Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer

René Traoré* Hugo Caselles-Dupré* Timothée Lesort* Te Sun Natalia Díaz-Rodríguez David Filliat

Autonomous Systems and Robotics Laboratory (ENSTA Paris), Theresis Laboratory (Thales), AI Lab (Softbank Robotics Europe), * Equal contribution

Abstract

We focus on the problem of teaching a robot to solve tasks in a continual learning (CL) scenario. The robot should learn to solve each task encountered, without forgetting past ones. Our approach is built on state representation learning (SRL), reinforcement learning (RL), policy distillation and transfer from simulation to real life.

Contribution

1) Applying SRL [1] into a continual learning setting of reinforcement learning. The SRL method learns a compact and efficient representation of data that facilitates learning a policy [2].
2) Proposing a CL algorithm based on distillation that does not need manual task indicator at test time, but learns to infer the task from observations only.
3) Successfully applying the learned policy on a real robot.

Tasks

We present 2 different 2D navigation tasks to a 3 wheel omni-directional robot. We want it to learn to solve them sequentially. The robot has first access to task 1 only, and then to task 2 only. It should learn a single policy that solves both tasks and be applicable in a real life scenario. The robot can perform 4 high level discrete actions (move left/right, move up/down).

Task 1: Target Reaching (TR): Reaching a red target randomly positioned.

Task 2: Target Circling (TC): Circling around a fixed blue target. Reward function:

$$R_t = \lambda(1 - \lambda(\|z_t\| - r_{circle})^2) * \|z_t - z_{t-k}\|_2^2 + \lambda^2 R_{t,bump} \quad (1)$$

where z_t is the agent's 2D position at timestep t, r_{circle} is the radius of the circle, and λ a balancing coefficient to induce circular movement. At each episode, the robot starting position z_0 changes randomly. k is the duration of the window history, in timesteps, to assure movement, here $k = 10$. $R_{t,bump}$ is a reward = -1 when hitting the border.

Approach

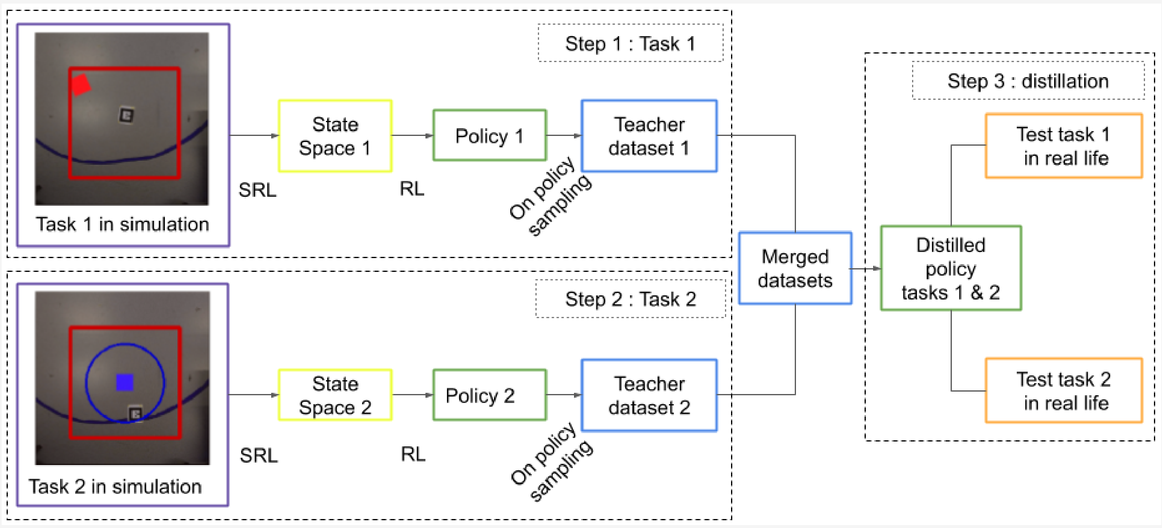

Figure 1: Step 1 and 2 correspond to learn respectively policies for tasks 1 and 2, and using those to create distillation datasets. Step 3 is the distillation of the two policies into a single policy, which can be deployed in simulation and on the real robot.

Method: The SRL model is trained with an auto-encoder and an inverse model. Once the SRL model is trained on task i, we only keep its encoder E_i and use it to learn our policy π_t on top with PPO2 [3]. We generate then a dataset D_{π_t} on policy with π_t. When all tasks have been learned, we merge all D_{π_t} and distill the knowledge into a new model.

