Aligning Human Preferences with Baseline Objectives in Reinforcement Learning

Daniel Marta, Simon Holk, Christian Pek, Jana Tumova, and Iolanda Leite

***Abstract*— Practical implementations of deep reinforcement learning (deep RL) have been challenging due to an amplitude of factors, such as designing reward functions that cover every possible interaction. To address the heavy burden of robot reward engineering, we aim to leverage subjective human preferences gathered in the context of human-robot interaction, while taking advantage of a baseline reward function when available. By considering baseline objectives to be designed beforehand, we are able to narrow down the policy space, solely requesting human attention when their input matters the most. To allow for control over the optimization of different objectives, our approach contemplates a multi-objective setting. We achieve human-compliant policies by sequentially training an optimal policy from a baseline specification and collecting queries on pairs of trajectories. These policies are obtained by training a reward estimator to generate Pareto optimal policies that include human preferred behaviours. Our approach ensures sample efficiency and we conducted a user study to collect real human preferences, which we utilized to obtain a policy on a social navigation environment.**

I. Introduction

Deep RL has shown great success in a variety of tasks, namely in control tasks, ranging from locomotion [1], to robot grasping [2], robot navigation [3], and human-robot interaction [4]. However, only a subset of these may be compliant with human preferences. If humans are not taken into account, deploying intelligent robots in the real world may lead to unforeseen consequences in safety [5], which holds true when considering deep RL [6]. In many robotic environments, including social robot navigation, it is often impossible to determine the ideal socially compliant reward function in advance. This may be due to the requirement of a large dataset of expert demonstrations, or the complexity of the task, which may make it nearly impossible to model without input from a human agent. Possible solutions consider human-in-the-loop approaches [7], [8], [9], where humans offer feedback within a learning loop to help optimize an agent. Although many approaches incorporate significant human input, our goal is to find a balance between reward engineering [10] and taking into account the necessary amount of human feedback. In many tasks, the general goals and penalties are known *a priori*. These can be modelled as a baseline reward to obtain policies which can be further guided by a human teacher. We consider our work to be a step towards including human preferences with baseline objectives to obtain human-aligned agents [11]. Robotic tasks typically have unique specifications that should not be significantly deviated from. However, there should be room to consider additional sub-objectives alongisde human preferences. Thus, at the core, human-aligned agents are fundamentally multi-objective [12]. The best reward function will achieve compromise between globally optimizing for both human preferences and specifications. This work poses itself as a step in the direction of including humans as part of the design of robotic tasks.

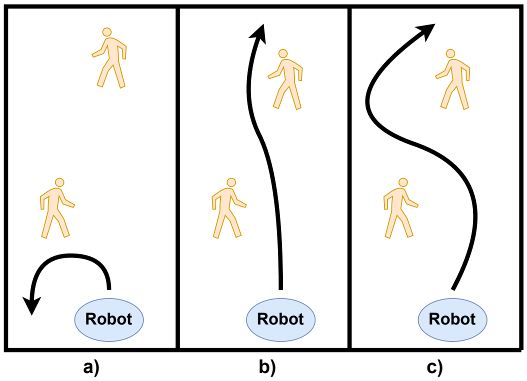

Fig. 1: Examples of different policies in a social navigation task. Information on how to optimize these policies may be both obtained in a hard constraint fashion for safety requirements, and through human feedback for perceived safety.

We use a robot navigation task scenario as motivation for the importance of correctly incorporating subjective human preferences to achieve socially compliant [13], [14] trajectories. Robot trajectories can be easily inspected visually, enabling humans to provide preferences between pairs of trajectories. Let us begin with an initial policy that results in collisions with walls (see 1.a). In such situations, it is unnecessary to rely on human feedback to correct simple behaviors, and a baseline reward can be designed with straightforward goals and penalties to address this. However, to incorporate the subtle differences between trajectories (see 1.b and 1.c) that stem from subjective human preferences, we need a solution to integrate this information with a baseline.

II. Related Work

Learning objectives for RL. Reward engineering [10] relates to handcrafting reward functions to serve a specific task or purpose. However, these can be exploited by RL

This research has been carried out as part of the Vinnova Competence Center for Trustworthy Edge Computing Systems and Applications at KTH, and partially supported by the Swedish Foundation for Strategic Research (SSF FFL18-0199) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

All of the authors are with the Division of Robotics, Perception and Learning, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden. The authors are also affiliated with Digital Futures. Mail addresses: {dlmarta, sholk, pek2, tumova, iolanda}@kth.se

algorithms leading to undesirable behaviours [5]. Alternatively, inverse RL [15] extracts reward functions from optimal policy rollouts. The functions are extracted, e.g., using heuristics alongside linear combination of known features [16] or by maximum entropy of utility functions [17]. Similar approaches focus on actively encoding human's preferences [18] or simplifying reward function design [19]. To avoid deriving reward functions solely on pre-designed sets of expert demonstrations, the authors in [20] suggest learning priors from different tasks. Alternative methods infer human objectives by evaluating hypothetical behavior [21] or by designing generative models of the policy [22]. Closer work to ours leverages reward function inference through Bayesian approaches [23], [24], alongside user input such as risk tolerance to optimize a robust policy that balances both. However, our work differs from the above by allowing the combination of baseline objectives and human-in-the-loop preferences to be decided as a trade-off in a Pareto frontier. Our work fills an important gap of allowing a flexible combination of *a priori* reward functions with completely data-driven models of preferences, which avoid over-querying humans.

