Accelerating the Convergence of Human-in-the-Loop Reinforcement Learning with Counterfactual Explanations

Jakob Karalus [1] **Felix Lindner** [1]

Abstract

The capability to interactively learn from human feedback would enable robots in new social settings. For example, novice users could train service robots in new tasks naturally and interactively. Human-in-the-loop Reinforcement Learning (HRL) addresses this issue by combining human feedback and reinforcement learning (RL) techniques. State-of-the-art interactive learning techniques suffer from slow convergence, thus leading to a frustrating experience for the human. This work approaches this problem by extending the existing TAMER Framework with the possibility to enhance human feedback with two different types of counterfactual explanations. We demonstrate our extensions' success in improving the convergence, especially in the crucial early phases of the training.

1. Introduction

Classical Machine Learning approaches like supervised learning, or reinforcement learning can solve complex tasks with sophisticated architectures, but they allow for little to no interaction possibilities with the human throughout their training. The field of Human-in-the-loop reinforcement learning (HRL) aims to integrate the human into the learning process, allowing the agent to learn directly from the human. For example, this integration of humans into the training process allows non-experts to train or modify a robots' behavior in a natural way. The requirements for huge data sets like in supervised learning or an environmental reward signal for reinforcement learning are not needed.

While the integration of humans into the learning process opens up new possibilities, they also introduce different challenges. Mainly the integration of different feedback modalities and the convergences of the algorithm are challenging. When teaching an agent, humans naturally want to enrich their feedback with explanations, and a too simplistic interface can lead to frustration on the human side, see (Krening & Feigh, 2018; Stumpf et al., 2007). In most previous work (e.g., (Knox & Stone, 2008)), humans can only give simple binary feedback or control a single dimension (e.g., (Celemin & del Solar, 2015)).

Besides, human feedback is more costly than automated feedback coming from the environment or a simulator. For example, state-of-the-art reinforcement learning needs millions of episodes with billions of frames (see (Milani et al., 2020)). It is insufficient to use the same algorithms and replace the environmental reward with the human reward since it is not feasible for a human to watch the agent for weeks or months of training. In addition to the time constraint, if progress is slow, it can lead to early frustration for the human if they see no impact of his feedback in the training. Therefore, the algorithms' speed of convergence is crucial for a positive user experience.

Throughout this work, we want to enable humans to give richer feedback through additional, optional counterfactual examples. This is motivated by recent developments in the field of "eXplainable Artificial Intelligence" (XAI) (see (Miller, 2019)), which aims at explaining black-box models and their predictions towards (novice) users. We reverse the direction of explanations: while the focus in XAI is on AI systems providing explanations of their reasoning as output, we consider how a learning agent can make use of explanations provided by the human as input. In concrete, we focus on the inclusion of counterfactuals in the HRL feedback loop to provide the user with a natural way to highlight important issues, without the need to "open the black box" (Mc Grath et al., 2018). We show that the inclusion of counterfactuals also leads to faster convergence.

2. Related Work

Human-in-the-Loop Reinforcement Learning deals with the integration of humans into the learning cycle. That means the reinforcement learning algorithm cannot depend on the environmental reward and instead has to learn with the hu-

[1]Institute of Artificial Intelligence, Ulm University, Ulm, Germany. Correspondence to: Jakob Karalus <jakob.karalus@uni-ulm.de>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

man feedback. One of the first proposed frameworks is the TAMER framework (Knox & Stone, 2008), which we use as a basis for our work. Most of the work done in the field of Human-in-the-Loop Reinforcement Learning (or Interactive Reinforcement Learning) focuses on creating more capable and more sophisticated frameworks and models. Throughout this work, we use the common extension DeepTAMER (Warnell et al., 2018), which enables the usage of Deep Neural Nets as a function approximator.Multiple extensions of TAMER tackle the problem of combining human feedback with environmental feedback. TAMER+RL (Knox & Stone, 2010) and continuous TAMER+RL (Knox & Stone, 2012) both re-introduce the environmental reward and enable learning in setting from multiple feedback sources. A common trend is introducing reinforcement learning frameworks into TAMER: Arakawa et al. (2018) combine it with Deep-Q learning, and Vien et al. (2012) use an actor-critic setting to enable the usage in continuous-actions spaces. While all these approaches enable the usage of TAMER in additional types of environments, their basic capabilities essentially stay the same as vanilla TAMER.

