OCCUPY & SPECIFY: INVESTIGATIONS INTO A MAXIMUM CREDIT ASSIGNMENT OCCUPANCY OBJECTIVE FOR DATA-EFFICIENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The capability to widely sample the state and action spaces is a key ingredient toward building effective reinforcement learning algorithms. The trade-off between exploration and exploitation generally requires the use of a data model, from which novelty bonuses are estimated and used to bias the return toward wider exploration. Surprisingly, little is known about the optimization objective followed when novelty (or entropy) bonuses are considered. Following the "probability matching" principle, we interpret here returns (cumulative rewards) as set points that fixate the occupancy of the state space, that is the frequency at which the different states are expected to be visited during trials. The circular dependence of the rewards sampling on the occupancy/policy makes it difficult to evaluate. We provide here a variational formulation for the matching objective, named MaCAO (Maximal Credit Assignment Occupancy) that interprets rewards as a log-likelihood on occupancy, that operate anticausally from the effects toward the causes. It is, broadly speaking, an estimation of the contribution of a state toward reaching a (future) goal. It is constructed so as to provide better convergence guaranties, with a complementary term serving as a regularizer, that, in principle, may reduce the greediness. In the absence of an explicit objective occupancy, a uniform prior is used, making the regularizer consistent with a MaxEnt (Maximum Entropy) objective on states. Optimizing the entropy on states in known to be more tricky than optimizing the entropy on actions, because of an external sampling through the (unknown) environment, that prevents the propagation of a gradient. In our practical implementations, the MaxEnt regularizer is interpreted as a TD-error rather than a reward, making it possible to define an update in both the discrete and continuous cases. It is implemented on an actor-critic off-policy setup with a replay buffer, using gradient descent on a multi-layered neural network, and shown to provide significant increase in the sampling efficacy, that reflects in a reduced training time and higher returns on a set of classical motor learning benchmarks, in both the dense and the sparse rewards cases.

1 PROBLEM STATEMENT

Learning in the real world implies to deal with very large, potentially unlimited environments, over which the data to collect is seemingly infinite. *Efficient* exploration is thus one of the key aspects of open-ended learning, when no final model of the environment can feasibly be expected to be carried out. On the one side, having access to unlimited data is very beneficial for the training of complex multi-layered perceptrons, for they are known to rely on large datasets to improve their performance. On the other side, the circular dependence between the learning algorithm and the data on which it operates renders the learning very tricky, at high risk of data overfitting and trapping in local optima. The open-ended learning problem is generally addressed through the lens of the reinforcement learning framework (Sutton et al., 1998), where rewards are collected during the interaction, and the selection of action is fit so as to maximize the total number of positive rewards, and prevent the encounter of negative ones. Fitting behaviour to rewards is however at the risk of ignoring important data from the rest of the environment, where putatively more rewarding regions may be neglected. The agreement of reward-seeking (that is exploitation) with data collecting (that is exploration), is still one of the fundamental issues of modern artificial intelligence.

An important effort has recently been put on reframing the reinforcement learning setup into a more general probabilistic inference framework, allowing to link rewards seeking and data modelling under a single perspective (Furmston & Barber, 2010; Levine, 2018; Haarnoja et al., 2018; Abdolmaleki et al., 2018; Fellows et al., 2019). This greater focus over the data collection problem is linked to an important set of training algorithms, that contain some forms of exploration bonuses, including the "curiosity" drives Schmidhuber et al. (2009); Pathak et al. (2017), intrinsic rewards Oudeyer et al. (2007) and pseudo-counts (Bellemare et al., 2016; Tang et al., 2017). However, at the difference of the classic optimization on rewards alone, where the Bellman optimum is well defined, there is still no consensus about the objective followed when doing both rewards and data collecting optimization under a variational perspective Eysenbach & Levine (2019). The data collection problem is effectively shadowed by the reward maximization objective, under which it is still considered as an auxiliary helper. An important body of work has recently been devoted to addressing the data collection problem as such, with the notable design of the MaxEnt algorithm (Hazan et al., 2019), State Marginal Matching (Lee et al., 2019) and E3D (Daucé, 2020), that aim at fitting the distribution of the states encountered to a uniform distribution, in the absence of definite rewards. This is here referred as a MaxEnt-on-state principle (or MaxEnt to be short), not to be confounded with the MaxEnt-on-actions principle implemented in the soft actor critic (Haarnoja et al., 2018) for instance. A MaxEnt objective considers the *occupancy* of the (state, actions) space as the main subject over which the optimization should be performed, with the occupancy distribution being the distribution of states over the different pathways followed by a policy, independently of the time (or the sequence) at which they are visited. We develop in the following a possible extension of the MaxEnt principle toward a Bayesian variational optimization setup, framed over the reinforcement learning setup, that brings a considerable simplification in the expression of the evidence lower bound (ELBO) with regards with the existing literature (Furmston & Barber, 2010; Abdolmaleki et al., 2018; Fellows et al., 2019). An intriguing property of the resulting ELBO formula is that the future states (the ones that are visited after the current observation) play the role of a *model* for the current data, participating in the elaboration of the returns collected under the current policy. This gives ways toward optimizing the policy with respect to the distribution of the data, and provides a principled justification to the use of intrinsic rewards in the design of reinforcement learning algorithms.