Learning From Human Feedback. In [25] human feedback is used to remove bias of extracted skills from offline datasets sets and produce more human-aligned skills. Human feedback can also take more subtle forms, such as implicitly from facial features to learn reward rankings [26]. Human-robot collaborative manipulation policies can also be learned from datasets of human-human collaboration, such as learning handover tasks from conversations to obtain diverse strategies of human-robot collaboration [27], or to improve backchanneling behaviours [28] for social robots from behaviours such as nodding. Human feedback can also take the form of observing preferences of how scenes of objects are spatially arranged [29]. However, these methods require detailed datasets of human feedback and/or demonstrations, which can be costly or impossible to obtain if not readily available. We found inspiration in ideas such as repeatedly inferring reward functions from demonstrations in repeated IRL [30] [31] [32], and learning from high-level preferences, which benefits from a large body of literature [33]. Moreover, hybrid approaches such as combining both demonstrations and preferences [34], [35], were considered as a starting point for our work. Deep RL from human preferences [8] presented itself as a framework to teach agents from *tabula rasa* solely on human preferences from pairs of trajectories. A similar approach to ours, also learns from preferences with a human-in-the-loop framework [36], allowing humans to interactively teach agents through tailored feedback between clips of demonstrations to infer a reward function. A closer work to ours [37] acknowledges that the problem of inverse reinforcement learning can often be succinctly represented by a simple reward plus additional constraints, instead of fully estimating a complex reward from *tabula rasa*.

III. Background

Multi-Objective RL. We consider continuous deep RL setups with state spaces $\mathcal{S} \subseteq \mathbb{R}^n, n \in \mathbb{N}_+$, and compact and convex action spaces $\mathcal{A} \subset \mathbb{R}^m, m \in \mathbb{N}_+$. The underlying problem of the chosen task is modelled as an infinite dimensional multi-objective Markov decision process (MOMDP) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{S}_0, \boldsymbol{\mathcal{R}}, \boldsymbol{\gamma})$, where $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ returns the transition probability of ending in state $s_{t+1} \in \mathcal{S}$ when applying an action $a \in \mathcal{A}$ in state $s_t \in \mathcal{S}$, $\mathcal{S}_0$ is the initial state distribution from which s_0 is drawn, and $\boldsymbol{\mathcal{R}} : \mathcal{S} \times \mathcal{A} \rightarrow [r_1, \dots, r_k], k \in \mathbb{N}_+$ is a vectorized scalar reward, where k represents the number of different objectives for a given state transition with chosen action. The vector of discount factors $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_k], \gamma_k \in [0, 1]$ represents the discount factor for each objective k. A policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ in an MOMDP has a specific vector of expected returns $\boldsymbol{\mathcal{J}^\pi} = [\mathcal{J}^\pi_{r_1}, \dots, \mathcal{J}^\pi_{r_k}]$ associated with the vectorized reward function $\boldsymbol{\mathcal{R}}$, such that $\mathcal{J}^\pi_{r_k} = \mathbb{E}\left[\sum_{t=0}^{T} \gamma_k^t r_k(s_t, a_t) | s_0 \sim \mathcal{S}_0, a_t \sim \pi(s_t)\right]$. Thus, $\mathcal{J}^\pi_{r_k}$ represents the return of r_k, where T represents the trajectory's horizon.

Proximal Policy Optimization (PPO) [38]. PPO represents a family of policy optimization algorithms based on TRPO [39]. PPO showed better sample complexity empirically, when compared to other RL algorithms [38], and is well suited for continuous state and action spaces. It was also the algorithm used in learning from human preferences [8]. We consider both the actor (policy π) and critic (V) to be approximated by deep neural networks, with their weights parameterized by ϕ. Moreover, we use the clipped surrogate objective $\mathcal{L}^{\text{CL}}$ for the actor, since it yields better results on simulated robotic continuous control environments:

$$\mathcal{L}^{\text{CL}}(\phi) = \hat{\mathbb{E}}(\min(r_t(\phi)\hat{A}_t, \text{clip}(r_t(\phi)), 1-\epsilon, 1+\epsilon)\hat{A}_t), \quad (1)$$

where $r_t(\phi)$ is the probability ratio between the new and old policy and $\hat{A}$ is an estimator of the advantage function. To train the critic, we use the common squared-error loss:

$$\mathcal{L}^{\text{V}}(\phi) = \hat{\mathbb{E}}(V_\phi(s_t) - V_t^{target})^2 \quad (2)$$

where $V_\phi(s_t)$ is an estimation of the value for s_t and V_t^{target} the value computed with an s reward.

Learning from Human preferences [8]. A robot agent follows a policy π which in turn generates a dataset of trajectories such that $\mathcal{D}_\tau = \{\tau^1, ..., \tau^j\}, j \in \{1, \dots, N_{\text{trajectories}}\}$. Trajectories may be further divided into groups of trajectory segments. Trajectory segments are simply pairs of state-action transitions such that $\tau = (\sigma^1, \dots, \sigma^i), i \in \{1, \dots, N_{\text{segments}}\}$, and in turn each segment is represented by $\sigma^i = (s_t, a_t), \dots, (s_{t+k}, a_{t+k}), k \in \mathbb{N}_+$, where k is the length of the trajectory segment. The objective is to query humans on segments as opposed to single state-action pairs, generalizing their feedback while requesting less prompts. Pairwise comparisons of segments (σ^1, σ^2) are forwarded to humans for preference gathering. To achieve the latter, preferences need to map concrete effects in the reward function $\mathcal{R}(s_t, a_t)$. If we prefer segment σ^1 over segment σ^2, $\sigma^1 \succ \sigma^2$ (where $\succ$ is a preference operator between any two tensors), then we can verify that the sum of rewards from time t to $t+k$ for the actions taken in state s^1_t through s^1_{t+k}

