



Quantifying the effect of feedback frequency in interactive reinforcement learning for robotic tasks

Daniel Harnack¹ · Julie Pivin-Bachler² · Nicolás Navarro-Guerrero¹

Received: 2 March 2022 / Accepted: 12 October 2022

© The Author(s) 2022

Abstract

Reinforcement learning (RL) has become widely adopted in robot control. Despite many successes, one major persisting problem can be very low data efficiency. One solution is interactive feedback, which has been shown to speed up RL considerably. As a result, there is an abundance of different strategies, which are, however, primarily tested on discrete grid-world and small scale optimal control scenarios. In the literature, there is no consensus about which feedback frequency is optimal or at which time the feedback is most beneficial. To resolve these discrepancies we isolate and quantify the effect of feedback frequency in robotic tasks with continuous state and action spaces. The experiments encompass inverse kinematics learning for robotic manipulator arms of different complexity. We show that seemingly contradictory reported phenomena occur at different complexity levels. Furthermore, our results suggest that no single ideal feedback frequency exists. Rather that feedback frequency should be changed as the agent's proficiency in the task increases.

Keywords Interactive reinforcement learning · Human-aligned reinforcement learning · Guided exploration · Intrinsic feedback homology

1 Introduction

Reinforcement Learning (RL) has become widely used in modern robotic technologies. One reason is the compelling simplicity and generality of the framework. In short, the behavior an agent is expected to learn is encoded by a reward function. Through interaction with the

environment, the agent will learn to maximize the reward by performing actions that have proven to be beneficial. However, this seeming simplicity has many pitfalls and subtleties. One common shortcoming of most algorithms is the very low data efficiency: complex problems might require millions of agent-environment interactions to be solved [1].

One strategy to accelerate this procedure is interactive reinforcement learning (IRL) [2]. Interaction augments the sources of information provided to the learning agents by teacher feedback, which can be a human or another type of agent [3]. In the latter case, it is also known as the *agents teaching agents* subfield of transfer learning [4]. There are numerous alternatives to implement IRL as described by Arzate Cruz and Igarashi [2]. A graphical overview is provided in Fig. 1. Firstly, the teacher feedback can be classified into critique (binary), scalar values, action advice and guidance. Further, this feedback can be used to modify different aspects of the learning model, i.e., the reward function (reward shaping [5]), the policy (policy-shaping

✉ Nicolás Navarro-Guerrero
nicolas.navarro@dfki.de

Daniel Harnack
daniel.harnack@dfki.de

Julie Pivin-Bachler
julie.pivin-bachler@univ-tlse3.fr

¹ Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH, Robert-Hooke-Straße 1, 28359 Bremen, Bremen, Germany

² Robotics and Interactive Systems – UPSSITECH, University Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse, Occitanie, France

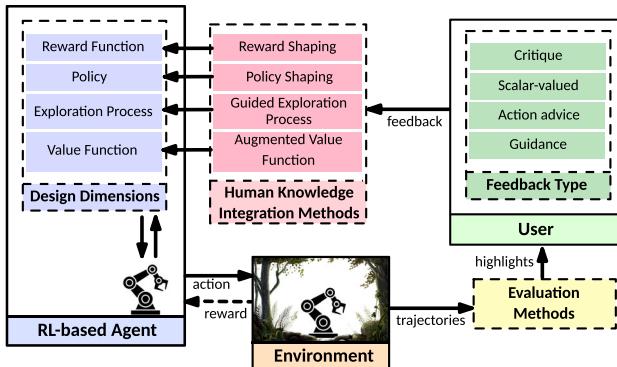


Fig. 1 Summary of alternative implementations of interactive reinforcement learning. Adapted from [2]

[6]), the exploration process (guided exploration process [7]), and the value function (augmented value function).

The literature on interactive reinforcement learning is extensive, such that many combinations shown in Fig. 1 have been explored already. One consistent result is that interaction in any form can perform better than vanilla RL agents. However, it is still unclear how much the different aspects contribute to overall performance gains or how different feedback strategies can be combined, and in which proportions, to optimize users' experience, agents' task performance, or both.

IRL algorithms are mainly developed with human teachers in mind. Empirical evidence indicates that people are strongly biased to use evaluative feedback communicatively rather than as reinforcement [8, 9]. In other words, humans use *evaluative feedback as communication* as a policy-shaping strategy rather than reward shaping. Arguably, this type of feedback favors IRL strategies based on *policy shaping* and *guided exploration*, because it would make the interaction both more engaging for the teacher and more effective for the learning agent [6, 9, 10]. Results from Ho et al. [8], showing that people consistently use feedback communicatively even when interacting with reward-maximizing agents, further support this claim. Moreover, using feedback signals as rewards and punishments when interacting with reward-maximizing RL agents can lead to *reward hacking* [11]. Reward hacking is a consequence of misspecified reward functions, which lead to undesired behaviors, such as when action sequences leading to the reward from the human are repeated at the expense of learning to complete the task more generally.

Despite efforts to improve the study of interaction on RL agents with human teachers, we believe there is still much to be learned using simulated teachers and oracles as suggested by Bignold et al. [12]. Moreover, human feedback varies in accuracy, availability, concept drift, reward bias, cognitive bias, knowledge level, latency, etc. [12],

which makes it very challenging to isolate the effects of different interaction strategies in reinforcement learning agents. Fortunately, pre-trained agents or hard-coded heuristics can be used as feedback sources without requiring modifications to the learning algorithms. These types of ‘teachers’ are primarily used in theoretical research since it allows for better controllable and more easily implementable experiments.

Different strategies have been compared regarding policy-shaping, such as early advising, alternating or stochastic advising, importance advising, and mistake correction. Mistake correction consistently outperformed the other methods, both in discrete [13, 14] and continuous state and action spaces [13]. Taylor et al. [13] also noted that mistake correction is more robust to changes in feedback quality than alternate advising. In addition, mistake correction and predictive advising are most robust to changes in the state representation between teacher and agent. However, as noted by Cruz et al. [14], mistake correction in policy shaping would be the most difficult strategy to implement in real-world scenarios with human teachers since it requires the teacher to detect the mistake, revert it, and suggest a better alternative action.

A more straightforward way is using mistake correction for guided exploration. Here, the teacher must detect and revert the error but not necessarily suggest a better action. In addition, despite its popularity, we believe that policy-shaping strategies might hinder the learning of robust policies by reducing exploration, which leads to good performance primarily in the neighborhood of the behaviors demonstrated by the teacher [15]. Limiting the exploration in this manner can result in poor performance in other areas of the state space not or rarely encountered during training [15].

In the literature on feedback-guided exploration, Stahlhut et al. [7, 16] found that mistake correction does not help to increase the learning speed in simple tasks but only starts to have a measurable effect as the task complexity increases. It was also observed that higher feedback frequencies lead to more robust agents, i.e., that the average agents' performance for the same hyperparameters has a smaller standard deviation across different random seeds. Moreover, feedback can offset the detrimental effect of poorly tuned hyperparameters as a byproduct of this increased robustness. This effect becomes stronger as feedback frequency increases, regardless of the complexity of the problem. Stahlhut et al. also speculate that interaction has a more significant impact during early learning. In addition, they hypothesize that feedback may be indispensable to achieve sufficient performance in very complex tasks, in agreement with Suay and Chernova's hypothesis [15].

