Zhu, Y., Rossiter, J., & Hauser, H. (2019). Learning in Growing Robots: Knowledge Transfer from Tadpole to Frog Robot. In *Conference on Biomimetic and Biohybrid Systems: Living Machines 2019: Biomimetic and Biohybrid Systems* (pp. 378-382). (Lecture Notes in Computer Science; Vol. 11556). Springer, Cham. https://doi.org/10.1007/978-3-030-24741-6_42

# Learning in Growing Robots: Knowledge Transfer from Tadpole to Frog Robot

Yiheng Zhu[1,2], Jonathan Rossiter[1,2], and Helmut Hauser[1,2]

[1] Department of Engineering Mathematics, University of Bristol, Bristol, UK,
[2] Bristol Robotics Laboratory, Bristol, UK
{yiheng.zhu, jonathan.rossiter, helmut.hauser}@bristol.ac.uk

**Abstract.** Inspired by natural growing processes, we investigate how morphological changes can potentially help to lead and facilitate the task of learning to control a robot. We use the model of a tadpole that grows in four discrete stages into a frog. The control task to learn is to locomote to food positions that occur at random positions. We employ reinforcement learning, which is able to find a tail-driven swimming strategy for the tadpole stage that transitions into a leg-driven strategy for the frog. Furthermore, by using knowledge transferred from one growing stage to the next one, we were able to show that growing can benefit from guiding the controller optimization through morphological changes. The results suggest that learning time can be reduced compared to the cases when learning each stage individually from scratch.

**Keywords:** biomimetic robotics · knowledge transfer · reinforcement learning · morphological computation

## 1 Introduction

Biological systems are remarkably robust and adaptive, and they are able to control their highly complex bodies seemingly with ease. One of the reasons is that some biological systems grow. Their morphological structures are simple in the early stages and can therefore be easily controlled. Consequently, it is almost trivial to find a good and simple control policy. During growth, the complexity of the body increases and the corresponding control structure adapts by exploiting previous experience, which means it does not have to learn from scratch. A potential interpretation is that the morphological changes introduced by the growing process serve as guidances for the controller optimization. Work by Bongard [1] shows that such controllers are more robust and can be learnt faster. However, in that work a genetic algorithm was used, which is a population-based method, to obtain these controllers. To improve the learning efficiency, we propose to adopt reinforcement learning and show how this could be advantageous in the case of a robot model that grows stepwise from a tadpole to a frog structure. This concept of reusing and transferring knowledge has been explored previously by shaping [2], lifelong learning [3], curriculum learning [4], and transfer learning [5]. We form a sequence of four structurally ordered robot models inspired by the metamorphosis of the frog (Xenopus Laevis). These range from the tadpole form (actuated tail), through the froglet form (actuated tail and legs), and finally to the frog form (actuated legs only). The task is to repeatedly reach an item of food spawned at random positions. The reinforcement learning

algorithm applied in the robots is proximal policy optimization (PPO) [6]. The policy and value networks trained from the previous stage are transferred directly to bias the learning process for later stages, which is compared to learning from scratch at each stage. We experimentally show that knowledge transfer in growing robots serves as desirable parameter initialization for later stages, which accelerates learning, compared to learning from scratch.

