Human-Robot Gym: Benchmarking Reinforcement Learning in Human-Robot Collaboration

Jakob Thumm, Felix Trost, and Matthias Althoff

***Abstract*— Deep reinforcement learning (RL) has shown promising results in robot motion planning with first attempts in human-robot collaboration (HRC). However, a fair comparison of RL approaches in HRC under the constraint of guaranteed safety is yet to be made. We, therefore, present `human-robot gym`, a benchmark suite for safe RL in HRC. We provide challenging, realistic HRC tasks in a modular simulation framework. Most importantly, `human-robot gym` is the first benchmark suite that includes a safety shield to provably guarantee human safety. This bridges a critical gap between theoretic RL research and its real-world deployment. Our evaluation of six tasks led to three key results: (a) the diverse nature of the tasks offered by `human-robot gym` creates a challenging benchmark for state-of-the-art RL methods, (b) by leveraging expert knowledge in form of an action imitation reward, the RL agent can outperform the expert, and (c) our agents negligibly overfit to training data.**

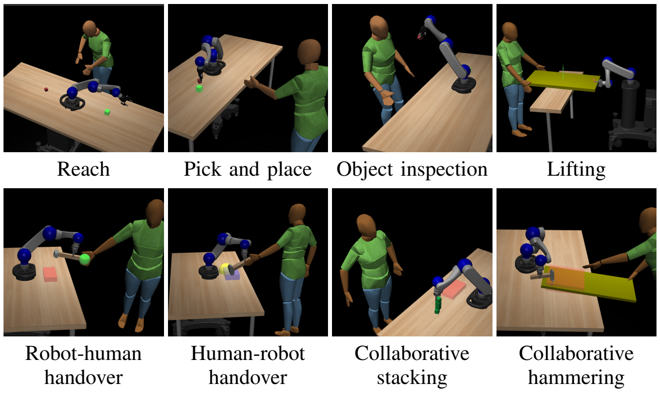

Fig. 1. `Human-robot gym` presents eight challenging HRC tasks.

I. Introduction

Recent advancements in deep reinforcement learning (RL) are promising for solving intricate decision-making processes [1] and complex manipulation tasks [2]. These capabilities are essential for human-robot collaboration (HRC), given that robotic systems must act in environments featuring highly nonlinear human dynamics. Despite the promising outlook, the few works on RL in HRC confine themselves to narrow task domains [3]. Two primary challenges impeding the widespread integration of RL in HRC are safety concerns and the diversity of tasks. The assurance of safety for RL agents operating within human-centric environments is a hurdle as agents generate potentially unpredictable actions, posing substantial risks to human collaborators. Current HRC benchmarks [4], [5] circumvent these safety concerns by focusing on interacting with primarily stationary humans.

In this paper, we propose `human-robot gym`[1], a suite of HRC benchmarks that comes with a broad range of tasks, including object inspection, handovers, and collaborative manipulation, while ensuring safe robot behavior by integrating SaRA shield [6], a tool for provably safe RL in HRC. With its set of challenging HRC tasks, `human-robot gym` enables training RL agents to collaborate with humans in a safe manner, which is not possible with other benchmarks. `Human-robot gym` comes with pre-defined benchmarks that are easily extendable and adjustable. We then track all relevant performance and safety metrics to allow an extensive evaluation of the solutions. Our benchmark suite features the following key elements that lower the entry barrier into the field of RL in HRC:

- Pre-defined tasks, see Fig. 1, with varying difficulty, each with a set of real-world human movements.
- Available robots: Panda, Sawyer, IIWA, Jaco, Kinova3, UR5e, and Schunk.
- Provable safety for HRC using SaRA shield in addition to static and self-collision prevention.
- High fidelity simulation based on MuJoCo [7].
- Support of joint space and workspace actions.
- Highly configurable and expandable benchmarks.
- Environment definition based on the OpenAI gym standard to support state-of-the-art RL frameworks, such as stable-baselines 3 [8].
- Pre-defined expert policies for gathering imitation data and performance comparison.
- Easily reproducible baseline results, see Sec. V.

This article is structured as follows: Sec. II introduces previous work in RL for HRC, compares `human-robot gym` to other related benchmarks in the field, and gives a short overview of imitation learning approaches. Sec. III presents our benchmark suite in detail. We then present additional tools supporting users to solve `human-robot gym` tasks in Sec. IV. Sec. V evaluates our benchmarks experimentally and discusses the results. Finally, we conclude this work in Sec. VI.

The authors are with the Department of Computer Engineering, Technical University of Munich, Germany. `jakob.thumm@tum.de`, `felix.trost@tum.de`, `althoff@tum.de`

[1] `human-robot gym` is available at `https://github.com/TUMcps/human-robot-gym`

II. Related Work

Semeraro et al. [3] summarize recent efforts in machine learning for HRC. They identify four typical HRC applications: collaborative assembly [9], [10], object handover [11]–[13], object handling [14], [15], and collaborative manufacturing [16].

Recent developments in RL evoke the need for comparable benchmarks in various applications. One of the most used benchmark suites for robotic manipulation is robosuite [17], which offers a set of diverse robot models, realistic sensor and actuator models, simple task generation, and a high-fidelity simulation using MuJoCo [7]. Further notable manipulation benchmarks are included in Orbit [18], which focuses on photorealism; Behavior-1K [19], which provides 1000 everyday robotic tasks in the simulation environment OmniGibson; and meta-world [20] for meta RL research.