Millán-Arias et al. [17] further investigated the effect of feedback frequency used to guide the exploration process. They observed that too much feedback might lead to delayed onset of learning. Despite that, the performance of highly interactive agents converges at the same time as the performance of less interactive ones. In addition, they speculate that too much binary advice, even if 100% correct, can be counterproductive and slow down learning, particularly in noisy environments. They conclude that intermediate interaction frequencies are optimal.

Summarizing previous findings, there are strikingly contradictory accounts of the optimal choice of feedback frequencies. At the same time, some authors suggest that more feedback is better [7, 16], while others indicate that the cost of high feedback frequencies does not justify the gains [12]. In contrast, others advocate for intermediate feedback frequencies and report even detrimental effects of frequent feedback [17]. It was also suggested that the feedback frequency should not be stationary but adjusted as the agents' proficiency increases during training [7, 16, 18].

We assume that the reason for the disagreement is a lack of knowledge about the differential effects of varying feedback frequencies at different levels of agent proficiency and task complexity. Consolidating these previous findings without further experimentation is complicated since most results only show cumulated reward or sequence length. Moreover, effect sizes, average performance, and statistical analyses are not reported in most cases. Also, the complexity of the setup might make it impossible to isolate the effects of the different parameters used [18]. In addition, the most common testbeds in IRL research are grid worlds and other low-dimensional discrete state and action spaces. Whereas the small size of these testbeds allows for short experiments, we believe results obtained in those testbeds might not generalize well to more complex problems with real-world implications [2].

Thus, in this paper, we aim to isolate and quantify the effect of feedback frequency on learning performance for different task complexities and agent proficiencies, to shed some light on seemingly contradictory results. As testbeds, we use robotic tasks of varying complexity and continuous action and state spaces. We focus on feedback as mistake correction to guide the exploration process since it does not demand expert knowledge of the task. In our experiments, we reproduce various seemingly contradictory findings about the optimal choice of interaction frequencies and relate them to a differential effect of the teacher interaction on task complexity. We also show that optimal feedback frequencies typically exhibit temporal drifts, making it difficult to recommend a single range of feedback frequencies for any task. We instead conclude, in line with previous suggestions [7, 16, 18], that an adaptive interaction regime, which changes with the

agents' proficiency, is likely optimal. Finally, we discuss a simple heuristic for choosing a close-to-optimal temporal trajectory for the interaction rate.

2 Methods

This section details all experimental and analysis methods used in the paper.

2.1 Environment

Inspired by Stahlhut et al. [7, 16], we study the effect of feedback frequency as exploration guidance in an inverse kinematic learning task. The environment dynamics were implemented by the forward kinematics models of the NAO and KUKA (LBR iiwa 14 R820) robots.

For the NAO robot, the forward kinematics model described by Kofinas et al. [19] was used. Two conditions for the NAO robot were defined: a 2 degrees of freedom (DoFs), and a 4 DoFs condition. For the 2 DoFs configuration, the elbow and shoulder roll were actuated. In the 4 DoFs condition, all four joints are used, i.e., shoulder pitch, shoulder roll, elbow yaw and elbow roll.

The KUKA LBR iiwa 14 R820 kinematics were simulated with the model described by Busson et al. [20]. For the KUKA arm, three conditions were studied, i.e., 2 DoFs, 4 DoFs, and 7 DoFs conditions. For the 2 DoFs configuration, joints 2 and 4 were actuated while keeping the other joints in their respective zero-position. For the 4 DoFs configuration, the first four joints were actuated while keeping the other joints in their respective zero-position, and all 7 joints were actuated in the 7 DoFs condition.

The 2 DoFs models of the NAO and KUKA are used to study the role of feedback frequency in two-dimensional task spaces. In addition, The 4 DoFs and 7 DoFs conditions of NAO and KUKA are examples of more complex three-dimensional task spaces.

2.2 Task and reward

All experiments aim to generate controllers that can reach arbitrary goal zones in task space while controlling the robot arms in joint space. A sparse reward function is used, i.e., reaching the goal zone leads to a reward of 1. All other actions result in a reward of 0. Such a reward function allows us to isolate the effect of feedback and analyze the learning dynamics more easily. Moreover, adding other rewards signals, such as punishment signals, might have a detrimental effect on learning speed [21, 22], which makes both analysis and design of the reward function difficult.

The *Goal Zone Radius* (GZR) for both NAO configurations is 17.5 and 150 mm for the KUKA arm conditions.

2.3 Interactive RL setup

We use the Continuous Actor-Critic Learning Automaton (CACLA) [23] as the underlying reinforcement learning framework. The agent has an Ask Likelihood (\mathcal{L}) parameter, representing the likelihood of the agent asking/receiving guidance from the teacher. The teacher judges the agent's last action based on the Euclidean distance between the end-effector and the goal. However, the feedback to the agent is binary, and it is only used to guide the exploration process and not as an additional reward or to shape the policy directly. In particular, if the last action increases the distance to the goal, it is considered a mistake. When the teacher reports a mistake, the agent undoes the action and explores a new one, after which the cycle is repeated. This process is illustrated in Fig. 2.

The stochastic feedback strategy used here is a good analogy for teachers taking breaks when providing feedback. Although this type of feedback is easy to automate, it might not always be correct. For instance, in the presence of obstacles or complex task spaces, it might be required first to move away from the goal position to reach it. Thus, \mathcal{L} cannot be equal to 1. To prevent the agent from potentially getting stuck, we keep the maximum value of 0.99 used by Stahlhut et al. [7, 16].

2.4 State and action spaces

The state space \mathcal{S} for all conditions consists of the corresponding joint positions (proprioception) and the Cartesian coordinates of the target position (exteroception). For all conditions, the action space is limited to the maximal allowed joint displacement per time step of $\pi/10$. I.e. $\mathcal{A} = [-\pi/10, \pi/10]^{DoF}$.

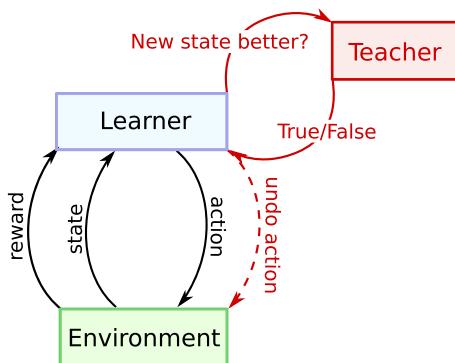


Fig. 2 Diagram representing the basic RL setup, with the interactive components colored in red

2.5 The controller

The Actor and Critic are implemented with two separate multilayer perceptrons (MLPs). The networks share the same input vector. However, the networks are tuned separately using hyperparameter optimization as described in Sect. 2.9. Thus, the learning rate, number of hidden layers and outputs, may differ between the Actor and the Critic. All input and output values are scaled to the range $[-1, 1]$. The activation function for the output units is linear. The networks were implemented in PyTorch 1.10.0 and trained using Adam [24].

2.6 Episodes

Based on both the maximum range of motion of the joints and the maximal action size, the smallest number of steps needed to traverse the entire task space was computed as follows:

$$\text{Steps}_{\min} = \frac{\text{Range}_{\max}}{\text{Action}_{\max}}$$

Steps_{\min} was then used to define the episode length Steps_{\max} as $3 \times \text{Steps}_{\min}$ rounded to the next tenth.

The minimum number of goals zones G_{\min} to cover the entire workspace was used to determine the number of episodes N_{train} per epoch. G_{\min} was calculated using optimal disc (2D task space) or ball (3D task space) packing in the task space volume. N_{train} results from $10 \times G_{\min}$ rounded to the second leading digit. Table 1 shows a summary of the episode parameters conditions.

