Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Learning Safe Behaviour via Justified Human Preferences and Hypothetical Queries

Ilias Kazantzidis (ik3n19@soton.ac.uk)
University of Southampton
Timothy J. Norman (t.j.norman@soton.ac.uk)
University of Southampton
Yali Du (yali.du@kcl.ac.uk)
King's College London
Christopher T. Freeman (ctf1@soton.ac.uk)
University of Southampton

Research Article

Keywords:

DOI: https://doi.org/

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Learning Safe Behaviour via Justified Human Preferences and Hypothetical Queries

Ilias Kazantzidis[1*], Timothy J. Norman[1], Yali Du[2] and Christopher T. Freeman[1]

[1]School of Electronics and Computer Science, University of Southampton, Building 32, University Road, Southampton, SO17 1BJ, United Kingdom.
[2]Department of Informatics, King's College London, Bush House, 30 Aldwych, London, WC2R 2LS, United Kingdom.

*Corresponding author(s). E-mail(s): ik3n19@soton.ac.uk;
Contributing authors: t.j.norman@soton.ac.uk; yali.du@kcl.ac.uk; ctf1@soton.ac.uk;

Abstract

Although reinforcement learning is a powerful paradigm for agent sequential decision-making, it cannot be used in its traditional form in most safety-critical environments. Human feedback can enable an agent to learn a good policy while avoiding unsafe states, but at the cost of human time. We present JPAL-HA, a model for safe learning in safety-critical environments that is grounded on two novel ideas: (i) human preferences over a choice of actions are augmented with *justifications* such as one action is preferred because the other is unsafe; and (ii) we use these justifications to guide the generation of future queries over *hypothetical actions* to enable the agent to more effectively map out unsafe regions. Our algorithm comes with experimental safety and performance guarantees during *training*, and so offers a practical tool for users to train their agent safely and according to the level of safety their application requires. We evaluate it using a modified version of the Island Navigation environment from the AI Safety Gridworlds. We show that it significantly improves safety compared to existing human-in-the-loop safe reinforcement learning approaches and outperforms state-of-the-art learning algorithms in terms of sample efficiency. Further, we evaluate

our method with people training a mobile robot, demonstrating that a diverse group of people can complete training in only a few minutes and that *hypothetical actions* not only improve safety, but also slightly decrease the training time and keep it to admissible levels.*

Keywords: Safe Reinforcement Learning, Learning from Human Preferences, Human-Agent Collaboration, Human-Robot Interaction

1 Introduction

Inexperienced interaction with the real world has its dangers. Fortunately, in virtual environments an agent can simply 'restart' and no harm will be caused. This is the reason why traditional Reinforcement Learning (RL) has mastered some of the most difficult board [1] and video games [2, 3], where we do not care about continuous blundering of Queens in chess or deaths of virtual creatures. In real-world applications, such as autonomous vehicles and robotics, however, we do mind if a self-driving car (agent) crashes even once into an obstacle or even worse a pedestrian (environment). Such examples motivate the development of Safe RL, where in addition to the pursuit of the optimal policy, safety constraints have to be respected [4]. The main problem Safe RL tries to address is that the agent cannot naïvely take explorative actions in order to gain new information. This Safe Exploration problem, combined with the problem of catastrophic forgetting, where the agent gradually forgets things it previously learnt, makes Safe RL a very difficult task with, at present, no general solution.

Numerous techniques have been proposed to address this problem, such as those augmenting the objective function with constraints in the single-agent domain [5–8] or those that incorporate norms in the multi-agent domain [9]. Our solution lies in the context of techniques that involve a human in the learning process who teaches the agent using Human Preference queries. Our proposed algorithm, called ***J**ustified Human **P**references for **A**ctive **L**earning with **H**ypothetical **A**ctions* (***JPAL-HA***), is close to Frye and Feige's [10] *Parenting* algorithm which also uses the technique of Human Preferences (the human providing feedback to the agent by giving a preference for one state-action pair) and learns directly in a supervised way from the preferences. We augment Human Preferences with two novel ideas:

1. *Justifications*: a binary warning signal expressed by a human as a choice between two actions, such as one action is preferred because the other is unsafe. Justifications are simple 'Yes' or 'No' answers and require marginally more time for a human than asking for a preference. However, the impact on safety is significant.

*The code is available at https://github.com/ilkaza/JPAL-HA.

2. *Hypothetical Actions* (*HAs*): we use justifications to guide the generation of future queries regarding hypothetical actions that could have been taken, with the goal of enabling the agent to more effectively map out unsafe regions.

As discussed in previous research [11], in JPAL-HA, in contrast to traditional RL, the 'reinforcement' is not the traditional environmental reward. First, in many real-world tasks, such as robotics, the goals are complicated and such rewards are hard or even impossible to design. Moreover, even if they exist, they may be sparsely populated or delayed. Additionally, traditional RL is, in general, very sample-inefficient (sometimes needing millions of steps), which makes training impossible in the real world where every step may not take just milliseconds as in simulations. Further, even if there is a well-defined reward function to be optimised, the final agent behaviour will not align with what the human really wants the agent to do (constituting an *agent alignment* or *reward specification* problem [12]). Finally, and most importantly, due to safe exploration, environmental rewards may also not be helpful in real-world safety-critical domains such as robotics; e.g. informing an agent-robot that 'you got -100 points for destroying your gripper, the assembly and harming a person', is obviously not desirable.

In JPAL-HA, the reinforcement comes from human feedback before an action is taken, and is not in the form of a numerical reward value; instead it takes the form of preference and justification answers. Moreover, we do not use these preferences to build a reward model but to learn directly from an increasing dataset of human preferences with supervised learning. These choices are discussed in more detail in Section 6.1 with comparison to related work.

Although obtaining human guidance improves the performance and safety of the algorithm, it introduces the issue of human burden. Overall, we assess the general efficiency of an agent learning approach by three factors: (i) performance, i.e. does the algorithm converge to an optimal or near-optimal policy; (ii) sample (or data) efficiency, i.e. how much time does training take; and (iii) safety, i.e. will the agent avoid the unsafe states until it has learnt the optimal policy. Performance is evaluated by the return, and safety by the number of catastrophic actions. Sample efficiency would normally be measured by the number of overall interactions (e.g. steps) until the optimal policy is found. However, when a human is employed, what takes additional time is the response time of the human. Hence, for sample efficiency we also consider the human burden; i.e. how many questions are asked to the human (and how hard they are) until optimality is achieved.

The contributions of this work are:

- We present in detail two new ideas that can be generalised and incorporated in models that use Human Preferences. First, learning with *Justifications*, which does not depend on the general framework of JPAL-HA or a specific environment. Its benefit is reflected by significantly improving safety.

Second, *HAs* follows Justifications, and further improves safety. It also significantly compensates for human burden, assuming that Justifications and HAs place a marginal additional cost.

- We introduce a novel algorithm called JPAL-HA, a learning from Human Preferences approach, which incorporates both of the aforementioned features of Justifications and HAs. JPAL-HA benefits from experimental safety and performance guarantees *during training*, and as such constitutes a practical tool for users to train their agent safely and also according to the level of safety their application requires. The user can tune the level of safety they want based on a hyperparameter that trades safety with human burden.
- We evaluate our algorithm not only in simulation environments but also in a real-world mobile-robot task, achieving successful learning in only a few minutes by a diverse group of people. We also demonstrate comparisons made between different methods in this real-world task.

The *Justifications* idea is introduced on the *JPAL* algorithm (Section 4.1), which builds on the Parenting algorithm [10] (Section 3); *Hypothetical Actions* is introduced on *JPAL-HA* (Section 4.2), which builds on JPAL. Before presenting these models, we provide an overview of the safe exploration problem and how we model it in this paper.

2 The Safe Exploration Problem

Safe Exploration [4] is the key challenge for Safe RL. One of the fundamental concepts of reinforcement learning is balancing exploitation with exploration during learning. Unrestricted exploration can, however, lead to unsafe states, or even catastrophic failures in the environment due to the agent's action.

Leike et al. [13] represent this problem with a mini-gridworld environment, which, despite its small state-space, captures the semantics of safety well. The environment, called Island Navigation, was adapted by Frye and Feige [10] to the more challenging version illustrated in Figure 1. We further created an arena based on this simulated environment for training a mobile robot using real human input, presented in Figure 8. The agent A has four moves: UP, DOWN, LEFT and RIGHT, is initialised in the red cell and tries to reach the goal cell (gold). For every step it takes, it receives a -1 reward from the environment and if it reaches the goal state it receives a reward of +50. There are cells of free road (grey). If the agent 'hits' a wall cell (green) it does not move, but receives -1 reward. The agent cannot swim and 'dies' when entering a water cell (blue), and the episode ends with an added reward of -50. Our goal is to avoid as many 'deaths' as possible (ideally avoiding any death) until the agent has learnt the optimal policy and is ready for deployment. Safe exploration is, therefore, defined by minimising the number of deaths during training.

We solve this problem in the following setting. First, the unsafe states (water) are unknown to the agent and we have access to neither the state transition dynamics (i.e. there is no available simulator to sample trajectories[1]), nor to the hand-crafted reward function described above[2]. This is the general case of a challenging real-world environment. The only source of information is the human trainer (user), who must have a reasonable understanding of the environment (can correctly answer all preference and justification queries — simple for the Island Navigation environment, as we will see later) and recognise unsafe states. With the latter assumption, a lot of important safety-critical applications can be tackled. However, for very complex applications, such as helicopter flight where there is a restricted number of human trainers who know the correct actions, more advanced methods are yet to be developed. Finally, querying the user is expensive.

Specifically, for the Island Navigation environment we also assume that the agent can sense (observe) its exact location (coordinates x, y) and the whole state of the board. In other settings, we could assume that the agent (e.g. a robot) uses an attached camera to obtain the observations [14]. Finally, here we do not consider any debatable (normative) occasions, i.e. different opinions regarding the correct human input. This could occur in some cases and needs extra modelling [15]; however, it goes beyond the scope of this paper.

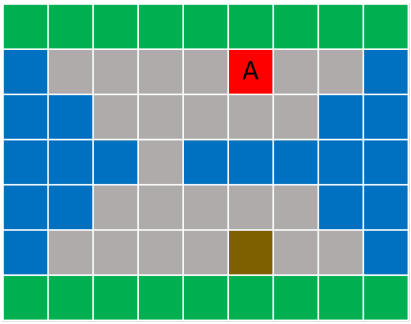


Fig. 1 The Island Navigation Environment. **A**: Agent, **Red**: Initialisation cell, **Gold**: Goal, **Blue**: Water, **Green**: Wall, **Grey**: Free road

3 The Parenting approach

Our newly-proposed methods, JPAL and JPAL-HA, build upon the parent-child (human-agent) intuition of Feige's *Parenting* model [10]. The child (untrained agent) is unfamiliar with the dangers of most states and asks the parent (the human) a lot of questions. The more familiar the child becomes

[1] Even though we simulate experiments to draw conclusions, we assume they should run in the real world, with a real human and robot in order to avoid the reality gap as done in Section 5.9.

[2] We only use this reward function for evaluation in order to compare with methods that use it.

with a state, the fewer queries it needs to ask about it. As the child develops experience, it may take actions independently without asking the parent when it is confident about the safety of its action. The child asks for the parent's preference between two actions that they think are the best amongst all possible actions.

Algorithm 1 presents Parenting more formally[3]. The goal is to train a policy model $\pi(a \mid s)$, i.e. the probability of choosing action a from state s, from which the agent then picks the greedy action and acts on the environment during deployment. This model is the average of: (i) a Convolutional Neural Network (CNN) with input being the state of the board; and (ii) a Feed-Forward Neural Network (FNN) with input being the type of the four neighbouring cells (in Up, Down, Left and Right directions). It is trained by supervised learning from human preferences stored in an *Embraced Memory* $\mathcal{X}$ (essentially $\pi(a \mid s)$ is a predictor of a human's preference). For our learning paradigm, the output has four neurons as the number of possible actions.

In each step of the algorithm and until the optimal policy has been found (Lines 6–16) the agent from state s_t, asks a *Before-The-Fact-Query (BTFQ)*, i.e. a query issued to the human before taking an action, with probability $(p_{BTFQ})^{f(s_t)}$, where $f(s_t)$ (*familiarity*) is the number of queries that have been issued from s_t and $p_{BTFQ} \in [0, 1)$ is a hyperparameter normally set close to 1, to ensure the agent is cautious in new states (Lines 7–8). Note that $(p_{BTFQ})^{f(s_t)}$ gradually decreases as the agent asks more queries from s_t, reducing human burden.