None of the benchmarks mentioned above include humans in the simulation. There are, however, some benchmarks that provide limited human capabilities with a specific research focus. First, the robot interaction in virtual reality [21] and SIGVerse [22] benchmarks include real humans in real-time teleoperation through virtual reality setups. Unfortunately, this approach is unsuitable for training an RL agent from scratch due to long training times. Closest to our work are AssistiveGym [4] and RCareWorld [5]. These benchmark suites provide simulation environments for ambulant caregiving tasks. RCareWorld provides a large set of assistive tasks using a realistic human model and a choice of robot manipulators. However, AssistiveGym and RCareWorld focus on tasks where the human is primarily static or only features small, limited movements. Comparably, our work focuses on collaborative tasks, where the human and the robot play an active role, and the human movement is thus complex. Furthermore, one primary focus of `human-robot gym` is human safety, which other benchmarks only cover superficially. Also closely related to our work is HandoverSim [23], which investigates the handover of diverse objects from humans to robots. Here, prerecorded motion-capturing clips steer the human hand. However, these movements only capture the hand picking up objects and presenting them to the robot. From that point onward, the hand remains motionless [23]. Compared to our work, HandoverSim (a) does not supply motion data while the handover is ongoing, (b) has a much narrower selection of tasks, and (c) excludes safety concerns.

We utilize learning from experts [24] to provide the first results on our benchmarks. Currently, we mainly rely on two techniques: reference state initialization, which lets the agent start at a random point of an expert trajectory [25], and state-based imitation reward, which additionally rewards the agent for being close to the expert trajectory [26]. We explicitly decided against behavior cloning techniques [27] as they merely copy the expert behavior and often fail to generalize to the task objective [24].

III. Benchmark suite

We base `human-robot gym` on robosuite [17], which already provides adjustable robot controllers and a high-fidelity simulation environment with MuJoCo. Primarily, our environment introduces the functionality to interact with a human entity, define tasks with complex collaboration objectives, and evaluate human safety. In the following subsections, we describe our benchmarks, the typical workflow of `human-robot gym`, and its elements in more detail.

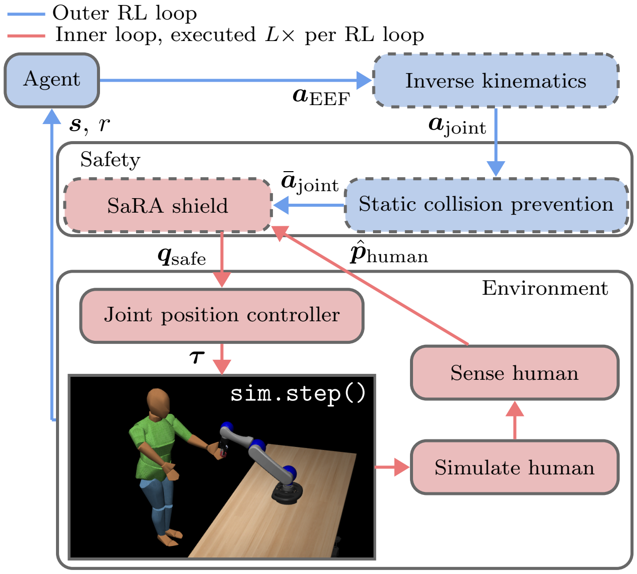
Outer RL loop
Inner loop, executed $L\times$ per RL loop
Agent
Inverse kinematics
$\boldsymbol{a}_{\text{EEF}}$
$\boldsymbol{s}$, r
$\boldsymbol{a}_{\text{joint}}$
Safety
$\bar{\boldsymbol{a}}_{\text{joint}}$
SaRA shield
Static collision prevention
$\boldsymbol{q}_{\text{safe}}$
$\hat{\boldsymbol{p}}_{\text{human}}$
Environment
Joint position controller
$\boldsymbol{\tau}$
sim.step()
Sense human
Simulate human

Fig. 2. A typical workflow of an RL cycle in `human-robot gym`. Optional elements are depicted with dashed borders, and the inner loop of the environment step is executed L times, e.g., $L = 25$. In this example, the agent returns an action in Cartesian space corresponding to a desired end effector position, which is converted to a desired joint position using inverse kinematics. Our collision prevention alters the action if the desired joint position results in a self-collision or a collision with the static environment. The shield calculates the next safe joint positions, which the joint position controller converts into joint torques that are then executed in simulation.