2 PRINCIPLES

2.1 PROBABILITY MATCHING RL

We assume an agent acting in a fully observable environment. The state of the environment is provided by an observation $s \in \mathcal{S}$, with $\mathcal{S}$ the set of all possible states. The agent can act on the environment through its actuators. Such a motor command is described by $a \in \mathcal{A}$, with $\mathcal{A}$ the set of all motor commands. The decision of which action to choose relies on a *policy*, that maps the current observation to the action space, generally expressed in a conditional probabilistic form $\pi(a|s)$.

A reinforcement learning problem consists in finding a policy π^* that maximizes a certain objective function, without knowing the physical or mechanical properties of the environment. It is supposed here, for simplicity, that the dynamics of the environment is Markovian (no hidden states). Moreover, the environment is providing an auxiliary signal called the reward. Sending an action to the environment makes it possible to access to a new state s', and to obtain a reward $r \in \mathbb{R}$. A classic objective in learning is to maximize the global return, generally described as a discounted sum of future rewards over all possible trajectories. Let us now denote by s_t the state visited at time t and $\tau(s_t) = (s_{t+1}, ..., s_{t+T}, ...)$ a certain pathway that is visited after observing s_t. During this visit, a certain number of rewards can be collected, and $R(\tau)$ is the (discounted) return obtained over τ, i.e. $R(\tau) = \sum_t \gamma^t r_t$, with $\gamma \in [0, 1[$ a discounting factor that sums up the rewards up to an "horizon" of the order of $\frac{1}{1-\gamma}$. This said, the dynamic programming objective (Bellman, 1966), is described like: $\pi^* = \max_\Pi \mathbb{E}_{s\sim p(S_0), \tau\sim p_\pi(\tau|s)} R(\tau)$, with $p(S_0)$ the distribution of initial states, and Π the set of all conditional policies. When a state transition model $p(S'|s, a)$ is provided, the unique solution is given by the dynamic programming recurrent equation in the discrete case (Bellman, 1966). On the contrary, a large panel of reinforcement learning techniques allow to approach the solution in the model-free setup, assuming an effective sampling of all state-action pairs (Sutton et al., 1998).

We are here interested in a different class of objective function, that rely on interpreting the reward signal as a set point (in a distribution space) rather than an objective. A reward should indicate in which proportion the different states (and actions) should be visited (and selected) during trials (with the idea that the states providing high return should be visited more often than the ones providing low returns). Solving the reinforcement learning problem then means to match the external cue to an actual distribution of visit over states and actions, where a differential in rewards only indicates a difference in the number of visits, allowing to seek for rewards in a flexible way (so it is also referred as to "soft" reinforcement learning (Haarnoja et al., 2018)). This idea stems back from empirical observations on human and animal behaviors, and was coined the "matching law" in the operant conditioning literature (Herrnstein, 1961; Eysenbach & Levine, 2019).

2.2 State occupancy

Matching rewards to probabilities can be done in many different ways. We frame here the probability matching reinforcement learning problem into a *state occupancy matching problem*. It relies on the use of an occupancy distribution, that is a density of state visit under a certain policy. Importantly, it ignores the time order at which the different states are visited, still conserving some aspects of causality between states in the form of conditional probabilities as we see later. The occupancy distribution is designed so as to match with the gamma-vanishing distribution of states observed on a particular trajectory of the MDP. Let $\tau = (s_0, s_1, ..., s_t, ...)$ a certain trajectory observed on the MDP under the policy π. The set of all possible trajectories is noted $\mathcal{T}$, $p_\pi(\tau)$ is a measure over the trajectories for a certain policy π, and $p_\pi(S|t)$ the distribution of states that is a restriction of the trajectories to a certain time t. Then, the gamma discounted occupancy on states $\rho_\pi(S)$ (eq. 1) is constructed for $(1-\gamma)\sum_t \gamma^t p_\pi(s|t)$ to be a Monte Carlo estimate of $\rho_\pi(s)$. In short, the occupancy density provides a distribution on states that is discounted in proportion to the average temporal distance of s from the initial state. In other words, it is a time-agnostic distribution over states on (softly) bounded trajectories, with horizon $\frac{1}{1-\gamma}$.

Following the definitions of (Puterman, 2014; Ho & Ermon, 2016; Hazan et al., 2019), a gamma-discounted state occupancy of a Markov Decision process (with a policy π) is the (discounted) density of visit of the states — or (state, action) pairs — of the environment when starting from the initial distribution $p(S_0)$. It is defined, as:

$$\begin{cases} \rho(s) = (1-\gamma)p_0(s) + \gamma \sum_{s',a'} p(s|s',a')\pi(a'|s')\rho(s') \\ \rho(s,a) = \pi(a|s)\rho(s) \end{cases} \tag{1}$$

so that any policy π settled on an MDP defines a (gamma-discounted) occupancy on the states of that MDP. It comes that, inversely, any valid (state, action) occupancy (meaning that this occupancy is effectively feasible in a given agent/environment setup), defines a unique corresponding policy:

$$\pi(a|s) = \frac{\rho(s,a)}{\rho(s)} \tag{2}$$

that is a softmax (stochastic) conditional policy over the states.