BTFQs are constructed by the agent sampling two distinct actions $a_t^{(0)}$ (1^{st} sampled action) and $a_t^{(1)}$ (2^{nd} sampled action) from $\pi(a \mid s)$ and querying the human. The human answers with a preference $P \in \{1^{\text{st}}, 2^{\text{nd}}, either, neither\}$, where 1^{st} and 2^{nd} means preference for the first or the second sampled action, respectively; *either* means that both actions are equivalently preferable and safe (not necessarily the best ones though); and *neither* means that both actions are catastrophic (Line 2). The agent from P can obviously find the recommended action a^* which will be executed (Lines 3, 5). If $P = either$, a random action between the two is executed; if $P = neither$, resampling takes place from $\pi(a \mid s)$ until a non-catastrophic one is identified. Moreover, the human's input P drives the agent's learning. In order to do that, the ground truth label μ is derived, with $\mu^{(0)} = \mu$ denoting how much the first action is preferred to the second, and $\mu^{(1)} = 1 - \mu$ denoting the opposite. The definition is $\mu \in \{0, 0.5, 1\}$, where 0 and 1 represent the preference for the 1^{st} or 2^{nd} action, respectively, and 0.5 means no preference (Line 3). Then, the entry:

[3]We change the terminology and make certain technical improvements to Parenting [10], described in Sections 3, 4.3 and 5.1; these optimisations are, however, irrelevant to the core contributions of Justifications and HAs. When we refer to Parenting from now on, we will mean this optimised version.

$(s_t, a_t^{(0)}, a_t^{(1)}, \mu, a^*)$ is stored in $\mathcal{X}$, from which the agent will learn at the end of the step (Line 4)[4].

If a BTFQ is not asked (meaning the agent 'knows the state well', and so can take a safe action with high probability) (Lines 9–14), then Parenting continues in the same manner as JPAL and JPAL-HA at this point. It potentially *records and asks an After-The-Fact-Query (ATFQ)* (Lines 10–13)[5] and it executes the greedy action from $\pi(a \mid s)$ (Line 14). A step finishes by learning via supervised learning from the human preferences stored in $\mathcal{X}$ (Line 15). Before giving the details of Recording/Asking ATFQs and Learning (Section 4.3), we introduce the novel features of JPAL and JPAL-HA.

Algorithm 1 Parenting (optimised)

Input: $p_{BTFQ} \in [0, 1)$: hyperparameter close to 1, p_{REC}: probability of recording, p_{ATFQ}: probability of asking an ATFQ
Output: $\pi(a \mid s)$: agent's policy model

1: **function** ask_BTFQ(s_t)
2: sample $a_t^{(0)}$ and $a_t^{(1)}$ and receive P from human
3: find a^* and μ from P (if both actions catastrophic, resample)
4: add entry $(s_t, a_t^{(0)}, a_t^{(1)}, \mu, a^*)$ to $\mathcal{X}$
5: execute a^*

6: **repeat**
7: **if** $(p_{BTFQ})^{f(s_t)} > r \overset{\text{iid}}{\sim} U[0, 1]$ **then**
8: ask_BTFQ(s_t)
9: **else**
10: **if** $p_{REC} > r \overset{\text{iid}}{\sim} U[0, 1]$ **then**
11: record a <*greedy, random*> action pair from s_t in $\mathcal{R}$
12: **if** $p_{ATFQ} > r \overset{\text{iid}}{\sim} U[0, 1]$ **then**
13: ask an ATFQ from $\mathcal{R}$ and add entry to $\mathcal{X}$
14: execute greedy action drawn from $\pi(a \mid s)$
15: train policy $\pi(a \mid s)$ with gradient descent minimising Equation 1
16: **until** optimal policy is found

[4] a^* is stored simply to avoid bothering the human with the same queries and also avoid multiple copies in $\mathcal{X}$. For an identical previously-asked query, the agent will be able to retrieve the answer from $\mathcal{X}$. This mechanism was not mentioned to be implemented in [10].

[5] This mechanism, although inspired by [10], has been significantly modified and is described in detail in Section 4.3.

4 Learning with Justifications and Hypothetical Actions

The ideas that we add to the parent-child allegory of Section 3 are: (i) *Justifications*: the parent not only gives advice about which action is better but also warnings (a kind of justification for the expressed preference[6]) regarding safety. That is, if one of the actions proposed is catastrophic, it is valuable for the child to know. Providing this guidance is a small additional effort for the parent, but helps to promote safer behaviour; (ii) *Hypothetical Actions*: if a justification is 'warning', which means that the child asked a preference query that included at least one catastrophic action, that predisposes the child to think that more actions could lead to a catastrophe from that state ('where there's smoke, there's fire'). So the child may want to ask further questions regarding *hypothetical actions* that could have been taken from that state, and as such map out potentially unsafe regions of the environment faster. This is beneficial not only for the child's safety, but also for the parent's effort that may be more focused on dangerous regions. Moreover, the child, when faced with a state of multiple unsafe behaviours, creates and stores 'deadly patterns' and is always suspicious about them, in order both to avoid them in another state and not to bother the parent again. A deadly pattern could be, for example, the combination of moves (Left, Right) when crossing a bridge. Both actions would lead to the cliff and the child should be aware of it as soon as possible, both for its safety and for not continuously bothering the parent.

4.1 JPAL

More formally, in JPAL (Algorithm 2), $P \in \{1^{\text{st}}, 2^{\text{nd}}, equal\}$, where 1^{st} and 2^{nd} means preference for the first or the second sampled action, respectively; and *equal* means no preference (either because they are equivalently preferable and safe, or because they are both catastrophic). However, in addition to a preference P, the agent requests and gets a justification over this preference $J_P \in \{w, n\}$, where w means 'warning', i.e. at least one proposed action is catastrophic (e.g. leads into water), and n means 'no warning', i.e. both actions are safe (Line 2).

The agent from P can firstly again find a^* (Line 3). However, in JPAL (and JPAL-HA), we can make a more efficient mapping $P \times J_P \rightarrow \mu$, where $\mu \in \{0, 0.25, 0.5, 0.75, 1\}$ (Line 3), that directly improves the safety of the agent. We use $\mu = 1$ to denote total preference for the 1^{st} action implying a catastrophe for the 2^{nd} one (i.e. $P = 1^{\text{st}}, J_P = w$), $\mu = 0.75$ to denote simple preference for the 1^{st} action (i.e. $P = 1^{\text{st}}, J = n$), $\mu = 0.5$ to denote equal preference (i.e. $P = equal, J_P : arbitrary$) and $\mu = 0$ and 0.25 similarly to indicate preference for the 2^{nd} action. In the case that both actions are

[6] The general idea of using the term *justifications* is for the human to provide a rationale for the expressed preference. In this context we only consider warnings, but there may be others such as use of limited resources, etc. A concrete example is discussed in Section 6.3. Hence, we use the more general term justifications.

catastrophic (i.e. $P = equal$, $J_P = w$), we resample until a non-catastrophic action is identified. Having mapped P and J_P to μ, we store in $\mathcal{X}$ the entry: $(s_t, a_t^{(0)}, a_t^{(1)}, \mu, J_P, a^*)$ (Line 4). Finally, a^* is executed. At this point, JPAL continues the same as Parenting and JPAL-HA (Section 4.3).

Algorithm 2 JPAL

Input: $p_{BTFQ} \in [0,1)$: hyperparameter close to 1, p_{REC}: probability of recording, p_{ATFQ}: probability of asking an ATFQ
Output: $\pi(a\,|\,s)$: agent's policy model

1: **function** ask_BTFQ(s_t)
2: sample $a_t^{(0)}$ and $a_t^{(1)}$ and receive P and J_P from human
3: find a^* and μ from P and J_P (if both actions catastrophic, resample)
4: add entry $(s_t, a_t^{(0)}, a_t^{(1)}, \mu, J_P, a^*)$ to $\mathcal{X}$
5: execute a^*

6: **repeat**
7: **if** $(p_{BTFQ})^{f(s_t)} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
8: ask_BTFQ(s_t)
9: **else**
10: **if** $p_{REC} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
11: record a <*greedy, random*> action pair from s_t in $\mathcal{R}$
12: **if** $p_{ATFQ} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
13: ask an ATFQ from $\mathcal{R}$ and add entry to $\mathcal{X}$
14: execute greedy action drawn from $\pi(a\,|\,s)$
15: train policy $\pi(a\,|\,s)$ with gradient descent minimising Equation 1
16: **until** optimal policy is found

4.2 JPAL-HA

As we have said, JPAL-HA (Algorithm 3) introduces the idea of Hypothetical Actions and builds on JPAL. The description in Section 4.1 holds for JPAL-HA, except for the case of $J_P = w$. Specifically, if $J_P = n$, then JPAL-HA works exactly like JPAL (Lines 18–21). However, if $J_P = w$ (at least one of $a_t^{(0)}$ or $a_t^{(1)}$ is catastrophic) the agent will *generate hypothetical actions* (Lines 22-23). Specifically, it creates a set of safe actions $\mathcal{S}$ and a set of catastrophic actions $\mathcal{C}$, where initially $a_t^{(0)}$ and $a_t^{(1)}$ are added based on whether they are safe or not; this can be determined automatically combining P and J_P (Lines 2–3). Then the following loop (Lines 4–13) is executed: a new action a_{new} (*hypothetical action*) is sampled and *only* a single justification over the new action $J_{a_{new}} \in \{w, n\}$ is asked from the human, where w means 'warning' (a_{new} is a catastrophe) and n means 'no warning' (a_{new} is safe) (Line 5). If

$J_{a_{new}} = w$, then $\mathcal{X}$ is extended to include all the combinations of a_{new} and the actions already in $\mathcal{S}$ and $\mathcal{C}$ — leaving a^* blank temporarily[7] (Line 7). *Crucially*, since we know that one action is already catastrophic, no extra human input is needed for these added entries to $\mathcal{X}$, i.e. μ and J_P of the new entries can be inferred *automatically*. Additionally, a_{new} is added in $\mathcal{C}$ (Line 8). If $J_{a_{new}} = n$, then $\mathcal{X}$ can again be *automatically* extended to include the combinations of a_{new} and the actions already in $\mathcal{C}$ — leaving a^* blank temporarily (Line 10). However, regarding a_{new} and the actions already in $\mathcal{S}$, an additional preference query P_{add} is asked between a_{new} and the best (mostly preferred) action in $\mathcal{S}$ until that loop iteration a_{best}, leading to another entry in $\mathcal{X}$ — again leaving a^* blank temporarily (Line 11). Potentially a_{best} can now be substituted with a_{new} in case a_{new} is preferred (Line 12). Finally, a_{new} is added to $\mathcal{S}$ since it is safe (Line 13).

One may use different conditions to stop the loop (Lines 4–13) based on the environment. For example, a simple rule we used was to stop when two safe actions are sampled overall (so that the region looks safe) or when all actions are sampled (Line 4). However, more sophisticated rules can be formulated for environments with higher action spaces, such as to stop when the size of $\mathcal{S}$ becomes double of the size of $\mathcal{C}$. When the loop finishes, the correct action a^* added to $\mathcal{X}$ is equal to a_{best} for *all* the combinations of actions that were involved during the loop[8] (Line 14) and a_{best} is also executed (Line 24). Overall, the aim of HAs is to (i) more quickly investigate a dangerous area with minimal user input and without interrupting them repeatedly for the same state; and (ii) potentially improve the executed action, and thus find the goal state quicker.

Additionally, the procedure of HAs is augmented with the mechanism of dynamically *detecting deadly patterns*. Specifically, when a combination of at least two catastrophic actions from the same state is detected (whether from the beginning of two catastrophic sampled actions or during the run of the loop), then a deadly pattern *dp* is created and stored in the *Danger Memory* $\mathcal{D}$. Memory $\mathcal{D}$ is usually small and contains patterns that will be checked both during training to avoid unnecessary queries to the human and during deployment as safety constraints. Essentially, $\mathcal{D}$ is a set of dynamically created constraints that have to be respected throughout training and deployment, such as 'do not go left or right when crossing a bridge' (Figure 5a).

