strategy ($x \sim 1/2$), while higher rates will generate higher fluctuations in the vicinity of pure strategies, as shown in Appendix A.

These insights can be used to improve the dynamics' convergence by treating the learning rate as a dynamical variable, which can be tuned according to a time schedule [46]. The approximation in terms of an Itô stochastic equation allows us to use Itô's lemma to derive the optimal scheduling of the learning rate. This turns out to be $\alpha(t) \propto 1/\sqrt{t}$, which is consistent with the results for the so-called Exp3 algorithm [41], all details of the derivation can be found in Appendix B.

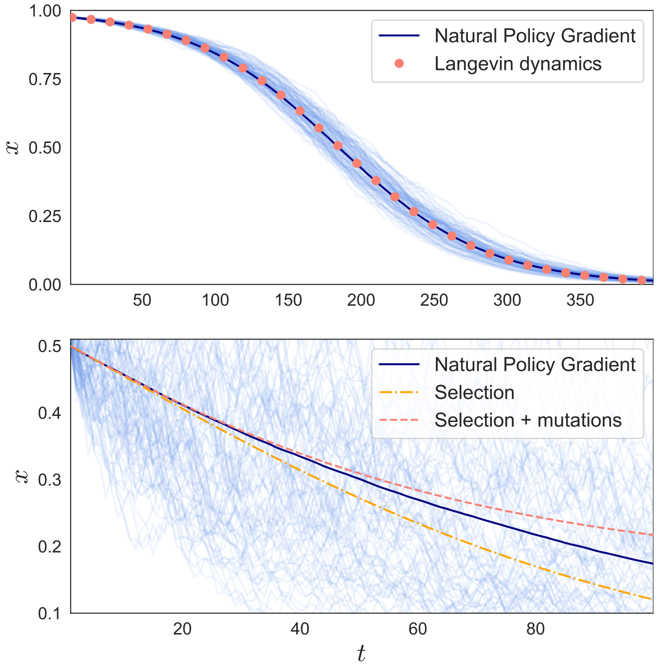

FIG. 1. Top: 10^2 lightly shaded trajectories of the action probability x generated by a natural policy gradient for the 2-armed bandit, along with their mean, compared to the Langevin dynamics (8). The rewards are distributed as $\mathcal{N}_{1(2)}(r = \pm 1, \sigma = 1)$, while the learning rate is $\alpha = 0.01$, and the initial policy is close to the worst one $x_0 = 0.975$. Bottom: The contributions of mutation and selection on the average Langevin dynamics near the boundaries, compared to the natural policy gradient. Rewards are distributed as $\mathcal{N}_{1(2)}(r = \pm 1, \sigma = 9)$, $\alpha = 0.01$, $x_0 = 0.5$.

All the obtained results can be easily generalized for the case in which the agent has k possible actions and their probabilities follow a k-dimensional drift-diffusion motion:

$$
d\pi_a = u_a(\pi)dt + \sum_{ab}^{N} \sigma_{ab}(\pi)dW_b \quad , \tag{11}
$$

expressed here in the Itô form. The resulting coefficients for this motion are

$$u_a = \alpha\pi_a\left(r_a - \sum_b r_b\pi_b\right) + \frac{\alpha^2}{2}\left(\sigma_a^2(1-\pi_a)(1-2\pi_a) - \sum_{b\neq a}\sigma_b^2(1-2\pi_b)\pi_a\right),$$

$$D_{ab} = \frac{\alpha^2}{8}\pi_a\pi_b\left(\delta_{ab}\frac{\sigma_a^2}{\pi_a} + \sum_{c\neq a,b}\pi_c\sigma_c^2 - (1-\pi_a)\sigma_a^2 - (1-\pi_b)\sigma_b^2\right). \tag{12}$$

They drive the trajectory towards the best action by a so-called replicator dynamics [47] proportional to α, and away from pure strategies by the mutation term proportional to α^2. In addition, the diffusion term scatters the trajectory proportionally to the rewards' variances. A thorough derivation of these results is reported in the Supplemental Material.