2.7 Performance metrics

The following metrics were used to quantify the effect of feedback frequency. Lower values indicate better performance:

Table 1 Boundary conditions for the episodes. Steps_{\min} : min. # of steps to cover the task space, Steps_{\max} : max. # of steps the agent can take to reach the target, G_{\min} : min. # of goals to cover the task space and N_{train} : # of episodes per epoch

	NAO		KUKA		
	2 DoFs	4 DoFs	2 DoFs	4 DoFs	7 DoFs
GZR	17.5 mm	17.5 mm	150 mm	150 mm	150 mm
Steps_{\min}	6	13	19	19	19
Steps_{\max}	20	40	60	60	60
G_{\min}	22	224	38	261	261
N_{train}	220	2200	380	2600	2600

- *Positioning error*: mean Euclidean distance to the target divided by the target radius.
- *Failure rate*: percentage of missed targets during evaluation, which is equivalent to
 $\text{Failure rate} = 1 - \text{average cumulative reward}$.

The performance learning curves are analyzed with respect to the cumulative steps instead of epochs since it better reflects the total amount of interaction with the environment. The slope of the failure rate is used as an indicator of the *improvement speed*.

We furthermore consider thresholded performance profiles. Note that for visual clarity, the error bars represent the standard error of the mean and not a confidence interval. Here, the steps of the \mathcal{L} agent reaching the corresponding *failure rate* threshold first are used as reference. Then the best *failure rate* up to this step count is compared across different \mathcal{L} values. For this analysis, we used a two-sided Wilcoxon rank sum test with respect to the $\mathcal{L} = 0.0$ condition. This temporal threshold strategy copes better when dealing with conditions that cannot achieve an arbitrary performance threshold than the *time to threshold* [25] strategy.

2.8 Datasets

Following similar practices as in supervised learning, three datasets were used: a training, a validation, and a test set. Both the training set and test set are of the same size N_{train} , while the validation set is 1/5 the size of the training set, see Table 1. The datasets consist of pairs of initial positions for the agent and the target. These positions are generated randomly from a uniform distribution in joint space. The targets are represented in Cartesian coordinates and result from feeding random joint configurations into the forward kinematics model of the corresponding robotic arm. Target coordinates that lie within the goal zone of the corresponding initial position are rejected. One epoch is defined as training for all pairs in the training set once.

2.9 Hyperparameter optimization

Hyperopt [26] was used to determine the best hyperparameters out of 100 hyperparameter sets for each experimental condition. Preliminary tests showed signs of significant performance improvement by the 10th (2 DoFs) or 20th (4 DoFs and 7 DoFs) epoch. Thus, during hyperparameter selection, the 2 DoFs conditions agents were trained for 10 epochs while the 4 DoFs and 7 DoFs conditions were trained for 20 epochs. In all cases, we used the corresponding training set. The best hyperparameters set

was selected based on the lowest *positioning error* in the validation set at the respective final epoch.

Prior tests showed that optimizing for the lowest *positioning error* or fastest convergence speed leads to similar results. In real-world scenarios, it is arguably more important to have the robotic arm performing the defined task precisely – with minimal possible error – than learn to perform quickly but with low repeatability or precision. Thus, here the hyperparameters were optimized for the lowest *positioning error*.

The hyperparameter search can be done at least in two manners: 1) optimizing the hyperparameters for each Ask Likelihood (\mathcal{L}) value independently, or 2) optimizing the hyperparameters only for $\mathcal{L} = 0.0$ and evaluating the performance for increasing values of \mathcal{L} . The latter was selected for two reasons. Firstly, hyperparameter optimization is computationally expensive. Secondly, this strategy allows for quantifying the gain of using a particular feedback frequency in an existing system of vanilla RL ($\mathcal{L} = 0.0$).

Table 2 summarizes the hyperparameters and optimization boundaries. The last five hyperparameters corresponding to the neural network configuration are optimized independently for the Actor and Critic, but share the same ranges.

2.10 Training and testing

The agents are trained on the same training set used for the hyperparameter optimization, but this time the agents' performance is evaluated on the test set. All agents are trained for a fixed number of epochs. Eleven agent versions for each condition were trained, including the baseline agents for $\mathcal{L} = 0.0$ (non-interactive) and ten other agents sets with \mathcal{L} values increasing in increments of 0.1, with the

Table 2 List of hyperparameters to be optimized and their interval of possible values

Hyperparameter	Range
Actor learning rate	$[10^{-4}, 10^{-2}]$
Critic learning rate	$[10^{-4}, 10^{-2}]$
Exploration rate	$[0.2, 0.9]$
Discount factor	$[0.75, 1.0]$
Zeta	$[10^{-4}, 10^{-1.3}]$
Initial variance	$[1.0, 3.0]$
# of hidden layers	1, 2 or 3
# of neurons on 1st layer	$[10, 100]$ step of 10
# of Neurons on 2nd layer	$[5, 100]$ step of 5
# of Neurons on 3rd layer	$[5, 100]$ step of 5
Activation function	ReLU, SeLu, or Softplus

last agent having $\mathcal{L} = 0.99$ (fully interactive). The \mathcal{L} values do not change during training. During testing, no learning occurs, no interaction is possible, and no undo action is performed. Statistics are taken over 20 randomly initialized agents for each task and each \mathcal{L} value.

3 Results

In the following section, we present the temporal evolution of the *failure rate* on the test sets for all experimental conditions. The *failure rate* value at each point is the average over the 20 randomly initialized agents.

In addition, for each experimental condition, we compare the *failure rate* for different \mathcal{L} values at various threshold levels to assess the optimal \mathcal{L} value at different stages in training.

3.1 NAO 2 DoFs Experiment

Figure 3a shows the performance evolution for the NAO 2 DoFs condition. The circles indicate the best *failure rate* value achieved during training for each \mathcal{L} value. The *failure rate* continuously improves with higher \mathcal{L} values, even reaching perfect performance in late training for $\mathcal{L} \geq 0.7$, whereas the performance for low \mathcal{L} starts to diverge around 10^4 steps. The *failure rate curves' rate of improvement* is comparable for all \mathcal{L} values during the first few epochs.

Figure 3b shows the thresholded *failure rate* performances in the NAO 2 DoFs condition. The cumulative step counts corresponding to the thresholds are marked by blue arrows in Fig. 3a. At the first threshold levels, there is a trend toward a significant lower *failure rate* as \mathcal{L} values increase. However, the behavior is dynamic, and no single \mathcal{L} value remains the best through training. Within the tested time horizon, the final performance favors the highest \mathcal{L} values.

3.2 NAO 4 DoFs experiment

Figure 4a shows the performance evolution for NAO 4 DoFs condition. Values up to $\mathcal{L} = 0.7$ convergence to a similar value. In contrast, the best performance of higher \mathcal{L} is reached earlier, after which the *failure rates* start to diverge slowly.

The *improvement speed* is initially faster for higher \mathcal{L} values before they start to diverge.

Figure 4b shows the corresponding time thresholded *failure rate* analysis. Here the highest effect on the *failure rates* is observed in the first half of training at very high \mathcal{L} values. In the later phase of training, at the 5% threshold, the optimal \mathcal{L} shifts toward medium and high \mathcal{L} values.

