Gradient dynamics in reinforcement learning

Riccardo Fabbricatore and Vladimir V. Palyulin

1Skolkovo Institute of Science and Technology, 121205, Moscow, Russia

(Dated: April 11, 2022)

Despite the success achieved by the analysis of supervised learning algorithms in the framework of statistical mechanics, reinforcement learning has remained largely untouched. Here we move towards closing the gap by analyzing the dynamics of the policy gradient algorithm. For a convex problem, we show that it obeys a drift-diffusion motion with coefficients tuned by learning rate. Furthermore, we propose a mapping between a non-convex reinforcement learning problem and a disordered system. This 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.

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 [14], and in many other fields [15]. At the same time, reinforcement learning [16, 17] has proven to be a valuable tool for finding optimal jet grooming strategies [18], in the pursuit of the conformal bootstrap program [19], or in the engineering of smart active matter [20]. Nonetheless, numerous questions about the algorithms’ functioning remain unanswered [21]. 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 [22–24]. It has been shown how the stochastic gradient descent algorithm [25, 26] is prone to lead the network’s weights towards a needed suboptimal, robust, and well-generalizing region [27, 28]. 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) [29, 30]. 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 [31], natural language processing [32], and games [33]. 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.

THE REINFORCEMENT LEARNING FRAMEWORK

The typical reinforcement learning setting, the so-called Markov decision process [34], 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 $s \rightarrow s'$. 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 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 $\gamma$, $0 \leq \gamma < 1$, which tunes the importance of memory. Note that we used capital let-
ters 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 $\pi$ reads:

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

(1)

Reinforcement learning aims to efficiently find a policy $\pi$ that maximizes $J_\pi$. 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.

In this Letter we analyze policy gradient algorithm [16]. It exploits the well-known idea of gradient ascent to find the maximum of the return function (1). In this case the policy $\pi(a|s, \theta)$ is parametrized with a $d$-dimensional set of numbers $\theta = \{\theta_1, \ldots, \theta_d\}$. 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 $\partial_\theta J(\theta) = E_{\pi,q,p}[Q_\pi(s,a)|\partial_\theta \log \pi(a|s, \theta)]$. 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 $(G(s,a) - h(s))\partial_\theta \log \pi(a|s, \theta)$, 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 $\theta(t+1)$ will be derived from the current ones $\theta(t)$ by adding the gradient, multiplied by a coefficient $\alpha$, 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 [35, 36]:

$$\theta_{(t+1)} = \theta_{(t)} + \alpha F^{-1}_{(t)} \left( G(s_{(t)}, a_{(t)} - h(s_{(t)})) \right) \times \partial_\theta \log \pi(a_{(t)}|s_{(t)}, \theta_{(t)}),$$

(2)

where

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

(3)

The matrix $F$ is the Fisher information metric of the policy for the parameters $\theta$ [37]. 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 [29].

**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 [38] 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: $A = \{1, 2\}$. Since the gradient is not affected by the particular parametrization choice, we will use the convenient softmax function:

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

(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 $\gamma = 0$:

$$J(\theta(t)) = x(t)R_1 + (1 - x(t))R_2,$$

(5)

where $R_a$ represent the stochastic reward extracted from its corresponding distribution $R_a \sim q_a = N(r_a, \sigma_a)$. 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 this Letter.

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 [39, 40]. 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:

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

(6)

where $\eta_t$ is white Gaussian noise with zero mean and correlation $E_\tau[\eta_t \eta_{t+\tau}] = \delta(\tau - \tau')$. To this end, we expand the policy for small $\alpha$ by Taylor series:

$$dx(t) = \frac{dx}{d\theta} \bigg|_{\theta = \theta(t)} d\theta(t) + \frac{1}{2} \frac{d^2 x}{d\theta^2} \bigg|_{\theta = \theta(t)} d\theta(t)^2 + o(\alpha^2).$$

(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, $u(x) = E_\tau[x(t)|x(t)]$, and $D(x) = \text{Var}_\tau[x(t)|x(t)]/2$. 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 $\alpha$:

$$u(x) = \alpha x(1-x)(r_1 - r_2) + \frac{\alpha^2}{2}(1 - 2x) m,$$

(8)

$$D(x) = \frac{\alpha^2}{2} x(1-x)d_1 + \frac{\alpha^4}{4}(1 - 2x)^2d_2.$$
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:

\[
m = (1 - x) \left( \sigma_{R1}^2 + l_1^2 \right) + x \left( \sigma_{R2}^2 + l_2^2 \right),
\]

\[
d_1 = (1 - x) \sigma_{R1}^2 + x \sigma_{R2}^2 + [(1 - x)l_1 + xl_2]^2,
\]

\[
d_2 = (1 - x)^2 \frac{(3 - x)c_1^2 - 2l_1^4}{x^2} + x^2 \frac{(2 + x)c_2^2 - 2l_2^4}{1 - x},
\]

where \(c_a = \sigma_a^2 + l_a^2 \) and \(l_a = r_a - b \).

