Morphological Computation as Intrinsic Reward for Reinforcement Learning

Omar Baiazid, Jan Dohmen, Nihat Ay and Manfred Eppe
Institute for Data Science Foundations
Hamburg University of Technology
Hamburg, Germany
Email: {omar.baiazid, jan.dohmen, nihat.ay, manfred.eppe}@tuhh.de

***Abstract*—We propose morphological computation (MC) – the amount to which an agent exploits the physical dynamics of its body morphology and its environment – as intrinsic reward for a reinforcement learner. We show that a small amount of MC-based intrinsic reward improves the learning process for simulated environments using an off-policy actor-critic algorithm.**

I. Introduction

Morphological computation considers an agent's body morphology and the physical dynamics of its environment to simplify the computation required for action-selection. An extreme example with a high degree of morphological computation is the passive walker by McGeer, walking autonomously without any electronic computation, simply by exploiting physical laws of gravity, force and momentum (see Figure 1). Here, the behavior of the agent is completely inherent in its mechanical design and, therefore, self-organized.

As an example involving a low amount of morphological computation, consider a similar walker, where each joint is controlled by a motor, so that each motor requires an individual control signal that needs to be computed electronically for letting it walk.

Fig. 1. The passive walker by McGeer does not require any control signal for walking. It is an extreme example of morphological computation (https://www.youtube.com/watch?v=-YYVH36wDGE).

In this work, we ask in how far morphological computation can improve the performance of a reinforcement learning agent that learns to interact with its environment. We hypothesize, that encouraging at least a small amount of morphological computation improves the learning process. We will implement this encouragement as an intrinsic reward for the reinforcement learner.

II. Background

In this work, we consider fully observable Markov Decision Processes (MDP). With s_t, we denote a state representation at a discrete time step t, and with a_t an action at step t, causing a state transition to a successor state s_{t+1}, receiving a reward r_{t+1}. In this work, we use goal-conditioned RL and hindsight experience replay [1] to determine extrinsic rewards, denoted r_t^{ex}. It depends on how close a state s_t is to a given goal state g. The objective of RL is to find a policy π that maximizes the extrinsic rewards.

For our experiments, we use the soft actor-critic algorithm (SAC) [3] to achieve this objective. Here, the Q-function for a policy π is defined as

$$Q^\pi(s,a) \approx r + \gamma \left(Q^\pi(s', \tilde{a}') - \alpha \log \pi(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi(\cdot|s') \tag{1}$$

Many approaches exist to improve the reinforcement learning process by optimizing the exploration behavior of an agent through intrinsic rewards [4, 2]. As a novelty, we here use morphological computation (MC) to determine the intrinsic rewards. There a many ways to quantify MC [6]. In this work, we define MC as the KL-divergence

$$KL(p(s_{t+1}|s_t, a_t)||p(s_{t+1}|a_t)) \tag{2}$$

Intuitively, the MC-value determined in Equation 2 states that much MC is used if the influence of state s_t for the transition with a_t to s_{t+1} is high. In contrast, there is only little MC if the transition to s_{t+1} depends only on the action a_t, but not so much on the previous state s_t. Hence, we define the intrinsic reward r_{t+1}^{in} as the normalized KL-divergence $KL(p(s_{t+1}|s_t, a_t)||p(s_{t+1}|a_t))$. By normalized, we mean that we offset and scale the KL-divergence value, so that it is within the same numerical range as the extrinsic rewards.

In a similar approach [5], the authors consider Causal Action Influence (CAI) as intrinsic reward for RL. In their work, CAI is defined as $KL(p(s_{t+1}|s_t, a_t)||p(s_{t+1}|s_t))$. Here, the second parameter of the KL-divergence, is, unlike in our approach, the probability to achieve a successor state without considering the action. Also, the authors use the deep deterministic policy gradient (DDPG) method instead of SAC.

III. Morphological Computation as Intrinsic Reward for Reinforcement Learning

To determine the MC-value for the intrinsic reward according to Equation 2, we require probabilistic state transition models to determine $p(s_{t+1}|s_t, a_t)$ and $p(s_{t+1}|a_t)$. We refer to the former as the state-action model and to the latter as the action-model. We implement these models using probabilistic neural networks trained with maximum likelihood estimation.