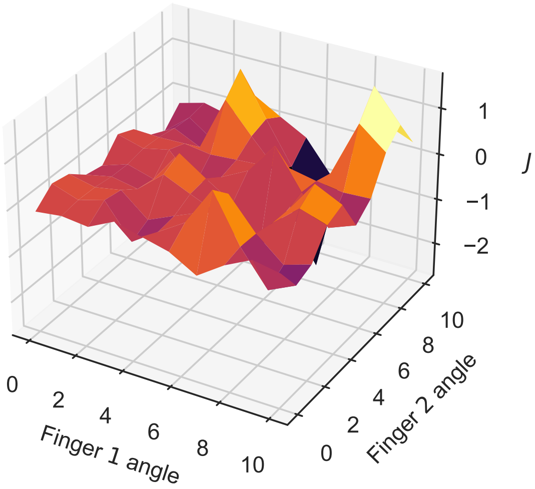

FIG. 2. An example of the return (energy) landscape of a robotic hand bending two fingers. Each finger can bend to 11 different angles, the return J is a function of the overall configuration.

IV. P-DIMENSIONAL K-ARMED BANDIT

The k-armed bandit can be viewed as a special case of a more general model in which the return is expressed as

$$J = \sum_{i_1,i_2,\ldots,i_p=1}^{K} R_{i_1 i_2\ldots i_p}\pi_{i_1}\cdot\pi_{i_2}\cdot\ldots\cdot\pi_{i_p}, \tag{13}$$

where $\sum_{i=1}^K \pi_i = 1,\ \pi_i \geq 0\ \forall i \in \{i_1,\ldots,i_p\}$. Each probability distribution π_i is defined over a distinct set of K actions. All p such sets are independent. This picture can be viewed simply as a factorization of the overall distribution $\pi = \prod_i \pi_i$. It arises naturally when one deals with an agent performing a set of actions at each time step and the task is to optimize the resulting overall behavior. For instance, robotics deals with a multitude of artificial joints flexed simultaneously [34, 48], producing a highly non-convex cost landscape, as portrayed in Fig. 2. Furthermore, this model describes p interacting agents, each performing independently their set of K actions [49]. The reward coefficients of each agent $R_{i_1\ldots i_p}$ could be different in this case, but for equal constant coefficients, this is a generalization of the random replicant model [50–52]. Another useful interpretation arises when an agent is performing a sequence of actions in a state-changing environment so that for each state s_t, π_t is the policy over the set of its K actions. The ordered set $(\pi_1,\pi_2,\ldots,\pi_p)$ then corresponds to the sequence of policies undertaken.

What is remarkable about this model is that now we have a clear way to map a reinforcement learning problem to a disordered system. This can be achieved by taking the instantaneous rewards to be normally distributed around their mean values $\mathcal{N}(\overline{R}_{i_1 i_2\ldots i_p}, \sigma_{i_1 i_2\ldots i_p})$, and considering the system described by the Hamiltonian $H \equiv -\overline{J}$, obtained substituting mean rewards in (13). Its temperature $T(\sigma)$ is defined by the specific learning algorithm, and for a policy gradient is proportional to the diffusion coefficient of the Langevin dynamics (6). A complete analogy with the physics of a magnet is described in the Supplemental Material.

PG dynamics is described by a system of p multidimensional Langevin equations, navigating through the rough landscape of (13). To evaluate the effect of the learning rate on this motion, we will shift our perspective from the probabilities π to the parameters θ. The latter form a basis defined by

$$d\boldsymbol{\theta} \propto \alpha\nabla\ln\pi = \nabla\ln\phi, \qquad \phi = \pi^\alpha. \tag{14}$$