It is interesting to highlight the similarity with an evolving population of competing species/genotypes, described by the Kimura equation \([41, 42]\):

\[
u_K(x) = x(1 - x)(f_1 - f_2) - \mu_{12} x + \mu_{21}(1 - x),
\]

\[
D_K(x) = \frac{1}{2N} x(1 - x),
\]

where \(f_i \) is the fitness of the genotype \(i \), \(\mu_{ij} \) 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 effects of the drift coefficient on the gradient dynamics. The two terms correspond to natural selection and mutations, and can be tuned with the learning rate. For a large learning rate, the policy is pushed away from pure strategies, i.e. vertices of the probability simplex. Conversely, for small learning rates, the policy tends to converge to the best action. The intrinsic stochasticity of the algorithm appears in the diffusion coefficient \((8)\): small learning rates confine stochasticity to the bulk of the strategy simplex \((x \approx 1/2)\), while higher rates will generate higher fluctuations in the vicinity of pure strategies, as shown in appendix II.

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 \([43]\). 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 \(o(t) \propto 1/\sqrt{t} \), which is consistent with the results for the so-called Exp3 algorithm \([38]\), all details of the derivation can be found in appendix I.

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:

\[
d\pi_a = u_a(\pi) dt + \sum_{ab} \sigma_{ab}(\pi) dW_b,
\]

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

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

\[
D_{ab} = \frac{\alpha^2}{8} \pi_a \pi_b \left( \delta_{ab} \frac{\sigma_a^2}{\pi_a} + \sum_{c \neq a,b} \pi_c \sigma_c^2 \right.
\]

\[
\left. - (1 - \pi_a)\sigma_a^2 - (1 - \pi_b)\sigma_b^2 \right).
\]

They drive the trajectory towards the best action by a so-called replicator dynamics \([44]\) proportional to \(\alpha \), and
away from pure strategies by the mutation term proportional to $\alpha^2$. 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.

**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

$$J = \sum_{i_1, i_2, \ldots, i_p} R_{i_1 i_2 \ldots i_p} \pi_{i_1} \cdot \pi_{i_2} \cdot \ldots \cdot \pi_{i_p},$$

(13)

where $\sum_{i=1}^{K} \pi_i = 1$, $\pi_i \geq 0 \forall i \in \{i_1, \ldots, i_p\}$. Each probability distribution $\pi_i$ 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 $\pi \equiv \prod_i \pi_i$. 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 [31, 45], producing a highly non-convex cost landscape, as portrayed in Fig. 2. Furthermore, this model describes agents performing independently their set of $K$ actions [46]. The reward coefficients of each agent $R_{i_1 i_2 \ldots i_p}$ could be different in this case, but for equal constant coefficients, this is a generalization of the random replicant model [47–49]. Another useful interpretation arises when an agent is performing a sequence of actions in a state-changing environment so that for each state $s_t$, $\pi_t$ 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 $\mathcal{N}(R_{i_1 i_2 \ldots i_p}, \sigma_{i_1 i_2 \ldots i_p})$, and considering the system described by the Hamiltonian $H \equiv -\sum_{i=1}^{K} \pi_i R_{i_1 i_2 \ldots i_p}$, 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).

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 $\pi$ to the parameters $\theta$. The latter form a basis defined by

$$d\theta \propto \alpha \nabla \ln \pi = \nabla \ln \phi, \quad \phi = \pi^\alpha.$$  

(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,

$$H = -\sum_{i_1, i_2, \ldots, i_p} \pi_{i_1 i_2 \ldots i_p} \phi_{i_1}^{1/\alpha} \phi_{i_2}^{1/\alpha} \ldots \phi_{i_p}^{1/\alpha}.$$  

(15)

We take $K$ to be large and mean rewards to be self-averaging, i.e. distributed as $\bar{R} \sim \mathcal{N}(0, \sigma)$ with $\sigma^2 \sim 1/K$. This allows us to conveniently exploit methods of mean-field theory to analyze this landscape [50]. Its average partition function over the variables $\pi$ will look similar to the partition function of the spherical $p$-spin [51, 52] with planar rather than spherical constraints:

$$\langle Z^\alpha \rangle = \int_0^\infty \prod_{i=1}^{K} d^p \pi_i \delta^p \left( \sum_i \pi_i - K \right) \times \prod_{i=1}^{\infty} d\Phi_{i_1 i_2 \ldots i_p} \times \exp \left[ -\sum_{i_1 i_2 \ldots i_p} K^p \pi_{i_1 i_2 \ldots i_p} + \beta \sum_{i_1 i_2 \ldots i_p} \pi_{i_1} \ldots \pi_{i_p} \right],$$

(16)

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

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

(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

$$Z = \int \prod_{a,b}^{n} \prod_{i}^{K} dQ_{ab} d\Lambda_{ab} d\xi_{i}^{a} d\xi_{i}^{b} \cdot \exp \left[ \frac{\beta^{2} K}{4} \sum_{ab} Q_{ab}^{2} + K \sum_{ab} Q_{ab} \Lambda_{ab} - \sum_{i} \sum_{a,b} \Lambda_{ab} \pi_{i}^{a} \pi_{i}^{b} - \sum_{i} \xi_{i}^{a} \pi_{i}^{a} + K \sum_{a} \xi_{a} \right].$$

(18)

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

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

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

**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, 53, 54]) to PG algorithms.

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 [55–59] between two players, each having its own reward matrix $R_{1(2)}$. It has been shown [60], 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, $R_{1} \not= -R_{2}^{T}$, dynamics can converge to pure strategies, but such equilibria have been shown to give birth to a spin glass phase for low values of $u$ [60].

