DYNAMIC ROBOT SYSTEMS GROUP
OXFORD ROBOTICS INSTITUTE

APPLIED ARTIFICIAL INTELLIGENCE LAB
OXFORD ROBOTICS INSTITUTE

Robot Learning for Autonomous Assembly

Chia-Man Hung, Sasha Salter, Ingmar Posner, Ioannis Havoutis

UNIVERSITY OF OXFORD

Learning complex manipulation tasks by playing.

Problem

- Starting from a random initialisation, learn to perform manipulation tasks on the **Human Support Robot (HSR)**.
- We formulate it as a **reinforcement learning (RL)** problem with sparse reward.

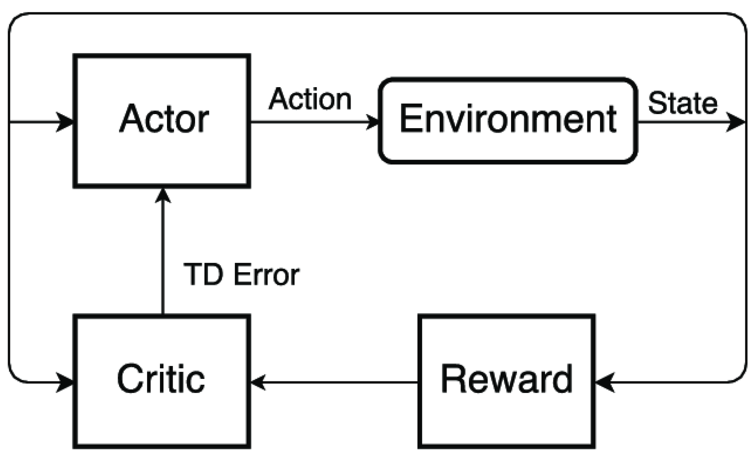


Fig. 1: Schematic of an actor-critic agent.

Scheduled Auxiliary Control

Code: https://github.com/ascane/sacq-hsr

Key idea

- High-level scheduling of auxiliary tasks and the execution of auxiliary policies to explore efficiently ([1]).

Learning the policy (Actor θ)

- The action-value function $Q_{\mathcal{T}}(s_t, a_t)$ for task $\mathcal{T}$

$$Q_{\mathcal{T}}(s_t, a_t) = r_{\mathcal{T}}(s_t, a_t) + \gamma \mathbb{E}_{\pi_{\mathcal{T}}}\left[\sum_{t'=t+1}^{\infty} \gamma^{t'} r_{\mathcal{T}}(s_{t'}, a_{t'})\right]$$

where $\mathcal{T} \in \mathcal{A} \cup \{\mathcal{M}\}$, $\pi_{\mathcal{T}} = \pi_\theta(a|x, \mathcal{T})$.

- To learn the parameters, we optimise

$$\mathcal{L}(\theta) = \mathcal{L}(\theta; \mathcal{M}) + \sum_{k=1}^{|\mathcal{A}|} \mathcal{L}(\theta; \mathcal{A}_k)$$

with $\mathcal{L}(\theta; \mathcal{T}) = \sum_{\mathcal{B} \in \mathcal{A} \cup \{\mathcal{M}\}} \mathbb{E}_{p(s|\mathcal{B})} \left[Q_{\mathcal{T}}(s, a) | a \sim \pi_\theta(\cdot|s, \mathcal{T})\right]$.

Learning the Q-function (Critic ϕ)

- Since the policy parameters are constantly being updated, the trajectories are generated by different behaviour policies.
- The off-policy evaluation Retrace [2] is used to optimise the estimator $\hat{Q}^{\pi}_{\mathcal{T}}(s, a; \phi)$.

Learning the scheduler

- To determine the current intention of the agent based on previous intentions.

$$R_{\mathcal{M}}(\mathcal{T}_{0:H-1}) = \sum_{h=0}^{H} \sum_{t=h\xi}^{(h+1)\xi-1} \gamma^t r_{\mathcal{M}}(s_t, a_t)$$

$$\pi_{\mathcal{S}}(a_t|s_t, \mathcal{T}_{0:h-1}) = \sum_{\mathcal{T}} \pi_\theta(a_t|s_t, \mathcal{T}) P_{\mathcal{S}}(\mathcal{T}|\mathcal{T}_{0:h-1})$$

$$\mathcal{L}(\mathcal{S}) = \mathbb{E}_{P_{\mathcal{S}}}\left[R_{\mathcal{M}}(\mathcal{T}_{0:H-1}|\mathcal{T}_h \sim P_{\mathcal{S}}(\mathcal{T}|\mathcal{T}_{0:h-1}))\right]$$