In other words, we move from a picture in which the learning rate is affecting the parameters' change to the one where the learning rate is affecting the slope of the probability manifold. We can define the following Hamiltonian for this new landscape,

$$H = -\sum_{i_1,i_2,\ldots,i_p}^{K} \overline{R}_{i_1 i_2\ldots i_p}\phi_{i_1}^{1/\alpha}\phi_{i_2}^{1/\alpha}\ldots\phi_{i_p}^{1/\alpha}. \tag{15}$$

We take K to be large and mean rewards to be self-averaging, i.e. distributed as $\overline{R} \sim \mathcal{N}(0,\sigma)$ with $\sigma^2 \sim 1/K$. This allows us to conveniently exploit methods of mean-field theory to analyze the free energy averaged over all possible rewards $\langle F\rangle = -T\langle\ln Z\rangle$, where Z is the partition function [53]. The mean partition function will look similar to the one of the spherical p-spin”, ”” [54, 55] with planar rather than spherical constraints:

$$\langle Z\rangle = \int_0^{\infty} \prod_{i=1}^{K} d^p\pi_i\, \delta^p(\sum_i \pi_i - K) \times \int_{-\infty}^{+\infty} \prod_{i_1,\dots,i_p} d\overline{R}_{i_1\dots i_p} \tag{16}$$

$$\times \exp\left[-\overline{R}^2_{i_1\dots i_p} K^p + \beta \overline{R}_{i_1\dots i_p}\pi_{i_1}\dots\pi_{i_p}\right],$$

where $\beta = 1/T$. The free energy expression can be rendered tractable by the replica trick $\langle \ln Z\rangle = \lim_{n\to 0}\frac{1}{n}\ln\langle Z^n\rangle$ in order to compute its mean value [56].

$$\langle Z^n\rangle = \int D\pi \exp\left[\frac{\beta^2}{4K^{p-1}}\sum_{a,b}^{n}\left(\sum_i^K \pi_i^a\pi_i^b\right)^p\right], \tag{17}$$

where $\int D\pi$ is a shorthand for the measure $\prod_{a=1}^n\prod_{i=1}^K d\pi_i^a\,\delta(\sum_j \pi_j^a - K)$. Introducing $Q_{ab} = \sum_i \pi_i^a\pi_i^b$ by inserting the identity $1 = \int \delta(Q_{ab} - \sum_i \pi_i^a\pi_i^b)\, dQ_{ab}$, and changing to Fourier representations for all delta functions, we obtain

$$\overline{Z^n} = \int \prod_{a,b}^{n}\prod_{i}^{K} dQ_{ab}d\Lambda_{ab}d\xi^a d\pi_i^a \cdot$$
$$\cdot \exp\left[\frac{\beta^2 K}{4}\sum_{ab} Q_{ab}^p + K\sum_{ab} Q_{ab}\Lambda_{ab} \right. \tag{18}$$
$$\left. - \sum_i\sum_{ab}\Lambda_{ab}\pi_i^a\pi_i^b - \sum_{ia}\xi^a\pi_i^a + K\sum_a \xi_a\right].$$

For large $K\to\infty$, the integral is dominated by the saddle point of the exponent's argument, thus the free energy can be recovered by solving a system of equations.

In the neighborhood of a pure strategy (where $\pi_a \approx 1$, $\pi_b \approx 0\ \forall\ b \neq a$), the partition function for the Hamiltonian (15) can be recovered from Eq. (17) by substituting $p \to p/\alpha$. This will affect the saddle point equation containing the temperature

$$0 = \frac{p}{4T^2\alpha} Q_{ab}^{\frac{p}{\alpha}-1} + \Lambda_{ab} \tag{19}$$

in a fundamental way: It will get modified by $T \to \sqrt{\alpha}T$. Thus, $\sqrt{\alpha}$ acts as an effective temperature that modifies the shape of the free energy landscape.