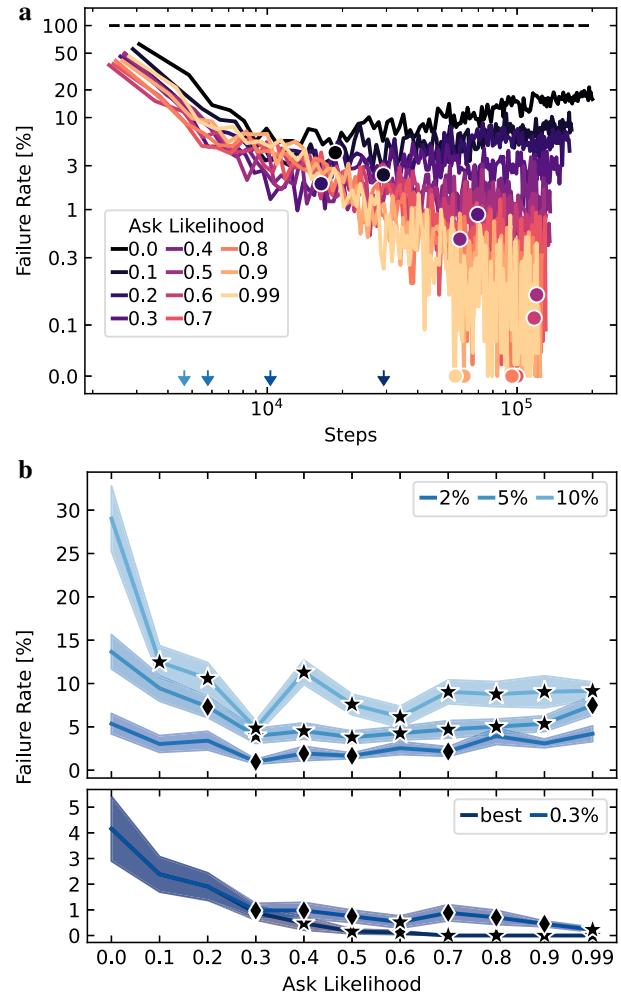


Fig. 3 **a** Failure rate evolution for the NAO 2 DoFs experiment in log scale. The circle indicates the best performance for the corresponding \mathcal{L} value for the whole training. The blue arrows show the number of environment steps needed for the fastest \mathcal{L} agents to reach 10, 5, 2 and 0.3% failure rate. **b** Time thresholded failure rates for the NAO 2 DoFs condition. Statistical significance with respect to the $\mathcal{L} = 0.0$ condition was computed using a two-sided Wilcoxon rank sum test. The \blacklozenge show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the \star show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$

Finally, a significant benefit is mostly absent for the strictest threshold of 3% and beyond. However, for $\mathcal{L} \geq 0.9$, the effect is significantly detrimental, as indicated by the red markers in Fig. 4b.

3.3 KUKA 2 DoFs experiment

Figure 5a shows the performance evolution for the KUKA 2 DoFs condition. The overall *failure rate* in this condition is relatively high. However, a similar trend to that of the NAO 2 DoFs condition can be observed, i.e., the *failure rate* continuously improves with a higher \mathcal{L} value. In

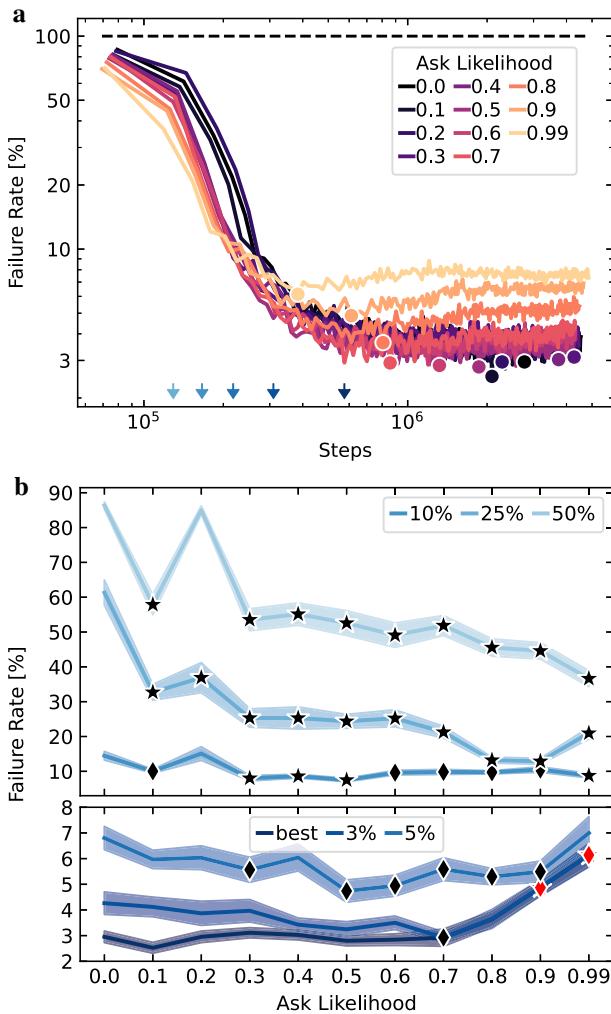


Fig. 4 **a** Failure rate evolution for the NAO 4 DoFs experiment shown in log scale. The circle indicates the best performance for the corresponding \mathcal{L} value. The blue arrows show the number of environment steps needed for the fastest \mathcal{L} agents to reach 50, 25, 10, 5 and 3% failure rate. **b** Time thresholded Failure rates for NAO 4 DoFs. Statistical significance was computed using a two-sided Wilcoxon rank sum test. The ♦ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the ★ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$. Red markers indicate significantly detrimental effects

contrast, the performance for low \mathcal{L} values starts to diverge after the respective best performance is reached.

The *improvement speed* for low to medium \mathcal{L} agents is initially higher. Whereas the *improvement speed* for higher \mathcal{L} values is slightly lower, it is maintained almost constantly throughout the tested time horizon.

Figure 5b shows the time thresholded *failure rate* for the KUKA 2 DoFs condition. Unlike in both NAO conditions, here, early in training, the interaction does not yield any benefits. Interaction is even significantly detrimental for very high \mathcal{L} values. Only toward the end of training does interaction significantly reduce the *failure rate*.

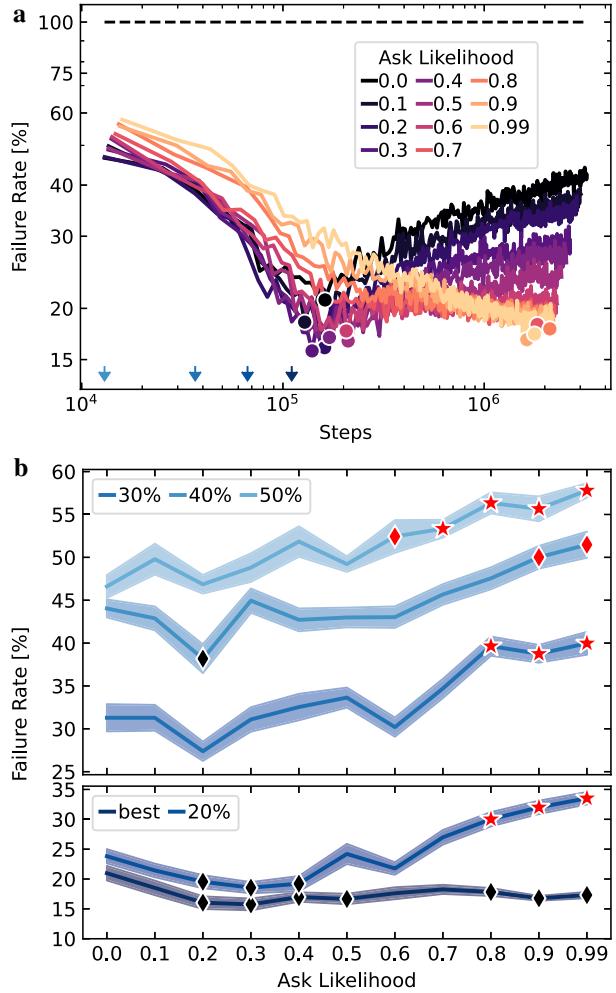


Fig. 5 **a** Failure rate evolution for the KUKA 2 DoFs experiment shown in log scale. The circle indicates the best performance for the corresponding \mathcal{L} value. The blue arrows show the number of environment steps needed for the fastest \mathcal{L} agents to reach 50, 40, 30 and 20% failure rate. **b** Time thresholded failure rates for the KUKA 2 DoFs condition. Statistical significance was computed using a two-sided Wilcoxon rank sum test. The ♦ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the ★ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$. Red markers indicate significantly detrimental effects