is greater than the sum of rewards for the actions taken in state s_t^2 through s_{t+k}^2 for segment σ^2. Humans observe both segments and provide a feedback of preference on a segment over the other. Preferences are denoted as $\zeta = (\zeta_1, \zeta_2)$ and are tuples of the form $(1, 0)$ if $(\sigma^1 \succ \sigma^2)$, or $(0, 1)$ otherwise. If no preference is observed, a feedback of $(0.5, 0.5)$ is given, and if segments are not related $(0, 0)$. A human prefers a segment over another $(\sigma^1 \succ \sigma^2)$ if the sum of rewards for σ^1 is greater than σ^2, i.e. $\sum_t \mathcal{R}(s_t^1, a_t^1) > \sum_t \mathcal{R}(s_t^2, a_t^2)$. We denote the estimation of $\mathcal{R}$ as $\hat{\mathcal{R}}$. Considering two segments σ^1 and σ^2 of equal length k, the probability of a human choosing one over the other, such that $\sigma^1 \succ \sigma^2$ can be given by a combination of softmax functions $\hat{\mathrm{p}}(\sigma^1 \succ \sigma^2)$, such that:

$$\hat{\mathrm{p}}(\sigma^1 \succ \sigma^2) = \frac{\exp(\sum_t \hat{\mathcal{R}}(s_t^1, a_t^1))}{\exp(\sum_t \hat{\mathcal{R}}(s_t^1, a_t^1)) + \exp(\sum_t \hat{\mathcal{R}}(s_t^2, a_t^2))} \quad (3)$$

Each pair of segments (σ^1, σ^2) can be queried by either humans or a synthetic oracle, and the resulting preference is concatenated alongside the pair as $(\sigma^1, \sigma^2, \zeta)$. The queries are stored on a query dataset $\mathcal{D}_\zeta$ which is used for gradient descent. The loss function of the reward prediction estimator can be represented by the cross entropy between both predictions $\hat{\mathrm{p}}(\sigma^1 \succ \sigma^2)$ and $\hat{\mathrm{p}}(\sigma^2 \succ \sigma^1)$ thus:

$$\mathcal{L}(\hat{\mathcal{R}}) = -\sum_{(\sigma^1, \sigma^2, \zeta)} \zeta_1 \log\hat{\mathrm{p}}(\sigma^1 \succ \sigma^2) + \zeta_2 \log\hat{\mathrm{p}}(\sigma^2 \succ \sigma^1) \quad (4)$$

IV. Towards Combining Human-Aligned Preferences with Baseline Objectives

Combining diverse objectives [40] [41] has experienced recent research interest. However, many challenges arise: combining reward functions with different scales overwhelm some objectives over others (or even completely overruling them); optimizing for specific objectives over others might be desirable when trying to accomplish different robotic tasks. We present our approach to obtain posterior policies which reflect human subjective preferences as a multi-objective problem. To differentiate preferences, we use ζ for the human preferences used in preference RL, and $\mathbf{\Omega}$ as the objective preferences in multi-objective RL. Baseline and human reward estimations are identified with the subscript β and $\mathcal{H}$ respectively. Our approach is as follows (see Fig 2):

- Step 1 (starting from an initial specification): Initially, an agent is trained on a baseline reward function $\mathcal{R}_\beta$ provided by an expert. This function captures the most essential aspects of a task, such as goals and penalties, until a baseline performance level is achieved. The resulting policy is identified as π^*_β. In situations where only a high-level heuristic is accessible to evaluate trajectory quality, we can use the heuristic as a deterministic oracle (0% error rate) to label preferences and calculate $\hat{\mathcal{R}}_\beta$ through eq. 4 to obtain π^*_β.
- Step 2 (estimating subjective human preferences $\hat{\mathcal{R}}_\mathcal{H}$): Segments of rollouts from π^*_β are sampled (uniformly at random) to form queries which are then presented to humans. The resulting query dataset $\mathcal{D}_\zeta$ is used to train $\hat{\mathcal{R}}_\mathcal{H}$ (see Sec. IV-A), leading to step 3.

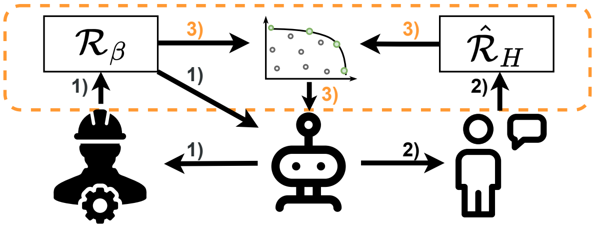

Fig. 2: Step 1) A baseline objective is provided $\mathcal{R}_\beta$ (or inferred $\hat{\mathcal{R}}_\beta$) from expert human feedback to obtain π^*_β. Step 2) Rollout examples of π^*_β are sampled and queries are prepared to be evaluated by humans. Step 3) Both $\mathcal{R}_\beta$ (or $\hat{\mathcal{R}}_\beta$) and $\hat{\mathcal{R}}_\mathcal{H}$ are combined to generate Pareto optimal policies.

- Step 3 (obtaining Pareto optimal solutions): We maintain several policies with different objective preferences. We combine $\mathcal{R}_\beta$ (which remains constant) and $\hat{\mathcal{R}}_\mathcal{H}$ according to $\mathbf{\Omega}$. Until we obtain Pareto optimal policies, we sample new rollouts and repeat the process on step 2 to continuously update $\hat{\mathcal{R}}_\mathcal{H}$ for each objective Ω_i.