Consider for instance the series of rewards encountered when following τ. It comes that:

$$\mathbb{E}_{s_0 \sim p_0} V(s_0) = \mathbb{E}_{\tau \sim p_\pi(\mathcal{T})} \sum_t \gamma^t r(s_t, a_t) \approx \mathbb{E}_{\substack{s \sim \rho_\pi(S) \\ a \sim \pi(A|s)}} r(s,a) \sum_t \gamma^t = \mathbb{E}_{s,a \sim \rho_\pi(S,A)} \frac{r(s,a)}{1-\gamma}$$

so that $\forall t$, $\frac{r(s,a)}{1-\gamma}$ is interpreted as an estimator of $V(s_0)$.

2.3 Conditional occupancy

Following the same reasoning, we show here that for any t, it comes that $\forall t' > t$, $\frac{r(s_{t'}, a_{t'})}{1-\gamma}$ is an estimator of $Q(s_t, a_t)$. Let $\rho_\pi(S^+|s...)$ the *conditional occupancy* be defined recursively:

$$\forall s' \in \mathcal{T}_\pi(s), \rho_\pi(s^+|s...) = p_\pi(s^+|s) + \gamma \sum_{s'} p_\pi(s^+|s')\rho_\pi(s'|s...)$$

Note that the triple dots (...) are use here to well distinguish the one-step distribution $p_\pi(S'|s)$ from the long-term distribution $\rho_\pi(S^+|s...)$. This conditional distribution provides a description of the

"future" of s_t, describing the sequence of states that will most probably follow s_t. Those future states will generally be noted s^+, with the '+' exponent meaning the state being measured "further away in time".

Then

$$Q(s_t, a_t) = \mathbb{E}_{\substack{\tau \sim p_\pi(\mathcal{T}) \\ s_t \in \tau}} \sum_{t' > t} \gamma^{(t'-t)} r(s_{t'}, a_{t'}) \approx \mathbb{E}_{\substack{s^+ \sim \rho_\pi(S^+|s_t, a_t ...) \\ a^+ \sim \pi(A|s^+)}} \frac{r(s^+, a^+)}{1-\gamma}$$

In that setup, the rewards are interpreted as value samples. This means, in short, that each future reward $r(s^+, a^+)$ contains a "guess" about the total return that is averaged in proportion to the conditional occupancy $\rho_\pi(S^+, A^+|s, a...)$.

2.4 Probability template matching

Assume now $\pi(a|s) = \frac{\exp \beta Q(s,a)}{\sum_{a'} \exp \beta Q(s,a')}$ the softmax distribution, with β the "inverse temperature". The softmax equation is a way to "map" the state-action value to a probability so that it provides simple identity: Assuming $\exists! K(.)$ so that $\log \pi(a|s) = \beta Q(s,a) - K(s)$, we define :

$$\overline{R(s^+, a^+)} \triangleq \frac{r(s^+, a^+)}{1-\gamma} - \frac{1}{\beta}(K(s) - \log \rho(s)) \tag{3}$$

as the "extended" return composed of the return estimator plus a virtual baseline. Then, because the policy and the occupancy are exchangeable from eq.(2), each reward collected after (s, a) contains an information not only about the policy, but also (indirectly) about the occupancy, i.e.:

$$\log \rho(s, a) \approx \mathbb{E}_{\substack{s^+ \sim \rho_\pi(S^+|s, a...) \\ a^+ \sim \pi(A|s^+)}} \beta \overline{R(s^+, a^+)}$$

Like for the Q function, each future reward $r(s^+, a^+)$ also contains a "guess" about the current log occupancy that is averaged in proportion to the conditional occupancy $\rho_\pi(S^+, A^+|s, a...)$.

2.5 Optimization

Assume now π^* be the policy that solves:

$$\pi^* = \underset{\Pi}{\text{argmax}}\, \mathbb{E}_{s,a \sim \rho_\pi(S,A)} \frac{r(s,a)}{1-\gamma}$$

The optimal policy comes with an optimal occupancy $\rho^*(S)$ and a set of conditional occupancies $\{\rho^*(S^+|s, a...)\}_{s,a}$.

In the absence of a model, one can not evaluate ρ_π. The optimization needs to be done by sampling over the trajectories of the dynamics, i.e. by stochastic gradient ascent. Assuming (s, a) a sample of the dynamics, the update is obtained by sampling on the forthcoming trajectory, containing many $(s^+, a^+) \sim q_\pi(S+, A^+|s, a...)$, that is: $Q(s,a) \overset{s^+, a^+}{\sim} \frac{r(s^+, a^+)}{(1-\gamma)}$. Equivalently, the rewards behave as "templates" from which the objective occupancy is "matched", i.e. $\log \rho(s,a) \overset{s^+, a^+}{\sim} \beta \overline{R(s^+, a^+)}$.

Then, assuming π a current policy, q_π the corresponding occupancy, and taking $\langle \widetilde{\log \rho}(s,a) \rangle_{A^+, S^+, \pi}$ as a shorthand for $\mathbb{E}_{\substack{s^+ \sim q_\pi(S^+|s_t, a_t ...) \\ a^+ \sim \pi(A|s^+)}} \beta \overline{R(s^+, a^+)}$, the sampling-based optimization writes:

$$\pi^* = \underset{\Pi}{\text{argmax}}\, \mathbb{E}_{s,a \sim q_\pi(S,A)} \langle \widetilde{\log \rho}(s,a) \rangle_{A^+, S^+, \pi} \tag{4}$$

which is a cross-entropy objective that aims at fitting $q_\pi(S, A)$ with $\langle \widetilde{\rho}(S, A) \rangle$. It should be noted that $\langle \widetilde{\rho}(S, A) \rangle$ is not equal to $\rho^*(S, A)$ but only a proxy, for the rewards are sampled on the current policy, not the optimal one.