We would like to thank Antonio Celani, Andrea Mazzolini and Enrico Malatesta for the thoughtful discussions and precious insights on the topic.

**Appendix I: 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’s defined as

$$\mathcal{R}_{T} = \max_{a \in \{1,\ldots,k\}} \sum_{t=1}^{T} R_{a}^{t} - \sum_{t=1}^{T} \sum_{b=1}^{k} \langle \pi_{b}^{t} \rangle R_{b}^{t}.$$  

(20)

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

$$\rho_{a}^{t} = R_{a}^{t} - \sum_{b=1}^{k} \pi_{b}^{t} R_{b}^{t}.$$  

(21)
The overall regret for that specific arm will then simply be $R_{T,a} = \sum_{t=1}^{T} \langle \rho_a^t \rangle$, and therefore the total regret of the policy is the maximum of this quantity over all arms $\mathcal{R}_T = \max_{a \in \{1, \ldots, k\}} R_{T,a}$. 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

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

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

$$\frac{d}{dt} \langle \log \pi_a^t \rangle = \left( \frac{u_t}{\pi_a^t} \right) + \left( \frac{D_{\alpha,a}^t}{\pi_a^t} \right) = \alpha_t \langle \rho_a^t \rangle + \frac{\alpha_t^2}{2} \left( \langle \rho_a^t \rangle^2 - \sum_b \langle R_b^t \rangle^2 (1 - \langle \pi_b^t \rangle) \right)$$

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

$$\langle \rho_a^t \rangle \leq \frac{1}{\alpha_t} \frac{d}{dt} \langle \log \pi_a^t \rangle + \frac{\alpha_t}{2} (k - 1) R_M^2.$$  \hspace{1cm} (23)

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

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

(24)

Where we have discarded negative terms. For any final probability distribution $\pi_a^T$, 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:

$$\mathcal{R}_{a,T} \leq \frac{\log k}{\alpha_T} + \frac{R_M^2}{2} (k - 1) \int_0^T dt \alpha_t.$$ \hspace{1cm} (25)

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 contribution are equally weighted and the expression can be rewritten as

$$\mathcal{R}_{a,T} \leq \left( \frac{\log k}{A} + \frac{R_M^2}{A} (k - 1) A \right) \sqrt{T}.$$ \hspace{1cm} (26)

The function $\alpha = A/\sqrt{T}$ can be refined specifying the coefficient $A$ so that the bound is minimised. It’s 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 minimising this bound:

$$\mathcal{R}_T \leq 2R_M \sqrt{(k - 1) \log k} \sqrt{T} \quad \alpha_t = \frac{1}{R_M} \sqrt{\log k / (k - 1) t}.$$ \hspace{1cm} (27)

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

*ri.fabbricatore@gmail.com
v.palyulin@gmail.com

[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).
Supplementary material: Gradient dynamics in reinforcement learning

Detailed derivation of drift-diffusion coefficients

Two actions

In our setting, an agent has a discrete set of actions $A$, which he performs with a certain probability measure $\pi$ at each turn, receiving a payoff $J(\pi)$ by the ambient he is immersed in. Our task is to describe the dynamics of the learning algorithm in the space of its parameters. Having only two actions $A_1$ and $A_2$, performed with probabilities $\pi_1$ and $\pi_2$ respectively, we can use the normalization constraint $\sum_{a=1}^2 \pi_a = 1$ to reduce the problem to a one-dimensional motion described by the variable $x \equiv \pi_1 = 1 - \pi_2$. We will suppose that the two actions at each turn yield stochastic rewards i.i.d. form normal distributions $R_1 \sim q_1 \equiv N(r_1, \sigma_1)$ and $R_2 \sim q_2 \equiv N(r_2, \sigma_2)$. Consequently, the payoff for a strategy $x$ is equal to

$$J(x) = xR_1 + (1-x)R_2.$$  \hfill (S1)

This dynamic is driven by a gradient ascent $\dot{x} \sim \nabla J(x)$, which reinforces (hence the framework name) the better performing action. Such ascent can not be freely implemented on a compact space as $[0, 1] \ni x$. This can be circumvented by the use of a parameter $\theta \in \mathbb{R}$, on which we can freely perform the ascent $\dot{\theta} \sim \nabla J(\theta)$, mapped to $[0, 1]$ by the compactification

$$x = \frac{1}{1 + e^{-\theta}}.$$  \hfill (S2)