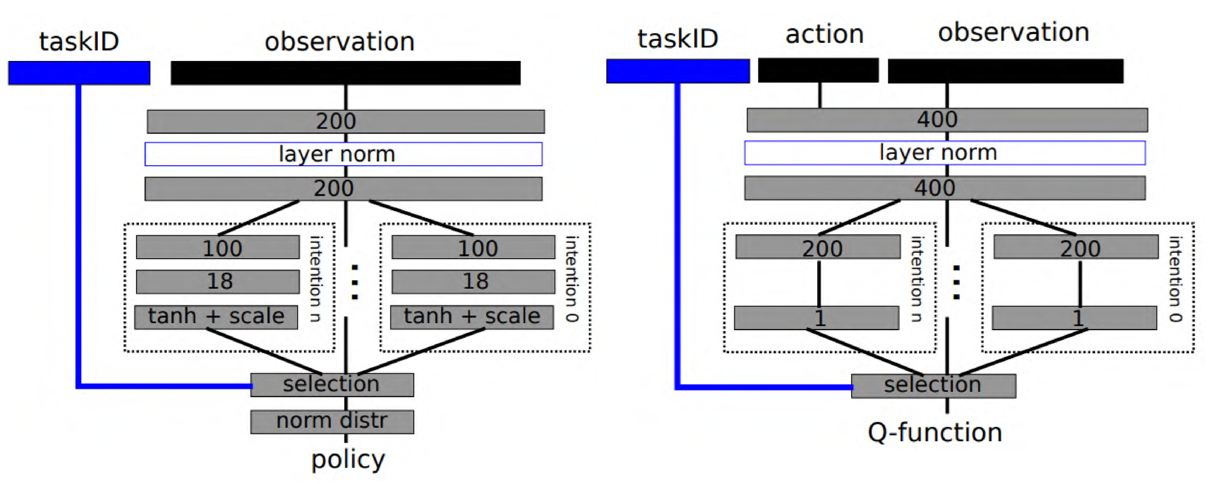
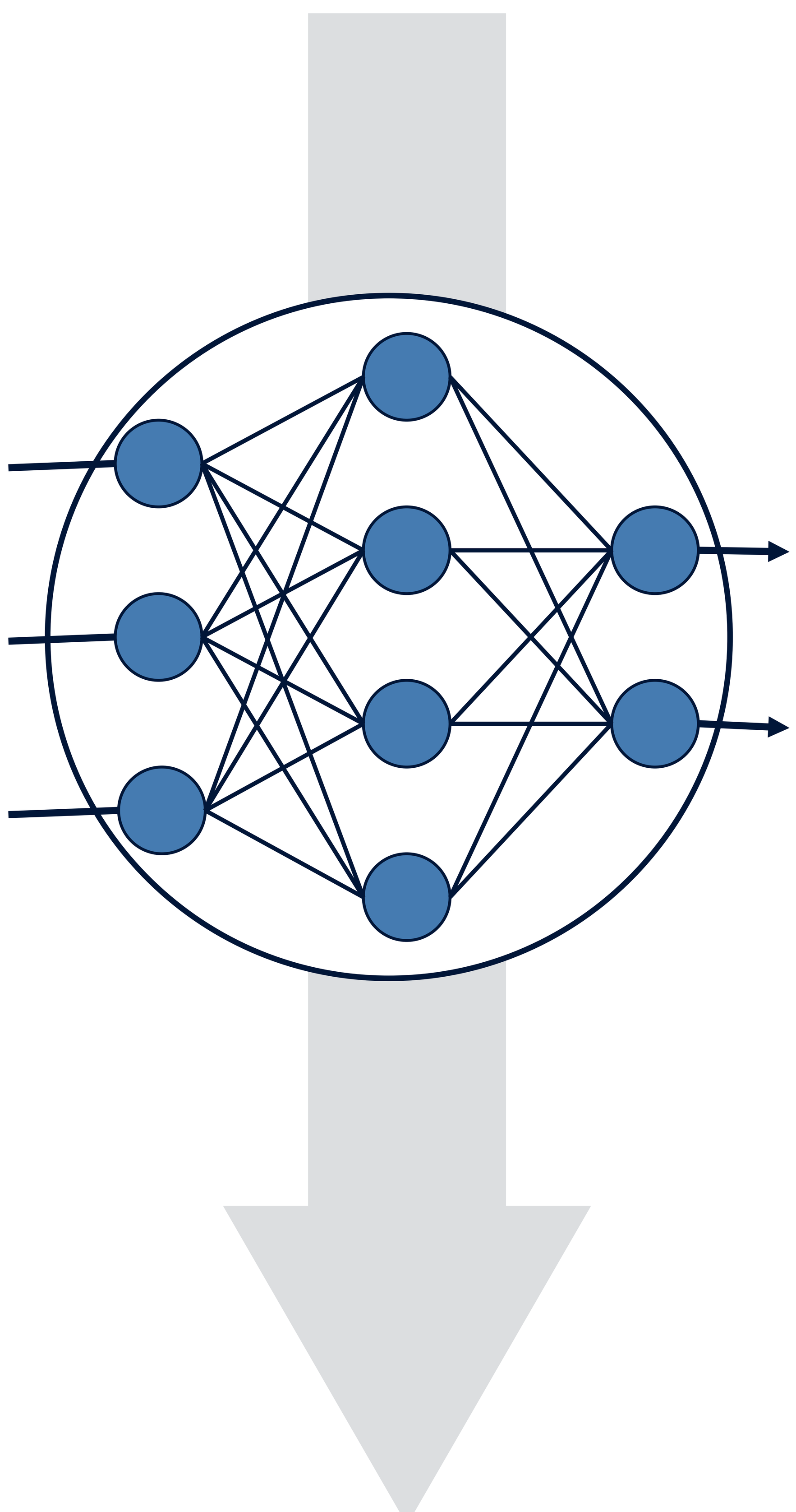