The Soft Actor Critic method (Haarnoja et al., 2018), considers instead a (negative) Kullback-Leibler divergence:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{s,a \sim q_\pi(S,A)} \langle \widetilde{\log \rho}(s,a) \rangle_{A^+, S^+, \pi} - \log q_\pi(s,a) \tag{5}$$

that contains a complement term $-\log q_\pi(s,a)$, still pursuing the same softmax probability matching objective. The supplementary entropy term acts here as a regularizer, and the loss has a much better form, for the gradient is expected to be parsimonious as the loss would shrink to zero as the optimization progresses. Still, the fundamental mismatch between the approximate and the true optimal log occupancy is not addressed, and the convergence toward the optimum can not be guaranteed.

2.6 Occupancy probability matching

Now we can make a step further by trying to estimate how far is $\langle\widetilde{\log\rho}(s,a)\rangle$ from $\log\rho^*(s,a)$. Interestingly, for any $q(S^+,A^+)$, the following inequality holds:

$$\begin{aligned}\log\rho^*(s,a) &\geq \mathbb{E}_{s^+,a^+\sim q(S^+,A^+)}\log\rho^*(s,a) - \mathcal{D}_{\mathbf{KL}}(q(S^+,A^+)||\rho^*(S^+,A^+|s,a...))\\ &\approx \mathbb{E}_{s^+,a^+\sim q(S^+,A^+)}\beta\overline{R(s^+,a^+)} - \mathcal{D}_{\mathbf{KL}}(q(S^+,A^+)||\rho^*(S^+,A^+|s,a...)) \qquad (6)\end{aligned}$$

providing a variational (log-) evidence lower bound (ELBO) interpretation of the maximization on $\beta\overline{R(s^+,a^+)}$. Following the formal correspondence with Bayesian inference, the future states (s^+,a^+) are now interpreted as a "model" for "explaining" the data (s,a), and q reflects the inference of a model from the data. In order to maximize an ELBO, one needs to estimate q from its samples. At the convergence of the ELBO (obtained by gradient ascent over the parameters of q), the distribution q is expected to match the posterior ρ^*, and the inequality would becomes an equality, i.e. .

$$\log\rho^*(s,a) = \mathbb{E}_{s^+,a^+\sim\rho^*(S^+,A^+)}\beta\overline{R(s^+,a^+)} - \mathcal{D}_{\mathbf{KL}}(\rho^*(S^+,A^+|s,a...)||\rho^*(S^+,A^+|s,a...))$$

This variational formulation (6) of the optimization is not directly usable, because the posterior ρ^* is not specified, but provides new hints into interpreting the current occupancy at the light of its future effects.

The actual optimization is done on $\overline{R(s^+,a^+)}$ solely, irrespective of the divergence bias. This bias is expected to fade away with the progress of the training, making the sampling of the rewards more and more accurate at estimating the parameters of the policy/occupancy. This however, is not guaranteed, and the optimization on the (pseudo) cross entropy is at risk of keeping a high divergence throughout the optimization, hindering the convergence toward the optimum. This illustrates a more general problem that is the lack of efficacy in sampling the data (the posterior occupancy), a more robust upper bound guarantee would be preferred, even at the risk of a lesser final optimality with regard to the Bellman optimum.

2.7 Maximal Credit Assignment Occupancy

We now introduce the main insight of our Maximal Credit Assignment Occupancy (MaCAO) model. We have seen that a state s^+ and an action a^+, that are seen further on in the iteration of the dynamics provide retrospectively an estimate of $\log\rho^*(s,a)$ (the objective occupancy) through the measure of $r(s^+,a^+)$. Assume here that goal states manifest in a high objective occupancy on pathways that should conduct to them. This conducts to define goal pathways as *modes* of the objective occupancy distribution. On contrary, adverse states should manifest on a lower occupancy on the pathways that head toward them, conducting to a *through* in the objective occupancy distribution. Each reward collected at (s^+,a^+) is thus supposed to exert a control on the occupancy of the previous states, and the "width" at which it is influential depends on the number of pathways that head toward (s^+,a^+).

Let us introduce a new conditional distribution, namely $\rho(S,A|...s^+,a^+)$, that is the frequency at which (s,a) may precede (s^+,a^+) in the iteration of the dynamics. This distribution expresses an "anti-causal" relationship between the future states and the current observations, that is the exact measure of how much (s,a) is instrumental in reaching (s^+,a^+). This is also said the "credit assignment" in the reinforcement learning literature (Sutton, 1988; Harutyunyan et al., 2019).

Let us now rewrite the previous ELBO formula (6) in its usual probabilistic formulation:

$$\mathbb{E}_{s^+,a^+\sim q(S^+,A^+)}\left[\log\rho^*(s,a|...s^+,a^+)\right] - \mathcal{D}_{\mathbf{KL}}(q(S^+,A^+)||\rho^*(S^+,A^+)) \leq \log\rho^*(s,a) \qquad (7)$$

Here $\rho^*(s,a|...s^+,a^+)$ represents the objective credit assignment. From the Bayesian perspective, it is interpreted as a log-likelihood of the "data" (s,a), given the "model" (s^+,a^+), that is a way to

say that (s^+, a^+) exerts a control on (s, a). Following this analogy, we now consider that:

$$\mathbb{E}_{s^+,a^+\sim q(S^+,A^+)}\frac{\beta r(s^+,a^+)}{1-\gamma} - \mathcal{D}_{\mathbf{KL}}(q(S^+,A^+)||\rho^*(S^+,A^+)) \leq \beta Q^*(s,a) \tag{8}$$

This is equivalent to augmenting the return with a supplementary divergence term on the conditional occupancy. Then, identifying $\log \rho^*(s,a)$ (or, equivalently $\beta Q^*(s,a)$) from rewards sampling becomes identical to maximize the ELBO (7) from variational inference.