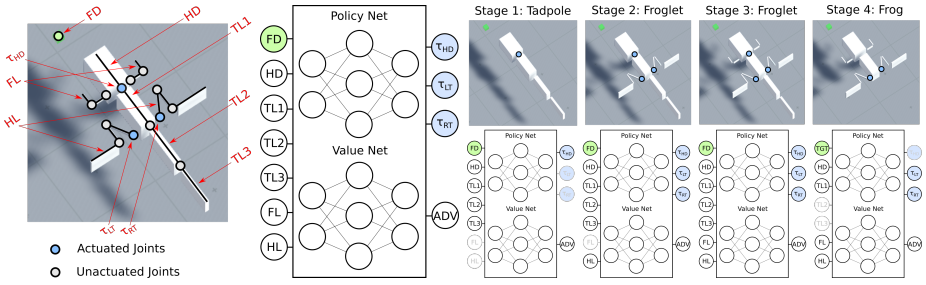



Fig. 1: Model robot growth and network change. (a) Model notations: HD (head), TL1 (tail 1), TL2 (tail 2), TL3 (tail 3), FL (forelimb), HL (hindlimb), FD (food), $\tau_{\text{HD}}$ (head torque), $\tau_{LT}$ (left thigh torque), $\tau_{RT}$ (right thigh torque), and ADV (action value advantage). (b) Sensor inputs on the left, neural networks in the middle, actuator outputs on the right. Policy network for decision making (upper) and value network for stabilizing cumulative reward estimation (lower). (c) Models and networks of 4 stages for tadpole to frog metamorphosis: tail-driven tadpole (stage 1), tail-leg-driven froglet without forelimbs (stage 2), tail-leg-driven froglet with forelimbs (stage 3), and leg-driven frog (stage 4).

## 2 Method

**Model Structure** The Unity ML-Agents toolkit [7] is used for physical simulation and learning. The body of the tadpole/frog robot consists of four cuboids in sequence, which are head, tail 1, tail 2, and tail 3 (Fig. 1(a)). Two three-link hindlimbs are connected to tail 1 where the feet have large areas for providing thrust while thighs and calves are relatively slim. Two two-link forelimbs are connected to the head, which have limited contribution to swimming [8]. Passive springs in all hinge joints between adjacent cuboids bring the robot back to its default position when no external force is applied. The learnt controller needs to exploit the unactuated degrees of freedom. We approximate the interaction between robot and water by applying linear and angular drag to all body parts. In this way, realistic underwater locomotion for the tadpole/frog robot is simulated with relatively low computational complexity.

**Model Inputs and Outputs** We define the sensor inputs (states) and actuator outputs (actions) as follows. For sensing, the robot's state consists of the positions, velocities, angles, and angular velocities of all body parts, which is close to a complete description of its configuration, and we can therefore say the Markov property is approximately satisfied. In terms of actuation, the tadpole/frog robot can apply continuous torques, $\tau \in [-2, 2]\ Nm$, to three hinge joints: (i) the joint between head and tail 1, (ii) the joint between tail 1 and left thigh, and (iii) the joint between tail 1 and right thigh (blue nodes in Fig. 1).

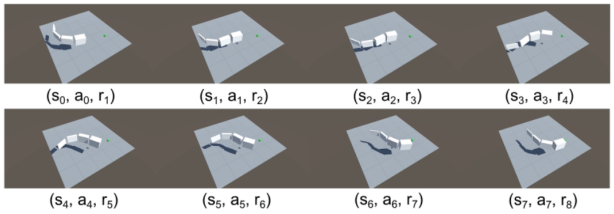



Fig. 2: A sample trajectory with 8 frames: $s$ for state, $a$ for action, and $r$ for reward. Subscripts denote frame counts in the current trajectory.

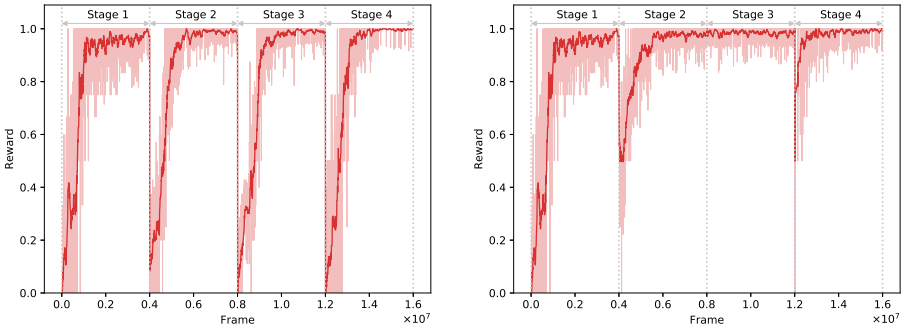
**Sampling** Reinforcement learning algorithms repeat between sampling and optimization to iteratively improve a policy, in which the policy maps sensor inputs (state) to actuator outputs (action). Fig. 2 is a sample trajectory of the tadpole robot swimming from the frame when the food is spawned ($s_0$) to the frame when it reaches the food ($s_7$). There are 8 frames in this sample trajectory, but the lengths of trajectories can vary depending on when the tadpole robot reaches the food or hits the arena's boundary. The robot's 2D position is defined by $x \in (-5, 5)$ $m$ and $y \in (-5, 5)$ $m$. Each frame is treated as a sample tuple, $(s_t, a_t, r_{t+1})$, where the tadpole robot at state $s_t$ takes action $a_t$ based on its policy and gets a scalar reward $r_{t+1}$. The reward value is 1.0 for the frame when the tadpole robot reaches the food ($s_7$ in Fig. 2), otherwise the reward value is always 0.0. Each time the food is reached, it is respawned at a random position in $x \in (-4, 4)$ $m$ and $y \in (-4, 4)$ $m$ with equal probability.