For the state-action model, this is achieved as follows (and similarly for the action-model):

- Define and randomly initialize a feed-forward neural network f_θ with input s_t, a_t and outputs $\mu_{s_{t+1}}, \sigma_{s_{t+1}}$, denoting mean and standard deviation of a normal distribution over the estimated successor state s_{t+1}.
- Collect triples (s_{t+1}, s_t, a_t) by exploration.
- train f_θ with gradient descent according to the loss function $\ell = -\log p(s_{t+1}|\mu_{s_{t+1}}, \sigma_{s_{t+1}})$. That is, given mean and standard deviation of the probability distribution over s_{t+1}, as estimated by the neural network, the loss is the negative log likelihood that s_{t+1} is indeed the successor state.

Based on the probabilistic dynamics models trained with MLE, we compute the intrinsic reward according to Equation 2. We balance the intrinsic rewards r_t^{in} with the extrinsic rewards r_t^{ex} using a parameter α as follows to obtain the total reward r_t^{total}.

$$r_t^{total} = (1-\alpha) r_t^{ex} + \alpha r_t^{in}. \tag{3}$$

This reward is then provided to the soft actor-critic's Q-function (cf. Equation 1) to train the agent. The proposed approach is tested on several tasks, as described in the following experiments section.

IV. Experiments and Results

We perform experiments in simulated environments illustrated in Figure 2

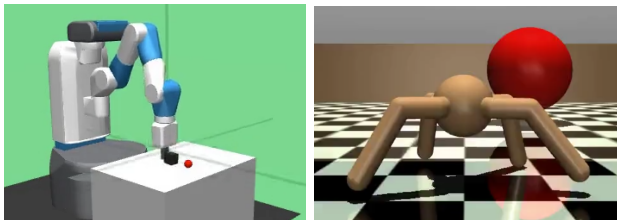

Fig. 2. Experimental environments: FetchPush (left) and AntReacher (right). The goals are indicated with the red sphere. In FetchPush, the agent must learn to push the black box towards the goal. In AntReacher, the agent must learn to walk to the goal position.

To evaluate whether MC as intrinsic reward improves the learning, we tested different values for the balancing parameter α in Equation 3. For both environments, $\alpha = 0.2$ was most successful and outperformed the original SAC algorithm without intrinsic reward, as illustrated in Figure 3.

The plots show that the agent learns the task significantly faster if MC as intrinsic reward with the balance parameter $\alpha = 0.2$ is used. For larger values $0.2 < \alpha < 0.6$ results were marginally better than for $\alpha = 0$. For even larger values, the performance dropped significantly below $\alpha = 0$.

V. Conclusion

We consider intrinsic rewards based on morphological computation (MC) as a form of self-organization that helps to maximize the extrinsic rewards. We showed that using MC as intrinsic reward can lead to beneficial results regarding the learning performance.

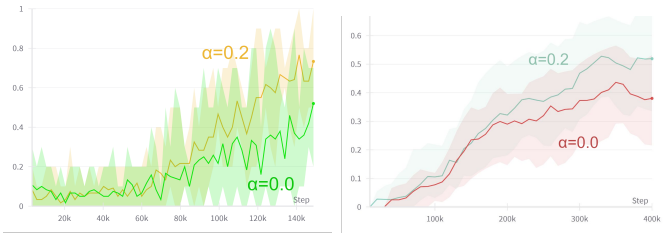

Fig. 3. Results of training FetchPush (left) and AntReacher (right). The x-axis represents the number of action steps and the y-axis represents the success rate, i.e., the fraction of episodes where the target has been reached.

We evaluated our method for the case of soft actor-critic and two simulated environments, but the general approach is agnostic to the RL-algorithm and task or environment used. As future work, it would be interesting to see how the method behaves when applied to other RL algorithms and tasks.

Acknowledgment

The authors acknowledge funding by the German Research Foundation via the projects MoReSpace/IDEAS (Nr.402776968), and LeCAREbot (Nr. 433323019).

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Neural Information Processing Systems (NIPS)*, pages 5048–5058, 2017.

[2] Manfred Eppe, Christian Gumbsch, Matthias Kerzel, Phuong D H Nguyen, Martin V Butz, and Stefan Wermter. Intelligent problem-solving as integrated hierarchical reinforcement learning. *Nature Machine Intelligence*, 4(1), 2022.

[3] 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 (ICML)*, pages 1861–1870. PMLR, 2018.

[4] Frank Röder, Manfred Eppe, Phuong D. H. Nguyen, and Stefan Wermter. Curious hierarchical actor-critic reinforcement learning. In *International Conference on Artificial Neural Networks (ICANN)*, pages 408–419, 5 2020.

[5] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal Influence Detection for Improving Efficiency in Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 22905–22918, 2022.

[6] Keyan Zahedi and Nihat Ay. Quantifying morphological computation. *Entropy*, 15(5):1887–1915, 2013.