3.4 KUKA 4 DoFs experiment

Figure 6a shows the performance evolution for the KUKA 4 DoFs condition. Again, the overall *failure rate* in this condition is relatively high. In contrast to all other experimental conditions, all \mathcal{L} values lead to a monotonically improving *failure rate*. Lower \mathcal{L} values initially show a faster rate of improvement but slow down as learning progresses. In contrast, very high \mathcal{L} values display a lower rate of improvement, which is, however, almost constant throughout the tested time horizon.

Figure 6b shows the time thresholded *failure rate* for the KUKA 4 DoFs condition. Early in training, interaction

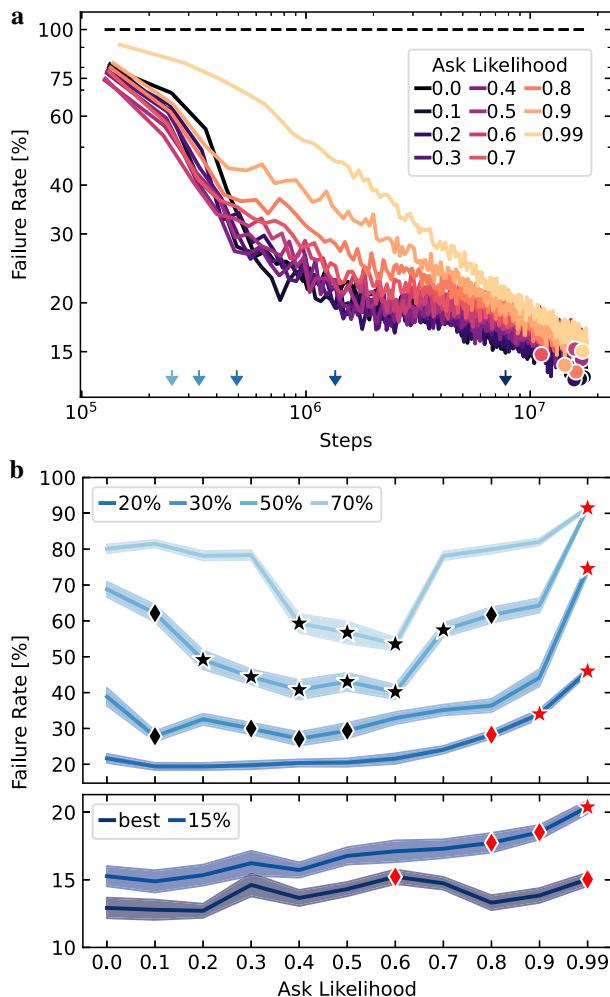


Fig. 6 **a** Failure rate evolution for the KUKA 4 DoFs experiment shown in log scale. The circle indicates the best performance for the corresponding \mathcal{L} value. The blue arrows show the number of environment steps needed for the fastest \mathcal{L} agents to reach 75%, 50%, 30%, 20%, and 15% failure rate. **b** Time thresholded failure rates for the KUKA 4 DoFs condition. Statistical significance was computed using a two-sided Wilcoxon rank sum test. The \blacklozenge show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the \star show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$. Red markers indicate a significantly detrimental effect

leads to a significant reduction in *failure rate* primarily for intermediate \mathcal{L} values. In contrast, the highest \mathcal{L} values have a significantly detrimental effect on performance throughout the tested time horizon. However, from the data, it cannot be judged what the very long-term behavior of the agents will be since the performance has not converged.

3.5 KUKA 7 DoFs experiment

Figure 7a shows the performance evolution for the KUKA 7 DoFs condition. As for the 2 DoFs conditions, low to medium \mathcal{L} agents reach a local optimum around 2×10^6

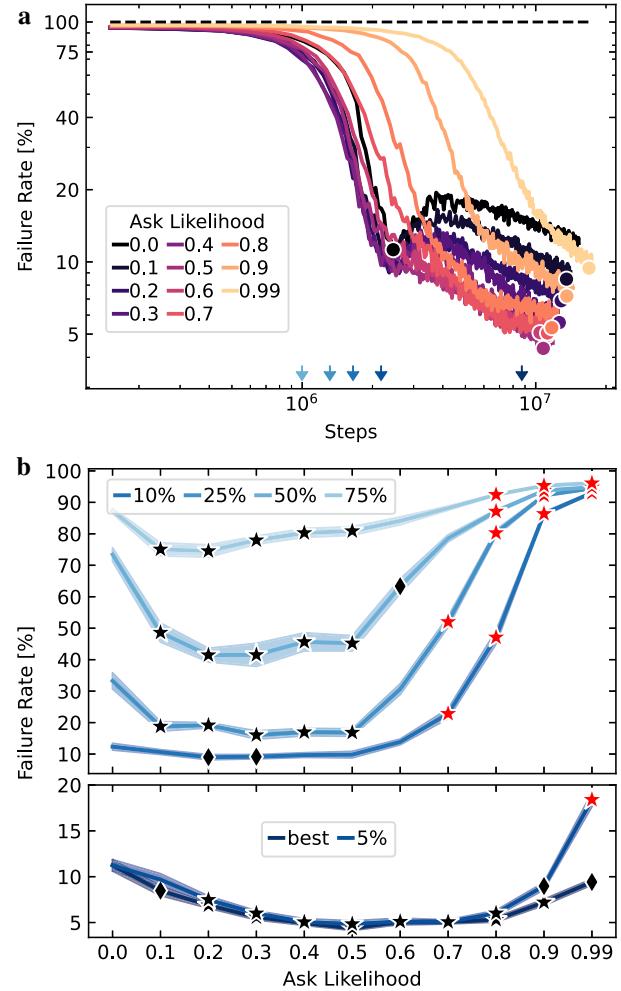


Fig. 7 **a** Failure rate evolution for the KUKA 7 DoFs experiment shown in log scale. The circle indicates the best performance for the corresponding \mathcal{L} value. The blue arrows show the number of environment steps needed for the fastest \mathcal{L} agents to reach 75%, 50%, 25%, 10 and 5% failure rate. **b** Time thresholded failure rates for the KUKA 7 DoFs condition. Statistical significance was computed using a two-sided Wilcoxon rank sum test. The \blacklozenge show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the \star show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$. Red markers indicate a significantly detrimental effect

steps, after which the performance temporarily deteriorates. However, in contrast to the 2 DoFs conditions, the performance continues to improve beyond the initial local optimum. Higher \mathcal{L} agents have a lower rate of improvement but do not experience any divergent behavior, at least within the tested time horizon.