V. DISCUSSION

Our analysis sheds light on the ability of policy gradient to overcome obstacles in complex reward landscapes. It appears that the dynamics of policies under PG follows a drift-diffusion motion with parameters strongly influenced by the learning rate. Higher values of the latter allow the policy to scatter and overcome obstacles. This picture is corroborated by our mean-field analysis of the free energy landscape for a complex reward scenario, with multiple local minima. The learning rate appears to act as an effective temperature smoothing the free energy landscape. It follows that scheduling of this parameter is essential to ensure the convergence to high value maxima. Furthermore, it follows that this scheduling corresponds to the physical process of annealing. This paves the road to a plethora of physics-inspired optimizations (as proposed, for instance, in [3, 57, 58]) to PG algorithms.

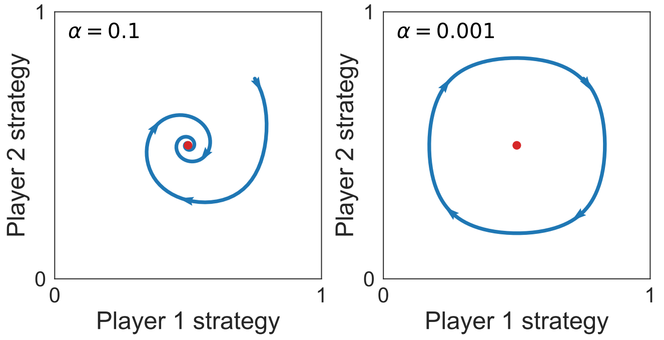

FIG. 3. Two average trajectories of the Natural Policy Gradient in a zero-sum game, corresponding to two different learning rates. Each point on the trajectories represents a pair $(\pi^1(t), \pi^2(t))$. The average rewards are $\overline{R}_1 = ((1,-1),(-1,1))$, $\overline{R}_2 = ((-1,1),(1,-1))$. The variance for all the rewards is equal to $\sigma = 1$. The starting point is $(0.75, 0.75)$, while the Nash equilibrium is at $(1/2, 1/2)$.

The p-dimensional k-armed bandit introduced here serves as a handy model to unify the description of partitioned policies, multi-state environments, and multi-agent interactions, by mapping them to a disordered system at finite temperature. This can be particularly well illustrated in the case of $p = 2$, which can be interpreted as a Matrix Game [59–63] between two players, each having its own reward matrix $R_{1(2)}$. It has been shown [64] that replicator dynamics with cooperation pressure u does not converge to all Nash equilibria below a critical value of u, unless we deal with a zero-sum game, i.e. $R_1 = -R_2^T$. On the other hand, the cooperation pressure acts in the replicator equation as the mutation term acts in the Langevin approximation of PG. In the case of a zero-sum game, the replicator trajectories can only factorize into a number of converging spirals as shown in the left side of Fig. 3, since Nash equilibria for pure strategies are suppressed for $K \to \infty$. If, instead, $R_1 \neq -R_2^T$, dynamics can converge to pure strategies, but such equilibria have been shown to give birth to a spin glass phase for low values of u [64].

ACKNOWLEDGMENTS

We would like to thank Andrea di Gioacchino, Antonio Celani, Andrea Mazzolini and Enrico Malatesta for the thoughtful discussions and precious insights on the topic. The work was supported by the Analytical center under the RF Government (subsidy agreement 000000D730321P5Q0002, Grant No. 70-2021-00145 02.11.2021).

Appendix A: The effect of the second-order expansion of the diffusion coefficient

Fig. 4 shows the comparison between average trajectories of the action probability x for the 2-armed bandit updated according to the Langevin dynamics.