A. Benchmark definition

We define a benchmark in `human-robot gym` by its robot ($\mathcal{R}$), reward[2] ($\mathcal{C}$), and task (Θ) following the definition in [28, Eq. 1]. All our benchmarks are described by modular configuration files via the Hydra framework [29], which makes `human-robot gym` easily configurable and extendable. Each benchmark has a main configuration file, consisting of pointers to configuration files for the task and reward definition, robot specifications, environment wrapper settings, expert policy descriptions, training parameters, and RL algorithm hyperparameters.

a) Robot: We currently support seven different robot models: Panda, Sawyer, IIWA, Jaco, Kinova3, UR5e, and Schunk.

b) Reward: The reward in our environments can be sparse, e.g., indicating whether an object is at the target position, and dense, e.g., proportional to the Euclidean distance of the end effector to the goal. Furthermore, environments can have a delayed sparse reward signal, which should mimic a realistic HRC environment, where the agent receives the task fulfillment reward shortly after the action that completes

[2] We can convert our rewards into costs used in [28] by $c = -r$.

the task. An example of a delayed reward is when a handover was successful, but the human needs a short time to approve the execution. The reward delay serves as an additional challenge for the RL agents.

c) Task definition: Each task in human-robot gym is defined by a safety mode, objects, obstacles, human motions, and a set of goals, adding to the task definition of [28]. Human-robot gym features tasks that reflect the HRC categories introduced in [3]. Additionally, we selected two typical coexistence tasks: reach as well as pick and place. Furthermore, we provide a pipeline to generate new human movements from motion capture data, which allows users to define their own tasks and extend human-robot gym. Table I displays the default settings of each task that we use in our experiments, and a subjective estimate of the authors on the relative difficulty of each task regarding manipulation, length of the time horizon, and human dynamics. The details of the safety modes are discussed in Sec. IV-A.

B. *Typical workflow*

Fig. 2 displays a typical workflow of an RL cycle in human-robot gym. The actions of the RL agent can be joint space $\boldsymbol{a}_{\text{joint}}$ or workspace actions $\boldsymbol{a}_{\text{EEF}}$. If $\boldsymbol{a}_{\text{EEF}}$ is selected, the inverse kinematics wrapper determines $\boldsymbol{p}_{\text{joint, desired}}$ from $\boldsymbol{p}_{\text{EEF, desired}}$ and returns the joint action. The workspace actions can include the end effector orientation in $\mathrm{SO}(3)$. However, in our experiments, we only use the desired positional difference of the end effector in Cartesian space $\boldsymbol{a}_{\text{EEF}} = \boldsymbol{p}_{\text{EEF, desired}} - \boldsymbol{p}_{\text{EEF}}$ as actions, where the gripper is pointing downwards. This simplification to a four-dimensional action space (three positional actions and a gripper action) is common in literature [2], [30], [31]. Training in joint space showed similar performance in first experiments but required significantly more RL steps until convergence due to the larger action space.

The RL action might violate safety constraints. Users can, therefore, implement safety functionalities as part of the outer RL loop or the inner environment loop. We present how our additional tools use both variants to prevent collisions with static obstacles and guarantee human safety in Sec. IV. The step function of our environment executes its inner loop L times. Every iteration of the inner loop runs the optional inner safety function, the robot controller, one fixed step of the MuJoCo simulation, and the human measurement. After executing the action, the environment returns an observation and a reward to the agent.

C. *Human simulation*

Our simulation moves the human using motion capture files obtained from a Vicon tracking system. All movements are recorded specifically for the defined tasks and include task-relevant objects in the scene, ensuring realistic behavior. A limitation of using recordings are instances where the recording must be paused until the robot initiates a specific event, e.g., in a handover task. Previous works show an unnatural human behavior in these cases. To address this limitation, we incorporate idle movements representing the human waiting for an event to trigger. For each recording, keyframes can designate the start and end of an idle phase. Once reached, the movement remains idle until an event predicate σ_E is true, at which point it progresses to the successive movement. The predicate σ_E is true when the robot achieves a task-specific sub-goal and thereafter, e.g., handing over an object. Instead of simply looping the idle phase, which would lead to jumps in the movement, we alter the replay time of the recording by a set of D superimposing sine-functions:

$$t_A = \begin{cases} t, \text{ if } t \leq t_I \vee \sigma_E \\ t_I + \sum\limits_{i=1}^{D} \upsilon_i \sin\left(\left(t - t_I\right)\omega_i\right), \text{ otherwise}, \end{cases} \tag{1}$$

where υ_i and ω_i define the amplitude and frequency of the i-th sine-function during idling respectively and both are randomized at the start of each episode. The replay time can also reverse in the idling phase. The recordings to replay are randomly selected at the start of each episode, and their starting position and orientation are slightly randomized to avoid overfitting.

D. *Observation*

Human-robot gym features typical task-related and robotic observations, as shown in Table II. Objects, obstacles, goals, and human bodies have a measurable pose $\mathbf{T} \in SE(3)$. These objects are observable through the following projections (adapted from [28, Tab. II]): position in world (W) and end effector (E) frame $\boldsymbol{p}_{\text{W}} : SE(3) \rightarrow \mathbb{R}^3$, $\boldsymbol{p}_{\text{E}} : SE(3) \rightarrow \mathbb{R}^3$, Euclidean distance to the end effector $\boldsymbol{d} : SE(3) \rightarrow \mathbb{R}^+$, and the orientation in world frame given through quaternions $\boldsymbol{o}_{\text{W}} : SE(3) \rightarrow SO(3)$. The task-specific elements in Table II include those necessary to fulfill the task, i.e., $\mathbf{T}_{\text{obj},a}, a = 1, \dots, A$, $\mathbf{T}_{\text{obs},b}, b = 1, \dots, B$, $\mathbf{T}_{\text{goal},c}, c = 1, \dots, C$, and $\mathbf{T}_{\text{body},d}, d = 1, \dots, D$, with A objects, B obstacles, C goal poses, and D human bodies. The robot information contains its joint positions and velocities as well as the end effector position, orientation, and aperture. In our experiments, we found that reducing the number of elements in the observation, e.g., only providing measurements of the human hand positions instead of the entire human model, is beneficial for training performance. To emulate real-world sensors, users can optionally add noise sampled from a compact set and delays to all measurements, further reducing the gap between simulation and reality. In addition to the physical measurements, the user can define cameras that observe the scene and learn from vision inputs.