The concrete form we will choose for the ascent is:

$$\dot{\theta} = \alpha F^{-1} \nabla J(\theta) = E_\pi[\alpha F^{-1} R \nabla \ln \pi],$$  \hfill (S3)

where $\alpha$ is the so-called learning rate $F$ and is used to account for the uneven paste yield by the chosen parametrization. It is equal to the Fisher information metric $F = E_\pi[\partial_\theta \ln \pi \partial_\theta \ln \pi]$. This constitutes the algorithm known in literature as natural policy gradient (NPG). At each time step, the algorithm will feel a “local” gradient, based only on the rewards $R_t$, and its trajectory will thus fluctuate following the stochasticity of the rewards:

$$\dot{\theta}(t) = \alpha F(t)^{-1} R(t) \nabla \ln \pi(t).$$  \hfill (S4)

The standard approach for the description of stochastic dynamics is that of a Langevin equation:

$$\frac{dx}{dt} = u(x) + \sqrt{2D(x)} \cdot \eta(t),$$  \hfill (S5)

where $\eta(t)$ is white Gaussian noise with zero mean and correlation $E_t[\eta(\tau)\eta(\tau')] = \delta(\tau - \tau')$. Approximating stochastic fluctuations by Gaussian noise is not always accurate. For instance, the noise of neural networks’ stochastic gradient descent is often described by heavy-tailed distributions, which appears to be a crucial characteristic for ensuring convergence to flat minima. Nonetheless, our reinforcement learning setting supposes Gaussian noise in the rewards, which translates to a canonical Brownian motion of the policy gradient, as shown in Fig.S1.

In order to obtain the coefficients $u(x)$ and $D(x)$, we will expand $x_t \equiv x(\theta_t)$ supposing a slow learning, i.e. $d\theta \ll 1$, given by $\alpha \ll 1$:

$$dx = \frac{dx}{d\theta} d\theta + \frac{1}{2} \frac{d^2 x}{d\theta^2} d\theta^2 + o(\alpha^2).$$  \hfill (S6)

The drift coefficient is then found by taking the average

$$u(x) = E_t[\dot{x}_t|x_t] = E_q[E_\pi[\dot{x}_t|x_t]].$$  \hfill (S7)

All the handy relations we will use later on are:

$$\frac{dx}{d\theta} = x(1-x); \quad \frac{d^2 x}{d\theta^2} = x(1-x)(1-2x)$$  \hfill (S8)
The rewards are distributed as $\mathcal{N}(0, 0.1)$ and the initial probability is $x_0 = 0.5$. The linear relation shows that the probability diffuses according to a Brownian motion.

$$F = \sum_\pi \left( \frac{\partial \ln \pi}{\partial \theta} \right)^2 \pi = x(1 - x)$$  \hspace{1cm} (S9)

$$\dot{\theta}_t = \alpha F^{-1} \frac{\partial}{\partial \theta} \ln \pi \cdot R = \alpha F^{-1} \left( 1_1 R_1 \frac{\partial \ln x}{\partial \theta} + 1_2 R_2 \frac{\partial \ln (1 - x)}{\partial \theta} \right)$$

$$\dot{\theta}_t = \alpha \left( \frac{R_1}{x} - 1_2 \frac{R_2}{1 - x} \right)$$  \hspace{1cm} (S10)

$$\dot{\theta}_t^2 = \alpha^2 \left( \frac{R_1^2}{x^2} + 1_2 \frac{R_2^2}{(1 - x)^2} \right) + \ldots$$  \hspace{1cm} (S11)

We omitted terms that become irrelevant after the averaging of indicator functions $1_1 = 1$ with probability $x$ (i.e. $E_\pi[1_1] = x$), $1_2 = 1$ with probability $1 - x$ (i.e. $E_\pi[1_2] = 1 - x$), and $E_\pi[1_11_2] = 0$, in order to express the probability density in terms of discrete variables. Now, supposing $E_q[R_a] = r_a$, $E_q[R_a^2] = \sigma_a^2 + r_a^2$, we obtain

$$u(x) = E_q[\frac{dx}{d\theta} \dot{\theta}_t + \frac{1}{2} \frac{d^2x}{d\theta^2} \theta_t^2 | x_t]$$

$$= \alpha x(1 - x)(r_2 - r_1) + \frac{\alpha^2}{2}(1 - 2x)[(1 - x)(\sigma_1^2 + \sigma_2^2) + x(\sigma_1^2 + r_1^2)].$$  \hspace{1cm} (S12)

Analogously, for the diffusion term:

$$2D(x) = \text{Var}_\pi[\frac{dx}{d\theta} \dot{\theta}_t + \frac{1}{2} \frac{d^2x}{d\theta^2} \theta_t^2 | x_t]$$

$$= E_q[\frac{dx}{d\theta} \dot{\theta}_t + \frac{1}{2} \frac{d^2x}{d\theta^2} \theta_t^2 | x_t] - E\pi[\frac{dx}{d\theta} \dot{\theta}_t + \frac{1}{2} \frac{d^2x}{d\theta^2} \theta_t^2 | x_t]^2].$$  \hspace{1cm} (S13)