The loss is composed of two complementary terms, a first term being the occupancy matching on cumulative rewards (that is consistent with the softmax optimization), while the second term is the explicit probability matching of a posterior with a prior. The prior takes the role of a supervision signal, that aims at putting a constraint on the conditional occupancy (that represents the exploration pattern). In a full information setting, it would help the policy to converge faster to the optimum. In practical setups however, the access to the information is only partial. When rewards are provided, the prior distribution is generally unknown. On contrary, when priors are provided, the rewards are generally absent, like in inverse reinforcement learning (Ho & Ermon, 2016). We concentrate here on the first case, with no prior provided whatsoever.

By construction of the loss, the right term is made to shape the posterior occupancy, that conducts the evaluation of the return. In the absence of a definite prior, one needs to make a guess. This is formally analog to the case of Bayesian inference where the prior serves as a regularizer that tends to counteract the overfitting of the data. This sort of regularizer is highly expected in reinforcement learning that is known to be prone to overfitting. The analogy with Bayesian inference suggests for instance to consider a Gaussian or a *uniform* distribution. This has important consequences though. Taking a prior that is not the objective occupancy strongly modifies the interpretation of the loss, that is now composed of two *concurrent* terms. The likelihood part aims at fitting the occupancy with the rewards collected, while the divergence part aims at fitting the posterior occupancy with an arbitrary prior. This breaks the original symmetry, for the implies to concurrently follow two different objectives. For instance, the role of regularizer that would be devoted to a uniform prior introduces a bias in the probability matching, for the objective occupancy is not anymore the softmax-Bellman optimum, but rather an intermediary occupancy that combines the Bellman optimum and a uniform occupancy. This is expected to be beneficial for a wider exploration, helping to avoid overfitting, but this is at the cost of a relaxed constraint on fitting to the optimum.

This expression of a variational upper bound is reminiscent of the standard variational bounds considered in the reinforcement learning literature (Furmston & Barber, 2010; Abdolmaleki et al., 2018), from which it could be interpreted as a temporally mixed-up variant. By breaking the temporal chain dependence, however, it surprisingly provides a justification for considering an entropy *on states* in the optimization of the policy. Indeed, the classic variational optimization operates on chained sequences of observations, for which the state transitions vanish during the optimization of π (Haarnoja et al., 2018). In our case, ignoring the sequential time order establishes both the future states and the future actions as parameters of the policy, over which it should be optimized. This allows to address the data efficiency through explicitly optimizing on the exploration path when maximizing the entropy of the conditional occupancy. It moreover gives room to the maxent-on-states optimization, as proposed in (Hazan et al., 2019), as a complementary regularizer in a more general expression. This conceptual shift has however an important consequence on the objective followed, for the objective occupancy is not anymore equal to the softmax Bellman optimum.

3 METHOD

The main ingredients for an efficient implementation is the access to a wide variety of samples (s, a, r, s', a') over which optimization can be carried out on parameterized policy π_θ (said the "actor") and a parameterized action value function Q_ψ (said the "critic"). It is here implicitly assumed that both the actor and the critic consist of multi-layered perceptrons, containing many parameters and organized in layered weights, over which a gradient descent is operated on losses expressed as negative objectives.

Assuming an off-policy approach, we consider a replay buffer containing many samples of states, actions and rewards as observed from interacting with the environment. In a variational setup, one can assume an alternation between two complementary steps. A first step, said the "estimation"

step, consists in evaluating a distribution over the parameters of the model, that is here assumed to be a distribution over future states, identified as the conditional occupancy under the current policy q_π. This distribution is then exploited in a second step, said the "maximization" step, where the parameters of the policy/Q-function are updated so as to maximize the proximal objective (that is fitting a policy to a distribution of returns obtained from the current occupancy).

We assume here the estimation of q_π being obtained from a parametric or non-parametric method over a sufficiently large sample of recent states (or state, action pairs), and concentrate on the optimization of the actor under the guidance of the critic. Knowing q_π, and given a sample (s, a, r, s', a'), an element of optimization is given by the log-difference $\log q_\pi(s', a') - \log \rho^*(s', a')$ that is a point estimate of the Kullback-Leibler divergence, that is expected to be minimized during the optimization of the policy. This term is supposed to be differentiated with regards to π, providing a first gradient direction that should contribute to improving the policy toward a wider occupancy on future states. A second and independent element of optimization is the parameterized action-value mapping $Q(s, a)$, relying on the TD-error construct Sutton (1988), based on approximating the future rewards with the best current estimate at $t+1$, defining a gamma-discounted proximal objective value $\tilde{Q}(s, a) = r + \gamma Q(s', a')$. The squared difference $\lambda(\tilde{Q}(s, a) - Q_\psi(s, a))^2$, with λ a precision hyperparameter, is known as the mean-squared Bellman error (MSBE), providing a second gradient that aims at maximizing the return with regards to the policy parameters. The concurrence of both gradients contains the necessary element to combine exploration and exploitation in an principled way.