In contrast to TAMER's reward-shaping approach, the second-most common HRL approach is policy-shaping (Griffith et al., 2013). Policy-shaping uses the feedback directly as labels for the corresponding policy instead of interpreting it as a reward signal. Guan et al. (2020) integrate human explanation into the HRL by using saliency-based attention annotation combined with Deep Convolutions Networks. This method showed that the inclusion of explanations in HRL could lead to better performance. However, this method can only be applied in specific cases due to Saliency Maps and Deep Neural Nets (solving vision problems with Convolution Neural Nets). In contrast, our current proposed solution is mostly independent of the input shape and the approximator's chosen architecture. Additionally, their work requires an environmental reward, while traditional TAMER and our extensions work purely on human feedback.

Lu et al. (2020) has shown that an integration of counterfactuals in traditional reinforcement learning (without human feedback) can be beneficial to the label efficiency. They use Structural Causal Models (SCM) to generate counterfactuals for existing data points as a data augmentation technique. While they showed that this approach could enable the usage in low-sample areas like healthcare, their reliance on model-based RL and SCM is a constraint since both approaches require additional and careful development efforts.

There has been extensive work on the generation of explanations, especially for counterfactuals. However, most of the work is centered around explaining a black box towards a human. Smith & Ramamoorthy (2020) have shown that dynamically generating counterfactuals and including them in the training process increases their image-based robotic control tasks' robustness. They train a GAN-inspired network to generate and discriminate between real and generated images. The generator is then used on a regressor for a model predictive control task, which leads to an increase in robustness to unseen obstacles. Their work shows the impact counterfactuals can have on the quality of regressors, and the increased robustness shows promise for more dynamic tasks like reinforcement learning. Other work successfully used counterfactuals in partial observable settings to increase the learning speed, and robustness of their reinforcement learning approaches (Jin et al., 2018; Buesing et al., 2019). Foerster et al. (2018) use a counterfactual loss in a multi-agent reinforcement setting to increase their performance. Menglin et al. (2020) use counterfactual experiences to solve inefficiencies with exploration in the early stages of SARSA with counterfactuals.

The concept of counterfactuals has been included in different applications with success. With this work, we want to formulate a general method for the inclusion of counterfactuals in HRL.

3. Background and Methods

In this section, we explain the relevant background and highlight our extension of the existing method. First, we will explain TAMER, introduce counterfactuals, and investigate the convergence of TAMER from a reinforcement learning perspective. Afterwards we use this perspective as motivation to introduce counterfactuals in TAMER.

3.1. Human-in-the-Loop Reinforcement Learning

In a traditional Reinforcement Learning setting, the designer has to define an environmental reward, which the agent is trained on. Human-in-the-loop Reinforcement Learning (HRL, also sometimes referred to as "Interactive Reinforcement Learning") instead allows the agent to learn from human feedback directly and incrementally, even in environments where no other reward signal is available. A major difference of human feedback is the potential inconsistency and sub-optimality of the human feedback. Throughout the training, humans can give feedback at various stages, but the agent cannot depend on a fixed and reliable reward (unlike with the normal environmental reward). HRL is also advantageous in very complex environments where a specification of a solid reward function is hard, since even slight miss-specifications of the reward function can lead to unexpected side-effects in real use cases (Dulac-Arnold et al., 2019).

3.2. The TAMER Framework

Our work is based on the TAMER framework, first proposed by Knox & Stone (2008). TAMER is a framework for interactive learning from human feedback without environmental rewards. A TAMER instance is given by a set of states S, a set of actions A, and a set of possible human feedbacks R_h. The goal is to learn a policy π which maximizes the expected human reward $H : S \times A \mapsto R_h$. H is unknown to the agent and thus the agent has to rely on human feedback only. The human is given the opportunity to give positive or negative feedback to an observed action a performed in state s. Like in most work, we use a $(-1, +1)$ range for the human feedback. This feedback is used to train the H function to approximate human reward for action a in state s. This learned function of the human reward is then used to greedily select actions with the highest expected human reward at each time-step, i.e., $argmax_a(H(s, a))$. TAMER does not make use of any environmental reward, methodical exploration, or longer-term planning mechanism.

We use a DeepTAMER architecture throughout this work, which approximates the H function with a Deep Neural Network and is training through a simple replay buffer (see (Warnell et al., 2018)). We use two shared fully-connected layers with $ReLU$ activation and, for each action, a separate, fully-connected head with a $tanh$ activation. To train the network, we combine new feedback samples with old feedback from a replay buffer into a batch and perform a single update step with *ADAM* for each feedback batch.

3.3. Counterfactuals

Counterfactual explanations describe a causal connection between two possible events in the form: "If event B rather than A had happened, then outcome Y rather than X would have happened" (Byrne, 2019). Counterfactuals thus refer to a *fact* that represents the event that actually has happened and caused some outcome, and to the *contrast*, which would have resulted in a different outcome (the *foil*) (Miller, 2019). Extensive research has been performed on different methods for generating counterfactuals from trained models by finding realistic contrasts that are as close as possible to the fact but still provide a different outcome (Stepin et al., 2021).