Figure 7b shows the time thresholded *failure rate* for the KUKA 7 DoFs condition. Similarly, as in the previous KUKA conditions, low to medium \mathcal{L} values lead to significantly better performance early in training. In contrast, very high \mathcal{L} values lead to significantly worse *failure rates*. The detrimental effect becomes stronger the higher the \mathcal{L} value. Only for longer training horizons do high \mathcal{L} values

start to become significantly beneficial, albeit not optimal. The long-term behavior cannot be clearly judged here since the performance has not converged.

Table 3 shows a combined summary of the statistical analyses of the effects on the *failure rate* thresholds for all tested robotic tasks and \mathcal{L} values. The effect size reported is the difference of means.

4 Discussion

Our thorough investigation of the Ask Likelihood's effect on the evolution of the *failure rate* over time allows us to make more nuanced statements on task-dependent effects of the interaction rate than previously reported. Furthermore, our experiments can unify seemingly contradictory statements on the best choice of \mathcal{L} . In summary, across the different experimental conditions, we make three main observations: 1) *policy robustness*, 2) *optimal long-term \mathcal{L}* , and 3) *optimal \mathcal{L} trajectory*.

Table 3 Effect Size for different values of \mathcal{L} across robot tasks, computed as difference of means.

	Failure Rate	Ask Likelihood									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
NAO 2 DoFs	10	-16.55★	-18.45★	-24.18★	-17.73★	-21.43★	-22.86★	-19.98★	-20.25★	-19.98★	-19.86★
	5	-	-6.34♦	-9.70★	-9.14★	-9.86★	-9.39★	-8.95★	-8.64★	-8.30★	-6.16♦
	2	-	-	-4.39♦	-3.43♦	-3.70♦	-	-3.20♦	-	-	-
	0.3	-	-	-3.18♦	-3.18♦	-3.41♦	-3.64★	-3.27♦	-3.45♦	-3.70♦	-3.93★
	best	-	-	-3.27♦	-3.68★	-4.00★	-4.05★	-4.16★	-4.16★	-4.16★	-4.16★
NAO 4 DoFs	50	-28.51★	-	-32.84★	-31.23★	-33.80★	-37.29★	-34.5★	-40.82★	-41.78★	-49.75★
	25	-28.61★	-24.42★	-36.00★	-36.01★	-36.94★	-36.09★	-40.06★	-48.10★	-48.43★	-40.36★
	10	-4.46♦	-	-6.45★	-5.87★	-6.96★	-4.86♦	-4.65♦	-4.75♦	-3.97♦	-5.71★
	5	-	-	-1.24♦	-	-2.07♦	-1.86♦	-1.21♦	-1.51♦	-1.31♦	-
	3	-	-	-	-	-	-	-1.34♦	-	0.60♦	1.88♦
KUKA 2 DoFs	best	-	-	-	-	-	-	-	-	1.92★	3.19★
	50	-	-	-	-	-	5.83♦	6.74★	9.74★	9.05★	11.20★
	40	-	-5.86♦	-	-	-	-	-	-	5.97♦	7.43♦
	30	-	-	-	-	-	-	-	8.36★	7.46★	8.68★
	20	-	-4.26♦	-5.26♦	-4.59♦	-	-	-	6.20★	8.18★	9.71★
KUKA 4 DoFs	best	-	-4.93♦	-5.22♦	-3.99♦	-4.32♦	-	-	-3.14♦	-4.20♦	-3.66♦
	70	-	-	-	-20.78★	-23.34★	-26.56★	-	-	-	11.40★
	50	-6.68♦	-19.67★	-24.42★	-28.04★	-25.79★	-28.63★	-11.38★	-7.19♦	-	22.68★
	30	-11.05♦	-	-8.87♦	-11.76♦	-9.46♦	-	-	-	-	35.71★
	20	-	-	-	-	-	-	-	6.60♦	12.35★	24.30★
KUKA 7 DoFs	15	-	-	-	-	-	-	-	2.47♦	3.25♦	5.12★
	best	-	-	-	-	-	2.31♦	-	-	-	2.11♦
	75	-12.23★	-12.76★	-9.33★	-7.02★	-6.42★	-	-	5.12★	8.01★	8.80★
	50	-24.79★	-31.95★	-31.89★	-27.76★	-28.21★	-10.08♦	-	13.71★	20.35★	21.39★
	25	-14.47★	-14.08★	-17.21★	-16.28★	-16.41★	-	18.76★	47.02★	58.95★	61.16★
KUKA 10 DoFs	10	-	-3.33♦	-3.18♦	-	-	-	10.48★	34.77★	74.01★	80.70★
	5	-	-3.76★	-5.23★	-6.19★	-6.40★	-6.14★	-6.18★	-5.23★	-2.27♦	7.16★
	best	-2.77♦	-4.35★	-5.66★	-6.30★	-6.88★	-6.17★	-6.19★	-5.93★	-4.04★	-1.82♦

Statistical significance was determined by a two-sided Wilcoxon rank sum test. The ♦ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.05$, while the ★ show significance with respect to $\mathcal{L} = 0.0$ at $p < 0.001$. Bold markers indicate significantly detrimental effects

(1) Policy robustness: *With the exception of the KUKA 4 DoFs condition, low \mathcal{L} agents are prone to suffer from performance divergence after reaching an initial local optimum. High \mathcal{L} agents in the same condition do not show this behavior—see the KUKA 7 DoFs, NAO 2 DoFs, and KUKA 2 DoFs conditions.*

The divergence could be caused by the limited time horizon of the hyperparameter optimization. In all cases where divergence occurs, it sets in after the number of epochs used for the optimization. In this case, high \mathcal{L} agents not suffering from divergence would be in line with Stahlhut et al. [7, 16], who report that high \mathcal{L} values lead to more robust policies and are less sensitive to optimally tuned hyperparameters.

However, we note that the seemingly fast divergence is in part attributed to the log-log scale of the figures: the number of steps at which the agents stay close to their local optimum is relatively large compared to the initially fast convergence. An alternative explanation is that this divergence happens regularly but is rarely observed or reported since the training is stopped automatically when initial convergence is reached (early stopping). The reason for this divergence could be the phenomenon called *capacity loss*, which was only recently described. Lyle et al. [27] show that agents can lose the ability to adjust their value function approximator in light of new prediction errors. Capacity loss is attributed to a state representation collapse in the function approximators. This collapse seems most prevalent in temporal difference learning algorithms, using neural networks as function approximators, and sparse and non-stationary rewards.

An exciting question is whether high \mathcal{L} agents are more robust to capacity loss. However, this investigation is beyond the scope of this paper.

(2) Optimal long-term \mathcal{L} : *High \mathcal{L} values are mostly beneficial in the long run, except for the NAO 4 DoFs condition. In all other conditions, high \mathcal{L} agents reach either the best or comparable to the best performance observed for other \mathcal{L} values, albeit at later stages in training.*

This observation agrees with Stahlhut et al. [7, 16], who report increasing performance with increasing interaction frequency. However, in the NAO 2 DoFs case, there is only a small benefit of the highest \mathcal{L} value over the others in the long run. Thus, the gain can be considered not very large, in accordance with Bignold et al. [12], who argue that the increased effort of very high frequent interaction does not justify the small performance gains.