The first part of $2D(x)$ is:

$$E_q[\frac{dx}{d\theta} \dot{\theta}_t^2 | x_t] - E\pi[\frac{dx}{d\theta} \dot{\theta}_t | x_t]^2] =$$

$$= \alpha^2 x^2(1 - x)^2 \left( \frac{\sigma_1^2 + \sigma_2^2}{x} + \frac{\sigma_1^2 + r_1^2}{1 - x} \right) - \alpha^2 x^2(1 - x)^2(r_2 - r_1)^2$$

$$= \alpha^2 x(1 - x) \left[(1 - x)\sigma_1^2 + x\sigma_2^2 + ((1 - x)r_1 + xr_2)^2 \right].$$  \hspace{1cm} (S14)
The second part is:

$$E_q[\pi_2 \left( \frac{d^2 x}{d \theta^2} \right)^2 | x] - 2E_q \left( \frac{dx}{d \theta} \right)_t | x] E_q \left( \frac{d^2 x}{d \theta^2} \right) | x] \approx 0.$$  \hspace{1cm} \text{(S15)}$$

This term is proportional to $x(1 - x)(1 - 2x)$, and, thus, almost everywhere equal to zero.

The third part is:

$$E_q[\pi_2 \left( \frac{d^2 x}{d \theta^2} \right)^2 | x] - E_q \left( \frac{dx}{d \theta} \right)_t | x] E_q \left( \frac{d^2 x}{d \theta^2} \right) | x] \approx \frac{\alpha^4}{4} \left( 1 - 2x \right)^2 \left( \left( 1 - x \right)^2 \sigma_1^2 + \frac{\sigma_2^2 + \sigma_3^2}{x^2} \right)^2 - \frac{2r_1^4}{1 - x}.$$ \hspace{1cm} \text{(S16)}$$

We clearly see how the dynamic is driven towards a deterministic solution ($x = 1$ or $x = 0$), by the term of $u(x)$ proportional to $\alpha$, whilst it is repelled from it by the term proportional to $\alpha^2$. The diffusion analogously affected by $\alpha$: whilst its first power governs diffusion in the bulk of the probability simplex ($x \approx 1/2$), $\alpha^2$ is responsible for the diffusion next to solutions. A remarkable behavior is underlined by this expansion: the diffusion, although dumped by the square of the learning rate, augments in the vicinity of a solution. The difference between Langevin dynamics with first and second-order diffusion coefficients is shown in Fig. S2. The second-order term of $D$ is the main dissimilarity between the Kimura equation and NPG’s Langevin approximation. It is clear that it acts repelling the probability $x$ from boundaries, whereas the first-order term vanishes.

**K actions**

When dealing with $k$ independent actions, the dynamics will follow a $k$-dimensional drift-diffusion motion that can be described in the Langevin or Itô form. This time we will use the latter, but for our purposes they are equivalent.

$$d\pi_a = u_a(\pi) dt + \sum_{ab} \sigma_{ab}(\pi) dW_b \hspace{1cm} \text{(S17)}$$
in which $W_b$ are the components of a $k$-dimensional Wiener process.

The generalization to a set composed of $k$ independent actions is achieved assigning $k$ parameters $\theta_a \in \{1, \ldots, k\}$ to their respective probabilities by the mappings

$$\pi_a(\theta) = \frac{e^{\theta_a}}{\sum_{b=1}^k e^{\theta_b}} .$$  \hspace{1cm} (S18)

Fluctuations in the probability space $W_\pi$ are not uncorrelated. Correlations arise from the constraint $\sum_{a=1}^k \pi_a = 1$, which yields $\text{Cov}_\pi[W_{\pi_a}, W_{\pi_b}] = \rho_{ab}$. In order to describe the process in terms of uncorrelated Gaussian noises, we define noise in the parameters’ $\theta$ space, which is mapped to the probability space via $W_{\pi_a} = \sum_b \frac{\partial \pi_a}{\partial \theta_b} W_b$, where $\text{Cov}_\pi[W_a, W_b] = \delta_{ab}$.

It naturally follows, that the probabilities can be expanded as

$$d\pi_a = \sum_b \partial_b \pi_a d\theta_b + \sum_{bc} \partial_b \partial_c \pi_a d\theta_b d\theta_c + o(\alpha^2)$$  \hspace{1cm} (S19)

As before, the gradient ascent will be given by

$$\dot{\theta}_{(i)} = \alpha \left( F_{(i)}^{-1} R_{(i)} \nabla \log \pi(\theta_{(i)}) \right),$$  \hspace{1cm} (S20)

where $(F)_{ab} = E_{\pi} [\partial_a \ln \pi(\theta) \partial_b \ln \pi(\theta)] = \pi_a (\delta_{ab} - \pi_b)$. It is now clear why the metric $g_{ij} = F_{ij}$ is useful: the gradient is a covariant vector field on a smooth statistical manifold, i.e. a Riemannian manifold each of whose points is a probability distribution. Our manifold is defined by the map $\pi(\theta)$ and the local covariant basis of tangent vectors is defined by $e_i = \partial \log \pi / \partial \theta_i$. Having this in mind, it is easy to see how we need to account for the contravariance of $d\theta^i$, while $\partial_i \log \pi$ is covariant. We simply can’t equate two such vectors, since they behave in opposite ways under the curvature’s effects. Of course, with the help of the metric $g_{ij} = e_i e_j = F_{ij}$, we can transform covariant into contravariant vectors, thus obtaining $d\theta^i \propto g^{ij} \partial_j \log \pi$, or $d\theta \propto F^{-1} \nabla \log \pi$.