A. Human-aligned reward estimation from baseline behaviours

In learning from human preferences [8] a reward estimator $\hat{\mathcal{R}}$ is trained iteratively from *tabula rasa* until a performant policy is reached. We see our framework as a natural extension, adapted towards our final goal of combining both human ζ and objective preferences $\mathbf{\Omega}$. We start by training an optimal policy given a baseline reward $\mathcal{R}_\beta$ or an estimation $\hat{\mathcal{R}}_\beta$, which we continuously update from a synthetic reward or heuristic. After optimizing an actor and a critic through eqs. 1,2 until convergence, we obtain π^*_β. An important trade-off we underline in our approach is the fact we aim at querying humans solely on trajectories which present a strong baseline of performance, as opposed to querying humans on poor performing trajectories which happens when considering a completely data-driven approach. The reduced number of queries we are able to achieve relates to querying humans at a fraction of the state-space since $\mathcal{S}_{\pi^*_\beta} \subset \mathcal{S}$. We sample a large number of rollout trajectories $\tau \sim \pi^*_\beta$, to form a dataset $\mathcal{D}_\tau$. The dataset $\mathcal{D}_\tau$ is then used to compute queries of segments, which are picked uniformly at random, and the resulting preferences ζ are collected into $\mathcal{D}_\zeta$ which in turn is used to optimize a human reward estimator $\hat{\mathcal{R}}_\mathcal{H}$ by minimizing $\mathcal{L}(\hat{\mathcal{R}}_\mathcal{H} | \mathcal{D}_\zeta)$ eq. 4.

B. Human-aligned Pareto-optimal solutions

In many tasks, such as the social navigation scenario we present in Sec.V, there is an interesting balance to be explored between hard safety constraints (baseline reward function) and perceived safety (human preferences hard to model *a priori*). Balancing between both means exploring solutions in the Pareto frontier of both. To incorporate human preferences with baseline objectives, we adopt a multi-objective RL setting in our approach. This approach provides us with greater control over the extent to which we diverge from the baseline task or converge towards a policy that is fully aligned with human preferences, with a focus on robotic applications. As part of this process, we redefine the initial MDP used to acquire a baseline policy

π^*_β to a MOMDP. To this end, we are interested in policies on the Pareto frontier $\mathcal{F}$. A Pareto frontier, constitutes all policies π' which dominate all other policies $\pi' \succ \pi$ such that $\mathcal{F} = \{\pi' \mid \forall\pi : \boldsymbol{\mathcal{J}}^{\pi'}_{\boldsymbol{\mathcal{R}}} \geq \boldsymbol{\mathcal{J}}^{\pi}_{\boldsymbol{\mathcal{R}}}\}$. A common challenge in multi-objective RL is to combine reward functions of arbitrary scale. To handle potential scale differences between the baseline objective $\mathcal{R}_\beta$ and our estimated objective $\hat{\mathcal{R}}_\mathcal{H}$, we estimate the baseline using $\hat{\mathcal{R}}_\beta$. To maintain equivalent scales for $\hat{\mathcal{R}}_\beta$ and $\hat{\mathcal{R}}_\mathcal{H}$, we use a hyperbolic tangent function for the reward models. This ensures both objectives are estimated using a similar model and makes our approach independent of the scale of $\mathcal{R}_\beta$. Another challenge arises from the fact that both $\hat{\mathcal{R}}_\beta$ and $\hat{\mathcal{R}}_\mathcal{H}$ are estimated by neural networks. This makes the estimation of the Pareto frontier more challenging than using linear functions. To address this, we consider a single-policy MOMDP approach which allows us to apply a scalarization approach between objectives. Instead of trying to estimate the full Pareto frontier, we let the user define objective preferences $\Omega = (\Omega_\beta, \Omega_\mathcal{H}), \Omega_\beta$ and $\Omega_\mathcal{H} \in [0, 1]$ and $\Omega_\beta + \Omega_\mathcal{H} = 1$. Thus Ω_β is a scalar preference for the baseline objective and $\Omega_\mathcal{H}$ the objective preference for human preferences modelled by the reward estimator $\hat{\mathcal{R}}_\mathcal{H}$. We consider a subset of Pareto-optimal solutions which can be obtained by a linear combination of the different vectorized rewards. We are interested in a convex converage set (CCS) of the Pareto frontier such that:

$$\mathcal{F}^* = \{\pi' \in \mathcal{F} \,|\, \exists\Omega\forall\pi : \Omega_\beta \underset{\beta}{\mathcal{J}^{\pi'}} + \Omega_\mathcal{H} \underset{\mathcal{H}}{\mathcal{J}^{\pi'}} \geq \Omega_\beta \underset{\beta}{\mathcal{J}^{\pi}} + \Omega_\mathcal{H} \underset{\mathcal{H}}{\mathcal{J}^{\pi}}\} \tag{5}$$

where $\mathcal{J}^\pi_\beta$ and $\mathcal{J}^\pi_\mathcal{H}$, are the returns associated with the baseline objective $\hat{\mathcal{R}}_\beta$ and human preferences $\hat{\mathcal{R}}_\mathcal{H}$ respectively. Thus, the user may define several objective preferences, such that $\boldsymbol{\Omega} = \{\Omega_0, \ldots, \Omega_i\}, i \in \mathbb{N}_+$, which are used to generate a set of optimal policies (see Fig. 2) $\Pi^* = \{\pi^*_0, ..., \pi^*_i\}, i \in \mathbb{N}_+, \Pi^* \subset \mathcal{F}^*$. To reduce sample complexity, we decided to bootstrap each policy π from the optimal baseline policy π^*_β. The final parameterized weights are expected to be similar, since they portray a strong baseline of performance. In Section V we empirically motivate this decision to be worthwhile as opposed to using preference learning from *tabula rasa*. An additional challenge deals with both $\hat{\mathcal{R}}_\beta$ and $\hat{\mathcal{R}}_\mathcal{H}$ are intrinsically non-stationary, since they are updated with novel queries to adapt to the current state-space subset of the policy. To handle this, we rely on PPO's ability to robustly deal with the change in reward estimation [8]. At each training step we use the collected trajectories to compute advantages as used in eq. 1, which are generated by a linear combination of objectives.