(3) Optimal \mathcal{L} trajectory: *During training within one experimental condition, the best choice of \mathcal{L} depends on the proficiency level and typically changes over time. For instance, in the NAO 4 DoFs experiment, the optimal \mathcal{L} changes from intermediate values in early training to low*

values in late training. However, this pattern does not generalize across tasks. E.g., the optimal \mathcal{L} changes from intermediate to high values in the NAO 2 DoFs condition, in contrast to the NAO 4 DoFs experiment.

This observation can encompass the following findings:

- early feedback is beneficial [18], as seen in all but the KUKA 2 DoFs condition,
- intermediate feedback frequencies are optimal [17], as seen in the NAO 2 DoFs and KUKA 7 DoFs conditions across most of the training,
- and that too much early feedback leads to a delayed onset of learning [17], as seen in all 3 KUKA conditions.

Furthermore, the shift of optimal \mathcal{L} values during training leads us also to conclude that the interaction rate should be changed adaptively for optimal performance, as also suggested by Cruz et al. [18] and Stahlhut et al. [7, 16].

As a proof of concept, we trained agents on the KUKA 2 DoFs task, starting with $\mathcal{L} = 0.0$, switching to $\mathcal{L} = 0.5$ at epoch 14, and finally to $\mathcal{L} = 0.99$ at epoch 18. Figure 8 shows that it is possible to achieve the early convergence of the low \mathcal{L} agents in this task, combined with the long-term refinement of high \mathcal{L} agents. The switch of \mathcal{L} induces a short-term performance deterioration. However, measuring the performance as the area under the curve, the adaptive strategy is superior to the fixed \mathcal{L} agents.

An important question is how to choose the optimal adaptive \mathcal{L} -strategy without having to train agents for various \mathcal{L} -values before. Whereas it seems to be a good rule of thumb to switch to high \mathcal{L} in late training, the optimal values in early training are very diverse. Here, *low*

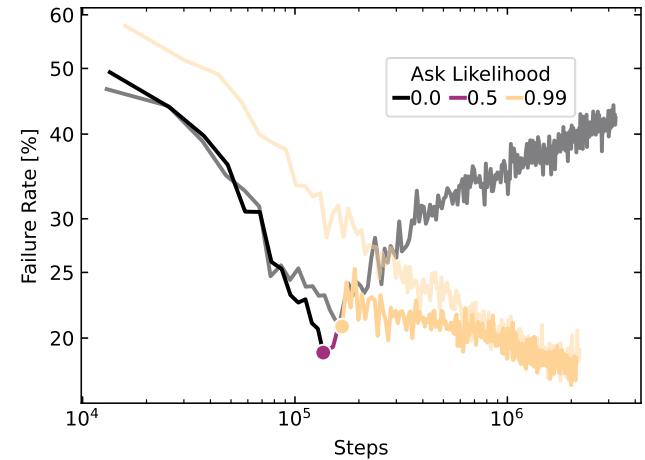


Fig. 8 Adaptive failure rate in the KUKA 2 DoFs task. The initial $\mathcal{L} = 0.0$ is changed to $\mathcal{L} = 0.5$ at epoch 14 and $\mathcal{L} = 0.99$ at epoch 18. The 50% opacity black and yellow curves show the original performance for fixed $\mathcal{L} = 0.0$ and $\mathcal{L} = 0.99$, compare Fig. 5a. The adaptive agents show both features of fast early training and long-term convergence of the fixed \mathcal{L} agents

\mathcal{L} values are optimal for the KUKA 2 DoFs condition, *intermediate* for KUKA 4 DoFs, KUKA 7 DoFs, and NAO 2 DoFs, or *high values* for the NAO 4 DoFs, spanning a range of $\mathcal{L} = 0.2$ to $\mathcal{L} = 0.8$ across experiments.

We hypothesize that the optimal \mathcal{L} in early training is influenced by how much the teacher simplifies the task. Expressly, if the task for high \mathcal{L} values becomes too easy in comparison to the $\mathcal{L} = 0.0$ task, the agent might fail to generalize and explore too little. Note that the agent effectively faces the $\mathcal{L} = 0.0$ situation during evaluation since it is not receiving feedback then. Specifically, consider a fully interactive agent. Here, the teacher will rarely allow actions that increase the distance of the end effector to the goal. This scenario simplifies the task during training but also entails that sub-optimal state-action pairs are seldom encountered. Note that this situation primarily applies to mistake-correcting teachers as used in this article.

We quantify the reduction in task complexity by \mathcal{L} as the average *failure rate* of a newly initialized agent on the training set with an interaction frequency \mathcal{L} , but without policy updates. All failure rates for $\mathcal{L} \neq 0$ are normalized to the performance of the baseline $\mathcal{L} = 0$ agent. Each value is averaged over 20 randomly initialized agents. We compare the relative task complexities with the best choices of \mathcal{L} at similar stages in early training. For this, we use the *failure rate* thresholds of 50% for KUKA 7 DoFs and KUKA 4 DoFs, 40% for KUKA 2 DoFs, 25% for NAO 4 DoFs, and 5% for NAO 2 DoFs (see Figs. 3b, 4b, 5b, 6b, 7b). Note that it is not feasible to use the same value for all experiments since the initial performance across experiments varies between $\approx 50\%$ and $\approx 90\%$.

Figure 9 shows the relative task complexity for all experiments and \mathcal{L} values, along with the best choices of \mathcal{L} at the mentioned thresholds and the significantly detrimental choices. Indeed, \mathcal{L} values that lead to relatively low

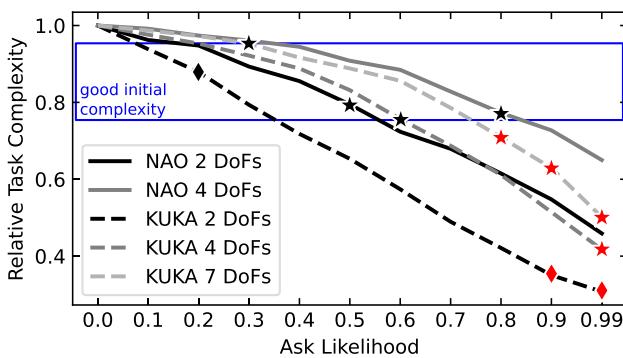


Fig. 9 Relative task complexity reduction induced by feedback frequencies for all experiments. Interaction has a differential effect on the complexity reduction, depending on the task. Black symbols show the best choice of \mathcal{L} for all five experiments during early training (compare Figs. 3b to 7b, second *failure rate* thresholds). Red symbols show significantly detrimental choices for \mathcal{L} at this *failure rate* threshold

task complexities are prone to have a detrimental effect. In contrast, the most beneficial choices of early \mathcal{L} values are those that lead to a relative task complexity between ≈ 0.78 to ≈ 0.95 . Thus, drawing an initial \mathcal{L} from that range for each task makes it considerably more likely to choose the initially optimal \mathcal{L} value than naively sampling from the range of $\mathcal{L} = 0.2$ to $\mathcal{L} = 0.8$.

Based on this observation, we argue against the claims that either high, intermediate, or low \mathcal{L} values are optimal in early training. Instead, optimality seems to be better predicted by the task complexity reduction induced by \mathcal{L} .

5 Conclusion

In this study, we conducted a thorough extension of previous research investigating the effect of feedback frequency on agent performance in continuous action and state spaces. The main contribution is the discovery that task complexity and performance threshold influence the interpretation of the best interaction rate with a teacher. Moreover, no single best solution exists across task conditions.