Distillation: The distillation process [4] consists of transferring knowledge from one or several neural network(s) (the teacher(s)) to another (the student). The teacher annotates a database with a soft label (actions probabilities). In our setting, the student is trained to fit these soft labels in order to learn a policy. This process is used here to distill our two policies into one model.

Results

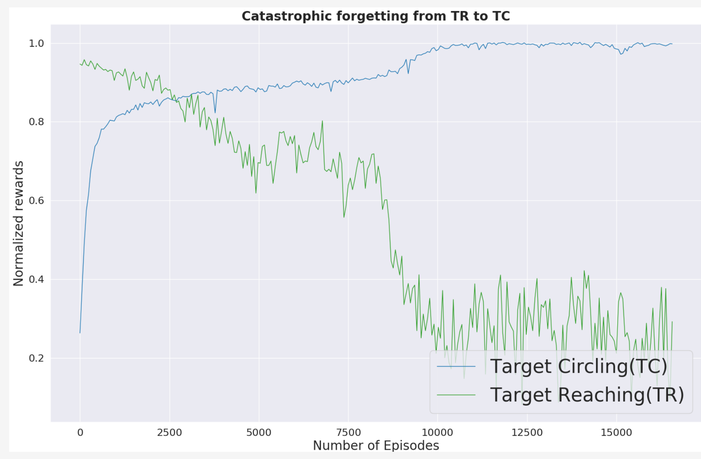

Figure 2: Demonstration of catastrophic forgetting while fine-tuning: The blue curve shows the progression of mean reward of the policy currently learned (TC), green curve represents its mean reward when evaluated on the task previously learned (TR). Both evaluations were made with 5 random seeds.

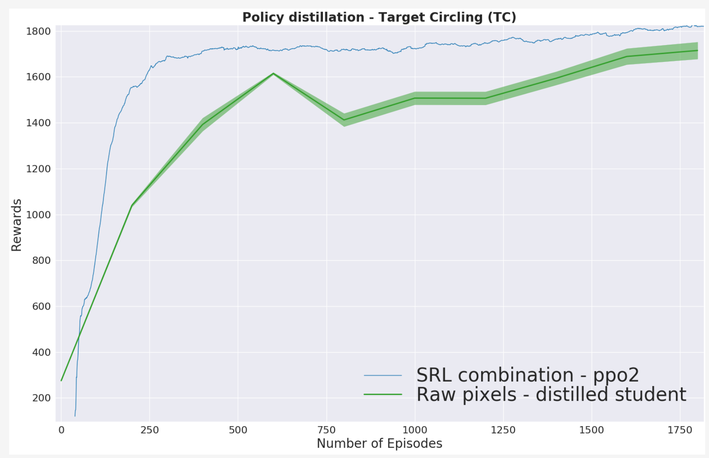

Figure 3: Demonstration of the effectiveness of distillation. Blue: RL training Green: distilled student policy (on 8 seeds). At each point of the curve (timestep), the blue policy is used to be distilled in a student policy.

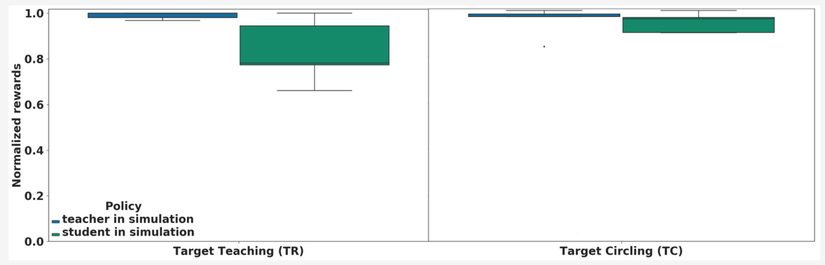

Figure 4: Comparison between performance (normalized mean reward and standard error) of a policy trained on one task only to distilled student policy on the two tasks. The student policy has similar performance on both tasks. **Left:** Target Reaching (TR). **Right:** Target Circling (TC) task.

Conclusion

In this work we experimented SRL with RL policy distillation to learn policies in a simulated continual setting and transfer them into a real life setting. These preliminary results show a promising research direction where a real robot can learn a sequence of tasks without forgetting.
Future work should experiment with more tasks, as well as aim at getting rid of the simulation stage.

- **Repository**: `https://github.com/kalifou/robotics-rl-srl/tree/circular_movement_omnibot`
- **Acknowledgement**: This work is supported by the EU H2020 DREAM project (Grant agreement No 640891).

References

[1] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, 2018.

[2] A. Raffin, A. Hill, K. R. Traoré, T. Lesort, N. Díaz-Rodríguez, and D. Filliat, "Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics," *Workshop on "Structure and Priors in Reinforcement Learning" (SPiRL) at ICLR*, 2019.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[4] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.