V. Experimental Results

In this section we evaluate our approach on several robotic tasks. We define the policies trained on a specific objective preference Ω as a preference policy π^*_Ω, initially bootstrapped from π^*_β (see Section. IV-B) . The main questions we aim at addressing are:

- Question 1 (Q1): Is the framework able to produce a set of stable policies $\Pi^*_\Omega \subset \mathcal{F}^*$ eq.5, which combine both objectives?
- Question 2 (Q2): Are the obtained preference policies Π^*_Ω compliant to human objective preferences Ω?
- Question 3 (Q3): How many human queries are required by bootstrapping from π^*_β without compromising performance?
- Question 4 (Q4): Is our approach applicable with real human feedback instead of using synthetic heuristics?

To address Q1, we train a baseline policy π^*_β on 1M steps, and each preference policy π_Ω on an additional 500k steps bootstrapped from π^*_β. We use environmental metrics (see Sec. V-B) to evaluate if the newly obtained policies follow the desired human behaviours. Q2 is answered by comparing metrics between a set of policies Π with different preferences. We address Q3 by comparing how many queries are saved with our approach versus when training policies from scratch with human feedback. In Fig. 3 and Tab. I, we show the results both in terms of metrics: to confirm human preferences were included in the baseline policy; and learning curves: to compare the effect of those preferences in the original policy. For the last question Q4, we performed a user study on the social navigation environment (see Fig. 5). We collect real feedback and test the applicability of our approach in Sec. V-C.

A. Implementation Details

We use the default reward for the social navigation environment (see Sec. V-B) and to test for a hypothetical absence of a reward function (replaced by heuristics), we use estimations of $\hat{\mathcal{R}}_\beta$ in Lunar Landing and Hopper to obtain an optimal policy π^*_β. To simulate human feedback, we use oracle reward functions $\mathcal{R}_\mathcal{H}$ when answering preference queries (see Sec. III). We define them as $\mathcal{R}_\mathcal{H} = \mathcal{R}_\beta + \mathcal{R}_p$, where $\mathcal{R}_p$ represents a subjective preferred behaviour on top of a general sense of performance $\mathcal{R}_\beta$ (e.g., for Lunar Landing, the general objective is to land the spaceship safely and, the subjective behaviour is to land from the right). To account for noisy feedback, we introduce a 10% error rate when answering queries to estimate $\hat{\mathcal{R}}_\mathcal{H}$. To train the different policies and reward estimators $\hat{\mathcal{R}}$ (see Sec. III), we used densely connected hidden layers of size {128,128} (for the actor and critic) and {256,256,256}, respectively. Both models have a hyperbolic tangent layer at the output.

B. Domains and Evaluation Metrics.

Hopper In Hopper, a human oracle is designed to prefer high jumps. In this case, we chose as metric the average height across time-steps the agent is able to achieve. In Fig. 3, we can observe a large impact on learning between different objective combinations. The impact of each objective preference is also verified in the metrics across time-steps in Fig 3, where the increase of the average height was in the range between 13.4% and 19.5%. Since there is no significant degradation in performance of the environmental objective $\mathcal{R}_\beta$ and we see noticeable improvements in the preference