4.3 Recording/Asking ATFQs and Learning

Occasionally, one may expect a child to ask slightly 'unconventional' random questions to a parent in order to explore new possibilities that may have not been investigated yet. This mechanism of *Recording and Asking ATFQs* was inspired by [10], but it was significantly modified and improved. It is also applied the same way for all Parenting, JPAL and JPAL-HA. Specifically,

[7] We will return to the reason it is left blank when the loop is finished.

[8] For example, an entry containing two catastrophic actions would have $\mu = 0.5$ and $a^* = a_{best}$, i.e. a safe good action.

continuing the algorithms (Lines 9–15 in Parenting and JPAL, and 28–34 in JPAL-HA), if the agent does not ask a BTFQ, it will select the greedy action from $\pi(a \mid s)$. Before executing it (Line 14 in Parenting and JPAL, and 33 in JPAL-HA), updating its policy (Line 15 in Parenting and JPAL, and 34 in JPAL-HA) and finishing the step, it performs two more tasks. First, it *records a pair* of the greedy action and a *randomly* chosen one and adds it to a *Temporary Memory* $\mathcal{R}$ with probability p_{REC} (Lines 10–11 in Parenting and JPAL, and 29–30 in JPAL-HA). Second, regardless of recording in that step, it could also, with probability p_{ATFQ}, pop from $\mathcal{R}$ a recorded pair (potentially added in $\mathcal{R}$ a couple of steps before) and *ask an ATFQ*, i.e. a query not related to the current state, but to a previously visited one (Lines 12–13 in Parenting and JPAL, and 31–32 in JPAL-HA). Asking an ATFQ works exactly as asking a BTFQ in terms of adding an entry to $\mathcal{X}$, with the difference that no action is decided now for execution for the current step. Essentially, recording and asking ATFQs is a simple mechanism of *safe exploration* (something like a safe ϵ-*greedy*), because the agent may succeed in bringing in $\mathcal{X}$ potentially good actions that otherwise would never be sampled from $\pi(a \mid s)$ and presented to the human due to their low probability. This is different from the approach taken by Frye and Feige [10] where they *executed* the explorative action, which could mean a 'straight suicide' if the agent was in a cell surrounded by three cells of water.

The learning at the end of each step (Line 15 in Parenting and JPAL, and 34 in JPAL-HA) is done by following the Bradley-Terry model [16] for estimating score functions from pairwise preferences. That is, we minimise, using gradient descent, the cross-entropy loss between the predictions

$$\frac{\pi\left(a_t^{(0)} \mid s_t\right)}{\pi\left(a_t^{(0)} \mid s_t\right) + \pi\left(a_t^{(1)} \mid s_t\right)}, \frac{\pi\left(a_t^{(1)} \mid s_t\right)}{\pi\left(a_t^{(0)} \mid s_t\right) + \pi\left(a_t^{(1)} \mid s_t\right)}$$

and the ground-truth labels μ, $1 - \mu$:

$$\mathcal{L} = -\sum_{s,a \in X} \mu \log \frac{\pi(a_t^{(0)}|s_t)}{\pi(a_t^{(0)}|s_t)+\pi(a_t^{(1)}|s_t)} + (1-\mu) \log \frac{\pi(a_t^{(1)}|s_t)}{\pi(a_t^{(0)}|s_t)+\pi(a_t^{(1)}|s_t)} \quad (1)$$

We minimise this loss using all samples from the dataset $\mathcal{X}$ in each step.

Algorithm 3 JPAL-HA

Input: $p_{BTFQ} \in [0,1)$: hyperparameter close to 1, p_{REC}: probability of recording, p_{ATFQ}: probability of asking an ATFQ
Output: $\pi(a \mid s)$: agent's policy model

function gen_hypoth_actions(s_t, $a_t^{(0)}$, $a_t^{(1)}$, P, J_P)
    create $\mathcal{S}$ and $\mathcal{C}$
    add to them $a_t^{(0)}$ and $a_t^{(1)}$ accordingly based on P and J_P
    while $len(\mathcal{S})<2$ and $len(\mathcal{S})+len(\mathcal{C})<num_all_actions$ **do**
        sample a_{new} and ask for $J_{a_{new}}$
        if $J_{a_{new}} = w$ **then**
            extend $\mathcal{X}$ with all $<a_{new}, action\ in\ \mathcal{S}\ and\ \mathcal{C}>$ pairs
            add a_{new} in $\mathcal{C}$
        else if $J_{a_{new}} = n$ **then**
            extend $\mathcal{X}$ with all $<a_{new}, action\ in\ \mathcal{C}>$ pairs
            ask P_{add} between a_{new} and a_{best} and extend $\mathcal{X}$
            update a_{best} if a_{new} is preferred to a_{best}
            add a_{new} in $\mathcal{S}$
    add $a^* = a_{best}$ to all action pairs in $\mathcal{X}$ of the loop (4–13)
    return a_{best}

function ask_BTFQ(s_t)
    sample $a_t^{(0)}$ and $a_t^{(1)}$ and receive P and J_P from human
    if $J_P = n$ **then**
        find a^* and μ from P and J_P
        add entry (s_t, $a_t^{(0)}$, $a_t^{(1)}$, μ, J_P, a^*) to $\mathcal{X}$
        execute a^*
    else if $J_P = w$ **then**
        $a_{best} \leftarrow$ gen_hypoth_actions(s_t, $a_t^{(0)}$, $a_t^{(1)}$, P, J_P)
        execute a_{best}

repeat
    if $(p_{BTFQ})^{f(s_t)} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
        ask_BTFQ(s_t)
    else
        if $p_{REC} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
            record a $<$*greedy, random*$>$ action pair from s_t in $\mathcal{R}$
        if $p_{ATFQ} > r \overset{\text{iid}}{\sim} U[0,1]$ **then**
            ask an ATFQ from $\mathcal{R}$ and add entry to $\mathcal{X}$
        execute greedy action drawn from $\pi(a \mid s)$
    train policy $\pi(a \mid s)$ with gradient descent minimising Equation 1
until optimal policy is found

5 Evaluation

5.1 Experimental Setup

To evaluate the impact of *Justifications* and *Hypothetical Actions* we first tuned and modified Parenting to a great extent so that the comparisons amongst *Parenting*, *JPAL* and *JPAL-HA* were fair. That is, all components such as model architecture, learning rate, learning epochs per step, and the techniques of *recording* and *asking ATFQs* were optimised for Parenting and also used for JPAL and JPAL-HA[9].

The policy model $\pi(a \mid s)$ is the one used in [10]. It is the average of two components. The first is a global component which is a CNN, mapping the board state to the actions; the board state is converted to an One-Hot Encoding form where, for each object (Agent, Goal, Wall, Free road, Water), a different matrix of zeros and ones is created (that is, the state's dimensions in the input layer is Height of board×Width of board×Number of objects). The CNN consists of at most four convolutional layers with number of kernels 16, 32, 64, 64 from first to last. The kernel size is 3×3 and the stride is 1. Appropriate padding, depending on the spatial dimensions of each gridworld, is used before the final convolution in order to bring the spatial dimensions to 2×2, regardless of the input board dimensions. Every convolutional layer is followed by a rectifier nonlinearity activation. The final hidden layer is fully-connected and consists of 64 rectifier units. The output layer is a fully-connected linear layer with 4 outputs as the number of actions. The second component is a local FNN with input being the type of object of the four neighbouring cells of the agent, one hidden layer of 64 units and four neurons at the output.

For optimisation we used the Adam optimiser with learning rate 0.001 and the default initialisation of parameters. Regarding the three hyperparameters of the JPAL algorithm, we experimented with a *conservative setting*: $p_{BTFQ} = 0.95$, $p_{REC} = 0.8$ and $p_{ATFQ} = 0.8$, and a *relaxed setting*: $p_{BTFQ} = 0.8$, $p_{REC} = 0.5$ and $p_{ATFQ} = 0.5$. The most important parameter is p_{BTFQ}, which essentially trades safety with human burden. As the number of samples in our learning dataset $\mathcal{X}$ grows in every step, we created a rule for the number of training epochs per step, instead of keeping them constant. Specifically, in steps that $\mathcal{X}$ was not extended we only train for 1 epoch, while in other cases we follow the rule:

$$\#epochs = \lceil w_{epoch} * (X_{current_step_size} - X_{previous_step_size}) \rceil, \qquad (2)$$

where w_{epoch} is a hyperparameter found to work best when equal to 9. We also found it beneficial to use a learning rate scheduler, whereby when the embraced memory $\mathcal{X}$ is not updated for 20 steps, the learning rate is increased by 10

[9] We immediately witnessed a significant improvement to the results of Parenting after optimising it, which was probably because Parenting from [10] explored carelessly (as described in Section 4.3) and additionally trained the agent only for 1 epoch per step. The results we report are from the optimised Parenting.

and the number of epochs is increased to 40 just for that step. In general, we observed that small changes in the hyperparameters p_{REC}, p_{ATFQ} and w_{epoch} negligibly affected the output (demonstrating the robustness of the algorithm) so we decided to tune them to a value that worked well and did not undertake an exhaustive sensitivity analysis.

To avoid rare trials in which the agent gets stuck in a loop of bumping into the same wall cell or going back and forth between two cells, we implemented a simple mechanism that sensed such occasions, i.e. staying at the same state for more than one step or returning to the previous state. This mechanism during a BTFQ kept the human preferred action and also replaced the un-preferred sampled action with a new action from the agent's policy. We did this in a loop until a new entry appeared in the embraced memory $\mathcal{X}$ which allowed the training to smoothly continue. We believe this is a sensible and practical assumption that could also be implemented for stochastic environments, e.g. by observing the local area, as the agent could utilise simple memory rules or the human could take action as a special case.

In Section 5.9, we trained a mobile robot with real human users and a robot. However, for convenience, we ran simulation experiments to draw more conclusions (Sections 5.2–5.8) and employed the exact behaviour of a rational human. To do this, we computed the optimal Q-values at all states of the board. Preferred actions have higher Q-values and catastrophic actions have negative Q-values.

5.2 Comparison with traditional RL

Initially, we verified our assumption that traditional RL algorithms utterly fail in terms of safety. For Q-learning [17] with $discount\ factor = 0.5$, $learning\ rate = 1$ and ϵ (a small number determining exploration) starting from 0.01 and decaying exponentially[10], we extracted the blue line of Figure 2 which illustrates the return for 100 episodes averaged over 50 trials. The red dashed line depicts the maximum possible return, which is +42 because +50 is the reward for reaching the goal and the agent needs 8 steps to reach it. We found that in 22.9 episodes on average when the optimal policy had been found (i.e. if we deployed the agent, it would give the fastest route), the agent had already suffered 16.54 ± 1.34 deaths over 50 trials[11], which is considered a very high number, comparing to safety-oriented methods. We also experimented with SARSA [18] which is considered 'safer' than Q-learning, but still could not drop the number of deaths below 15 which is again by no means satisfactory for safety. In contrast, JPAL-HA with the conservative setting (green line) finds the optimal policy much earlier on average (3 episodes) and suffers only 0.02 ± 0.15 deaths. We realise that with safety-oriented methods, like ours, the goal is to avoid even 1 death.

[10] The $discount\ factor$ was chosen after experimentation, the $learning\ rate$ was 1 as this is the optimal value for deterministic environments and the starting ϵ was set as a small value, in order to keep deaths as low as possible.

[11] We report the mean and 1 standard deviation values $\mu \pm \sigma$ for this and the next experiments.

Moreover, a subtle point that many works do not take into account is what would happen if the agent was left training after it could deploy the optimal policy. This is important because in many real-world scenarios, optimality cannot be observed automatically such as in simulations, where we can quickly check after each episode by testing if the return of the updated policy is equal to the maximum possible return or by checking if the actions are optimal while training. A safe way to observe that the optimal policy has been found in a real scenario is to deploy and supervise the agent at the end of every single episode. However, that would be very time-consuming. Having an understanding that optimality has almost been achieved, we could leave the agent to train more and only then deploy it while supervising it, when we would be more or less certain that we would not have to go over the same time-consuming procedure more than once. As a result, it is important to check that further training would not lead to any potential future deaths. We observe that for JPAL-HA right after the 3rd episode when the agent finds the optimal policy, if we continued training there would be no more deaths[12]. In contrast, Q-learning would still suffer deaths due to unsafe exploration (we can see that from the spikes after 22.9 in the axis of episodes). This observation concerning JPAL-HA is not without reason: as stated by Frye and Feige [10] "*Direct policy learning ensures the agent does not contradict previous human input*", or in other words, the human is not going to change their mind when they are asked specific questions, such as: 'Do you prefer A or B?' or 'Is X safe or not?'.