In our work, we allow humans to state counterfactual explanations of negative feedback to an observed event $\langle state, action \rangle$. The counterfactuals can take the form "If a had been performed in state s' rather than in state s, then my feedback would be positive rather than negative" and "If action a' had been performed s rather than action a, then my feedback would have been positive rather than negative."

As multiple studies show (Markman et al., 2008; Rim & Summerville, 2013), it is easier for most people to envision a world "better-off" than a world "worse-off". Additionally, positively-directed counterfactuals are usually of higher quality (Byrne, 2019). While we could enable counterfactual feedback for every reward, we limit the possibility for counterfactual feedback to the negative reward case. This allows the agent to learn how to get from a state with bad feedback into a state with good feedback.

3.4. The convergence of Reinforcement Learning algorithms

While most Reinforcement Learning algorithms converge, their speed is a critical factor. In Interactive Learning, this convergence speed is essential since humans are asked to give feedback to the learner. This is a tedious task we naturally want to minimize. An agent which learns too slow or shows no progress can limit the motivation of the human to give further feedback (Cakmak & Thomaz, 2010). In reinforcement learning, the convergence difference between an expert policy π^* and a sub-optimal π can be described by the distance between the optimal distribution of visited states and the current distribution (also known as *performance difference lemma* due to Kakade (2004)):

$$V^{\pi^*} - V^{\pi} = \sum_s \pi^*(s) \sum_a (\pi^*(a \mid s) - \pi(a \mid s)) Q^{\pi}(s, a) \tag{1}$$

The optimal policy, is denoted by π^*, $\pi^*(s)$ is the visited state distribution of π^*, and $\pi^*(a \mid s)$ the action distribution of the optimal policy. This indicates that there will always be a difference in performance, as long as there is a substantial difference in the visited states. While the original lemma was formulated for classical Q-learning, we apply the same concept to TAMER since its replacement of Q with H can be seen as identical in function and convergence. In a setting with a large state/action space, under a random policy π, the distribution of visited states will drastically differ from the expert distribution. This leads to a "chicken/egg" problem, where to learn quickly, an agent has to reach states similar to the states of the expert, but doing that is essentially the goal of learning.

To solve this problem, we use the previously introduced concept of counterfactuals. A positively-directed counterfactual comes directly from the expert distribution since it indicates which part of the current $\langle state, action \rangle$ pair should have been different. This allows the agent to reduce the difference $V^{\pi^*} - V^{\pi}$, which leads to faster convergence. To experimentally test this hypothesis, we introduce two different types of positively-directed counterfactuals into the TAMER framework and confirm that their inclusions lead to a faster convergence rate. Besides, we expect that counterfactuals' impact on convergence should increase with the size of the state or action space since with increasing size, the agent has a lower chance of randomly selecting good states, but the counterfactual should still provide a "shortcut" to minimize

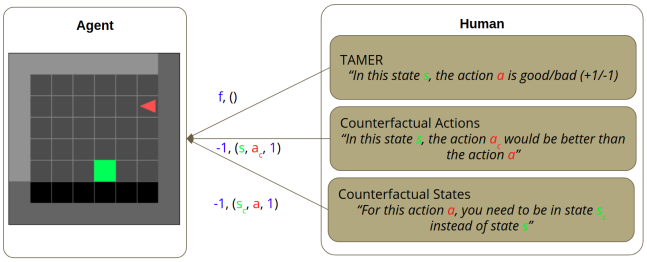

Figure 1. Left: Visualization of the Minigrid environment with the goal as a green square. Highlighted area indicates current viewpoint of the agent. Right: Verbalization of the various feedback mechanism in natural language.

the difference. Besides the above-discussed human reasons for "better-off" counterfactuals, the *performance difference* thus offers an additional viewpoint favoring our focus on positively-directed counterfactuals.

3.5. Using counterfactuals in TAMER

To introduce positively-directed counterfactuals, we have to change the *fact* of our current $\langle state, action \rangle$ pair, so that the new *contrast* would receive a positive reward. To achieve this *contrast*, we modify either the state or the action of our fact. We call the modified element *counterfactual state* or *counterfactual action*. A counterfactual state provides a state where the action would be a good solution $gather\_feedback(s, a) = (-1, \langle 1, s_c, a \rangle)$. This could be understood as *"Action a would be good if you were in state s_c instead of s"*.

In case of counterfactual actions, $gather\_feedback(s, a) = (-1, \langle 1, s, a_c \rangle)$ can be described in natural language with: *"In state s, action a_c would be better than action a"*. Both types of counterfactuals are exemplified in Fig. 1.