IV. Supporting tools

This section describes additional tools included in human-robot gym to provide safety and RL training functionality.

A. *Safety tools*

We can prevent static and self-collisions in the outer RL loop by performing collision checks of the desired robot trajectory using pinocchio [32]. If the trajectory resulting

TABLE I

BENCHMARK CHARACTERISTICS

Task	HRC category[1]	Safety mode[2]	Manipulation	Time-horizon	Dynamics	Reward	Reward delay	No. of motions
Reach	coexistence	SSM	easy	easy	easy	dense	no	12
Pick and place	coexistence	SSM	medium	medium	easy	sparse	no	12
Object inspection	object handling	SSM	medium	medium	medium	sparse	yes	8
Collaborative lifting	object handling	SSM	medium	medium	medium	dense	no	9
Robot-human handover	object handover	PFL	medium	medium	hard	sparse	yes	15
Human-robot handover	object handover	PFL	hard	medium	hard	sparse	yes	11
Collaborative hammering	object manufacturing	SSM	hard	medium	hard	sparse	yes	11
Collaborative stacking	object assembly	SSM	hard	hard	hard	sparse	yes	8

[1] from [3], [2] SSM: speed and separation monitoring, PFL: power and force limiting

TABLE II

OBSERVATION ELEMENTS

	Element	Observations
Robot ($\mathcal{R}$)	Joint angle	$\boldsymbol{q}$
	Joint velocity	$\dot{\boldsymbol{q}}$
	EEF aperture	φ
	EEF pose	$\boldsymbol{p}_{\mathrm{W}}(\mathbf{T}_{\mathrm{EEF}})^{\dagger}$, $\boldsymbol{o}_{\mathrm{W}}(\mathbf{T}_{\mathrm{EEF}})$
Task (Θ)	Objects	$\boldsymbol{p}_{\mathrm{W}}(\mathbf{T}_{\mathrm{obj}})$, $\boldsymbol{p}_{\mathrm{E}}(\mathbf{T}_{\mathrm{obj}})$, $\boldsymbol{d}(\mathbf{T}_{\mathrm{obj}})$, $\boldsymbol{o}_{\mathrm{W}}(\mathbf{T}_{\mathrm{obj}})$
	Obstacles	$\boldsymbol{p}_{\mathrm{W}}(\mathbf{T}_{\mathrm{obs}})$, $\boldsymbol{p}_{\mathrm{E}}(\mathbf{T}_{\mathrm{obs}})$, $\boldsymbol{d}(\mathbf{T}_{\mathrm{obs}})$, $\boldsymbol{o}_{\mathrm{W}}(\mathbf{T}_{\mathrm{obs}})$
	Goal poses	$\boldsymbol{p}_{\mathrm{W}}(\mathbf{T}_{\mathrm{goal}})$, $\boldsymbol{p}_{\mathrm{E}}(\mathbf{T}_{\mathrm{goal}})$, $\boldsymbol{d}(\mathbf{T}_{\mathrm{goal}})$, $\boldsymbol{o}_{\mathrm{W}}(\mathbf{T}_{\mathrm{goal}})$
	Goal joint angles	$\boldsymbol{q}_{\mathrm{goal}} - \boldsymbol{q}$
	Object gripped	σ_{grip}
	Object at target	σ_{target}
	Static collision	$\sigma_{\mathrm{col,\ stat}}$
	Body positions	$\boldsymbol{p}_{\mathrm{W}}(\mathbf{T}_{\mathrm{body}})$, $\boldsymbol{p}_{\mathrm{E}}(\mathbf{T}_{\mathrm{body}})$, $\boldsymbol{d}(\mathbf{T}_{\mathrm{body}})$
	Safe human contact	$\sigma_{\mathrm{contact}}$
	Critical human contact	σ_{crit}

† $\boldsymbol{p}$: position in world (W) or end effector (E) frame, $\boldsymbol{d}$: Euclidean distance, $\boldsymbol{o}$: orientation, σ: predicate

from the RL action is unsafe, we sample actions uniformly from the action space until we find a safe action.

Guaranteeing human safety in the outer RL loop is challenging, as the time horizon of RL actions is relatively long, e.g., $200\,\mathrm{ms}$. Hence, checking safety only once before execution would lead to a very restrictive safety behavior [33]. Therefore, we ensure human safety in the inner environment loop. We provide the tool SaRA shield introduced for robotic manipulators in [6], [34] and generalized to arbitrary robotic systems in [33]. First, SaRA shield translates each RL action into an intended trajectory. In the subsequent period of an RL action, the shield is executed L times. In each timestep, the shield computes a failsafe trajectory, which guides the robot to an invariably safe state. As defined in [6], an invariably safe state in manipulation is a condition where the robot completely stops in compliance with the ISO 10218-1 2021 regulations [35]. Next, the shield constructs a shielded trajectory combining one timestep from the planned intended trajectory with the failsafe trajectory. SaRA shield validates these shielded trajectories through set-based reachability analysis of the human and robot. For this, the shield receives the position and velocity of human body parts as measurements from the simulation. We assure safety indefinitely, provided that the initial state of the system is an invariably safe state, by only executing the step from the intended trajectory when the shielded trajectory is confirmed safe [6]. In the event of a failed safety verification, the robot follows the most recently validated failsafe trajectory, guaranteeing continued safe operation. Finally, SaRA shield returns the desired robot joint states for the next timestep to follow the verified trajectory. We then use a proportional–integral–derivative controller to calculate the desired robot joint torques.