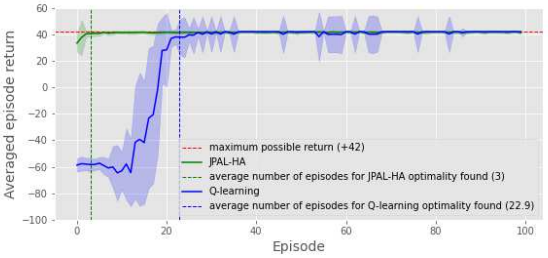


Fig. 2 Comparing JPAL-HA with Q-learning - Averaged episode return over 50 trials and 1 standard deviation

5.3 Comparing Parenting, JPAL and JPAL-HA

Next, we start testing the effect of our novel features, Justifications and Hypothetical Actions. For this section, we make two assumptions: (i) justifications is a marginal additional cost to preferences, which we do not consider; and (ii)

[12]There is a rare case (encountered for very few seeds) of a catastrophe after optimality, which may occur when the two sampled actions do not include the greedy one that leads to the goal (unlikely since the policy has already been shaped), so the agent starts going through completely new routes. A candidate simple fix to this would be one of the two sampled actions to always be the greedy one after some advanced point in time.

extra queries during a loop of hypothetical actions (Lines 4-13 in Algorithm 3) do not impose significant additional bother cost to the human users as they are concerned with the same state, and do not get bothered repeatedly at regular intervals at those same states. Hence, we wrap everything into a BTFQ. However, in Section 5.6 we do consider and model the additional human time potentially needed.

Table 1 shows the results of simulation experiments acquired over 1000 trials until the optimal policy is found. It is worth noting from the beginning that the trend in both settings (conservative and relaxed) is similar, which indicates the robustness of our algorithm to the different hyperparameters.

Initially, we compare Parenting with JPAL (i.e. using only *justifications*). For both the conservative and the relaxed setting, we observe an important decrease in the number of training deaths[13] (safety improved due to Justifications), but an increase in BTFQs[14], which represents extra human burden. We interpret the fewer BTFQs of Parenting in the following way: in Parenting the agent goes through the same 'unsafe' routes leading to more deaths while not asking the human many questions. As the familiarity $f(s)$ increases, the agent acts more and more independently. Thus, we conclude that although the BTFQs increase in JPAL, there is a good reason: safety.

Comparing JPAL with JPAL-HA (adding *hypothetical action* queries), we observe a further improvement in safety (more significant in the relaxed setting) and a significant decrease in bothering (BTFQs). This is happening because the agent asks many queries in one go when it is in a 'dangerous' area and the human does not need to be bothered repeatedly with such cases. We believe that in most cases it is reasonable to assume that asking a lot of queries about the same state at once is not as time-consuming for the human as multiple bothering BTFQs; nobody enjoys being interrupted every couple of minutes, whilst one does not mind quickly answering a few questions on the same subject[15]. As noted in Section 4.2, most of these hypothetical action queries (asked until a second safe action is found) do not require a preference answer P, but simply a single justification over a new (hypothetical) action $J_{a_{new}}$ (i.e. 'if I had taken that other action, would it be safe?').

We conducted significance tests for the variables of training deaths, BTFQs and Overall steps to observe if there is a statistical difference amongst Parenting, JPAL and JPAL-HA. We used the significance level $\alpha = 0.05$. Initially, we assured that the assumptions of *homoscedasticity* (equal variances) and *normality* (data generated from the normal distribution) do not hold for our data. We did that both by observation (plotted the bar charts) and by running the *Levene's test* [19] for homoscedasticity and *Shapiro–Wilk test* [20] for normality. These tests showed that none of the aforementioned assumptions

[13] Unlike other variables that the standard deviation is lower than the mean, regarding the number of training deaths the standard deviation is higher because deaths are rare having been averaged over 1000 trials.

[14] Until Section 5.6 we do not take into account ATFQs which as it will be described in Section 5.5 are not urgent to be issued and also the differences in their number amongst methods are marginal.

[15] We will return to this argument in Section 5.6.

holds. Thus, we conducted the non-parametric *Kruskal–Wallis one-way analysis of variance test* (which does not make these assumptions) to determine if there is a statistical difference for the means amongst the three different methods. As expected, p-values for all variables and for both the conservative and relaxed setting were much smaller than α, so we conclude that there is a clear significant difference amongst the three methods. The interesting question was to check the pairwise differences between methods. Following the rejection of the *Kruskal–Wallis* test, we conducted the post-hoc, non-parametric pairwise *Dunn's test* [21], which is usually preferred to the other option of *Mann–Whitney-Wilcoxon rank-sum U test* [22]. We also needed to tackle the Multiple comparisons problem (due to three pairwise hypotheses tests) by controlling in our case the False-wise error rate (FWER) (i.e., we wish to minimise the False Positives) instead of the False discovery rate (which gives better power, i.e. caring about a better True Positive rate). The correct adjustment in this case is the *Holm-Bonferroni* [23], which dominates the plain Bonferroni correction and is used under arbitrary assumptions. Table 2 shows the adjusted p-values which we compare with $\alpha = 0.05$. With reference to Table 1, in terms of training deaths, we observe a significant difference with the introduction of JPAL for both the conservative and relaxed setting and a significant difference with the introduction of Hypothetical Actions in the relaxed setting (a reduction of training deaths). In terms of BTFQs and overall steps the unfavourable for JPAL significant difference with Parenting (BTFQs and overall steps increased), is fixed with the introduction of JPAL-HA (favourable for JPAL-HA significant difference with JPAL — BTFQs and overall steps dropped), where in the relaxed setting we even observe a significant difference in favour of JPAL-HA to the initial Parenting ($p = 0.02$). In short, we achieved to improve safety without deteriorating BTFQs or overall steps both in the conservative and relaxed setting.

Table 1 Average number of training deaths, BTFQs, Recordings, ATFQs and overall steps in simulation experiments until the optimal policy is found (1000 trials for each with mean and standard deviation reported).

	Conservative setting			*Relaxed setting*		
	Parenting	*JPAL*	*JPAL-HA*	*Parenting*	*JPAL*	*JPAL-HA*
Training Deaths	0.07 ± 0.27	$\mathbf{0.04} \pm 0.23$	$\mathbf{0.02} \pm 0.15$	0.31 ± 0.61	$\mathbf{0.22} \pm 0.54$	$\mathbf{0.13} \pm 0.35$
BTFQs	26.25 ± 5.8	31.12 ± 8.21	25.92 ± 6.53	24.67 ± 5.54	29.63 ± 7.54	24.02 ± 5.98
Recordings	1.59 ± 1.45	1.97 ± 1.72	1.76 ± 1.64	3.85 ± 2.48	4.79 ± 3.09	4.13 ± 2.86
ATFQs	1.32 ± 1.32	1.63 ± 1.54	1.47 ± 1.45	2.61 ± 1.94	3.3 ± 2.36	2.74 ± 2.13
Overall steps	37.72 ± 14.25	41.94 ± 15.58	37.76 ± 14.5	39.86 ± 15.45	46.38 ± 17.48	40.51 ± 15.59

5.4 Analysing the choice of p_{BTFQ}

Regarding safety, it is clear that increasing p_{BTFQ} to a value such as 0.99, almost always yields 0 deaths even in 1000 trials (even though this is not proven

Table 2 Dunn's test (table shows the p-values of pairwise comparisons in each setting).

	Conservative setting			*Relaxed setting*		
	Training deaths	*BTFQs*	*Overall steps*	*Training deaths*	*BTFQs*	*Overall steps*
Parenting/JPAL	$< 10^{-2}$	0	$< 10^{-8}$	$< 10^{-3}$	0	$< 10^{-17}$
JPAL/JPAL-HA	0.1	0	$< 10^{-9}$	$< 10^{-4}$	0	$< 10^{-15}$
Parenting/JPAL-HA	$< 10^{-5}$	0.28	0.73	$< 10^{-15}$	0.02	0.5

theoretically since JPAL-HA is a probabilistic algorithm and gives experimental safety guarantees) and the human bothering would also not be continuous. That means that our method is useful even in highly safety-critical environments, where we should simply pick a very high p_{BTFQ} (even higher than 0.99 possibly). In contrast, in less dangerous cases, e.g. washing the dishes, we would not mind relaxing p_{BTFQ} for the sake of less human time and potentially sacrifice a few plates. Taking, for example, the *JPAL-HA* case of the relaxed setting, we observe that deaths do not markedly increase (0.13 in Table 1).

Figure 3 summarises the evaluation of the three methods in terms of safety (mean values extracted over 1000 trials). We keep constant the hyperparameters $p_{REC} = 0.8$ and $p_{ATFQ} = 0.8$ and we alter the most important one p_{BTFQ}. Firstly, if we compare the three methods at any p_{BTFQ} point (except for $p_{BTFQ} = 1$ which would normally be unwanted since it implies constant supervision), we observe that JPAL is significantly safer than Parenting, and JPAL-HA is safer than JPAL (the latter is more distinct in more 'relaxed settings' ($p_{BTFQ} \leq 0.9$)). The fact that this happens with consistency shows the robustness of our framework. Moreover, we observe that as we increase p_{BTFQ} all methods improve safety until the point that all of them achieve 0 deaths. This indicates the safety/human burden trade-off discussed earlier. To make it more evident, if we plot the BTFQs versus p_{BTFQ} curve for JPAL-HA we observe the almost linear ascending relationship of Figure 4a (similar for Parenting and JPAL). Hence, as expected, Figure 4b which plots the training deaths versus BTFQs curve (real trade-off) for JPAL-HA has almost the same trend as the one of Figure 3 and clarifies the trade-off between safety (training deaths) and human burden (BTFQs).

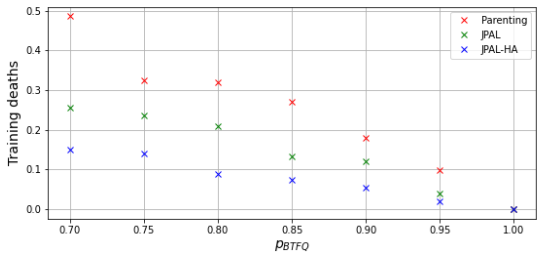
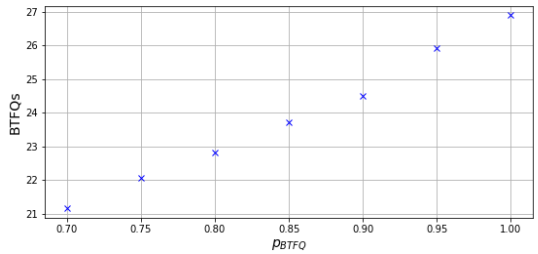
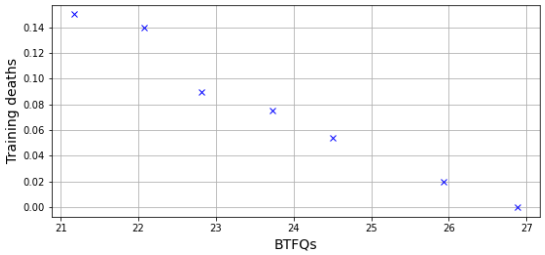


Fig. 3 Training deaths versus p_{BTFQ}.



(a) BTFQs versus p_{BTFQ}



(b) Training deaths versus BTFQs

Fig. 4 JPAL-HA safety/human burden trade-off.

5.5 Recording and asking ATFQs

As described in Section 4.3, recording and asking After-The-Fact-Queries is an additional mechanism for safe exploration. We noticed in some trials that when the agent was becoming very 'confident' from a limited number of actions in a particular state and would not sample other potentially better actions to present them for a query to the user, an ATFQ could help it escape from such a behavioural loop learnt by chance.