Useful relations for later on derivations include

$$\partial_b \pi_a = \pi_a (\delta_{ab} - \pi_b)$$  \hspace{1cm} (S21)

$$\partial_b \partial_c \pi_a = \pi_a \left[ (\delta_{ab} - \pi_b)(\delta_{ac} - \pi_c) - \pi_b (\delta_{bc} - \pi_c) \right]$$  \hspace{1cm} (S22)

$$\dot{\theta}_a = \alpha \sum_{b,c=1}^k F_{ab}^{-1} R_{c}(\delta_{bc} - \pi_b) \mathbf{1}_a,$$

$$\dot{\theta}_a = \alpha^2 \frac{R_a R_c}{\pi_b \pi_c} \mathbf{1}_a = \alpha^2 \frac{R_a R_c}{\pi_b \pi_c} \delta_{bc} \pi_b$$  \hspace{1cm} (S24)

The drift coefficient $u_a(\pi)$ is found by averaging:

$$u_a = E_q[E_\pi[\dot{\pi}_a | \pi]]$$

$$= E_q[E_\pi[\sum_b \partial_b \pi_a \dot{\theta}_b | \pi]] + E_q[E_\pi[\sum_{bc} \partial_b \partial_c \pi_a \dot{\theta}_b \dot{\theta}_c | \pi]] = u_a^1 + u_a^2$$  \hspace{1cm} (S25)

$$u_a^1 = \alpha \pi_a \left( r_a - \sum_b r_b \pi_b \right)$$  \hspace{1cm} (S26)
\[ u_a^2 = E_q[E_{\pi}[\frac{\alpha^2}{2} \sum_{bc} \partial_b \partial_c \pi_a \frac{R_b R_c}{\pi_b \pi_c} \mathbf{1}_b \mathbf{1}_c]] \]

\[ = E_q[\frac{\alpha^2}{2} \sum_{bc} \partial_b \partial_c \pi_a \frac{R_b R_c}{\pi_b \pi_c} \delta_{cd}] \]

\[ = \frac{\alpha^2}{2} \sum_c \pi_a \left[ (\delta_{ac} - \pi_c)(\delta_{ac} - \pi_c) - \pi_c(\delta_{ac} - \pi_c) \right] E_q[R_c^2] \]

\[ = \frac{\alpha^2}{2} \pi_a \sum_c \left( \delta_{ac}^2 - 2\pi_c \delta_{ac} + 2\pi_c^2 - \pi_c \delta_{ac} \right) E_q[R_c^2] \]

\[ u_a^2 = \frac{\alpha^2}{2} \left( E_q[R_a^2](1 - 2\pi_a) - \sum_b E_q[R_b^2](1 - 2\pi_b)\pi_a \right). \]  

(S27)

Supposing \( E_q = [R_a^2] = \sigma_a^2 \), be obtain

\[ u_a^2 = \frac{\alpha^2}{2} \left( \sigma_a^2(1 - \pi_a)(1 - 2\pi_a) - \sum_{b \neq a} \sigma_b^2(1 - 2\pi_b)\pi_a \right). \]  

(S28)

The diffusion coefficient is found upon taking the covariance of the increments:

\[ \frac{1}{2} \text{Cov}_{\pi_a, \pi_b} = \frac{1}{2} \text{Cov}_{\pi_a}[\text{Cov}_{\pi_a}[\pi_a, \pi_b]] \]

\[ = \frac{1}{2} \text{Cov}_{\pi_a}[\text{Cov}_{\pi_a}[\sum_{a} \sigma_{a} W_{a}, \sum_{b} \sigma_{b} W_{b}]] = \frac{1}{2} \sigma \sigma^T = D \]  

(S30)

Expanding \( \hat{\pi}_a \) up to the first order in \( \alpha \), we find

\[ 2D_{ab} = E_q[\text{Cov}_{\pi_a} \sum_c \partial_c \pi_a d\theta_c \sum_d \partial_d \pi_b d\theta_d] \]

\[ = E_q[\sum_{cd} \partial_c \pi_a \partial_d \pi_b d\theta_c d\theta_d] \]

\[ = E_q[\sum_{cd} \partial_c \pi_a \partial_d \pi_b \frac{1 \cdot 1_d}{\pi_c \pi_d}] \]

\[ = E_q[\sum_{cd} \partial_c \pi_a \partial_d \pi_b \frac{\delta_{cd} \pi_c}{\pi_c \pi_d}] \]

\[ = \sum_c \pi_a \pi_b (\delta_{ac} - \pi_c)(\delta_{bc} - \pi_c) E_q[R_c^2] \]

\[ = \pi_a \pi_b \sum_c (\delta_{ac} \delta_{bc} - \pi_c \delta_{ac} - \pi_c \delta_{bc} + \pi_c^2) \frac{E_q[R_c^2]}{\pi_c}. \]

(S31)

Supposing \( E_q = [R_a^2] = \sigma_a^2 \), be obtain

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

(S32)
To summarize, we found the coefficients governing the drift-diffusion motion:

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

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

(S33)

It is now clear that the drift is driven towards the maximum by a replicator dynamic, and away from pure strategies by the mutation term. In addition, the diffusion term, here expanded up to the second order in alpha, scatters the trajectory proportionally to the rewards’ variances.