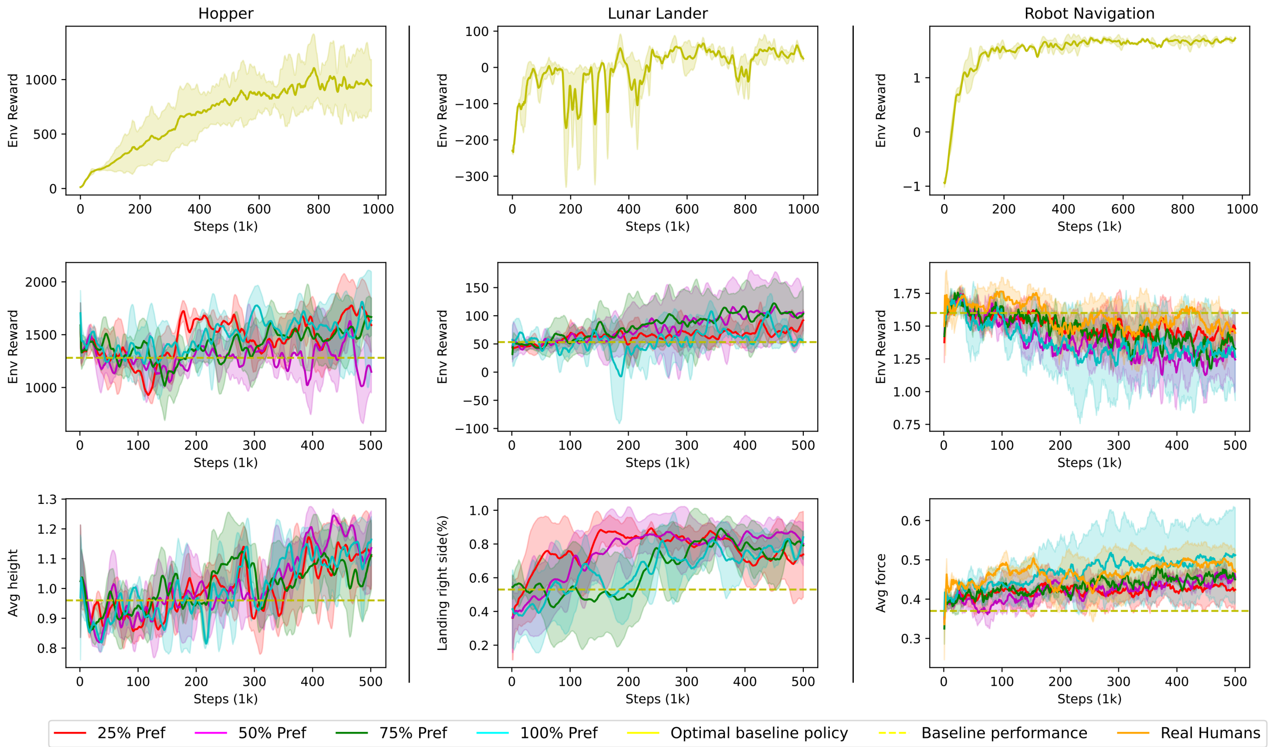

Fig. 3: Experiments for the different environments. Top row: Mean learning curve of the environmental expert policy using $\mathcal{R}_\beta$; Middle row: Represents the learning curves obtained with different objective preferences bootstrapped from π^*_β; Bottom row: Change in the metrics across training steps for the different obtained policies.

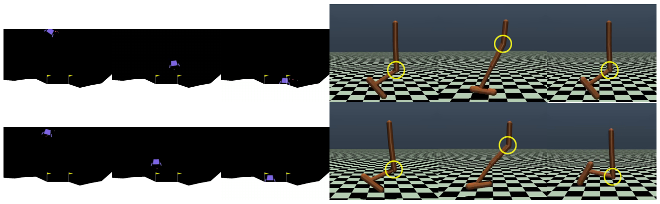

Fig. 4: Sequence of two rollouts on both Lunar Landing and Hopper for the same seed. Bottom sequence represents a rollout from an expert policy trained exclusively on an estimation of environmental reward $\hat{\mathcal{R}}_\beta$. Top sequence represents an optimal preference policy with an objective preference of $\Omega_\mathcal{H} = 1$.

metric, we find support for Q1. The behaviour is illustrated in Fig. 4 with a joint highlighted in circles; in order for the agent to reach higher average heights, it chooses not to bend one of the joints, remaining as straight as possible while jumping forward. Our new policies π^*_Ω adhere to human preferences (see Tab. I), confirming Q2. We then collect 20 queries every 20K step following our new policy up until we reach a total of 300 queries. When training π_β from scratch, 1400 queries are required, which means we save around 1100 queries per learned policy, showing support for Q3.

Lunar Landing In Lunar Landing, metrics are presented as a percentage which is defined by the ratio $r_p = \frac{\sum \Delta T_p}{\Delta T}$, where $\sum \Delta T_p$ represents the sum of all time-steps when the ship is approaching landing from the right, and ΔT the total environment steps of the policy π^*_Ω trained under a certain preference Ω_i. In Fig 3 we see that the return of the environmental reward $\mathcal{R}_\beta$ shows a slight degradation when preferences are included, with the worst case when $\Omega_\mathcal{H} = 1$ (100% Pref). This can be explained by the need for the agent to heavily use the left thruster in order to approach landing from the right. For the metric r_p the improvements range between 19.2% and 54.1% where the lowest range stems from $\Omega_\mathcal{H} = 0.25$ and highest range from $\Omega_\mathcal{H} = 1.0$ when comparing to the baseline $\Omega_\beta = 1$ Tab. I. In Fig. 4 we can see an example of the difference when following π^*_β and π^*_Ω with $\Omega_\mathcal{H} = 1$. These results show that we can learn a stable set Π^*_Ω and Fig 3 shows that we do obtain policies that comply with the different objectives $\boldsymbol{\Omega}$ which supports Q1 and Q2. When collecting queries during the learning of a policy π_Ω, we first collect 100 queries following π^*_β to obtain an initial estimation of $\hat{\mathcal{R}}_\mathcal{H}$. We collect 100 initial queries and then train continuously until we reach a total of 300 queries. We saved the same amount of queries per policy as in Hopper, supporting Q3.

Social Navigation To test our hypothetical example given in Section I we designed a social navigation environment in Unity [42], similar to the one introduced in [43]. The main purpose of the environment is for an agent (mobile robot) to reach two navigation goals, while avoiding collisions. As for the dynamics of the environment, we implemented the Social Force Model (SFM) [13], [44], where humans are represented as directed force fields which influence the agent's trajectory. The action space $a_t \in \mathbb{R}^3$ of the agent is represented by the intended acceleration in both directions $a_1 = a_x$, $a_2 = a_y$, and a_3 is a scalar value directly proportional to the magnitude of the SFM force field. The state space of the agent is comprised of its position and velocity, velocity of other humans, and a stacked array. The stacked array contains one-hot encoded ray information of whether it detects a goal, wall or human, and normalized distance to it.

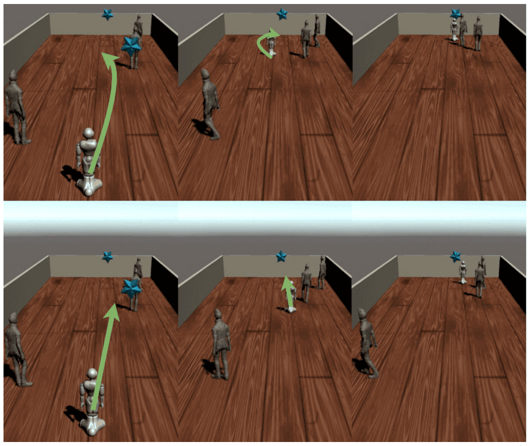

Fig. 5: Rollouts from π^*_β (bottom) and π^*_Ω, $\Omega_\mathcal{H} = 1$ (top), same seed. The robot takes into consideration human feedback which respects SFM.