Another benefit of ATFQs is characteristic to relaxed settings (lower p_{BTFQ} values). A relaxed setting is used when the human does not wish to be bothered consecutively by a lot of BTFQs, which need to be answered on the spot. Since ATFQs are queries about states visited in the past and recorded in the *Temporary Memory*, they do not need to be necessarily answered there and then, as they will not affect safety at that step (nothing will change if they are answered a few steps later). Indeed, as we can see in Table 1, although the aggregate of queries (BTFQs and ATFQs) is almost the same for both the conservative and relaxed setting, the additional ATFQs in the latter setting, gives extra 'relaxation' to the user to answer at their convenience.

5.6 Unwrapping a JPAL-HA query

Even though in Section 5.3 we claimed a significant difference between Parenting and JPAL-HA in terms of safety and no significant difference in terms of

BTFQs, this was based on the assumption that justifications and hypothetical actions place marginal additional burden, which we do not need to consider. The idea was that justifications take very little time and also that the human could finish off with a state in one go during hypothetical action queries, with no subsequent queries there. We do, however, need to more thoroughly test this assumption. We do this both here by 'unwrapping' the different types of queries, and then explore this further in our experiments with people in Section 5.9.

In JPAL-HA a query can take one of the following types:

1. $P^{(BTFQ)}$: preference between initial $<a_t^{(0)}, a_t^{(1)}>$ from a BTFQ
2. $P^{(ATFQ)}$: preference between initial $<a_t^{(0)}, a_t^{(1)}>$ from an ATFQ
3. $P_{add}^{(BTFQ)}$: additional preference between $<a_{new}, a_{best}>$ from a BTFQ
4. $P_{add}^{(ATFQ)}$: additional preference between $<a_{new}, a_{best}>$ from an ATFQ
5. $P_{add}^{(DP)}$: additional preference between $<a_{new}, a_{best}>$ from a DP
6. $J_P^{(BTFQ)}$: justification over $P^{(BTFQ)}$
7. $J_P^{(ATFQ)}$: justification over $P^{(ATFQ)}$
8. $J_{a_{new}}^{(BTFQ)}$: justification over a_{new} from a BTFQ
9. $J_{a_{new}}^{(ATFQ)}$: justification over a_{new} from an ATFQ
10. $J_{a_{new}}^{(DP)}$: justification over a_{new} from a DP

We assume that the human needs more cognitive effort to answer preference queries (1–5) than justification queries (6–10). Moreover, for simplicity we assume that justifications over two actions (6–7) require the same effort with justifications over a new (hypothetical) action (8–10). Parenting includes only query types 1 and 2, and JPAL includes queries 1, 2, 6 and 7. Table 3 shows the break-down of different types of queries, with the values acquired in the conservative setting.

Table 3 Comparing Parenting, JPAL and JPAL-HA with all types of queries

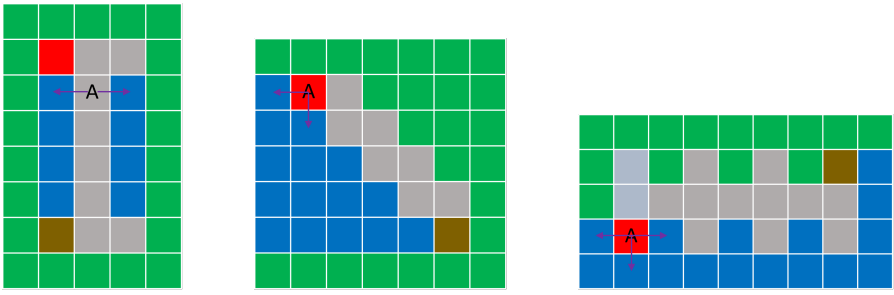
<table>
<tr><th></th><th colspan="2">Parenting</th><th colspan="3">JPAL</th><th colspan="3">JPAL-HA</th></tr>
<tr><td>$P^{(BTFQ)}$</td><td>26.25</td><td rowspan="6">$P^{(Parenting)} = 27.57$</td><td>31.12</td><td rowspan="5" colspan="2">$P^{(JPAL)} = 32.75$</td><td>25.93</td><td rowspan="5" colspan="2">$P^{(JPAL\text{-}HA)} = 30.98$</td></tr>
<tr><td>$P^{(ATFQ)}$</td><td>1.32</td><td>1.63</td><td>1.47</td></tr>
<tr><td>$P_{add}^{(BTFQ)}$</td><td>-</td><td>-</td><td>3.54</td></tr>
<tr><td>$P_{add}^{(ATFQ)}$</td><td>-</td><td>-</td><td>0.03</td></tr>
<tr><td>$P_{add}^{(DP)}$</td><td>-</td><td>-</td><td>0.01</td></tr>
<tr><td>$J_P^{(BTFQ)}$</td><td>-</td><td rowspan="6">-</td><td>31.12</td><td rowspan="2">$J_P^{(JPAL)} = 32.75$</td><td rowspan="6">$J^{(JPAL)} = 32.75$</td><td>25.93</td><td>$J_P^{(JPAL\text{-}HA)} = 27.4$</td><td rowspan="6">$J^{(JPAL\text{-}HA)} = 33.21$</td></tr>
<tr><td>$J_P^{(ATFQ)}$</td><td>-</td><td>1.63</td><td>1.47</td></tr>
<tr><td>$J_{a_{new}}^{(BTFQ)}$</td><td>-</td><td>-</td><td rowspan="3">-</td><td>5.57</td><td rowspan="3">$J_{a_{new}}^{(JPAL\text{-}HA)} = 5.81$</td></tr>
<tr><td>$J_{a_{new}}^{(ATFQ)}$</td><td>-</td><td>-</td><td>0.21</td></tr>
<tr><td>$J_{a_{new}}^{(DP)}$</td><td>-</td><td>-</td><td>0.03</td></tr>
</table>

Parenting takes the least time as it requires the fewest preference queries to be answered and does not seek justifications. However, we have already seen that JPAL significantly improves safety during learning and JPAL-HA also offers additional improvements in safety (Figure 3). Comparing JPAL and JPAL-HA with respect to time required to respond to queries, although the number of justification queries issued by JPAL-HA, $J^{(\text{JPAL-HA})}$, are very

slightly higher than those by JPAL, $J^{(JPAL)}$, JPAL-HA issues fewer preference queries, i.e. $P^{(\text{JPAL-HA})} < P^{(\text{JPAL})}$. This reduction in preference queries is observed in all settings, confirming that Hypothetical Action queries reduce the need for more preference queries. Thus, we hypothesise that HAs help to decrease human burden. Our analysis of the hypothesis, $t^{(\text{Parenting})} < t^{(\text{JPAL-HA})} < t^{(\text{JPAL})}$, is tested in Section 5.9 through experiments with people.

5.7 Detecting deadly patterns

In Section 4.2 we described as part of HAs the mechanism of detecting deadly patterns; i.e. dynamically storing patterns in the *Danger Memory* $\mathcal{D}$ and avoiding them, without the need of human input. Figure 5 shows three characteristic examples, where such a technique is useful. In Figure 5a the deadly pattern $dp = (Left, Right)$ is likely stored during the first step of 'crossing the bridge' and then checked all along the way. Similarly, in Figure 5b once the deadly pattern $dp = (Left, Down)$ is stored in $\mathcal{D}$, it is continuously automatically detected throughout the path to the goal state. This means that not only are such patterns always avoided, but also the human is not bothered answering a preference query about this combination (although they do need to provide information about the other cells, as in HAs). For 1000 trials ran using the conservative setting, both of these environments gave trends of training deaths and number of queries for Parenting, JPAL and JPAL-HA similar to the original configuration (Figure 1). The only difference was now that in JPAL-HA $P_{add}^{(DP)}$ and $J_{a_{new}}^{(DP)}$ were comparable or even exceeded $P_{add}^{(BTFQ)}$ and $J_{a_{new}}^{(BTFQ)}$, respectively. Similar observations were made in Figure 5c where the mechanism is even more useful, as once $dp = (Left, Right, Down)$ is detected, then only one $J_{a_{new}}^{(DP)}$ query about the fourth safe action is required when a pattern state is met, in order to map out the whole dangerous area. As a result, we can exploit this technique to both improve safety and reduce the number of queries issued.



(a) (b) (c)

Fig. 5 Characteristic examples of detecting deadly patterns

5.8 Queries or human intervention?

Before we present the evaluation of Parenting, JPAL and JPAL-HA with real human trainers and a robot in terms of actual training time, we compare our approach with a different class of human-in-the-loop safe learning methods that rely on the human teacher to intervene when an unsafe action is about to be taken, rather than the agent querying the human before taking an action. We compare JPAL-HA as closely as possible with Human Intervention RL (HIRL) [24]. We did not compare with ReQueST [14, 25], another human-in-the-loop method with a focus on safety, because the necessary approximations would make a comparison unrealistic; for example, ReQueST uses off-policy data and there is a question of how these data can be collected.

HIRL [24] learns in three stages: (i) the agent learns a policy using standard reinforcement learning methods with a human overseeing it, blocking catastrophic actions and proposing an alternative (normally the best when they understand the environment) action; (ii) the agent pauses and a binary catastrophe classifier is built from the record of interventions from Stage (i) using supervised learning; and (iii) the agent starts again, but now the human is substituted by the catastrophe classifier (a kind of governor). As the authors note, in Stage (i) the agent must slow down to steps of around 1s to leave enough time for the human to act. The Human's judgement of actions in each state as 'safe' or 'catastrophic' creates a dataset used to learn the governor in Stage (ii). During learning in Stage (i) the catastrophic action is also penalised with a high negative reward (-50 in our case) as if the action had been really taken so that the agent both learns a policy to avoid catastrophic actions and builds the governor. In this comparison, we used the same technique as the HIRL authors of passing the input state of the classifier through several convolutional layers to output a vector, and then concatenating that vector with the one-hot-encoding vector of the corresponding input action. After the concatenation, a few more linear layers were used. We used the sigmoid function at the output, Binary Cross Entropy loss as the objective function, 200 epochs, the ADAM optimiser with learning rate=0.01 and Q-learning as the RL algorithm with the same settings as in Section 5.2. The optimal human behaviour for HIRL was simulated in the same way as for JPAL-HA.

The comparison relies on the reasonable assumption that a preference P plus a justification J_p (or $J_{a_{new}}$ for simplicity) query with JPAL-HA requires more or less the same time as an *Intervention* (i.e. a block plus a proposal of a new action) in HIRL. Also, given both algorithms consider the trade-off between safety and human burden, we use safety as the top priority; i.e. minimising training deaths for both. We consider two environment configurations, the original (Figure 1) and the one illustrated in Figure 6. JPAL-HA had almost no training deaths in each configuration with the conservative setting, and for HIRL this depends on the steps the human would constantly supervise (or equivalently on the samples used to build the catastrophe classifier — Figure 7). We then compared the two methods using: (1) how much information needs to be extracted from the human (we compare $P^{(\text{JPAL-HA})}$

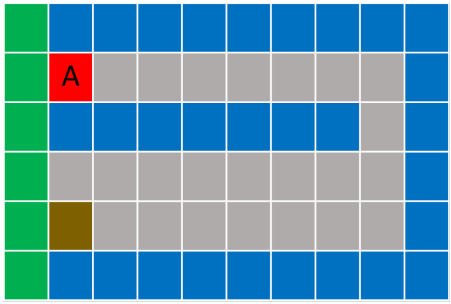


Fig. 6 Island Navigation Environment configuration with more dangerous states.

Table 4 Information required from, and time commitment of human trainer for JPAL-HA and HIRL for two environment configurations.

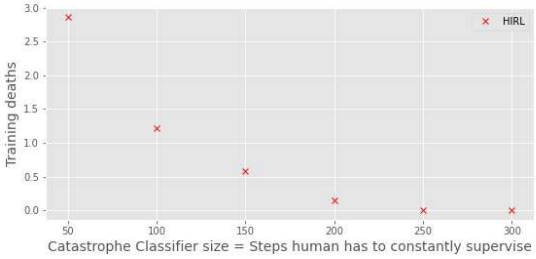
Config.	Criterion	JPAL-HA	HIRL
Figure 1	Information	$P^{(\text{JPAL-HA})} = 30.98$ $J^{(\text{JPAL-HA})} = 33.21$	$Interventions^{(\text{HIRL})} = 16.12$
	Time	$Steps^{(\text{JPAL-HA})} = 37.76$	$Steps^{(\text{HIRL})} = 250$
Figure 6	Information	$P^{(\text{JPAL-HA})} = 34.11$ $J^{(\text{JPAL-HA})} = 42.34$	$Interventions^{(\text{HIRL})} = 33.48$
	Time	$Steps^{(\text{JPAL-HA})} = 60.24$	$Steps^{(\text{HIRL})} = 350$

and $J^{(\text{JPAL-HA})}$ with $Interventions^{(\text{HIRL})}$); and (2) how long it takes to have the human oversee the agent (we compare the overall steps until optimality in JPAL-HA $Steps^{(\text{JPAL-HA})}$ with the overall steps until we can release the human in HIRL $Steps^{(\text{HIRL})}$.