Fig. 2: Network architectures for actor and critic taken from [1].

Observation



Action

Autonomous Intelligent Machines & Systems

EPSRC
Pioneering research and skills

Simulation Environment

Code: https://github.com/ascane/gym-gazebo-hsr

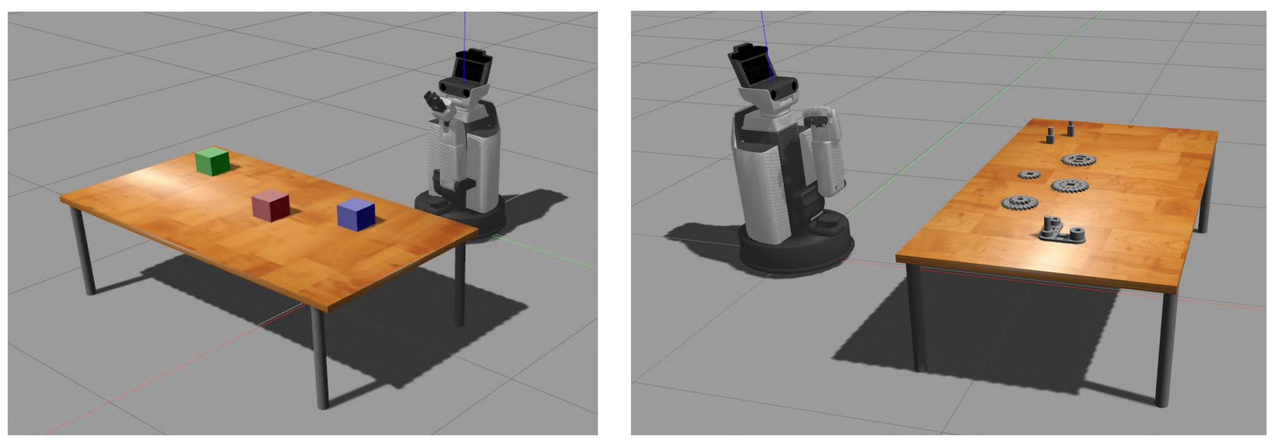


Fig. 3: Stacking boxes and the assembly challenge in simulation.

- We create an **OpenAI Gym environment** based on **Gazebo** (a physics engine, 3D modeling and rendering tool) and **ROS** (software frameworks to interact with the robot) [3].

Experiments

Stacking two boxes

- Stack the green box on top of the red one
- Three auxiliary task with sparse reward – Reach, Move, Lift.

Siemens Assembly Challenge

- Assemble different components to the end configuration as shown in Fig. 4.

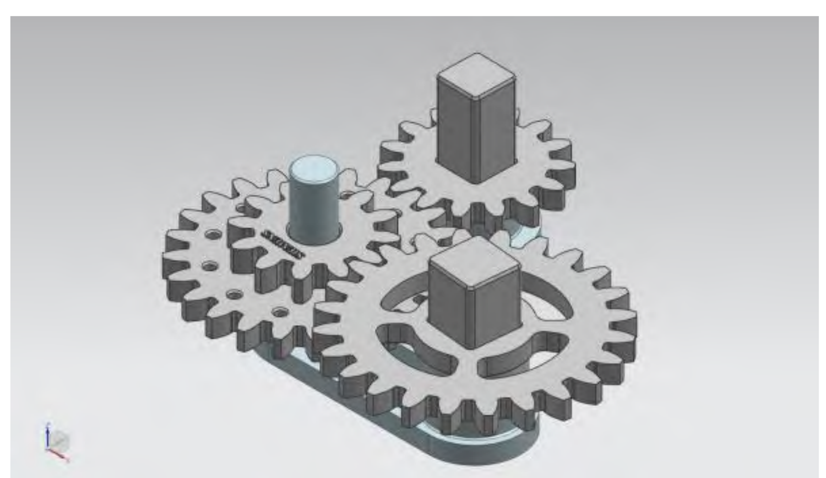


Fig. 4: The desired end configuration of the Siemens assembly challenge. Credits: Siemens Corporate Technology.