Algorithm 1 TAMER with Counterfactual Feedback

Require: H function approximator; total episodes N_e; maximum number of steps per episode N_s
  for e = 1 **to** N_e **do**
    for t = 1 **to** N_s **do**
      $s_t \leftarrow$ observe state
      $f, \langle f_c, s_c, a_c \rangle \leftarrow$ gather feedback (s_{t-1}, a_{t-1})
      if $(f < 0) \& (f_c)$ **then**
        update H with $\langle f_c, s_c, a_c \rangle$
      end if
      update H with $\langle f, s_{t-1}, a_{t-1} \rangle$
      $a_t \leftarrow \mathrm{H}(s_t)$
      perform action a_t
    end for
  end for

To include these counterfactuals in TAMER, we extend the human feedback function to return an optional triple $\langle f_c, s_c, a_c \rangle$. This triple contains the counterfactual feedback f_c, which is in our case always 1 (since we only use positively-directed counterfactual). Either s_c and a_c contains a counterfactual version of the original input, depending on the types of counterfactual feedback introduced above. If the human feedback contains this triple, we use it as an additional update towards H, combined with the standard TAMER update $\langle f, s, a \rangle$. The full procedure with all steps is also described in Algorithm 1.

4. Evaluation

To evaluate the impact of counterfactuals on TAMER's convergence, we perform a series of training runs with different conditions. The basic evaluation will focus on the performance of TAMER and the previously introduced counterfactual additions. To achieve this, we train each agent for 250 episodes with different random seeds for the environment initialization and agent initialization. Throughout the training, we evaluate the agents' performance after an episode in 10 fixed evaluation runs. In further runs, we evaluate how our changes perform with better/worse feedback quality. To be able to test all these conditions, we use a simulated environment with synthetically generated human feedback.

We will use TAMER as the baseline throughout this evaluation, with our two proposed additions of counterfactuals states (TAMER+CFS) and counterfactual actions (TAMER+CFA). We want to show that our additions can improve the existing TAMER baseline.

4.1. Environment Setup

The evaluation is run in multiple environments (MountainCar, LunarLander, Minigid), with different observation and action shapes. For most of the environment, we did not change any settings, except for the "Minigrid" environment (Chevalier-Boisvert et al., 2018), which we changed so that the goal position is random (instead of a static goal at the bottom-right corner.) For LunarLander and Minigrid we selected a DeepTAMER architecture, due to their more complex state/action space. For MountainCar a linear function with RBF-kernel is chosen (similar to the reference architecture in the original TAMER).

To evaluate the performance of the agent, we use an environmental reward as a measure of success in finding the goal. This reward is only used to evaluate the agent and is never seen or used in training. The agent fully learns from human feedback. In the following sections, we will refer to this reward for the evaluation of the algorithms. To track the learning progress we run after every episode an evaluation (without learning) in 10 different (fixed) seeds

of the same environment. The total reward of each run is collected and then presented as a mean. This allows us to track the convergence of an agent throughout its training. Since the convergence of reinforcement learning algorithms can be highly unstable, we perform multiple training runs per algorithm/environment with different random seeds to initialize the agent and the environments. The presented rewards are then the means (from different seeds) of the mean evaluation rewards (10 in each seed). This allows us to provide confidence intervals for the rate of convergence. In the following, all reward numbers, therefore, represent this mean of evaluation means.

In the first step, we want to evaluate the global performance of each variant and show that all approaches converge towards the respective maximum reward and that our changes lead to a faster convergence, especially in the early episodes. Afterward, we will introduce changes in the feedback frequency and the optimality of the feedback to show that the improvement with counterfactual holds under different feedback conditions.

4.2. Human-Feedback Oracle

While Interactive Learning aims to incorporate human feedback into the training loop, it can be cumbersome to collect human feedback. Especially throughout development or when comparing methods, it is costly to collect extensive human feedback at scale. This cost can hinder the development and evaluation of these algorithms since evaluations are often performed with only a few humans, leading to false interpretations due to statistical randomness. Besides, the feedback of humans can have multiple problems. It can have a bias or could be inconsistent. While handling these challenges is an ongoing research question, in this work, we focus purely on the improvements of the feedback from a learning standpoint. Therefore we use a synthetic oracle to imitate human feedback. To achieve this, we trained an agent for the oracle with proximal policy optimization (PPO, (Schulman et al., 2017)) algorithm on every environment for sufficiently many steps to solve the tasks. We then use this policy to give dynamical calculate (counterfactual) feedback for the actual learning algorithm. To transform this learned policy into an actual feedback oracle we use algorithm 2. While the determination of the basic feedback is straightforward (compare the performed action a in state s with the preferred action $a*$), the calculation of counterfactual feedback involves more steps. In the case of the counterfactual actions, we iterate through all possible actions for a given input $\langle s, a\rangle$ and select the best actions $a*$ as counterfactual action. Since the state space is usually much larger than the action space, we randomly sample states from a replay buffer $s*$ with the constrain that action a would be considered optimal on that state. In either case the learning agent can use the original tuple $\langle -1, s, a\rangle$ and the counterfactual tuple, $\langle +1, s, a'\rangle$ or $\langle +1, s', a\rangle$, to update the model. This provides us with a perfect feedback oracle.