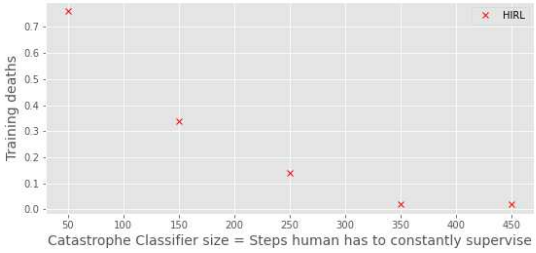
In the original configuration (Figure 1), for JPAL-HA we reproduce the results for $P^{(\text{JPAL-HA})}$, $J^{(\text{JPAL-HA})}$ and $Steps^{(\text{JPAL-HA})}$ from earlier in Table 4. For HIRL, we ran experiments to determine how many steps a human trainer will need to monitor the agent and intervene as necessary to achieve a similar number of training deaths on average. From Figure 7a, we need the human trainer to monitor for approx. 250 steps[16]. Across these 250 steps, the human trainer had to intervene an average of 16.12 times. Thus, HIRL needs to extract less information from the human trainer. However, JPAL-HA is far better in terms of the time the human has to 'wait in their chair'. Moreover, with JPAL-HA the human trainer can focus on other tasks with the agent being smart enough to ask the human when it predicts that it needs more information. Instead, with HIRL the human has to constantly monitor the agent and intervene (although the human still may relax if they are certain that the RL agent will be in a safe region for some time). Moreover, if the environment has many hazards, such as in the configuration illustrated in Figure 6, then JPAL-HA is competitive with respect to information need with $P^{(\text{JPAL-HA})}$ and $J^{(\text{JPAL-HA})}$ much closer to $Interventions^{(\text{HIRL})}$ (see Table 4). JPAL-HA continues to dominate HIRL with respect to time demands on the human trainer in this scenario.

[16] While the algorithm continues with the binary classifier and will take an average of 291.64 steps to complete, we assume that the time the agent operates by itself is not counted.

Moreover, the optimal policy (even if using the human only for the first 350 steps) was found after 533.92 steps on average, which illustrates again the sample-inefficiency of traditional RL.



(a) Original Configuration (Figure 1).



(b) Configuration of Figure 6.

Fig. 7 Safety/human burden trade-off in HIRL.

5.9 User experiments

In order to fully understand the impact of our novel human-in-the-loop learning approach, we conducted experiments with people answering queries to train a Thymio mobile robot [26]. The configuration of Figure 1 (the same as the one of the simulation experiments) was created with simple wooden pieces (Figure 8). The participants used a computer next to the arena to answer preference queries and give justifications using a simple graphical interface (Figure 9). The answers were relayed to a Raspberry Pi attached above Thymio that controls its movements. Thymio executed the actions at an average speed of 20 cm/s. ATFQs were presented to the laptop screen. After the execution of each episode, Thymio was placed at the initial state manually.

Our aim is to answer two questions: (i) Can training using human preferences and justifications be completed in a reasonable amount of time? And (ii) does our hypothesis from Section 5.6 that $t^{(\text{Parenting})} < t^{(\text{JPAL-HA})} < t^{(\text{JPAL})}$ hold? We identified seeds for Parenting, JPAL and JPAL-HA that returned the

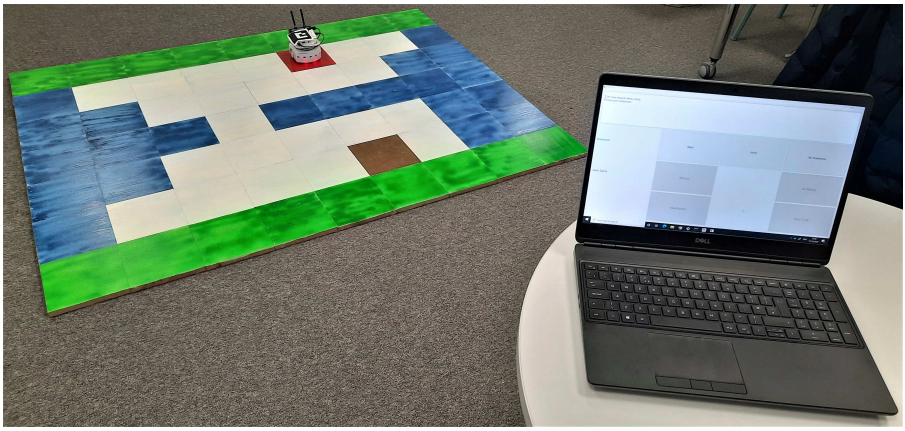


Fig. 8 Experiment configuration with a Thymio mobile robot.



Fig. 9 Instance of the graphical interface the participants used to interact with the robot in JPAL and JPAL-HA. *Alarm Signal* is equivalent to Justifications. With *Push To Talk* participants could give their answer through a microphone; however, we did not use this button during the experiments.

average values of *all* the variables involved in a trial (overall steps until optimality, number of preference/justification queries, etc.) based on the average values from the 1000 trials during of simulation experiments (Table 1). The same seeds and the conservative setting were used in all experimental trials.

We recruited 40 participants[17] overall for around one hour each, paying for their time. They consisted mainly of university students, but also included university staff and members of the general public. 65% of participants had backgrounds related to computer science, but in the other 35% we had a good diversity in terms of discipline, such as linguistics, medicine, engineering, law, administration, etc. The gender balance was 13 female and 27 male. The experiments were separated into two groups, with 18 participants in the first and 22 in the second. For each participant, we devoted around 20 minutes for them to practise in a smaller configuration to get used to interacting

[17] All experimental protocols were approved by University of Southampton Ethics and Research Governance Online (ERGO), the methods were carried out in accordance with the relevant guidelines and regulations, and informed consent was obtained from all participants.

with the robot through the interface. All participants were asked whether they were comfortable interacting with the robot. We recorded 'misses' (i.e. giving a sub-optimal or bad answer to a query) for all participants, and excluded two outliers from our analysis (Figure 10). We did this because our aim is to focus on training time rather than training with human input that varies in trustworthiness; cf. [27]. Although some 'misses' are to be expected, enforcing an average per participant to be within a low mean and tight distribution is, we believe, reasonable.

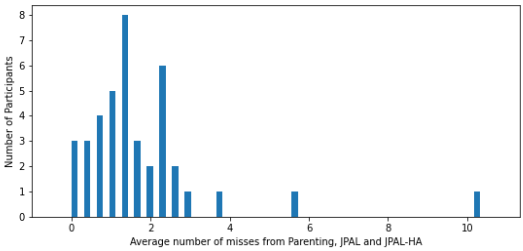


Fig. 10 Histogram of the average number of misses from all three methods. Misses are wrong answers to any type of query (preference or justification) that were mainly due to lack of attention. The two points on the right of the histogram were excluded as outliers from the results.

Although we provided initial training with the smaller configuration, there is a further training effect where participants tend to perform better in later trials. For the **first** group of participants, we reduced this training effect by permuting the order of the three methods and changing the configuration by rotating it 90 or 180 degrees to avoid memorisation. The participants ran each method once and the boxplot of each method is shown in Figure 11. These results clearly indicate that training with each of the three methods can be completed in a reasonable time (slightly more than 5 minutes with JPAL and JPAL-HA).

To determine whether there are statistically significant differences amongst the three methods, we initially ran *Levene's* [19] and *Shapiro–Wilk* [20] tests and, respectively, found that the assumptions of homoscedacity (equal variances) and normality (data generated from the normal distribution) hold. From an ANOVA test, we found that there is a significant difference amongst the three methods. Using Tukey's test, with the proper adjustments of the p-value, there is a significant difference between Parenting and the other two methods, but no significant difference between JPAL and JPAL-HA (Table 5). Although there is a significant difference between JPAL-HA and Parenting, this is around 30% on average. Further investigation into these results revealed that a significant training effect remained due to the extra features of JPAL and especially of JPAL-HA that the participants had to get used to; participants almost always took less time using the second method, regardless of whether this was JPAL or JPAL-HA.

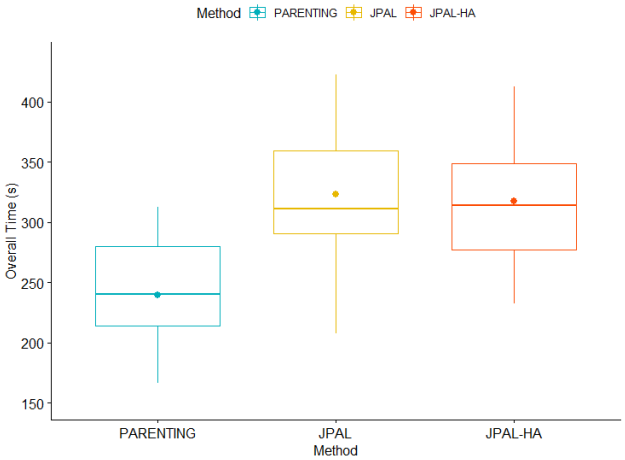


Fig. 11 Overall training time for the first group of participants.

Table 5 Adjusted p-values from Tukey's test for the first group and from the Student's t-test for the second group.

	First group	Second group
Parenting/JPAL	0.001	-
JPAL/JPAL-HA	0.960	0.332
Parenting/JPAL-HA	0.001	-

In order to investigate this further, we recruited the **second** group of participants, and, to reduce the training effect, we asked participants to run JPAL and JPAL-HA twice. They also started with Parenting to give additional practice. The participants then ran JPAL and JPAL-HA in different orders and by rotating the configuration. The results again showed subtle differences between them on the first try. However, on the second try, we found $t_{\text{JPAL}} = 242 \pm 33.3$ and $t_{\text{JPAL-HA}} = 231 \pm 34.9$ (Figure 12); i.e. participants took slightly less time on average using JPAL-HA than JPAL (approx. 11s). Again, the *Levene's* [19] and *Shapiro–Wilk* [20] tests indicated that the results had equal variances and were normally distributed. Analysing the results from these trials, there is no significant difference (Student t-test) between JPAL and JPAL-HA, but we do observe a shift in the p-value from the first group. Looking in depth at the data for these experiments, we observed that the average response time for $J_{a_{new}}$ queries (2.45s) (i.e. justifications over hypothetical actions in JPAL-HA) was significantly greater than for J_P queries (1.14s) (i.e. justifications over preferences used in both JPAL and JPAL-HA). Our explanation for this is that $J_{a_{new}}$ queries come sparsely and users do not expect them in every step like J_P queries, and that $J_{a_{new}}$ queries refer to a new action that the users have

not yet considered. This may be part of the explanation that we do not see a significant difference between JPAL and JPAL-HA with human participants.

Most importantly, the time required to train the agent using both JPAL and JPAL-HA are now significantly reduced from the average for the first group of participants (from around 5 minutes to around 4 minutes), which is close to the average for Parenting in the first group. It is also very close to the average for Parenting in the first group, even when only the participants who ran Parenting third in the row are considered (meaning they already had significant practice after running JPAL and JPAL-HA). The latter's average time was 219s which is only 12s less than $t_{\text{JPAL-HA}} = 231$. This means that, even without the assumption that justifications and hypothetical action queries are a marginal cost, the training time can be kept with JPAL and especially JPAL-HA to admissible levels, not very far from the one of Parenting.

