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**
- 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].

**Scheduled Auxiliary Control**
- Code: [https://github.com/ascane/sacq-hsr](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 \( \theta \))**
- The action-value function \( Q_T(s_t, a_t) \) for task \( T \)
  \[
  Q_T(s_t, a_t) = r_T(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} \left[ \gamma^r r_T(s_{t+1}, a_{t+1}) \right]
  \]
  where \( T \in A \cup \{M\} \), \( \pi_T = \pi_0(a|x, T) \).
- To learn the parameters, we optimise
  \[
  \mathcal{L}(\theta) = \mathcal{L}(\theta; A) + \sum_{k=1}^{|A|} \mathcal{L}(\theta; A_k) 
  \]
  with \( \mathcal{L}(\theta; T) = \sum_{s,T\in\mathcal{R}(T)} \mathbb{E}_{s\sim\pi}(Q_T(s,a) | a \sim \pi_0(|s, T]) \).

**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 \( Q^*_T(s, a; \phi) \).

**Learning the scheduler**
- To determine the current intention of the agent based on previous intentions.
  \[
  R_M(T_0, T_{-1}) = \sum_{k=1}^{|A|} \sum_{l=1}^{\mathcal{M}-1} \gamma^r \pi_M(s_0, a_0) \]
  \[
  \pi_S(a_0|s_0, T_0, T_{-1}) = \sum_{s, T} \pi(a_0|s_0, T) P_0(T|T_0, T_{-1}) 
  \]
  \[
  \mathcal{L}(S) = \mathbb{E}_{s_0} \left[ R_M(T_0, T_{-1}|T_0 \sim P_0(T|T_0, T_{-1})) \right] 
  \]

**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.

**Figures**
- Fig. 1: Schematic of an actor-critic agent.
- Fig. 3: Stacking boxes and the assembly challenge in simulation.
- Fig. 4: The desired end configuration of the Siemens assembly challenge. Credits: Siemens Corporate Technology.
- Fig. 5: HSR falls over / reaches forward too much / gets stuck under the table / runs away from the table.

**References**