The default mode of SaRA shield is speed and separation monitoring, which stops the robot before an imminent collision. This is too restrictive for close interaction tasks, such as handovers, as the robot must come into contact with the human. Therefore, we include a power and force limiting mode in the tool SaRA shield that decelerates the robot to a safe Cartesian velocity of $5\,\mathrm{mm\,s^{-1}}$ before any human contact, as proposed in [36, Def. 3]. Thereby, our power and force limiting mode ensures painless contact in accordance with ISO 10218-1 2021 [35]. As in the speed and separation monitoring mode, SaRA shield only slows down the robot if our reachability-based verification detects a potential collision. Otherwise, the robot is allowed to operate at full speed. We further plan to include a conformant impedance controller, as proposed in [37], in SaRA shield in the future.

B. Tools for training

To provide a perspective on the performance of RL agents in our environments, we provide both expert and RL policies with our tasks. In this work, we consider an RL agent that learns on a Markov decision process described by the tuple $(\mathcal{S}, \mathcal{A}, T, r, \mathcal{S}_0, \gamma)$ in both continuous or discrete action spaces $\mathcal{A}$ and continuous state spaces $\mathcal{S}$ with a set of initial states $\mathcal{S}_0$. Here, $T(\boldsymbol{s}_{k+1} \,|\, \boldsymbol{s}_k, \boldsymbol{a}_k)$ is the transition function, which denotes the probability density function of transitioning from state $\boldsymbol{s}_k$ to $\boldsymbol{s}_{k+1}$ when action $\boldsymbol{a}_k$ is taken. The agent receives a reward determined by the function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ from the environment. Lastly, we consider a discount factor $\gamma \in [0, 1]$ to adjust the relevance of future rewards. RL aims to learn an optimal policy $\pi^{\star}(\boldsymbol{a}_k \,|\, \boldsymbol{s}_k)$ that maximizes the expected return $R = \sum_{k=0}^{K} \gamma^k r(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{s}_{k+1})$ when starting from an initial state $\boldsymbol{s}_0 \in \mathcal{S}_0$ and following $\pi^{\star}(\boldsymbol{a}_k \,|\, \boldsymbol{s}_k)$ until termination at $k = K$ [38].

C. Pre-defined experts

We define a deterministic expert policy $\pi_{\mathrm{e}}(\boldsymbol{a}_k \,|\, \boldsymbol{s}_k)$ for each task to gather imitation data and compare performance. The experts are hand-crafted and follow a proportional control law with heuristics based on human expertise strategy, as described in full detail in the `human-robot gym` documentation.

To achieve diversity in our expert data, we add a noise term to the expert action, resulting in the noisy expert

$$\tilde{\pi}_{\mathrm{e}}(\boldsymbol{a}_k \mid \boldsymbol{s}_k, k) = \pi_{\mathrm{e}}(\boldsymbol{a}_k \mid \boldsymbol{s}_k) * f_{k,\mathbf{n}}\,, \tag{2}$$

where $*$ denotes the convolution of probability distributions, and $f_{k,\mathbf{n}}$ is the probability density function of the noise signal $\mathbf{n}$ at time k. To restrain the random process from diverting too far from the expert, we choose a mean-reverting process. In particular, we model $\mathbf{n}$ to be a vector of independent random variables n_i and discretize the univariate Ornstein–Uhlenbeck process [39] to retrieve an autoregressive model of order one. We can sample an expert trajectory $\chi = (\tilde{\boldsymbol{s}}_0, \dots, \tilde{\boldsymbol{s}}_K)$ by a Monte Carlo simulation, where we start in $\tilde{\boldsymbol{s}}_0 \in \mathcal{S}_0$, and subsequently follow $\tilde{\boldsymbol{s}}_{k+1} \sim T(\tilde{\boldsymbol{s}}_{k+1} \mid \tilde{\boldsymbol{s}}_k, \tilde{\boldsymbol{a}}_k)$ with $\tilde{\boldsymbol{a}}_k \sim \tilde{\pi}_{\mathrm{e}}(\tilde{\boldsymbol{a}}_k \mid \tilde{\boldsymbol{s}}_k, k)$ for $k = 0, \dots, K-1$. For each task in `human-robot gym`, we provide the expert policies π_{e} and $\tilde{\pi}_{\mathrm{e}}$ together with a set of M expert trajectories $\mathcal{B} = \{\chi_1, \dots, \chi_M\}$ sampled from $\tilde{\pi}_{\mathrm{e}}$.