Details of the mean-field method

The \( p \)-dimensional \( k \)-armed bandit can be viewed as an agent that at each time step is performing \( p \) independent actions, each one chosen from a unique set of \( K \) possible actions. An example of such agent is the robotic hand with its five fingers flexing independently of one another. Despite actions being independent, they yield an overall result that can be optimized with a Policy Gradient. If each finger can be flexed to \( K \) different angles, then we can assign a probability \( \pi_i \) to each angle \( i \in \{1, \ldots, K\} \). Each overall configuration will then yield a reward based on \( p \) distribution probabilities:

\[
    J = E_{\pi_1, \pi_2, \ldots, \pi_p} [R_{1, 2, \ldots, p}]
        = \sum_{\pi_1, \pi_2, \ldots, \pi_p} R_{i_1, i_2, \ldots, i_p} \pi_{i_1} \cdot \pi_{i_2} \cdot \ldots \cdot \pi_{i_p}.
\]

(S34)

If we suppose that the rewards are normally distributed around their averages \( \mathcal{N}(R_{i_1, i_2, \ldots, i_p}, \sigma_{i_1, i_2, \ldots, i_p}) \), we can consider the system described by the Hamiltonian \( H \equiv -\mathcal{J} \) obtained substituting mean rewards in \( (S34) \). 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 \( (S17) \). One can clearly see how this total reward has multiple local minima, generated by the quenched disorder of the coefficients \( R_{i_1, i_2, \ldots, i_p} \). In other words, it corresponds to the Hamiltonian of a planar \( p \)-spin model, i.e. with constraints \( \sum_{i_a} \pi_{i_a} = 1 \), \( \pi_{i_a} > 0 \), \( \forall a \in \{1, \ldots, p\} \). The analogy with a spin glass becomes clear if we imagine a magnet in which interatomic forces are extremely weak. Heating it up, atoms start to oscillate before spins. This will subsequently affect spins, since their interaction depends on their distance, resulting in an effective temperature for them, probed by the intensity of their Brownian motion. In order to analyze the structure of minima arising in this problem, one can study its free energy:

\[
    F = -\frac{1}{\beta} \ln \sum_{\{\pi\}} \exp\{\beta J[\pi]\} = -\frac{1}{\beta} \ln Z,
\]

(S35)

where \( \sum_{\{\pi\}} \) means summing over all possible values of \( \{\pi_1, \pi_2, \ldots, \pi_p\} \), each one corresponding to one configuration, generating a particular reward \( J[\pi] \). Including a temperature via the parameter \( \beta = 1/T \), lets us explore the landscape of \( J \), since for high temperatures all configurations are weighted equally in the resulting free energy, while lowering \( T \) permits us to see how local minima arise. Of course, every problem has its own energetic landscape defined by its own constants \( R_{i_1, i_2, \ldots, i_p} \). In order to study average properties, we will average over the disorder by considering it a self-averaging quantity, meaning that we will take each instance to be drawn from a normal distribution \( \mathcal{N}(0, \frac{1}{K}) \). To render the problem tractable, we will use the mean field method named replica trick: \( \ln Z = \lim_{n \to 0} \frac{1}{n} \ln Z^n \). Each of
the \( n \) replica of the system will have a mean partition function

\[
Z = \int_0^\infty \prod_{i=1}^K \prod_{a=1}^n d^p \pi_i \, \delta^p(\sum_i \pi_i - K) \int_{-\infty}^{+\infty} \prod_{i_1, \ldots, i_p} dR_{i_1 \ldots i_p}
\]

\[
\times \exp \left[ -R_{i_1 \ldots i_p}^2 K^p + \beta R_{i_1 \ldots i_p} \pi_{i_1} \cdots \pi_{i_p} \right],
\]

where the integration is performed over \( p \) sets of \( d\pi_1 d\pi_2 \ldots d\pi_K \), with \( p \) constraints \( \sum \pi = K \) enforced by \( p \) delta functions. Once we multiply \( n \) copies of the system, we integrate over the disorder variables, obtaining

\[
Z^n = \int_0^\infty \prod_{a=1}^n \prod_{i=1}^K d^p \pi_i^a \, \delta^p(\sum_i \pi_i^a - K) \exp \left[ \frac{\beta^2}{4K^p-1} \sum_{a,b} \pi_i^a \pi_{i_1} \cdots \pi_{i_p} \pi_i^b \right]
\]

\[
= \int D\pi_i^a \exp \left[ \frac{\beta^2}{4K^p-1} \sum_{a,b} \left( \sum_i \pi_i^a \pi_i^b \right)^p \right].
\]