Building a full parametric model of the occupancy is however a difficult task that should be undertaken with care. The building of such a probabilistic model is indeed at a non negligible cost of regressing parametric distributions from samples, that inherently contain design choices and a specific optimization on a set of latent parameters (like in the case, e.g., of auto-encoders). For the sake of simplicity, we consider here the case of a non-parametric estimator of the occupancy distribution q_π. This could appear counter-intuitive at first glance, for the update of the policy is supposed to rely on backpropagating gradients through the estimator. As a workaround, we provide in the appendix a method allowing to directly inject the gradient information in the design of the Q-function.

The main lines of our implementation are provided in algorithm 1 (see Appendix), that fits the pursuit of the MaCAO objective in an actor-critic setup. It relies on a wide use replay buffers (Mnih et al., 2013) to regularize the gradient over batches that mix the samples from many different trials. From this perspective, an important shortcut is our on-the-fly calculation of the log-occupancy, with the help of kernel-based density estimation method (Pedregosa et al., 2011) from an initial sampling of (future) states from the buffer at each start of an update sequence (line 10). This occupancy sample remains quite limited in number (about 1000) in order to avoid unnecessary computer overload.

4 RESULTS

In order to reach state-of-the art efficacy, many algorithmic improvements need to be included in supplement to the baseline algorithm[1]. This concerns in particular the use of target Q-networks updated at slower pace (Mnih et al., 2013), and the clipped double-Q trick (Fujimoto et al., 2018). Our implementation is moreover drawn over the "spinning-up" open source framework (Achiam, 2018), allowing for a direct comparison with the state of the art. We consider here for comparison the soft actor-critic method (SAC) (Haarnoja et al., 2018), proximal policy optimization (Schulman et al., 2017) and TD3 (Fujimoto et al., 2018). Our method was tested over several benchmark environments, as provided by the "Gym" suite (Brockman et al., 2016). We concentrate here on the continuous states and actions case, that is the most challenging one with regards to function approximation.

The different setups are compared on the basis of the returns collected during training. This is expressed as average return (that is the total sum of rewards gathered at the end of an episode), the average reward (total rewards collected divided by the episode length) and cumulative rewards (the total sum of rewards collected at a given stage of the training). The width of the occupancy over the state space is not compared here, for the other frameworks are not designed to optimize it. The different environments differ in scale, difficulty and rewards density. All continuous problems

[1] code freely available at http://github.com/xxx/yyy (to come).

proposed in the library provide dense rewards, that are a compound of negatively and positively weighted extrinsic informations, like the energy consumption, the speed of the agent or its elevation. The problems separate in two broad categories. A first class of problems provides only dense rewards. A second class of problems have, in addition, a supplementary sparse reward taking the form of an "end-of-episode" bonus or penalty. In that case, the dense rewards may (or may not) contain relevant information with regards to the task at hand.

From that prospect, the most unfavorable problem is the Continuous Mountain Car problem (first row of figure 4). Here the dense rewards only refer to the energy consumption, at the exception of a +100 end-of-episode bonus obtained at the hilltop. This inevitably conducts baseline algorithms to remain stucked at the bottom of the hill, where the energy consumption is low. Only our approach, that contains an explicit incentive for widening the occupancy of the state space, has the capability to reach the most rewarding states, finally providing a policy that solves the task.

The Bipedal Walker (second row of figure 4) is also a problem that combines dense and sparse rewards. A negative (-100) reward is undergone when the agent falls down, and a positive (+100) reward is gained when the agent reaches the end of the track. The continual dense rewards provide an incentive for staying upright and increase the velocity. This task reveals more tricky to train than expected, and contains enough variability for the agent to develop various gaits and locomotion patterns over the course of learning. Our approach shows here a clear advantage, that is maybe more obvious when comparing with the SAC. Like in the Mountain car, the problem is about reaching a final (distal) end-of-path objective, from which a strong bonus allows to "freeze" the behavior in a favorable locomotion pattern. The (S-shaped) discontinuity in the MaCAO and the SAC learning curves reflect the reaching of the distal objective, after which close-by policies are followed. The difference in the two curves is the *time* at which it is attained, that is less than 10^5 iterations in the first case, and more than 10^6 in the second. This one order difference illustrates the disadvantage of occupancy-agnostic optimization methods.