Why is it challenging?

- Gazebo is too slow for RL algorithms.
- Hard to design auxiliary tasks for more complex tasks.

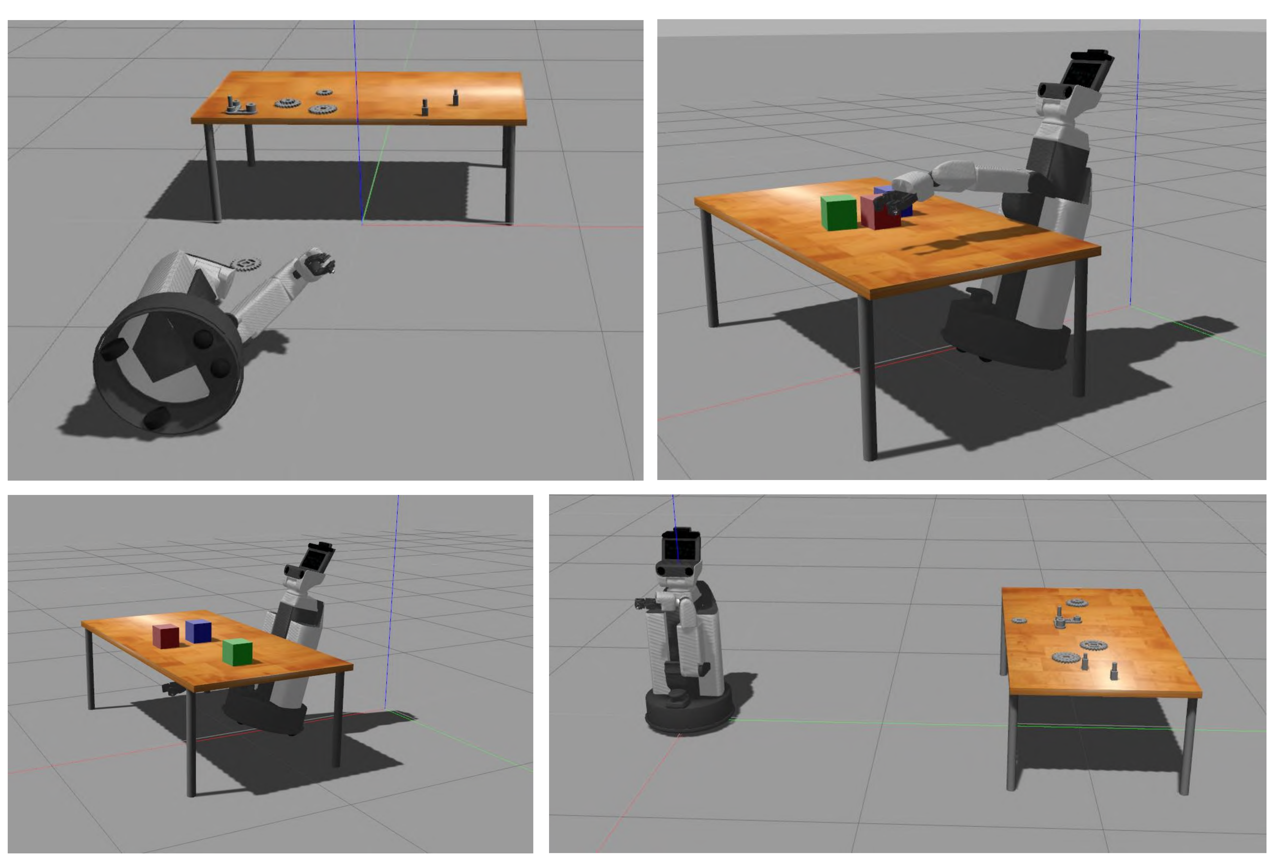


Fig. 5: HSR falls over / reaches forward too much / gets stuck under the table / runs away from the table.

[1] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van deWiele, V. Mnih, N. Heess, and T. Springenberg. Learning by playing - solving sparse reward tasks from scratch. arXiv preprint arXiv:1802.10567, 2018.
[2] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare. Safe and efficient off-policy reinforcement learning. In Advances in Neural Information Processing Systems, pages 1054–1062, 2016.
[3] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero. Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo. arXiv preprint arXiv:1608.05742, 2016.