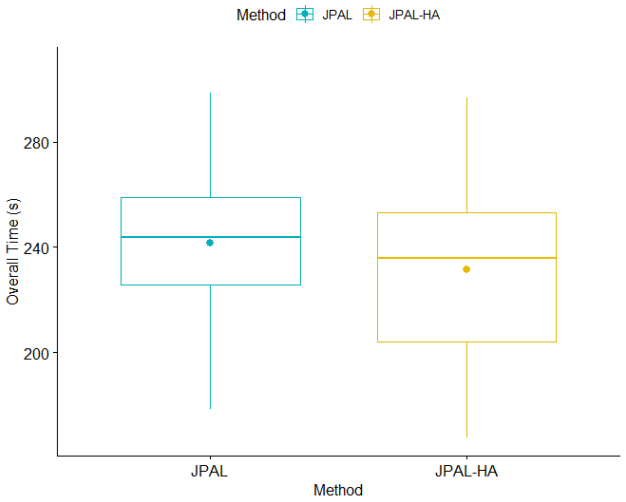


Fig. 12 Overall training time for the second group of participants for their second use of JPAL and JPAL-HA.

To summarise our findings, when given initial practise, the participants took around 30% longer to train the Thymio robot using either JPAL or JPAL-HA than for Parenting. When participants have additional training on providing justifications and answering hypothetical action queries, however, the training times were very similar. Our hypothesis, $t^{(\text{Parenting})} < t^{(\text{JPAL-HA})} < t^{(\text{JPAL})}$ is not supported, but it is reasonable to conclude that $t^{(\text{Parenting})} \simeq t^{(\text{JPAL-HA})} \simeq t^{(\text{JPAL})}$. Thus, the time required to train the agent is similar for people from a diverse set of backgrounds, but we have significant added benefits with respect to safety during the learning process.

6 Discussion

6.1 Related work

There is a variety of techniques that use a human-in-the-loop to help an agent learn a good policy. Most focus on sample efficiency, but a few also consider safety.

Our work is distinct from approaches which learn from human demonstrations such as *imitation learning* [28] or *inverse reinforcement learning*, [29–31], because rather than needing to be expert in executing the task, the human provides feedback signals. It is also different from *reward shaping*, where the feedback is a human reward over an action or trajectory, which can replace or be combined with the environmental reward [25, 27, 32–34]. In general, reward shaping does not deal directly with the problem of safety and it is also not always clear if a human can define whether an action or trajectory is good (or how good) or not and return a good reward signal. The *ReQueST* (Reward Query Synthesis via Trajectory optimisation) framework, however, uses reward shaping in [25] and also addresses safety through hypothetical behaviours that are generated by a generative model of initial states and a forward model of dynamics trained on off-policy data. These data can be safe human trajectories or past failures. Then the human assigns rewards to the hypothetical behaviours and a neural network is trained to predict them. This is the work from which Hypothetical Actions were inspired for JPAL-HA, with the main difference being that we sample hypothetical actions from the policy model directly in critical situations. The ReQueST framework was scaled up in [14] to complex 3D safety-critical environments using, instead of reward shaping, a more sample-efficient type of feedback, *reward sketching* [35]. In the latter, humans are shown a video sequence of frames and are asked to draw a 'reward curve' based on the progress of the task. Although reward sketching is probably the most efficient type of human feedback, it works only with raw pixels and requires a lot of engineering and fine-tuning. Moreover, although ReQueST has been evaluated in complex 3D simulated environments, it has not been tested in the real world, where different challenges may occur, such as the acquisition of adequate off-policy data.

An approach directly dealing with safety is *human intervention* reinforcement learning (HIRL), where the human continuously supervises the agent and blocks catastrophic actions. HIRL [24, 36] (described in more detail and compared to JPAL-HA in Section 5.8) builds a binary catastrophe classifier via supervised learning, which then takes over from the human trainer. Our work has a similar motivation, but we only seek input from the human trainer when the agent has less certainty about good action choices to make. However, HIRL has been shown to scale poorly to higher-dimensional state-action spaces due to the prohibitively large supervision time needed, deriving from the inherent sample inefficiency of RL (this was evident even for a small environment in Section 5.8).

Instead, provably a very sample efficient technique, which works in all domains, is learning from *human preferences*. It is essentially an *active learning* [37] method, with the goal to eliminate human feedback time in applicable magnitudes. Instead of picking *one* greedy action (or trajectory) from a policy model and expect from the human to give it a scalar reward to that, it would be more efficient to sample *two* actions (or trajectories) and ask the human which one they prefer. This would be more helpful and natural both for the agent which would learn to distinguish which action is better and for the human who may struggle to say how good an action is but definitely knows that one of them is better than the other[18]. This is also the natural way people communicate: a child having in mind two supermarkets to buy things from, they would not ask their parent: 'How good is the one I think is better?' (reward shaping), but they would ask: 'In which of the two supermarkets shall I go?' (human preferences). A noticeable work that formalises the framework of human preferences with Deep RL is [38], which fits a reward model to the comparisons between segments of sampled trajectories. Specifically, a behavioural policy (trained by traditional Deep RL) generates these trajectories and some segments (short sequences of a trajectory) are extracted from them. Then the human compares pairs of segments, giving their preference to the one or to the other. With these data, a deep *reward* model is trained by supervised learning, where the objective function to be minimised is the cross-entropy loss between the model's prediction and the human's preference. The output of that model (scalar reward) is then given to the agent to learn via a traditional Deep RL algorithm. The main differences between [38] and our work are:

1. Christiano et al. [38] use segments of trajectories starting from different initial states for a preference query; instead, we use state-action pairs. The latter is usually in favour of safety, but it can limit the sample-efficiency and generalisation.
2. JPAL-HA (as well as Parenting and JPAL) directly learns a policy by supervised learning from an increasing dataset of human preferences, omitting the reward model. As discussed in [10] building a reward model is less aligned with alleviating the safe exploration problem, because the maximisation of the reward function with RL can lead to unsafe states if it has not been encoded to the reward with a preliminary mechanism that unsafe states should be avoided. Instead, direct policy learning guarantees that the agent will not contradict previous feedback and move towards unsafe states.

With the technique of human preferences, the amount of human feedback is decreased by several orders of magnitude, so the sample efficiency is significantly improved. However, during RL the agent's interactions with the environment still remain a lot. Recent papers work on the direction of improving sample efficiency both in terms of the human burden and the number of overall interactions, using different intelligent ways, such as the use of

[18] For example, in a questionnaire that people mark a service from 1 to 10, it may be confusing what 7, 8 or 9 means; instead a much easier question allowing a concrete answer is 'do you prefer service A or service B?'.

human demonstrations for a good initialisation of policy [39], unsupervised pre-training and Off-policy RL with Relabelling [40], semi-supervised reward learning using a large number of unlabelled samples with data augmentation [41], exploration based on the learned reward's uncertainty [42] and adopting the performance of the Q-function as the learning target at the reward learning [43]. However, these works are not oriented to deal directly with the safe exploration problem.

Our work was most similar to Parenting [10] with the main similarities being the direct policy learning from human preferences and the query selection method. We optimised Parenting in a couple of ways such as the mechanisms of recording and asking ATFQs (Section 4.3), avoiding bothering the human with repeated queries (Section 3) and different details in the architecture (Section 5.1); we differentiated our approach from Parenting by adding the generalisable ideas, discussed throughout the paper, of Justifications and Hypothetical Actions.

6.2 Generalisation of JPAL-HA

We also made some preliminary experimentation on the generalisation of JPAL-HA using the standard method of *transfer learning*. We don't want to have to train from scratch and bother the human every time the configuration changes. We rotated the original configuration (Figure 1) by 90 degrees and trained again with JPAL-HA averaging over 1000 trials. As expected, we observed the same number of BTFQs and overall steps with negligible differences. Then, we took 10 different trained models from the original configuration, averaged their parameters (alternatively we could have also trained in different source tasks and averaged those models) and initialised the model of the rotated configuration (target task) before training it with JPAL-HA[19]. We executed 10 trials, and found a noteworthy decrease (both in the mean and in the standard deviation) of *BTFQs* from 25.92 ± 6.53 to 22.79 ± 2.37 and *Overall steps* from 37.76 ± 14.5 to 32.85 ± 4.91. That implies that the agent does not simply memorise the human preferences and as such, JPAL-HA responds well to standard generalising techniques to new environments such as transfer learning. We believe we could have dropped these values more, had we gone deeper with generalisation experiments, such as training the same policy sequentially (continual learning) or concurrently in similar source tasks, or even gradually from easier-to-harder source tasks (curriculum learning). We decided to leave such experiments for future work.

It is also important to note that, even if simulations are not available for training on the source tasks, such potentially long-running real-world safe training will only need to take place once (e.g. from the company that trains robots to wash the dishes), while only short-time real-world safe training will be needed from the actual users (e.g. based on the unique characteristics of their

[19] In cases that training a lot of models in source tasks would be costly in the real world, those models could be trained in simulations (even if not perfect) with traditional RL which does not require any human labour; then their knowledge could be transferred to the real world and to the target task where the human would put the finishing touches with JPAL-HA under safe conditions.

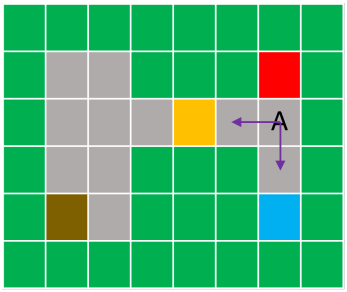
sink). Finally, we believe that if our method (as any other) was training not in a state-based way (from the coordinates x, y of the agent and the full state of the board), but from raw pixels via a camera, then generalisation techniques like transfer learning, etc. would not be as necessary. We leave testing this as a topic of future work too.

6.3 Testing JPAL-HA in more environments

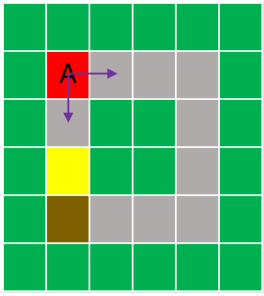
We also wanted to test JPAL-HA in some more gridworld environments (Figure 13) from [13]. Even though the main problem of Safe RL is Safe Exploration which was modelled by the Island Navigation Environment (Figure 1), there are more problems regarding the *agent alignment problem* (aligning the agent's behaviour with the human intentions) [12] that sometimes need to be considered. Since these problems mainly derive from a misspecified environmental reward function and JPAL-HA does not use such a reward, it was unlikely to face issues with these problems. However, we tried two of them.

Figure 13a illustrates the *Unsafe Interruptibility* problem modelled by the Off-switch Environment, where a traditional RL agent learning to go towards the goal is interrupted by 50% chance at the orange cell and receives a -1 reward until the end of an episode that it is interrupted. That would incentivise it to go to the light blue cell and press the button, which would disable the interruption. However, in the real world an interruption like that might be useful for the reasons of repair, upgrade, recharge the battery, etc. JPAL-HA which is driven by a human in a few only episodes learns to always avoid the light blue cell and go immediately towards the goal, for the reason that there is no reward in an episode that an interruption needs to be done. It is worth mentioning that if the environment was structured slightly differently, where pressing the button would be necessary to avoid an interruption, then as mentioned in Footnote 6, the idea of *Justifications* could be extended beyond Warnings. For example, if a robot had a low battery and needed to be recharged, the human operator would justify its preference to go left towards the orange cell due to the low battery. If, on the other hand, the robot had a full battery, then they should specify going down towards the button because of the full battery. That would be a different type of justification, which illustrates how Justifications can be used in another context.

Finally, the Absent Supervisor Environment in Figure 13b is an abstraction of the *Absent Supervisor* problem. Specifically, in the presence of a supervisor, the RL agent would go right taking the long route to the goal in order not to receive a punishment of -30 at the yellow cell. However, knowing that the supervisor is absent, it would take the short one knowing that there is no punishment. The supervisor is present with probability 50%. With JPAL-HA the agent does not get tempted to take the risky short route as the human teaches it to take the safe long one. As we observed upon deployment, after a few preference queries it never took the short way of punishment even when the supervisor was absent.



(a) The Off-switch Environment. **A**: Agent, **Red**: Initialisation cell, **Gold**: Goal, **Green**: Wall, **Grey**: Free road, **Orange**: Interruption, **Light Blue**: Button



(b) The Absent Supervisor Environment. **A**: Agent, **Red**: Initialisation cell, **Gold**: Goal, **Green**: Wall, **Grey**: Free road, **Yellow**: Punishment

Fig. 13 More environments in which JPAL-HA was evaluated.