Our results instead suggest that the optimal interaction rate changes over time and that the task complexity determines the specific optimal trajectory for \mathcal{L} . These observations allow us to consolidate previously contradictory claims on the optimal interaction frequency. Furthermore, we described a heuristic to choose the initial feedback frequency based on a measure of the relative task complexity changes induced by the teacher's feedback on an untrained agent.

Future work: A future goal is to probe the described heuristic further to predict the optimal trajectory before – and potentially adjust it during – training. Such a strategy has the potential to increase data efficiency significantly. Drawing such conclusions across an even more comprehensive range of tasks requires more extensive experimentation with more tasks of different complexity.

It is also necessary to determine the deeper cause of seemingly differential effects of task complexity reduction by teacher interaction and the relation to potential capacity loss in the agents' neural network function appropriators.

Finally, it will be helpful to quantify the interaction of the feedback frequency effects with other factors, such as fixed feedback budgets and advice quality, to go toward applicable scenarios with realistic human feedback.

Funding Open Access funding provided by the Projekt DEAL (Open access agreement for Germany). Research funding by the M-RoCK – Human–Machine Interaction Modeling for Continuous Improvement of Robot Behavior project funded by the Federal Ministry of Education and Research with grant no. 01IW21002.

Availability of data and material It can be found under the following link <https://doi.org/10.6084/m9.figshare.20027582>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate Not applicable

Consent for publication/informed consent Not applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A et al (2017) Mastering the game of go without human knowledge. *Nature* 550(7676):354–359. <https://doi.org/10.1038/nature24270>
- Arzate Cruz C, Igarashi T (2020) a survey on interactive reinforcement learning: design principles and open challenges. In: ACM designing interactive systems conference (DIS). Eindhoven, The Netherlands: Association for Computing Machinery; p. 1195–1209
- Tan M (1997) Multi-agent reinforcement learning: independent vs. cooperative agents. In: Readings in agents. Morgan Kaufmann Publishers Inc.. p. 487–494
- Da Silva FL, Warnell G, Costa AHR, Stone P (2019) Agents teaching agents: a survey on inter-agent transfer learning. *Auton Agents Multi-Agent Syst* 34(1):9. <https://doi.org/10.1007/s10458-019-09430-0>
- Ng AY, Harada D, Russell SJ (1999) Policy invariance under reward transformations: theory and application to reward shaping. In: International conference on machine learning (ICML). vol. Sixteenth. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; p. 278–287
- Griffith S, Subramanian K, Scholz J, Isbell C, Thomaz AL (2013) Policy shaping: integrating human feedback with reinforcement learning. In: International conference on neural information processing systems (NIPS). vol. 2. Lake Tahoe, NV, USA: Curran Associates, Inc.; p. 2625–2633
- Stahlhut C, Navarro-Guerrero N, Weber C, Wermter S (2015) Interaction in reinforcement learning reduces the need for finely tuned hyperparameters in complex tasks. *Kogn Syst*. <https://doi.org/10.17185/duepublico/40718>
- Ho MK, Cushman F, Littman ML, Austerweil JL (2019) People teach with rewards and punishments as communication, not reinforcements. *J Exp Psychol: Gen* 148(3):520–549. <https://doi.org/10.1037/xge0000569>
- Thomaz AL, Breazeal C (2008) Teachable robots: understanding human teaching behavior to build more effective robot learners. *Artif Intell* 172(6–7):716–737. <https://doi.org/10.1016/j.artint.2007.09.009>
- Loftin R, MacGlashan J, Peng B, Taylor M, Littman M, Huang J, et al. (2014) A strategy-aware technique for learning behaviors from discrete human feedback. In: AAAI conference on artificial intelligence. vol. 28 of AAAI Technical Track: Humans and AI. Québec City, Québec, Canada: Association for the Advancement of Artificial Intelligence. p. 937–943
- Knox WB, Stone P (2012) Reinforcement learning from human reward: discounting in episodic tasks. In: IEEE international symposium on robot and human interactive communication (RO-MAN). Paris, France. p. 878–885
- Bignold A, Cruz F, Dazeley R, Vamplew P, Foale C (2021) An evaluation methodology for interactive reinforcement learning with simulated users. *Biomimetics* 6(1):13. <https://doi.org/10.3390/biomimetics6010013>
- Taylor ME, Carboni N, Fachantidis A, Vlahavas I, Torrey L (2014) Reinforcement learning agents providing advice in complex video games. *Connect Sci* 26(1):45–63. <https://doi.org/10.1080/09540091.2014.885279>
- Cruz F, Wüppen P, Magg S, Fazrie A, Wermter S (2017) Agent-advising approaches in an interactive reinforcement learning scenario. In: Joint IEEE international conference on development and learning and epigenetic robotics (ICDL-EpiRob). Lisbon, Portugal. p. 209–214
- Suay HB, Chernova S (2011) Effect of human guidance and state space size on interactive reinforcement learning. In: IEEE international symposium on robot and human interactive communication (RO-MAN). Atlanta, GA, USA. p. 1–6
- Stahlhut C, Navarro-Guerrero N, Weber C, Wermter S (2015) Interaction is more beneficial in complex reinforcement learning problems than in simple ones. In: 4. Interdisziplinärer workshop kognitive systeme: mensch, teams, systeme und automaten. Bielefeld, Germany. p. 142–150
- Millán-Arias C, Fernandes B, Cruz F, Dazeley R, Fernandes S (2020) Robust approach for continuous interactive reinforcement learning. In: International conference on human-agent interaction (HAI). vol. 8th. Virtual Event USA: Association for Computing Machinery. p. 278–280
- Cruz F, Magg S, Weber C, Wermter S (2016) Training agents with interactive reinforcement learning and contextual affordances. *IEEE Trans Cogn Dev Syst* 8(4):271–284. <https://doi.org/10.1109/TCDS.2016.2543839>
- Kofinas N, Orfanoudakis E, Lagoudakis MG (2015) Complete analytical forward and inverse kinematics for the NAO humanoid robot. *J Intell Robot Syst* 77(2):251–264. <https://doi.org/10.1007/s10846-013-0015-4>
- Busson D, Bearee R, Olabi A (2017) Task-oriented rigidity optimization for 7 DoF redundant manipulators. *IFAC-PapersOnLine*. 50(1):14588–14593. <https://doi.org/10.1016/j.ifacol.2017.08.2108>
- Navarro-Guerrero N, Lowe R, Wermter S (2017) Improving robot motor learning with negatively valenced reinforcement signals. *Front Neurorobotics* 11:10. <https://doi.org/10.3389/fnbot.2017.00010>
- Navarro-Guerrero N, Lowe R, Wermter S (2017) The effects on adaptive behaviour of negatively valenced signals in reinforcement learning. In: Joint IEEE international conference on development and learning and epigenetic robotics (ICDL-EpiRob). Lisbon, Portugal. p. 148–155
- van Hasselt H, Wiering MA (2007) Reinforcement learning in continuous action spaces. In: IEEE symposium on approximate

- dynamic programming and reinforcement learning (ADPRL). Honolulu, HI, USA. p. 272–279
24. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International conference on learning representations (ICLR). 3rd. San Diego, CA, USA. p. 15
25. Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. *J Mach Learn Res* 10(56):1633–1685
26. Bergstra J, Yamins D, Cox DD (2013) Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms. In: Python in science conference (SciPy). Austin, TX, USA. p. 13–20
27. Lyle C, Rowland M, Dabney W (2022) Understanding and preventing capacity loss in reinforcement learning. In: International conference on learning representations (ICLR). vol. 10th. Virtual Event. p. 12

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.