D. Reinforcement learning agents

Soft actor-critic (SAC) [40] serves as a baseline for our experiments due to its sample efficiency and good performance on previous experiments [6]. We include three variants of imitation learning to investigate the benefit of expert knowledge for the RL agent. First, we use reference state initialization [26] to redefine the set of initial states to the set of states contained in the expert trajectories $\mathcal{S}_0 = \{\tilde{\boldsymbol{s}} \mid \tilde{\boldsymbol{s}} \in \chi, \chi \in \mathcal{B}\}$. Starting the episode from a state reached by the expert informs the agent about reachable states and their reward in long-horizon tasks.

Secondly, we evaluate a state-based imitation reward, where the agent receives an additional reward signal proportional to its closeness to an expert trajectory $\chi \in \mathcal{B}$ in state space $r_{\mathrm{SIR}}(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{s}_{k+1}, \tilde{\boldsymbol{s}}_k) = (1-\varsigma) r(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{s}_{k+1}) + \varsigma \,\mathrm{dist}(\boldsymbol{s}_k - \tilde{\boldsymbol{s}}_k)$, where $0 \leq \varsigma \ll 1$. For the distance function, we choose a scaled Gaussian function $\mathrm{dist}(\boldsymbol{x}) = 2^{-\kappa \|\boldsymbol{x}\|_2}$ with scaling factor $\frac{1}{\kappa}$ as suggested in [26]. We further apply reference state initialization when using the state-based imitation reward, as proposed in [26].

Finally, we adapt the state-based imitation reward method to an action-based imitation reward, where the agent receives an additional reward signal proportional to the closeness of its action to the expert action $r_{\mathrm{AIR}}(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{s}_{k+1}, \tilde{\boldsymbol{a}}_k) = (1-\varsigma) r(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{s}_{k+1}) + \varsigma \,\mathrm{dist}(\boldsymbol{a}_k - \tilde{\boldsymbol{a}}_k)$, with $\tilde{\boldsymbol{a}}_k \sim \tilde{\pi}_{\mathrm{e}}(\tilde{\boldsymbol{a}}_k \mid \boldsymbol{s}_k, k)$. When using action-based imitation rewards, we sample the expert policy alongside the RL policy in every step but only execute the RL action.

V. Experiments

This section presents the evaluated RL agents, shows the performance of the agents in `human-robot gym`, and discusses the results. Our experiments aim to answer three main research questions:

- Can RL be used to complete complex HRC tasks?
- How beneficial is prior expert knowledge in solving these tasks?
- Does the RL agent overfit to a limited amount of human recordings in training?

We present our results on six `human-robot gym` tasks: reach, pick and place, collaborative lifting, robot-human handover, human-robot handover, and collaborative stacking. The evaluation shows results for the Schunk robot with the rewards listed in Table I. Across all experiments, we execute $L = 25$ safety shield steps per RL step (empirically, training with $L = 50$ shows similar performance). Our training had an average runtime of $17.61\,\mathrm{s}$ per 10^3 RL steps[3]. To evaluate the benefit of expert knowledge in these complex tasks, we compare the four agents discussed in Sec. IV-D with the expert. Fig. 3 shows our main results, where we evaluate the performance every $2 \cdot 10^5$ training steps and trained all agents on five random seeds. We report the success rate, which indicates the rate at which the task was successful, and the reward normalized to the range between the minimal possible reward and the average expert reward. All plots show the mean evaluation performance during training and the $95\,\%$ confidence interval (shaded area) in the mean metric established with bootstrapping on 10^4 samples.

Our results show that the `human-robot gym` has a diverse set of tasks, from which some are already solvable, e.g., reach as well as pick and place, some show room for improvement, e.g., collaborative lifting and robot-human handover, and some are not solvable with the investigated approaches, e.g., human-robot handover and collaborative stacking. Comparing these results to the complexity estimate in Table I, we infer that the two main factors for the difficulty of a task are the complexity of the manipulation and the human dynamics. Handling these two areas will be among the main challenges for RL research in HRC.

The results in Fig. 3 further show that expert knowledge is beneficial in benchmarks with sparse rewards, with the action-based imitation reward (AIR) method showing higher or equal performance compared to the state-based one. In the pick and place task, the action-based imitation reward approach outperformed the expert policy and reached a nearly $100\,\%$ success rate. Unfortunately, constructing the action-based imitation reward requires an expert policy that can be queried online during training, which is not given in many manipulation tasks. Interestingly, the agent trained with a state-based imitation reward shows no significant improvement over the SAC agent trained only with reference state initialization in our evaluations. Our results indicate that starting the environment in meaningful high-reward states significantly improves performance in sparse reward settings. Future work could investigate if there are even more effective forms of reference state initialization that require little to no expert knowledge. Finally, expert knowledge does not

[3] Run on ten cores of an AMD EPYC™ 7763 @ 2.45GHz.

