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.

**Simulation Environment**

- Code: [https://github.com/ascane/gym-gazebo-hsr](https://github.com/ascane/gym-gazebo-hsr)

**Key idea**

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

**Learning the policy (Actor $\theta$)**

- The action-value function $Q_T(s_t, a_t)$ for task $T$

\[
Q_T(s_t, a_t) = r_T(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} \left[ \sum_{t'=t+1}^{\infty} \gamma^{t'} r_T(s_{t'}, a_{t'}) \right]
\]

- where $T \in \mathcal{A} \cup \{M\}$, $P_T = P_0(a|s, T)$.
- To learn the parameters, we optimise

\[
\mathcal{L}(\theta) = \mathbb{E}(\theta; \mathcal{A}) + \sum_{k=1}^{\mathcal{A}} \mathcal{L}(\theta; \mathcal{A}_k)
\]

**Learning the Q-function (Critic $\phi$)**

- 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 $\hat{Q}_T^*(s, a; \phi)$

**Learning the scheduler**

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

\[
R_M(T_{0:B-1}) = \sum_{b=1}^{B} \sum_{i=1}^{H} \gamma^i \mathbb{E}_{s_i} \left[ Q_T(s_i, a_i) \right]
\]

\[
\pi_S(a_i|s_i, T_{0:B-1}) = \sum_{s}\pi_0(a_i|s_i, T)P_B(T|T_{0:B-1})
\]

\[
\mathcal{L}(S) = \mathbb{E}_{P_T} \left[ R_M(T_{0:B-1}|T_0 \sim P_B(T|T_{0:B-1})) \right]
\]

**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.

**Why is it challenging?**

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

---

**References**