In order to render this expression tractable, we will make an ansatz on the form of \( Q_{ab} = \sum_{i=1}^K \pi_i^a \pi_i^b \). To enforce it, we insert the identity \( 1 = \int \delta(Q_{ab} - \sum_i \pi_i^a \pi_i^b) dQ_{ab} \). Delta functions can be dealt with by passing to their Fourier transform:

\[
\delta(\sum_i \pi_i^a - K) = \int_{-\infty}^{+\infty} \exp \left[ -ik \left( \sum_i \pi_i^a \xi - K \sum_i \xi^a \right) \right] \prod_a \frac{d\xi^a}{\sqrt{2\pi}},
\]

\[
\delta(Q_{ab} - \sum_i \pi_i^a \pi_i^b) = \int_{-\infty}^{+\infty} \exp \left[ -ik \left( \sum_{a,b} Q_{ab} \Lambda_{ab} - \sum_{a,b} \Lambda_{ab} \pi_i^a \pi_i^b \right) \right] \prod_{a,b} \frac{d\Lambda_{ab}}{\sqrt{2\pi}}.
\]

Absorbing \(-ik\) into \( \xi^a \) and \( \Lambda_{ab} \), we finally arrive at the complete expression for the replicated partition function

\[
Z^n = \int_{-\infty}^{+\infty} \prod_{a,b} \prod_{i=1}^K dQ_{ab} d\Lambda_{ab} d\xi^a \int_0^{+\infty} \prod_{a=1}^n \prod_{i=1}^K d\pi_i^a \times
\]

\[
\times \exp \left( \frac{\beta^2 K}{4} \sum_{ab} Q_{ab}^p + K \sum_{ab} Q_{ab} \Lambda_{ab} - \sum_i \sum_{a,b} \Lambda_{ab} \pi_i^a \pi_i^b \right)
\]

\[
- \sum_i \xi^a \pi_i^a + K \sum_{a} \xi^a
\]

\[
\right)
\]

For large \( K \to \infty \), the integral is dominated by saddle point. We can now impose various ansatzes to the form of \( Q_{ab}, \Lambda_{ab} \) and \( \xi^a \), the simplest of which is the replica-symmetric:

\[
Q_{aa} = q_0 \quad \Lambda_{aa} = \lambda_0 \quad \xi^a = \xi \quad \forall a,
\]

\[
Q_{ab} = q_1 \quad \Lambda_{ab} = \lambda_1 \quad \forall a > b,
\]

which yields a system of equations for the saddle point

\[
S = \text{ext}_{q_0, q_1, \lambda_0, \lambda_1, \xi} \left[ \frac{\beta^2}{4T^2} \sum_{ab} (q_1 + (q_0 - q_1)\delta_{ab})^p + q_0 \lambda_0 \right.
\]

\[
+ q_1 \lambda_1 + \xi + \ln \int \prod \, d\pi e^{-\left( \xi \sum \pi + \sum \Lambda_{ab} \pi_i^a \pi_i^b \right)}
\]

\[
\right)
\]

Such picture gets modified if we shift our frame of reference from the probabilities \( \pi \) to the parameters \( \theta \). The latter are performing an ascent along the gradient of the reward \( \theta \sim \alpha \nabla J \), which speed is governed by the slope of the surface \( J(\pi) \) and the learning rate \( \alpha \). From their perspective, they are affected by a resulting slope \( \theta_t \sim \)
Performing a change of variables $\pi = \phi^{1/\alpha}$ in S36, we obtain

$$\int D\phi_i^a \exp \left[ \frac{\beta^2}{4K^{p-1}} \sum_{a,b} \left( \sum_i (\phi_i^a \phi_i^b)^{1/\alpha} \right)^p \right], \quad (S42)$$

where $D\phi_i^a$ contains the Jacobian and the constraints. In the neighborhood of a pure strategy, i.e. when $\phi \rightarrow (1, 0, \ldots, 0)$, one can expand $(\sum_i \phi_i^a \phi_i^b)^{1/\alpha} \approx \sum_i (\phi_i^a \phi_i^b)^{1/\alpha}$, since all non-diagonal terms are suppressed, thus obtaining a saddle point equation analogous to (S41). This time, the equation for $Q_{ab}$ will read

$$0 = \frac{p}{4T^2\alpha} Q_{ab}^{p-1} + \Lambda_{ab} \quad (S43)$$

which means that we operate as if we had a new temperature $T \rightarrow \sqrt{\alpha}T$. 

$R_{1,2,\ldots,p} \nabla \ln(\pi_1\pi_2\ldots\pi_p)^{\alpha} = R_{1,2,\ldots,p} \nabla \ln \phi_1 \phi_2 \ldots \phi_p$, corresponding to a gradient over the reward $J_\alpha = E_{\phi^{1/\alpha}}[R]$. 