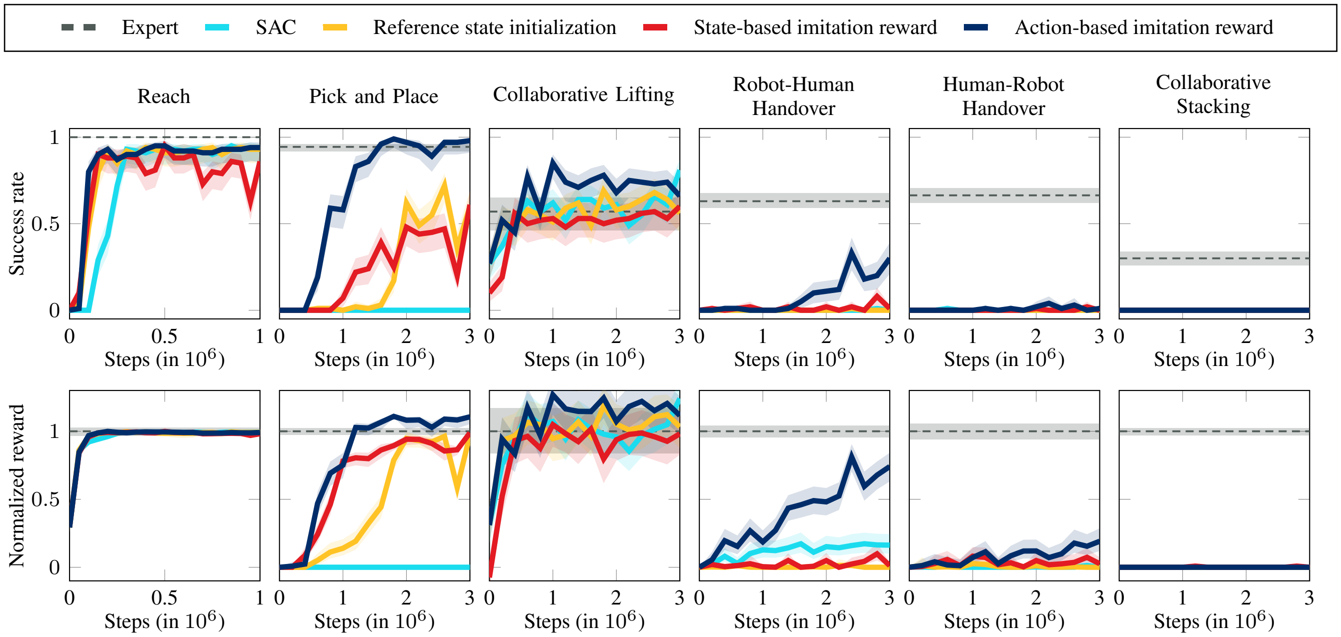

Fig. 3. Evaluation performance during training of RL agents on `human-robot gym`. The plots show the mean evaluation performance during training and the 95% confidence interval in the mean metric obtained with bootstrapping when training on five random seeds.

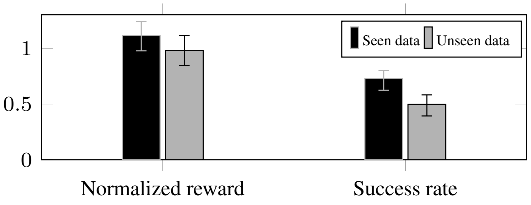

Fig. 4. Ablation study for overfitting to motion data in the training process.

improve performance in our experiments with dense reward settings, such as reaching or collaborative lifting. We assume this behavior stems from the fact that the additional action-based and state-based imitation rewards resemble the dense environment reward, yielding little additional information.

To address concerns related to overfitting to the limited amount of human motion profiles, we conduct an ablation study on the collaborative lifting task, which relies exceedingly on the human motion. This study aims to identify whether training an RL agent using a limited set of recordings instead of simulated behavior is satisfactory. Our dataset consists of nine unique human motion captures, seven of which we use as training data, reserving the remaining two for testing. We then perform a five-fold cross-evaluation, where we select different training and testing movements on each split and train RL agents on five random seeds per split. We report the average performance over the splits and seeds and the 95 % confidence interval in the mean metric of the trained SAC agent on the respective training movements (seen data) and test movements (unseen data) in Fig. 4. The reward performance of the trained agent on the unseen data is within the confidence interval of the performance on the training data. Both mean reward and success rate are only slightly lower on the unseen data, and the agent performs reasonably well. Therefore, we conclude that overfitting to the human movements is not a significant problem of `human-robot gym`.

VI. Conclusion

`Human-robot gym` offers a realistic benchmark suite for comparing performance of RL agents and safety functions in HRC. Its unique provision of a pre-implemented safety shield offers the opportunity to develop efficient HRC without designing a safety function. Our evaluation insights reveal the importance of expert knowledge in benchmarks with sparse rewards, showing that an action-based imitation reward is a promising approach if an expert is available online. In terms of practical application, it is noteworthy that an agent trained in `human-robot gym` was successfully deployed in actual HRC environments, as presented in our prior work [33]. These tests underline the critical role `human-robot gym` will play as an academic tool and as a practical approach for tangible robotic issues.

Acknowledgment

The authors gratefully acknowledge financial support by the Horizon 2020 EU Framework Project CONCERT under grant 101016007.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman *et al.*, "RT-1: Robotics Transformer for real-world control at scale," in *Proc. of Robotics: Science and Systems (RSS)*, 2022.

[2] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, and B. Dresp-Langley, "Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review," *Robotics*, vol. 10, no. 1, pp. 1–13, 2021.

[3] F. Semeraro, A. Griffiths, and A. Cangelosi, “Human–robot collaboration and machine learning: A systematic review of recent research,” *Robotics and Computer-Integrated Manufacturing*, vol. 79, pp. 1–16, 2023.

[4] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, “Assistive gym: A physics simulation framework for assistive robotics,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 10 169–10 176.

[5] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee, “RCareWorld: A human-centric simulation world for caregiving robots,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2022, pp. 33–40.