Algorithm 2 Provide (counterfactual) feedback with a trained policy

Require: state s, action a, trained policy $\pi*$, feedback rate fr, feedback optimality fo,
  if $shouldProvideFeedback(fr)$ **then**
    if $shouldProvideOptimalFeedback(fo)$ **then**
      preferred action = $\pi*(s)$
    else
      preferred action = random action
    end if
    if preferred action == a **then**
      return +1, (no counterfactuals)
    else
      if Counterfactual Actions **then**
        return -1, (+1, state, preferred action)
      end if
      if Counterfactual Actions **then**
        counterfactual state = sample from $\pi*$'s replay buffer where preferred actions applies
        return -1, (+1, counterfactual state, action)
      end if
    end if
  end if

The handling of inconsistent and sub-optimal feedback is one of the major challenges in HRL. Our evaluation accounts for inconsistency and sub-optimality by introducing two hyper-parameters, "feedback frequency" and "optimality" in the oracle. With the first one, we control how often the oracle gives feedback towards the agent to archive inconsistent feedback. With the second parameter, we allow the oracle to give random feedback, which results in sub-optimal feedback. To achieve sub-optimal feedback, every time the oracle decided to give feedback, an additional draw is performed, which determines if the feedback is optional or not. In non-optimal feedback, the oracle chooses a random action (instead of evaluating with the optimal action/strategy) and bases the feedback on the randomly chosen action. (The exact implementation is shown in Algorithm 2.)

4.3. Results

As shown in Fig. 2, TAMER extended with counterfactuals leads to a faster convergence compared to the original TAMER. The main advantage is in the early stages of training, where convergence is significantly faster than for the original TAMER. In later stages of the training, the difference can shrink, and as expected, all variants (vanilla TAMER and our extensions) reach, with enough episodes,

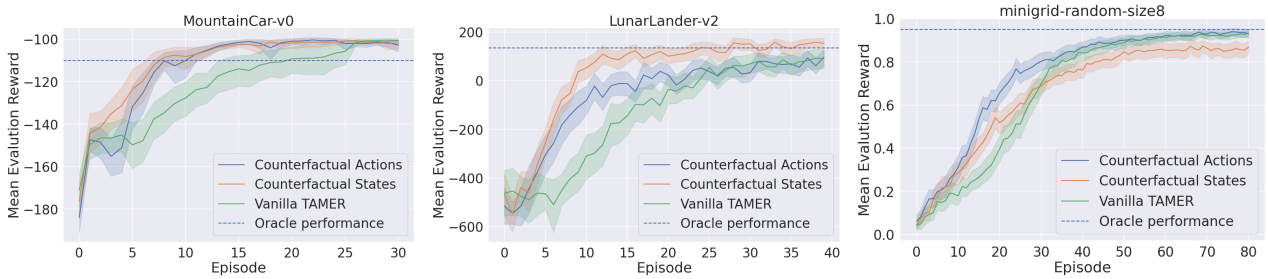

Figure 2. Mean reward of each feedback method on the evaluation set throughout training. Bold line is the mean reward of random seeds (n=50), area is the 95% confidence interval.

the maximum reward plateau. The focus of our work is to enable richer feedback mechanisms, which should reduce the absolute number of feedback steps needed for solving the task.

To highlight the faster convergence, we measure the absolute performance of each variant at early episodes for each environment and show the absolute performance of each variant in Table 4.3 (first section). While both counterfactual methods converge faster than vanilla TAMER, no clear ranking between the extensions can be drawn. While in the LunarLander environment the counterfactual states clearly perform better, it is the opposite in the minigrid environment, where counterfactual actions are more favorable.

Reduced feedback frequency. So far, all of the evaluations were performed with perfect feedback and a perfect feedback rate, which is not necessarily a realistic assumption for HRL. We investigate our extensions' impact on the convergence under more realistic conditions. To achieve this, we reduce the frequency in which the oracle gives feedback. This mimics one of the essential characteristics of HRL, namely that the feedback is not deterministic and can be inconsistent and sporadic (Cakmak & Thomaz, 2010). To achieve this, we change the feedback frequency of our oracle (as shown in Algorithm 2, and evaluate our extension under this changed condition.