6.4 Scalability of JPAL-HA

As described in Section 6.1, learning from human preferences is an established technique, which in contrast to other human-agent approaches (except reward sketching whose limitations were mentioned in Section 6.1) uses the human more efficiently for the exact reason of scaling better to more complex environments. We believe that our approach which uses human preferences inherits to some degree this scalability. Adding Justifications marginally affects scaling as it requires a single additional bit to answer a query (in the worst case it will double the time but still scale). Regarding adding Hypothetical Actions, we assume it slightly affects scaling, but this depends on when we intend to stop Hypothetical Action queries (stoppage rule of Hypothetical Actions loop — Line 4 in Algorithm 3). The key concern is how much safety we are willing to trade-off for human burden, and that may differ for different applications. Optimal stopping methods could have been explored, but this goes beyond the scope of this work. Additionally, the frequency of a human interacting with the agent depends on the term $(p_{BTFQ})^{f(s_t)}$, so being able to adjust p_{BTFQ} to exchange some safety for decreasing human burden gives extra flexibility to JPAL-HA. Our focus in this work was primarily on the evaluation of Justifications and HAs with a benchmark safety environment. Evaluation in more complicated environments is part of future work.

Moreover, although our algorithm has a fairly high percentage of human queries to overall agent interactions with the environment ($BTFQs/OverallSteps = 25.92/37.76$ for the conservative setting of JPAL-HA — Table 1), given that the number of these interactions is kept low, this should not be a problem. Some works, including [24] we analysed in Section 5.8, try to make this ratio of *Human Queries/Overall Interactions* lower, focusing

on just minimising the queries issued to the human; however, considering that the interactions of the agent with the environment (e.g. a move of the robot) take a substantial amount of time, it is unclear how they can scale when training has to take place in the real world. Finally, the scalability of JPAL-HA can be potentially improved if the queries are issued not on state-action pairs (1-step sequences), but on trajectories (sequences of states and actions). We leave this as an item of future work, in which justifications and hypothetical actions will refer to the last state and action of the trajectory.

6.5 Limitations of JPAL-HA and a possible future direction

We have claimed several advantages of our approach: (1) improving safety without significantly deteriorating human burden (Sections 5.3, 5.6 and 5.9); (2) available option to trade off safety with human burden by modifying p_{BTFQ} (Section 5.4); (3) ability to safely explore (Section 5.5); and (4) no more visits to unsafe states if training continues after optimality (Section 5.2). Some limitations of our approach are: (1) the assumption that a human knows the environment well and can answer agent queries correctly, which, however, is a reasonable assumption in a lot of practical real-world applications; (2) the fact that it does not come with theoretical safety guarantees — however, the trade-off safety/human burden mechanism guarantees a high level of safety for human labour exchange; (3) the myopic approach of querying over state-action pairs, which, however, has advantages in terms of safety and the agent alignment problem [12], and can be also extended to work with preferences and justifications in trajectories as described in Section 6.4.

Finally, based on the literature on Human-Based Safe RL, we notice that there is no work which learns a reward model from Human Preferences (the most standard Human-Based RL technique) and is directed at safety. Thus, one potential avenue for future research is to use the ReQueST framework (Section 6.1), (i.e. learning a dynamics model from expert safe trajectories, sampling trajectories from this model to generate queries for a human to provide feedback, and repeatedly improving the reward function before rendering it for traditional RL) but, instead of using Reward Shaping [25] or Reward Sketches [14] as the type of human feedback, to use human preferences. In that case, it would be able to exploit the safety benefits of justifications and hypothetical actions. In other words, safety will be encoded even better on the learnt reward function to be used for traditional RL at the last stage of ReQueST, and so more catastrophes will be avoided. We are planning to try this on the TIAGo robot [44] with a safety-critical environment we will set up, in order to train from raw pixels and tackle the generalisation problem. This idea has however some engineering challenges, such as who and for how long will teleoperate TIAGo to acquire the off-policy data (safe trajectories).

6.6 Current and future impact

In an outreach activity at the Southampton Science and Engineering Festival 2022, 60 participants, including both children and adults, successfully trained the Thymio robot [26] using the JPAL-HA model in a slightly smaller gridworld than that of the experiments. Similarly did nine high-school students aged between 16 and 19 during an educational trip at the University of Southampton. Participants could interact both via clicking on the laptop's interface or via natural language through a headset microphone. We observed an easy grasp of the training procedure from people of all ages and background knowledge and contributed to the public awareness of Human-in-the-loop AI, Human-Agent collaboration, AI Safety and other related notions.

RL is a very strong tool, but careless use of it can have unpredictable results. Accidents that have involved autonomous systems in the past have decreased the public's trust in them. All applications involving a learning agent for decision-making should take into account safety. The JPAL-HA framework, using a human in a standardised way to guarantee safety, points towards that direction of increasing the public's trust. Moreover, even though robots will most likely replace a lot of human jobs in the future, the human-agent interface we are proposing is directed at employing people to train agents efficiently. Apparently, people would be able to train intelligent agents with the only requirement to understand the environment and be able to answer preference and justification queries.

7 Conclusion

In the context of research which uses Human Preferences to train an agent safely, we proposed two novel ideas: the use of Justifications and Hypothetical Actions, which, combined into JPAL-HA, based on our evaluation improve safety significantly and consider bother cost. We focused on the Safe Exploration problem and demonstrated through experiments in a simulated environment that both the use of justifications and hypothetical actions significantly improves safety during the learning process. We showed that it does not affect the human burden under the assumption that justifications and hypothetical action queries place a marginal additional burden. In cases, where we do not make this assumption, such as in the user experiments with a mobile robot, the use of hypothetical actions still slightly reduces the human burden and also controls it to an admissible level, almost the same as if justifications and hypothetical actions had not been used at all.

Acknowledgments. We thank Jennifer Barnes-Nunn and Olaf Lipinski for their help with the outreach activities related to this research.

Declarations

Funding. This work was supported by the UK Research and Innovation Centre for Doctoral Training in Machine Intelligence for Nano-electronic Devices and Systems [EP/S024298/1].

Ethics approval. The user experiments involving human participants were ethically approved by University of Southampton Ethics and Research Governance Online (ERGO), with reference number ERGO/FEPS/69270.

Availability of data. All data supporting this study are openly available from the University of Southampton.

References

[1] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.*: Mastering the game of go with deep neural networks and tree search. *nature* **529**(7587), 484 (2016)

[2] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., *et al.*: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015)

[3] Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., *et al.*: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575**(7782), 350–354 (2019)

[4] Garcıa, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* **16**(1), 1437–1480 (2015)

[5] Achiam, J., Held, D., Tamar: Constrained policy optimization. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 22–31 (2017). JMLR. org

[6] Chow, Y., Nachum, O., Duenez-Guzman, E., Ghavamzadeh, M.: A Lyapunov-based approach to safe reinforcement learning. In: Advances in Neural Information Processing Systems, pp. 8092–8101 (2018)

[7] Cheng, R., Orosz, G., Murray, R.M., Burdick, J.W.: End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3387–3395 (2019)

[8] Turchetta, M., Kolobov, A., Shah, S., Krause, A., Agarwal, A.: Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems* **33**, 12151–12162 (2020)

[9] Gasparini, L., Norman, T.J., Kollingbaum, M.J.: Severity-sensitive norm-governed multi-agent planning. *Autonomous Agents and Multi-Agent Systems* **32**(1), 26–58 (2018)

[10] Frye, C., Feige, I.: Parenting: Safe reinforcement learning from human input. *arXiv preprint arXiv:1902.06766* (2019)

[11] Kazantzidis, I., Norman, T.J., Du, Y., Freeman, C.T.: How to train your agent: Active learning from human preferences and justifications in safety-critical environments. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pp. 1654–1656 (2022)

[12] Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., Legg, S.: Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871* (2018)

[13] Leike, J., Martic, M., Krakovna, V., Ortega, P.A., Everitt, T., Lefrancq, A., Orseau, L., Legg, S.: AI safety gridworlds. *arXiv preprint arXiv:1711.09883* (2017)

[14] Rahtz, M., Varma, V., Kumar, R., Kenton, Z., Legg, S., Leike, J.: Safe deep RL in 3D environments using human feedback. *arXiv preprint arXiv:2201.08102* (2022)

[15] Peschl, M., Zgonnikov, A., Oliehoek, F.A., Siebert, L.C.: Moral: Aligning AI with human norms through multi-objective reinforced active learning. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pp. 1038–1046 (2022)

[16] Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* **39**(3/4), 324–345 (1952)

[17] Watkins, C.J., Dayan, P.: Q-learning. *Machine learning* **8**(3-4), 279–292 (1992)

[18] Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. *Citeseer* **37** (1994)

[19] Brown, M.B., Forsythe, A.B.: Robust tests for the equality of variances. *Journal of the American Statistical Association* **69**(346), 364–367 (1974)

[20] Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). *Biometrika* **52**(3/4), 591–611 (1965)

[21] Dunn, O.J.: Multiple comparisons using rank sums. *Technometrics* **6**(3), 241–252 (1964)

[22] Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 50–60 (1947)

[23] Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 65–70 (1979)

[24] Saunders, W., Sastry, G., Stuhlmueller, A.: Trial without error: Towards safe reinforcement learning via human intervention. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2067–2069 (2018). International Foundation for Autonomous Agents and Multiagent Systems

[25] Reddy, S., Dragan, A., Levine, S., Legg, S., Leike, J.: Learning human objectives by evaluating hypothetical behavior. In: International Conference on Machine Learning, pp. 8020–8029 (2020). PMLR

[26] MOBOTS group: Thymio mobile robot. URL: https://www.thymio.org/ (2022)

[27] Guo, Z.., Norman, T.J., Gerding, E.H.: MTIRL: Multi-trainer interactive reinforcement learning system. In: Proceedings of the International Conference on Principles and Practice of Multi-Agent Systems, pp. 227–242 (2023)

[28] Hussein, A., Gaber, M.M., Elyan, E., Jayne, C.: Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* **50**(2), 1–35 (2017)

[29] Ng, A.Y., Russell, S.J., *et al.*: Algorithms for inverse reinforcement learning. In: ICML, vol. 1, p. 2 (2000)

[30] Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the Twenty-first International Conference on Machine Learning, p. 1 (2004)

[31] Brown, D.S., Cui, Y., Niekum, S.: Risk-aware active inverse reinforcement learning. In: Conference on Robot Learning, pp. 362–372 (2018). PMLR

[32] Warnell, G., Waytowich, N., Lawhern, V., Stone, P.: Deep tamer: Interactive agent shaping in high-dimensional state spaces. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[33] Arakawa, R., Kobayashi, S., Unno, Y., Tsuboi, Y., Maeda, S.-i.: DQN-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748* (2018)

[34] Sheidlower, I., Short, E.S., Moore, A.: Environment guided interactive reinforcement learning: Learning from binary feedback in high-dimensional robot task environments. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pp. 1726–1728 (2022)

[35] Cabi, S., Colmenarejo, S.G., Novikov, A., Konyushkova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., et al.: Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200* (2019)

[36] Goecks, V.G., Gremillion, G.M., Lawhern, V.J., Valasek, J., Waytowich, N.R.: Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 2462–2470 (2019)

[37] Settles, B.: Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences (2009)

[38] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Advances in Neural Information Processing Systems, pp. 4299–4307 (2017)

[39] Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., Amodei, D.: Reward learning from human preferences and demonstrations in atari. In: Advances in Neural Information Processing Systems, pp. 8011–8023 (2018)

[40] Lee, K., Smith, L., Abbeel, P.: Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In: International Conference on Machine Learning (2021)

[41] Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., Lee, K.: Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In: International Conference on Learning Representations (2021)

[42] Liang, X., Shu, K., Lee, K., Abbeel, P.: Reward uncertainty for exploration in preference-based reinforcement learning. In: International Conference on Learning Representations (2021)

[43] Liu, R., Bai, F., Du, Y., Yang, Y.: Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. In:

Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022). https://openreview.net/forum?id=OZKBReUF-wX

[44] PAL Robotics: TIAGo robot. URL: https://pal-robotics.com/robots/tiago/ (2022)

Figures



Figure 1



Figure 2



Figure 3



Figure 4



Figure 5



Figure 6



Figure 7



Figure 8



Figure 9



Figure 10



Figure 11



Figure 12



Figure 13



Figure 14



Figure 15



Figure 16



Figure 17



Figure 18

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- AnonymisedUserData.zip