TABLE I: Metrics, for each environment and each policy obtained with different preference objectives $\mathbf{\Omega}$: baseline: $\Omega = (1, 0)$; 25% HR.: $\Omega = (0.75, 0.25)$; 50% HR.: $\Omega = (0.5, 0.5)$; 75% HR.: $\Omega = (0.25, 0.75)$; 100% HR.: $\Omega = (0, 1)$;

Metrics	**Baseline**	25% HR.	50% HR.	75% HR.	100% HR.
Lunar	54.6%	73.8%	80.0%	79.1%	81.6%
Hopper	0.97m	1.1m	1.13m	1.11m	1.16m
Social Nav.	0.37	0.42	0.45	0.45	0.51

In this experiment we simulate human feedback as preferring trajectories which follow SFM, when possible. As metric we chose the normalized intensity of the force field due to SFM. In Fig. 3 and Tab. I we are able to observe that all policies Π^*_Ω consider human preferences and score higher metric-wise. A trade-off is verified between increasing the force field magnitude and some deterioration for the baseline reward $\mathcal{R}_\beta$. This is due to the agent following a less greedy policy by considering other human force fields, taking more time to reach navigation goals. We get an improved metric performance, ranging from an increase of 13.5% to 37.8% in SFM force field activation for $\Omega_\mathcal{H} = 0.25$ and $\Omega_\mathcal{H} = 1$ respectively. The change in behaviour is illustrated in Fig. 5, supporting both Q1 and Q2. For the results obtained in Tab. I, we collected a total of 1100 queries but tested with as low as 400 queries and still obtained satisfying results.

C. Applicability with Human feedback

To attest for the applicability of our approach and answer Q4 (see Sec. V), we have conducted a user study on the social navigation environment, to obtain real human feedback as opposed to using heuristics. To motivate our claim that human feedback is hard to model, the user study also aims at addressing an additional question: (Q5) Is real human feedback significantly different from an heuristic we thought appropriate? To train a policy with real human preferences, we start from the same baseline policy π^*_β as used in the simulation results, and set an objective preference of 100%, $\Omega_\mathcal{H} = 1$. Similarly to simulation, we collect trajectories from π^*_β to create queries. We divide preference learning in 4 loops. In each loop, we sample 200 random trajectory segments from a very large dataset of rollouts produced by π^*_β. We render these trajectory segments into 200 videos of length between 2 and 3 seconds, to form 100 queries. After collecting query preferences, we train a new reward estimator $\hat{\mathcal{R}}_\mathcal{H}$ and train a new preference policy π_Ω which concludes one loop. We repeat this process 4 times until we reach 400 queries. The study was performed on Amazon Mechanical Turk. There were 20 (5 per loop) unique participants (12 males, 8 females and none of other gender identities), each providing feedback for 20 queries. The age of participants ranged from 24 to 62 years old, with a median of 34. All participants were from the US and most completed college education (N=16). The majority of participants have seen a real life robot (N=12), and (N=6) have regular contact. In Fig. 3 and Tab. II we are able to observe that real human feedback did not substantially deteriorate the main baseline goal when compared to the considered 100% synthetic heuristic. We observe in Tab. II that the policy trained using real human feedback is better aligned with our chosen metric compared to the baseline policy. However, there may be many other metrics that humans consider important. To answer Q5, we compare real human feedback to the synthetic oracle directly on the same videos. From the 400 queries, in 273 (68.25%) humans agree with the synthetic oracle, while they disagree on 127. One of the major differences observed between real and synthetic feedback, is that while an heuristic has complete access to a reward function at each time step and can always decide between any 2 videos, humans were indecisive on 34 (8.5%). We performed a one-way Z-test for proportions on the null hypothesis of whether humans agree with the heuristic in 90% of the time (accounting for 10% error rate). The results were statistically significant $Z = -9.345$, $p < 0.001$, showing humans had significant different perceptions on the appropriate robot's behaviour when compared to our heuristic, showing support for Q5 and motivating our approach which promotes collecting real human feedback to model human's expectations. Videos of rollouts from the resulting policy (and other policies) can be found in the media attachment of this paper.

TABLE II: Comparison of metrics and avg. reward with a baseline policy, and a policy trained on a reward estimator with heuristics and real human preferences, on an objective preference of 100% HR.: $\Omega = (0, 1)$;

Social Nav.	Baseline	100% HR.	Real
Rwd.	1.6	1.26	1.39
Metric	0.37	0.51	0.47

VI. Conclusions

In this paper, we presented a multi-objective setting to combine subjective preferences with baseline objectives. Our results show our approach is able to produce a set of policies that adhere to preferences as shown by metrics, without substantially compromising the initial objective in a trade-off dynamic. By bootstrapping from a baseline policy, we were also able to reduce the necessary number of queries in about ~78%. We conducted a user study to collect human feedback in order to train a new policy. Our findings indicate that the policy trained using real feedback deviates from both the baseline and the heuristic we considered. This demonstrates that it is challenging to predict and model human feedback in advance, and it necessitates real human input.

References

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[2] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6284–6291.

[3] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5129–5136.

[4] B. Yang, G. Habibi, P. Lancaster, B. Boots, and J. Smith, "Motivating physical activity via competitive human-robot interaction," in *Conference on Robot Learning*. PMLR, 2022, pp. 839–849.

[5] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[6] J. Whittlestone, K. Arulkumaran, and M. Crosby, "The societal implications of deep reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1003–1030, 2021.