As shown in Table 4.3 (sections 2 and 3), the results are that even with reduced feedback rates, the general pattern of an increase in convergence with the inclusion of counterfactuals is similar to the results with perfect rate. It can be noted that some environments suffer more from the reduced rate of feedback, as visible in the LunarLander environment where with a frequency of 0.5, the convergence is not that much impacted (compared to perfect feedback), but with a much lower rate of 0.1, the convergence drops drastically for the baseline and our extensions.

Sub-optimal feedback. Another important characteristic of human feedback is that it does not have to be optimal at every point in time. Next, we evaluate the convergence behavior for non-optimal feedback, which we can create with our synthetic oracle as shown in Algorithm 2.

The results show that our additions still work with lower optimality and outperform the baseline. Naturally, the convergence is slower with less optimal feedback since the "signal strength" is lower. It can be noted that for the minigrid environment, the results with a feedback optimality = 0.5 (and frequency = 1.0) are similar to those with a frequency of 0.5 (and optimality = 1.0). In contrast to the LunarLander, lower quality feedback reduces performance much more than a lower feedback frequency. The hardest impact of the lower optimality can be seen in the MountainCar environment, where with an optimality of 0.5 counterfactual actions fail to converge (in a reasonable number of episodes).

4.4. Discussion

Faster convergence. We extended TAMER to allow the human to give additional counterfactual feedback and evaluated our extension in various environments and conditions. As shown in the previous sections, the inclusion of counterfactuals increases the convergence in every condition, which means that the human feedback is handled more label efficient. This behavior confirms the expectation stated in section 3.4: In the early stages of training, a large difference between the agent and the expert (human) policy exists. Therefore, the introduction of a single example from the expert policy can have a large impact. In later stages of the training, when the difference shrinks, the impact of additional expert samples becomes lower. A limiting side-effect of the convergences is that the agent sees less and less counterfactual in the learning process since we only give counterfactual feedback in negative cases.

Sub-optimal feedback. In cases where the feedback is sub-optimal (which also makes the counterfactual feedback sub-optimal), our extension still proves to increase the rate of convergence. In some smaller studies, we noticed that with counterfactual enriched feedback, the agent can converge in LunarLander and Minigrid with extremely sub-optimal feedback (feedback optimality < 0.1). Further research could be done to actually determine acceptable lower bounds

Table 1. Mean reward at selected episodes. Bold number indicate highest reward for each environment/condition section.

	ENVIRONMENT																	
ALGORITHM	MOUNTAINCAR						LUNARLANDER						MINIGRID					
EPISODE	1	2	3	5	10	20	2	5	10	15	20	25	2	5	10	15	20	25
	FEEDBACK FREQUENCY=1.0; FEEDBACK OPTIMALITY=1.0																	
TAMER	-149	-146	–146	-149	-127	-109	-462	-415	-295	-96	-5	32	0.08	0.15	0.18	0.26	0.4	0.51
TAMER + CFA	-147	-148	-155	-131	-110	**-100**	-491	-294	-108	-19	7	32	**0.13**	**0.2**	**0.31**	**0.53**	**0.66**	**0.75**
TAMER + CFS	**-144**	**-141**	**-135**	**-123**	**-108**	-101	**-410**	**-235**	**41**	**83**	**127**	**136**	0.10	0.17	0.29	0.41	0.52	0.61
	FEEDBACK FREQUENCY=0.5; FEEDBACK OPTIMALITY=1.0																	
TAMER	-162	-151	-145	-149	-140	-120	-480	-454	-416	-398	-312	-142	0.06	0.10	0.14	0.21	0.24	0.2
TAMER + CFA	**-149**	-146	-148	-150	-128	-104	-444	-372	-217	-78	-21	28	0.07	**0.13**	0.17	**0.30**	**0.40**	**0.47**
TAMER + CFS	-150	**-132**	**-130**	**-138**	**-122**	**-103**	**-429**	**-333**	**-207**	**17**	**134**	**118**	**0.08**	**0.13**	**0.21**	0.25	0.30	0.36
	FEEDBACK FREQUENCY=0.1; FEEDBACK OPTIMALITY=1.0																	
TAMER	-183	-176	-169	-166	-164	-145	-528	**-441**	-456	-395	-392	-411	0.02	0.04	0.05	0.06	0.07	0.10
TAMER + CFA	-178	-164	-171	-164	-148	-142	-568	-470	-425	-381	**-372**	-395	**0.04**	0.04	0.03	0.08	0.11	**0.15**
TAMER + CFS	**-164**	**-155**	**-150**	**-144**	**-135**	**-135**	**-516**	-448	**-415**	**-372**	-382	**-355**	0.03	0.04	**0.09**	**0.10**	**0.14**	0.14
	FEEDBACK FREQUENCY=1.0; FEEDBACK OPTIMALITY=0.75																	
TAMER	-157	-153	-155	-157	-128	-110	-434	-439	-325	-124	-33	29	0.09	0.11	0.18	0.25	0.45	0.55
TAMER + CFA	-178	-175	-152	-132	-122	-129	-429	-317	-86	-17	35	61	**0.12**	**0.24**	**0.38**	**0.48**	**0.62**	**0.73**
TAMER + CFS	**-138**	**-138**	**-149**	**-131**	**-103**	**-106**	**-428**	**-271**	**-15**	**95**	**131**	**144**	0.08	0.17	0.26	0.33	0.48	0.52
	FEEDBACK FREQUENCY=1.0; FEEDBACK OPTIMALITY=0.5																	
TAMER	**-156**	**-147**	-151	-146	-129	-111	-459	-414	-310	-216	-103	-42	0.04	0.11	0.18	0.20	034	0.40
TAMER + CFA	-190	-189	-182	-180	-179	-179	**-398**	-316	-138	-55	-29	-1	0.09	**0.24**	**0.24**	**0.33**	**0.39**	**0.43**
TAMER + CFS	-164	-148	**-140**	**-131**	**-110**	**-109**	-516	**-304**	**-81**	**-12**	**23**	**37**	**0.11**	0.17	0.19	0.17	0.19	0.30