[6] J. Thumm and M. Althoff, “Provably safe deep reinforcement learning for robotic manipulation in human environments,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 6344–6350.

[7] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[8] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-Baselines3: Reliable reinforcement learning implementations,” *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[9] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. Ben Amor, “A system for learning continuous human-robot interactions from human-human demonstrations,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 2882–2889.

[10] A. Cunha, F. Ferreira, E. Sousa, L. Louro, P. Vicente, S. Monteiro, W. Erlhagen, and E. Bicho, “Towards collaborative robots as intelligent co-workers in human-robot joint tasks: What to do and who does it?” in *Proc. of the Int. Symp. on Robotics*, 2020, pp. 1–8.

[11] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, “Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks,” *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[12] D. Shukla, Ö. Erkent, and J. Piater, “Learning semantics of gestural instructions for human-robot collaboration,” *Frontiers in Neurorobotics*, vol. 12, pp. 1–17, 2018.

[13] M. Lagomarsino, M. Lorenzini, M. D. Constable, E. De Momi, C. Becchio, and A. Ajoudani, “Maximising coefficiency of human-robot handovers through reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4378–4385, 2023.

[14] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi, “Model-based reinforcement learning variable impedance control for human-robot collaboration,” *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[15] Z. Deng, J. Mi, D. Han, R. Huang, X. Xiong, and J. Zhang, “Hierarchical robot learning for physical collaboration between humans and robots,” in *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2017, pp. 750–755.

[16] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, “Efficient model learning from joint-action demonstrations for human-robot collaborative tasks,” in *Proc. of the ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, 2015, pp. 189–196.

[17] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, “robosuite: A modular simulation framework and benchmark for robot learning,” 2020.

[18] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, “Orbit: A unified simulation framework for interactive robot learning environments,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[19] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei, “BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation,” in *Proc. of the Conf. on Robot Learning (CoRL)*, vol. 205, 2022, pp. 80–93.

[20] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, “Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning,” in *Proc. of the Conf. on Robot Learning (CoRL)*, 2020, pp. 1094–1100.

[21] P. Higgins, G. Y. Kebe, A. Berlier, K. Darvish, D. Engel, F. Ferraro, and C. Matuszek, “Towards making virtual human-robot interaction a reality,” in *Proc. of the Int. Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions (VAM-HRI)*, 2021, pp. 1–5.

[22] T. Inamura and Y. Mizuchi, “SIGVerse: A cloud-based VR platform for research on multimodal human-robot interaction,” *Frontiers in Robotics and AI*, vol. 8, pp. 1–19, 2021.

[23] Y.-W. Chao, C. Paxton, Y. Xiang, W. Yang, B. Sundaralingam, T. Chen, A. Murali, M. Cakmak, and D. Fox, “HandoverSim: A simulation framework and benchmark for human-to-robot object handovers,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 6941–6947.

[24] J. Ramírez, W. Yu, and A. Perrusquía, “Model-free reinforcement learning from expert demonstrations: A survey,” *Artificial Intelligence Review*, vol. 55, no. 4, pp. 3213–3241, 2022.

[25] I. Uchendu, T. Xiao, Y. Lu, B. Zhu, M. Yan, J. Simon, M. Bennice, C. Fu, C. Ma, J. Jiao, S. Levine, and K. Hausman, “Jump-start reinforcement learning,” in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2023, pp. 34 556–34 583.

[26] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “DeepMimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[27] B. Zheng, S. Verma, J. Zhou, I. W. Tsang, and F. Chen, “Imitation learning: Progress, taxonomies and challenges,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–16, (Early Access) 2022.

[28] M. Mayer, J. Külz, and M. Althoff, “CoBRA: A composable benchmark for robotics applications,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2024.

[29] O. Yadan, “Hydra - a framework for elegantly configuring complex applications,” 2019. [Online]. Available: https://github.com/facebookresearch/hydra

[30] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, “Hindsight experience replay,” in *Proc. of the Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5055–5065.

[31] R. Li, A. Jabri, T. Darrell, and P. Agrawal, “Towards practical multi-object manipulation using relational reinforcement learning,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 4051–4058.

[32] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, “The Pinocchio C++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,” in *Proc. of the Int. Symp. on System Integrations (SII)*, 2019, pp. 614–619.

[33] J. Thumm, G. Pelat, and M. Althoff, “Reducing safety interventions in provably safe reinforcement learning,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2023, pp. 7515–7522.

[34] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira, “Effortless creation of safe robots from modules through self-programming and self-verification,” *Science Robotics*, vol. 4, no. 31, pp. 1–14, 2019.

[35] ISO, “Robotics - safety requirements - part 1: Industrial robots,” International Organization for Standardization, Tech. Rep. DIN EN ISO 10218-1:2021-09 DC, 2021.

[36] D. Beckert, A. Pereira, and M. Althoff, “Online verification of multiple safety criteria for a robot trajectory,” in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2017, pp. 6454–6461.

[37] S. B. Liu and M. Althoff, “Online verification of impact-force-limiting control for physical human-robot interaction,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 777–783.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[39] G. E. Uhlenbeck and L. S. Ornstein, “On the theory of the Brownian motion,” *Physical Review*, vol. 36, no. 5, pp. 823–841, 1930.

[40] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2018, pp. 1861–1870.