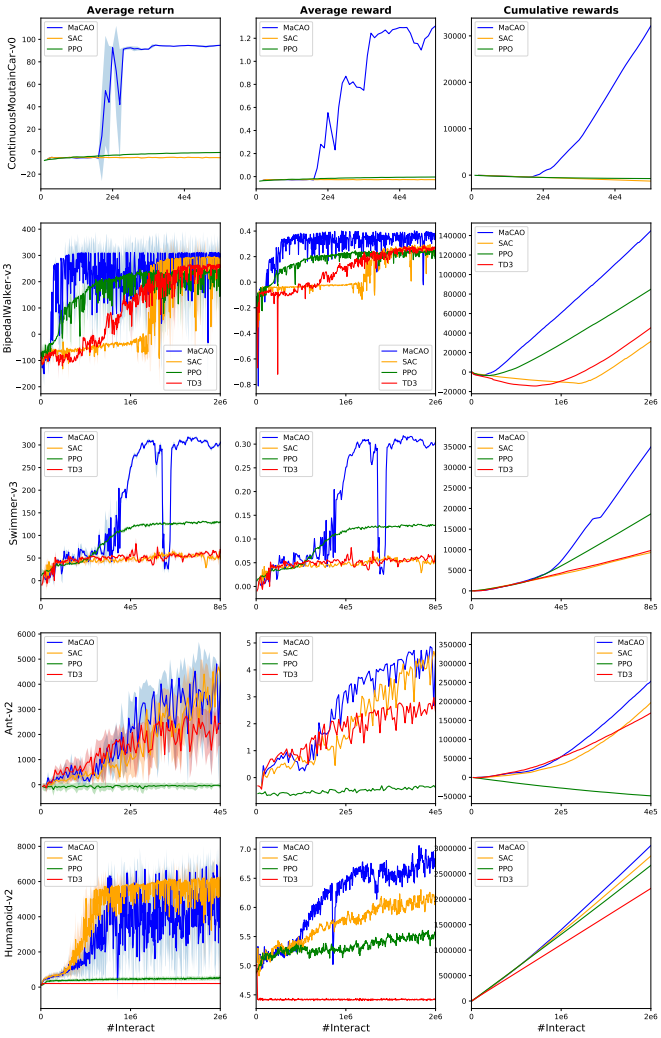

Figure 1: **Methods comparison** Average episode rewards, average rewards and cumulative rewards are compared in the course of learning for the MaCAO, SAC, PPO and TD3 frameworks, on 5 continuous state/continuous control problems. Row 1: Gym *Continuous Mountain Car* problem. $\beta = 10, \lambda = 0.1, \gamma = 0.99$, 2 hidden layers with $N = 32$ neurons. Row 2: Gym/MuJoCo *Swimmer* problem. $\beta = 30, \lambda = 0.3, \gamma = 0.995$, 2 hidden layers with $N = 32$ neurons. Row 3: Gym/Box2D *Bipedal Walker*. $\beta = 30, \lambda = 1, \gamma = 0.99$, 2 hidden layers with $N = 64$ neurons. Row 4: Gym/MuJoCo *Ant*. $\beta = 10, \lambda = 0.3, \gamma = 0.99$, 2 hidden layers with $N = 256$ neurons. Row 5: Gym/MuJoCo *Humanoid*. $\beta = 10, \lambda = 3, \gamma = 0.98$, 2 hidden layers with $N = 256$ neurons.

MuJoCo (Todorov et al., 2012) is a physic engine that has become the inevitable playground over which training algorithms are compared. The different MuJoCo tasks are mostly similar and mainly concerned with the development of locomotion patterns. Failures are not penalized, and the main differences lie in the number of degrees of freedom to be coordinated. The three tasks selected here are ranked by increasing order of difficulty. The swimmer task (third row) is concerned with the development of a locomotion pattern that is swimming in a liquid medium. The reward is only the speed at which an eel-like agent manage to swim over the place (that is coordinating segments in a periodic manner). This tasks contain a local optimum that corresponds to a rower pattern that coordinates the extremal segments, and a global optimum that corresponds to a classic swimming ripple from the head toward the tail. Despite its apparent simplicity, this ripple pattern is not contained in the reward signal, only an extensive exploration such as the one provided by our approach allows to reach the optimum.

The fourth task, known as the "ant" aims at controlling the locomotion of a 4-legged agent. The state space contains a detailed account of joint angle and torque moments plus contact sensors in a 111 dimension observation vector (Schulman et al., 2015), but the control space is more reduced (8 dofs). Here again the displacement speed is the main incentive, with a survival bonus, and an energy cost penalty. All 3 actor critic frameworks (namely MaCAO, SAC and E3D) are here capable to reach a decent locomotion pattern in about 400000 iterations of the dynamics, which can be considered data efficient here. A short advantage of our approach is reflected in the cumulative rewards, that are higher, mainly reflecting here a higher velocity attained. This high velocity is counterbalanced by a higher failure frequency, dimming the average episode return at the advantage of the other methods.

This more "risky" behavior is also observed in the humanoid task, that is mostly similar in its principle (move as fast as possible without falling down). This task is only more demanding in terms of coordination patterns, for it requires both to maintain a balance of the torso and the upper limbs while finding a way to coordinate the legs in order to move forward. The large number of degrees of freedom makes the task quite tricky to solve, and the unlimited number of possible locomotion patterns often result in strange-looking final gaits. Only the SAC and the MaCAO methods allow here to reach decent locomotion patterns in the limited number of steps considered. When looking in detail, the light advantage observed for the SAC algorithm on the average episode return is reversed when considering the average return. This apparent contradiction is explained when looking at the detailed behavior. Here, once again, the high-speed risky locomotion patterns developed in the MaCAO framework result in a higher number of early failures. This is not related to a risk-seeking incentive, but is rather explained by a tendency to maintain a high diversity of behavior while pursuing the reward-guided objective, which reveals to be more risky when the balance of the body needs to be maintained over time.