for the quality of the human feedback. An open question is why the extensions are much more sensitive to sub-optimal feedback in the MountainCar environment. We suggest this behavior is linked with the different architectural choices for MountainCar since a linear function approximation with RBF-kernels (like in the original paper) is used. In contrast to the DeepTAMER architecture used in LunarLander and minigrid, these changes were necessary due to the more complex state space (and no reference usage in the original paper exists).

Counterfactual states vs. actions. Both extensions, counterfactual actions and counterfactual states, increase the rate of convergence, and no clear preference over one or the other can be made. While the construction of counterfactual actions are technically easier, and they provide more immediate useful feedback (since the agent already knows how to reach state s, it can use the counterfactual action a' the next time the agent reaches it), it is unclear why they do not outperform counterfactual states in the LunarLander environment. We suspect that the preference of one extension over the other could be determined from the size of the observation and action space, which could be investigated in future research. The creation of state-based counterfactual provides more options since, in most cases, the state space is much larger than the actions space. This means that in no cases can the algorithm sample from a multitude of possible counterfactual states. Throughout this work, we only used a simplistic uniform sampling of states, but other more sophisticated sampling methods (for example, to sample state by similarity) might provide even better results. It is reasonable to assume that humans do not randomly create counterfactuals in the state space but with a more define pattern. We take it as future work to investigate how humans actually create counterfactuals in the state space.

Realistic oracles. With our implementation of a synthetic oracle, we took some first steps to make the synthetic oracle more realistic due to the randomness in feedback frequency and the reduced optimality of the feedback, but that is far from a perfect simulation of human feedback. While, in our opinion, this should be a focus for further studies, we believe that our reduced frequency and optimality are realistic enough to ensure that the results of our extensions will still hold in practice.

Handling of counterfactuals. While the results indicate that counterfactual feedback can speed up convergence, the integration of counterfactuals in the actual learning process could be further enhanced. Currently, counterfactuals are treated as additional samples without any special treatment towards other feedback samples. While this solution allows

for flexibility and is compatible with other modifications towards TAMER, more specialized handling of counterfactual feedback like contrasting or auxiliary losses would most likely enhance convergence. The integration of GAN-based approaches and counterfactuals (e.g., (Smith & Ramamoorthy, 2020)) could be another possibility for future research.

Environment & architectural choices. Throughout this work, we selected a single architecture of the H-function (for each environment) for this benchmark and refrained from a broad ablation study. The different environments required minor changes to the configuration in the network's first or last layers, but we refrained from larger changes for the sake of simplicity. Still, the general results that the inclusion of counterfactuals can benefit convergence, especially in early phases, should be similar with different architectures. As explained in section 1, the introduction of samples from the expert distribution (as counterfactuals do) leads to faster convergence in general since the distance between the current distribution and an optimal distribution is reduced. We are confident that our findings should be transformable to even more environments and domains. However, both previous points about the architecture and the environment show the dire need for proven and reliable benchmark methods and environments in Human-in-the-loop Reinforcement Learning. This would allow a more comparable evaluation of new developments in the field.