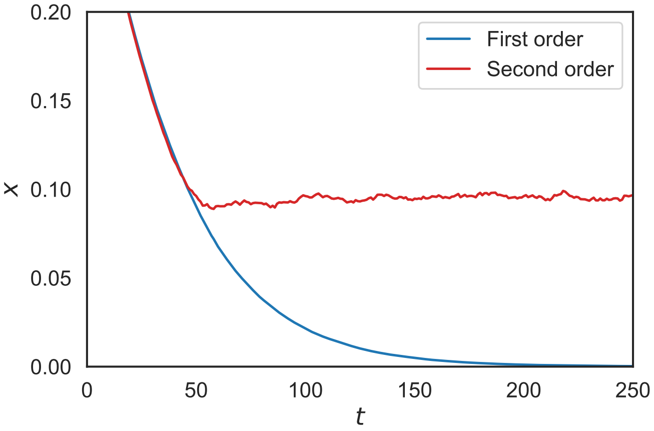

FIG. 4. Comparison between average trajectories of the action probability x for the 2-armed bandit updated according to the Langevin dynamics. The blue curve incorporates only the diffusion coefficient obtained by first-order expansion in α. The red one includes also the second-order α.

Appendix B: Regret bound and optimal learning rate scheduling

The regret of the Natural Policy Gradient is the difference between the reward obtained by a policy up to time T and the best possible reward one could obtain in the same time. In terms of the k-armed bandit problem it is defined as

$$\mathcal{R}_T = \max_{a\in\{1,\dots,k\}} \sum_{t=1}^{T} R_a^t - \sum_{t=1}^{T}\sum_{b=1}^{k} \left\langle \pi_b^t \right\rangle R_b^t. \tag{B1}$$

One can decompose this expression by introducing the *instantaneous regret* for an arm

$$\rho_a^t = R_a^t - \sum_{b=1}^{k} \pi_b^t R_b^t. \tag{B2}$$

The overall regret for that specific arm will then simply be $\mathcal{R}_{T,a} = \sum_{t=1}^{T} \left\langle \rho_a^t \right\rangle$, and therefore the total regret of the policy is the maximum of this quantity over all arms $\mathcal{R}_T = \max_{a\in\{1,\dots,k\}} \mathcal{R}_{T,a}$. We will consider the rewards to be independent stochastic variables, the only constraint being that they are bounded $R_a^t \in [0, R_M]$. Nonetheless, the result holds true also for correlated outcomes, non-stationary environments, and, the unluckiest configuration that one can imagine.

Itô's lemma states that if X_t is an Itô drift-diffusion process satisfying the diffusion equation

$$dX_t = u_t dt + \sqrt{2D_t} dW_t,$$

then any twice-differentiable function $f(X)$ can be expanded to the first order in time following

$$df = \left(u_t \frac{\partial f}{\partial x} + D_t \frac{\partial^2 f}{\partial x^2} \right) dt + \sqrt{2D_t} \frac{\partial f}{\partial x} dW_t + o\left(dt^2\right).$$

We will apply it to the average *log-policy* $\langle \log \pi_a \rangle$, expanding it to the form

$$\begin{aligned} \frac{d}{dt}\left\langle \log \pi_a^t \right\rangle = \left\langle \frac{u_a^t}{\pi_a^t} \right\rangle + \left\langle \frac{D_{a,a}^t}{(\pi_a^t)^2} \right\rangle = \\ \alpha_t \langle \rho_a^t \rangle + \frac{\alpha_t^2}{2} \left(\langle (\rho_a^t)^2 \rangle - \sum_b (R_b^t)^2 (1 - \langle \pi_b^t \rangle) \right) \end{aligned} \tag{B3}$$

and by making use of the fact that $\left\langle (\rho_a^t)^2 \right\rangle \geq 0$ and the rewards are bounded $R_b^t \leq R_M \ \forall b, t$, we can write the inequality

$$\left\langle \rho_a^t \right\rangle \leq \frac{1}{\alpha_t} \frac{d}{dt} \left\langle \log \pi_a^t \right\rangle + \frac{\alpha_t}{2} (k-1) R_M^2. \tag{B4}$$

We can now bound the single-arm regret using the latter equation:

$$\begin{aligned} \mathcal{R}_{a,T} \simeq \int_0^T dt \left\langle \rho_a^t \right\rangle \leq \left(\frac{\left\langle \log \pi_a^T \right\rangle}{\alpha_T} - \frac{\left\langle \log \pi_a^0 \right\rangle}{\alpha_0} \right) + \\ \frac{R_M^2}{2} (k-1) \int_0^T dt \alpha_t. \end{aligned} \tag{B5}$$