5 DISCUSSION

This work participates to a general trend toward the development of data models in reinforcement learning, that play the role of an auxiliary helper toward better exploring the world. This is known of practical use and has been largely exploited so far in the large family of curiosity-driven and maximum-entropy algorithms. Our contribution here is to provide a more detailed appraisal of the benefits and putative risks of such a construct. It is shown here to frame into a larger Bayesian/variational optimization where the future data plays the role of a model, and where an evidence lower bound is maximized through gradient ascent over the policy parameters. The general principles exposed point to the importance of an occupancy model that synthesizes the general distribution of the agent's environmental states over which it can act (defining a virtual "territory"). Those occupancy models are the subject of frequent updates as the exploration progresses and that new states are undisclosed during the course of the training. By making an additional uniform prior assumption on the occupancy, the resulting loss expresses a balance between two concurrent tendencies, namely the widening of the occupancy space and the maximization of the rewards, reminding of the classical exploration/exploitation trade-off. The consequence is a shift in the objective occupancy pursued, that relaxes the constraint on fitting the initial Bellman objective. Both are embodied in a MSBE Loss operating on a single Q-function in our implementation (though this is not necessary the case). Computer simulations illustrate the benefit of our conceptual developments, both in the case of sparse and dense rewards.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Joshua Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29:1471–1479, 2016.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Emmanuel Daucé. End-effect exploration drive for effective motor learning. In *International Workshop on Active Inference*, pp. 114–124. Springer, 2020.

Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

Matthew Fellows, Anuj Mahajan, Tim GJ Rudner, and Shimon Whiteson. Virel: A variational inference framework for reinforcement learning. *Advances in Neural Information Processing Systems*, 32:7122–7136, 2019.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Thomas Furmston and David Barber. Variational methods for reinforcement learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 241–248. JMLR Workshop and Conference Proceedings, 2010.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32:12488–12497, 2019.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Richard J Herrnstein. Relative and absolute strength of response as a function of frequency of reinforcement. *Journal of the experimental analysis of behavior*, 4(3):267, 1961.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Jurgen Schmidhuber et al. Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of SICE*, 48(1), 2009.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *31st Conference on Neural Information Processing Systems (NIPS)*, volume 30, pp. 1–18, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

A APPENDIX

Assume a parameterize Q-function Q_ψ that should undergo a dual optimization under two concurrent objectives. This implies in short that the value settled in the Q-function may arbitrate between the reward-seeking and the occupancy-seeking objectives, so that maximizing π with respect to Q only may be equivalent to the previous concurrent optimization. Considering (s, a, r, s', a') a sample, a simple way to retain the divergence information is to simply add the log difference information $\log q_\pi(s', a') - \log \rho^*(s', a')$ to the current Q-function, making it possible to recover the original formula. This sketch of idea implies the use of two concurrent update rules in the Q-function itself. A first term $L_{\text{ref}}(s, a, r, s', a')$ would be the traditional MSBE loss on rewards. The second loss needs to consider the log-difference itself as an *error*, for it to be "contained" (so to say) in Q after the update. Consider the implicit reward :

$$r_{\text{KL}}(s, a) \triangleq (1 - \gamma) \left[Q_\psi(s, a) - \frac{1}{\beta} (\log q_\pi(s', a') - \log \rho^*(s', a')) \right] \tag{9}$$

and let $\hat{Q}(s, a) = r_{\text{KL}}(s, a) + \gamma Q_\psi(s', a')$. Then a complementary MSBE loss is:

$$L_{\text{KL}}(s, a, r, s', a') = (r_{\text{KL}}(s, a) + \gamma Q_\psi(s', a') - Q_\psi(s, a))^2$$

So that the final loss expression is:

$$\begin{aligned} L_{\text{MaCAO}}(s, a, r, s', a') =& \lambda \left(r + \gamma Q_\psi(s', a') - Q_\psi(s, a) \right)^2 \\ &+ \left(\left[-\frac{(1-\gamma)}{\beta} (\log q_\pi(s', a') - \log \rho^*(s', a')) \right] + \gamma \left(Q_\psi(s', a') - Q_\psi(s, a) \right) \right)^2 \end{aligned} \tag{10}$$

The additional *precision* hyperparameter λ plays here the role of an extra regularizer, helping to adjust between both terms in case of highly sparse rewards.

Algorithm 1 Maximum Credit Assignment Occupancy (MaCAO)

Require: π_θ (actor), Q_ψ (critic), $\mathcal{B}$ (replay buffer), β, γ, λ (hyperparameters)
  while number of trials not exceeded **do**
    initialize the environment
    while trial not terminated **do**
      observe s
      choose $a \sim \pi_\theta(A|s)$
      read s', r
      store (s, a, r, s') in $\mathcal{B}$
    end while
    if $\mathcal{B}$ is full enough **then**
      randomly sample a batch of (next) states $\{s^+, ...\}$ from $\mathcal{B}$.
      estimate q with a nonparametric method.
      while number of batch updates not exceeded **do**
        randomly sample a batch of transitions $b = \{(s, a, r, s'), ...\}$ from $\mathcal{B}$.
        for all $(s, a, r, s') \in b$ **do**
          estimate $\log q(s')$ and set r_{KL} (eq. 9)
          sample $a' \sim \pi_\theta(A|s')$
          calculate $L_{\text{MaCAO}}(s, a, r, s', a')$ (eq. 10)
        end for
        update the critic Q_ψ by gradient descent over all losses.
        for all $s \in b$ **do**
          sample $a \sim \pi_\theta(A|s)$
          estimate $Q_\psi(s, a)$
          calculate $L_{\text{act}} = -Q_\psi(s, a) + \frac{1}{\beta} \log \pi_\theta(A|s)$ (using the reparameterization trick)
        end for
        update the actor π_θ by gradient descent over all losses.
      end while
    end if
  end while