**Optimization** TensorFlow [9] is used as the neural network library for policy optimization. We follow the policy gradient framework where the policy parameters are iteratively updated following the ascent direction of reward value. The policy gradient algorithm adopted is proximal policy optimization (PPO) [6], which is a first-order optimization method that strikes a balance among simplicity, data efficiency, scalability, and robustness. We use a feedforward policy network with 2 hidden layers, 128 hidden neurons per layer, and a value network of the same structure for caching and averaging rewards.

## 3 Results

To investigate the effect of knowledge transfer from the tadpole to the frog robot, we perform two experiments: (i) learning from scratch for all 4 stages separately and (ii) learning with policy and value networks transferred from previous stages.

Fig. 3 summarizes the results. On the left (Fig. 3(a)), all stages are learnt from scratch, while on the right (Fig. 3(b)), knowledge (policy and value networks) are reused from previous stages. It can be observed that knowledge transfer serves as a good strategy to accelerate convergence. Especially, from stage 2 to stage 3 we can see that reusing previous knowledge is beneficial. On the other hand, moving from stage 1 to stage 2 the advantage of knowledge transfer, although clearly seen, is not that prominent. The difference is that from stage 1 to stage 2 we add additional actuators and legs, which is a significant morphological change, while between stage 2 and stage 3 we only add small passive forelimbs, which do not change the dynamics significantly. This might point to the fact that we have to consider further, smaller steps between stage 1 and stage 2 to achieve smoother




(a) Without knowledge transfer (b) With knowledge transfer

Fig. 3: Cumulative reward versus frame. (a) Without knowledge transfer: each stage is trained from scratch. (b) With knowledge transfer: policy and value networks are transferred from the previous stage.

transfer of knowledge. Finally, interestingly, the morphological change between stage 3 and stage 4 is significant as well, as we remove almost a third of the body by removing the tail, however, this part only contributes to passive dynamics and, therefore, allows smooth knowledge transfer. We plan to investigate these questions further in future work.

## 4 Conclusion

In this paper, we study knowledge transfer in a growing tadpole/frog robot which learns an underwater food-seeking task. By directly transferring converged policy and value networks from one stage to the next, we show that learning is accelerated when knowledge is transferred between different physical morphologies, for example as a robot grows. This is expected to have advantage in future morphologically adapting robots.

## References

1. Bongard, J.: Morphological change in machines accelerates the evolution of robust behavior. *PNAS*, **108**(4), pp. 1234-1239. (2011)
2. Skinner, B.: Science and human behavior. Simon and Schuster. (1953)
3. Thrun, S.: Is learning the $n^{\text{th}}$ thing any easier than learning the first?. *NIPS*, pp. 640-646. (1996)
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. *ICML*, pp. 41-48. (2009)
5. Taylor, M., Stone, P.: Transfer learning for reinforcement learning domains: A survey. *JMLR*, pp. 1633-1685. (2009)
6. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. (2017)
7. Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mattar, M., Lange, D.: Unity: A General Platform for Intelligent Agents. *arXiv preprint arXiv:1809.02627*. (2018)
8. Tassava, R. A.: Forelimb spike regeneration in Xenopus laevis: testing for adaptiveness. *JEZ-A*, **301**(2), pp. 150-159. (2004)
9. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M.: TensorFlow: A system for large-scale machine learning. *USENIX*, pp. 265-283. (2016)