Where we have discarded negative terms. For any final probability distribution π_a^T, its logarithm will be negative and can be discarded leaving the bound unaltered. If we chose a uniform initial distribution $\pi_a^0 = 1/k \ \forall a$ and assume that $\alpha_T \leq \alpha_0$, we can rewrite the inequality substituting the latter:

$$\mathcal{R}_{a,T} \leq \frac{\log k}{\alpha_T} + \frac{R_M^2}{2} (k-1) \int_0^T dt \alpha_t. \tag{B6}$$

As we can see, the choice of scheduling function will influence the regret.

A convenient functional choice is $\alpha_t = A/\sqrt{t}$. In this way, both contributions are equally weighted and the expression can be rewritten as

$$\mathcal{R}_{a,T} \leq \left(\frac{\log k}{A} + R_M^2 (k-1) A \right) \sqrt{T}. \tag{B7}$$

The function $\alpha = A/\sqrt{T}$ can be refined specifying the coefficient A so that the bound is minimized. It is easy to see that such value is $A = \sqrt{\log k/(k-1)}/R_M$. Substituting this term, one finds the bound for the regret and the best scheduling of the learning rate for minimizing this bound:

$$\mathcal{R}_T \leq 2R_M\sqrt{(k-1)\log k\; T} \qquad \alpha_t = \frac{1}{R_M}, \sqrt{\frac{\log k}{(k-1)\; t}}. \tag{B8}$$

[1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[2] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).

[3] G. Zaránd, F. Pázmándi, K. F. Pál, and G. T. Zimányi, Phys. Rev. Lett. **89**, 150201 (2002).

[4] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).

[5] S. Franz, M. Leone, A. Montanari, and F. Ricci-Tersenghi, Phys. Rev. E **66**, 046120 (2002).

[6] A. K. Hartmann and M. Weigt, Journal of Physics A: Mathematical and General **36**, 11069 (2003).

[7] L. Kuśmierz and T. Toyoizumi, Phys. Rev. Lett. **119**, 250601 (2017).

[8] A. Montanari and R. Zecchina, Phys. Rev. Lett. **88**, 178701 (2002).

[9] M. I. Jordan and T. M. Mitchell, Science **349**, 255 (2015).

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT press, 2016).

[11] C. C. Aggarwal, *Neural Networks and Deep Learning* (Springer Cham, 2018).

[12] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, Nature **596**, 583 (2021).

[13] P. Baldi, P. Sadowski, and D. Whiteson, Nature Communications **5**, 1 (2014).

[14] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).

[15] H. Y. Sigaki, E. K. Lenzi, R. S. Zola, M. Perc, and H. V. Ribeiro, Scientific Reports **10**, 1 (2020).

[16] A. A. Pessa, R. S. Zola, M. Perc, and H. V. Ribeiro, Chaos, Solitons & Fractals **154**, 111607 (2022).

[17] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. **91**, 045002 (2019).

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning* (MIT press, 2018).

[19] 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.*, Nature **518**, 529 (2015).

[20] S. Carrazza and F. A. Dreyer, Phys. Rev. D **100**, 014014 (2019).

[21] G. Kántor, V. Niarchos, and C. Papageorgakis, Phys. Rev. D **105**, 025018 (2022).

[22] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, Phys. Rev. Lett. **118**, 158004 (2017).

[23] L. Zdeborová, Nature Physics **16**, 602 (2020).

[24] M. A. Larchenko, P. Osinenko, G. Yaremenko, and V. V. Palyulin, IEEE Access **9**, 159349 (2021).

[25] E. Gardner and B. Derrida, Journal of Physics A: Mathematical and General **21**, 271 (1988).

[26] E. Barkai, D. Hansel, and H. Sompolinsky, Phys. Rev. A **45**, 4146 (1992).