Conclusion

In this work, we have shown that the inclusion of counterfactual explanations in TAMER can significantly improve the speed of convergence. Both approaches, counterfactuals based on states and counterfactuals based around actions, show these improvements. This increase is evident in the early stages of the training, where counterfactuals can lead to large differences, while in later stages, the differences slowly phase out. Moreover, we performed multiple evaluations with environments and under realistic conditions. The experiments showed that our results hold in settings with realistic parameters. Since both extensions of TAMER (counterfactual states and counterfactual actions) outperform each other in some environments, no clear preference over one or the other can be made. Future work involves better handling of counterfactuals in the training's loop, and an empirical evaluation of real humans and their acceptance of the methods.

References

Arakawa, R., Kobayashi, S., Unno, Y., Tsuboi, Y., and Maeda, S. DQN-TAMER: Human-in-the-loop reinforcement learning with intractable feedback. *ArXiv*, abs/1810.11748, 2018.

Buesing, L., Weber, T., Zwols, Y., Heess, N., Racanière, S., Guez, A., and Lespiau, J. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Byrne, R. M. Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning. In *IJCAI*, pp. 6276–6282, 2019.

Cakmak, M. and Thomaz, A. L. Optimality of human teachers for robot learners. In *2010 IEEE 9th International Conference on Development and Learning*, pp. 64–69, 2010. doi: 10.1109/DEVLRN.2010.5578865.

Celemin, C. and del Solar, J. R. COACH: Learning continuous actions from COrrective Advice Communicated by Humans. *2015 International Conference on Advanced Robotics (ICAR)*, pp. 581–586, 2015.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, California*, 2019.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 2974–2982. AAAI Press, 2018.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. L. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2625–2633. Curran Associates, Inc., 2013.

Guan, L., Verma, M., and Kambhampati, S. Explanation augmented feedback in human-in-the-loop reinforcement learning. In *ICML Workshop on Human in the Loop Learning (HILL)*, 2020.

Jin, P. H., Keutzer, K., and Levine, S. Regret minimization for partially observable deep reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2347–2356. PMLR, 2018.

Kakade, S. M. *On the sample complexity of reinforcement learning.* PhD thesis, Gatsby Computational Neuroscience Unit, 2004.

Knox, W. and Stone, P. TAMER: Training an agent manually via evaluative reinforcement. *2008 7th IEEE International Conference on Development and Learning*, pp. 292–297, 2008.

Knox, W. B. and Stone, P. Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In *AAMAS*, 2010.

Knox, W. B. and Stone, P. Reinforcement learning from simultaneous human and MDP reward. In *AAMAS*, 2012.

Krening, S. and Feigh, K. M. Interaction algorithm effect on human experience with reinforcement learning. *J. Hum.-Robot Interact.*, 7(2), October 2018.

Lu, C., Huang, B., Wang, K., Hernández-Lobato, J. M., Zhang, K., and Schölkopf, B. Sample-efficient reinforcement learning via counterfactual-based data augmentation. In *Proceedings of Neural Information Processing Systems (NeurIPS) Workshop on Offline Reinforcement Learning*, 2020.

Markman, K., McMullen, M., and Elizaga, R. Counterfactual thinking, persistence, and performance: A test of the reflection and evaluation model. *Journal of Experimental Social Psychology*, pp. 421–428, 02 2008.

Mc Grath, R., Costabello, L., Le Van, C., Sweeney, P., Kamiab, F., Shen, Z., and Lecue, F. Interpretable Credit Application Predictions With Counterfactual Explanations. In *NIPS 2018 - Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy*, Montreal, Canada, December 2018.

Menglin, L., Jing, C., Shaofei, C., and Wei, G. A new reinforcement learning algorithm based on counterfactual experience replay. In *2020 39th Chinese Control Conference (CCC)*, pp. 1994–2001, 2020.

Milani, S., Topin, N., Houghton, B., Guss, W. H., Mohanty, S. P., Nakata, K., Vinyals, O., and Kuno, N. S. Retrospective analysis of the 2019 MineRL competition on sample efficient reinforcement learning. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, 2020.

Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Rim, S. and Summerville, A. How far to the road not taken? the effect of psychological distance on counterfactual direction. *Personality & social psychology bulletin*, 40, 11 2013.

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

Smith, S. C. and Ramamoorthy, S. Counterfactual explanation and causal inference in service of robustness in robot control. *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 1–8, 2020.

Stepin, I., Alonso, J. M., Catala, A., and Pereira-Fariña, M. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI07: 12th International Conference on Intelligent User Interfaces, pp. 82–91, 2007.

Vien, N. A., Ertel, W., and Chung, T. Learning via human feedback in continuous state and action spaces. *Applied Intelligence*, 39:267–278, 2012.

Warnell, G., Waytowich, N. R., Lawhern, V., and Stone, P. DeepTAMER: interactive agent shaping in high-dimensional state spaces. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1545–1554. AAAI Press, 2018.