[7] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman, "Interactive learning from policy-dependent human feedback," in *Int. Conf. on Machine Learning*. PMLR, 2017, pp. 2285–2294.

[8] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[9] S. H. Huang, I. Huang, R. Pandya, and A. D. Dragan, "Nonverbal robot feedback for human teachers," *arXiv preprint arXiv:1911.02320*, 2019.

[10] D. Dewey, "Reinforcement learning and the reward engineering principle," in *2014 AAAI Spring Symposium Series*, 2014.

[11] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, pp. 1–59, 2022.

[12] P. Vamplew, R. Dazeley, C. Foale, S. Firmin, and J. Mummery, "Human-aligned artificial intelligence is a multiobjective problem," *Ethics and Information Technology*, vol. 20, no. 1, pp. 27–40, 2018.

[13] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282–4286, 1995.

[14] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[15] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Int. Conf. on Machine Learning*, 2000, pp. 663–670.

[16] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Int. Conf. on Machine learning*, 2004, p. 1.

[17] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI*, 2008, pp. 1433–1438.

[18] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions." in *Robotics: Science and Systems*, 2017.

[19] E. Ratner, D. Hadfield-Menell, and A. Dragan, "Simplifying reward design through divide-and-conquer," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[20] K. Xu, E. Ratner, A. Dragan, S. Levine, and C. Finn, "Learning a prior over intent via meta-inverse reinforcement learning," in *Int. Conf. on Machine Learning*, 2019, pp. 6952–6962.

[21] S. Reddy, A. Dragan, S. Levine, S. Legg, and J. Leike, "Learning human objectives by evaluating hypothetical behavior," in *Int. Conf. on Machine Learning*, 2020, pp. 8020–8029.

[22] A. Ghadirzadeh, P. Poklukar, V. Kyrki, D. Kragic, and M. Björkman, "Data-efficient visuomotor policy training using reinforcement learning and generative models," 2020.

[23] D. Brown, S. Niekum, and M. Petrik, "Bayesian robust optimization for imitation learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2479–2491, 2020.

[24] Z. Javed, D. S. Brown, S. Sharma, J. Zhu, A. Balakrishna, M. Petrik, A. Dragan, and K. Goldberg, "Policy gradient bayesian robust optimization for imitation learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4785–4796.

[25] X. Wang, K. Lee, K. Hakhamaneshi, P. Abbeel, and M. Laskin, "Skill preferences: Learning to extract and execute robotic skills from human feedback," in *Conference on Robot Learning*. PMLR, 2022, pp. 1259–1268.

[26] Y. Cui, Q. Zhang, A. Allievi, P. Stone, S. Niekum, and W. B. Knox, "The empathic framework for task learning from implicit human feedback," *arXiv preprint arXiv:2009.13649*, 2020.

[27] C. Wang, C. Pérez-D'Arpino, D. Xu, L. Fei-Fei, K. Liu, and S. Savarese, "Co-GAIL: Learning diverse strategies for human-robot collaboration," in *Conference on Robot Learning*. PMLR, 2022, pp. 1279–1290.

[28] M. Murray, N. Walker, A. Nanavati, P. Alves-Oliveira, N. Filippov, A. Sauppé, B. Mutlu, and M. Cakmak, "Learning backchanneling behaviors for a social robot via data augmentation from human-human conversations," in *Conference on Robot Learning*. PMLR, 2022, pp. 513–525.

[29] I. Kapelyukh and E. Johns, "My house, my rules: Learning tidying preferences with graph neural networks," in *Proceedings of the 5th Conference on Robot Learning*, 2022, pp. 740–749.

[30] K. Amin, N. Jiang, and S. Singh, "Repeated inverse reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[31] G. Chou, D. Berenson, and N. Ozay, "Learning constraints from demonstrations," *arXiv preprint arXiv:1812.07084*, 2018.

[32] G. Chou, N. Ozay, and D. Berenson, "Learning parametric constraints in high dimensions from demonstrations," in *Conference on Robot Learning*. PMLR, 2020, pp. 1211–1230.

[33] C. Wirth, R. Akrour, G. Neumann, J. Fürnkranz *et al.*, "A survey of preference-based reinforcement learning methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

[34] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari," *Advances in neural information processing systems*, vol. 31, 2018.

[35] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," *arXiv preprint arXiv:1906.08928*, 2019.

[36] K. Lee, L. Smith, and P. Abbeel, "PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training," *arXiv preprint arXiv:2106.05091*, 2021.

[37] D. R. Scobee and S. S. Sastry, "Maximum likelihood constraint inference for inverse reinforcement learning," *arXiv preprint arXiv:1909.05477*, 2019.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[39] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Int. Conf. on machine learning*. PMLR, 2015, pp. 1889–1897.

[40] A. Abdolmaleki, S. Huang, L. Hasenclever, M. Neunert, F. Song, M. Zambelli, M. Martins, N. Heess, R. Hadsell, and M. Riedmiller, "A distributional view on multi-objective policy optimization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11–22.

[41] A. Abdolmaleki, S. H. Huang, G. Vezzani, B. Shahriari, J. T. Springenberg, S. Mishra, D. TB, A. Byravan, K. Bousmalis, A. Gyorgy *et al.*, "On multi-objective policy optimization as a tool for reinforcement learning," *arXiv preprint arXiv:2106.08199*, 2021.

[42] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," 2020.

[43] D. Marta, C. Pek, G. I. Melsión, J. Tumova, and I. Leite, "Human-feedback shield synthesis for perceived safety in deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 406–413, 2021.

[44] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proc. of the Royal Society B: Biological Sciences*, vol. 276, no. 1668, p. 2755–2762, 2009.