[27] H. Huang and Y. Kabashima, Phys. Rev. E **90**, 052813 (2014).

[28] H. Robbins and S. Monro, The Annals of Mathematical Statistics , 400 (1951).

[29] L. Bottou, in *Proceedings of COMPSTAT'2010* (Springer, 2010) pp. 177–186.

[30] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Proceedings of the National Academy of Sciences **113**, 10.1073/pnas.1608103113 (2016).

[31] Y. Feng and Y. Tu, Proceedings of the National Academy of Sciences **118**, 10.1073/pnas.2015617118 (2021).

[32] R. J. Williams, Machine Learning **8**, 229 (1992).

[33] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, in *Advances in Neural Information Processing Systems*, Vol. 12 (MIT Press, 2000).

[34] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, *et al.*, The International Journal of Robotics Research **39**, 3 (2020).

[35] R. Paulus, C. Xiong, and R. Socher, arXiv:1705.04304 (2017).

[36] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, *et al.*, arXiv:1912.06680 (2019).

[37] R. Bellman, Indiana Univ. Math. J. **6**, 679 (1957).

[38] S. M. Kakade, Advances in Neural Information Processing Systems **14** (2002).

[39] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, Automatica **45**, 2471 (2009).

[40] S.-i. Amari, *Information Geometry and Its Applications*, Vol. 194 (Springer Tokyo, 2016).

[41] T. Lattimore and C. Szepesvári, *Bandit Algorithms* (Cambridge University Press, 2020).

[42] M. Gurbuzbalaban, U. Simsekli, and L. Zhu, in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 139 (PMLR, 2021) pp. 3964–3975.

[43] Z. Xie, I. Sato, and M. Sugiyama, arXiv:2002.03495 (2020).

[44] M. Kimura, Journal of Applied Probability **1**, 177232 (1964).

[45] E. Baake and W. Gabriel, Annual Reviews of Computational Physics **7**, 203 (2000).

[46] C. Darken, J. Chang, and J. Moody, in *Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop* (1992) pp. 3–12.

[47] P. Schuster and K. Sigmund, Journal of Theoretical Biology **100**, 533 (1983).

[48] K. Lowrey, S. Kolev, J. Dao, A. Rajeswaran, and E. Todorov, in *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)* (2018) pp. 35–42.

[49] M. L. Littman, in *Machine Learning Proceedings 1994* (Morgan Kaufmann, 1994) pp. 157–163.

[50] S. Diederich and M. Opper, Phys. Rev. A **39**, 4333 (1989).

[51] M. Opper and S. Diederich, Phys. Rev. Lett. **69**, 1616 (1992).

[52] P. Biscari and G. Parisi, Journal of Physics A: Mathematical and General **28**, 4697 (1995).

[53] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, 1986).

[54] T. R. Kirkpatrick and D. Thirumalai, Phys. Rev. B **36**, 5388 (1987).

[55] A. Crisanti and H.-J. Sommers, Zeitschrift für Physik B Condensed Matter **87**, 341 (1992).

[56] V. Dotsenko, *An introduction to the theory of spin glasses and neural networks*, Vol. 54 (World Scientific, 1995).

[57] J. Houdayer and O. C. Martin, Phys. Rev. Lett. **83**, 1030 (1999).

[58] A. Möbius, A. Neklioudov, A. Díaz-Sánchez, K. H. Hoffmann, A. Fachat, and M. Schreiber, Phys. Rev. Lett. **79**, 4297 (1997).

[59] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)* (Princeton University Press, 2007).

[60] J. Nash, Annals of Mathematics **54**, 286 (1951).

[61] J. Berg and A. Engel, Phys. Rev. Lett. **81**, 4999 (1998).

[62] J. Berg and M. Weigt, Europhysics Letters (EPL) **48**, 129 (1999).

[63] J. Berg, Phys. Rev. E **61**, 2327 (2000).

[64] T. Galla, Europhysics Letters (EPL) **78**, 20005 (2007).