

Accelerating the Learning of TAMER with Counterfactual Explanations

Jakob Karalus
Institute of Artificial Intelligence
Ulm University
 Ulm, Germany
 jakob.karalus@uni-ulm.de

Felix Lindner
Institute of Artificial Intelligence
Ulm University
 Ulm, Germany
 felix.lindner@uni-ulm.de

Abstract—The capability to interactively learn from human feedback would enable agents in new settings. For example, even novice users could train service robots in new tasks naturally and interactively. Human-in-the-loop Reinforcement Learning (HRL) combines human feedback and Reinforcement Learning (RL) techniques. State-of-the-art interactive learning techniques suffer from slow learning speed, thus leading to a frustrating experience for the human. We approach this problem by extending the HRL framework TAMER for evaluative feedback with the possibility to enhance human feedback with two different types of counterfactual explanations (action and state based). We experimentally show that our extensions improve the speed of learning.

Index Terms—Human-in-the-loop Reinforcement Learning, Counterfactuals, Explainability

I. 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. This integration of humans into the training process allows non-experts to train or modify an agent’s behavior naturally.

While integrating humans into the learning process opens up new possibilities, doing so also introduces new challenges. In this work, we address two of them: the integration of an enhanced interface for evaluative feedback and the learning speed of the algorithm. 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 [1], [2]. In most previous work, humans can only give simple binary feedback [3] or control a single dimension [4]. Moreover, human feedback is more costly than automated feedback from the environment or a simulator. State-of-the-art Reinforcement Learning needs millions of episodes with billions of frames (see [5]). 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 their feedback in

training. Therefore, the algorithms’ learning speed is crucial for a positive user experience.

Since the flexible interfaces and learning speed are both crucial for the experience of the human, we want to investigate with this work if the inclusion of counterfactuals in TAMER can be beneficial for the learning speed. Counterfactuals are methods of explanations to describe a causal link between two possible outcomes in the form of: “If event B rather than A had happened, then outcome Y rather than X would have happened”. Since counterfactuals contain more information (due to the second outcome, see Section III-B), we believe they could be a solution to enhance the learning speed. In concrete, we focus on optional upward-directed counterfactuals, which should speed up learning and give humans more flexibility in their feedback when working with TAMER. We show that this type of feedback can be easily added to one of the most prominent methods for evaluative feedback, TAMER.

Source code and evaluations are available under: <https://anonymous.4open.science/r/InteractiveRL-25A1>.

II. RELATED WORK

One of the first proposed frameworks for HRL with evaluative feedback is the TAMER framework [3], which we use as a basis for our work (see chapter III). Throughout this work, we use the standard extension DeepTAMER [6], 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 [7] and continuous TAMER+RL [8] both reintroduce the environmental reward and enable learning from multiple feedback sources. A common trend is introducing Reinforcement Learning frameworks into TAMER: Arakawa *et al.* [9] combine it with Deep-Q learning, and Vien *et al.* [10] 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 assumption and interface for human feedback stay the same. Therefore our work would also be applicable to each of them.

COACH [11] assumes that the feedback given by the human is advantage-based (“How much better is the current policy than the previous one?”), which allows to learn a policy directly. Since the significant difference between COACH and

TAMER are the different assumptions about human feedback (forward-looking versus advantage-based) both variants are targeted toward different types of feedback. While our work is transferable to COACH, the crucially different assumption about how humans formulate feedback makes it hard to compare both methods fairly on a technical basis. At the same time, it is still open which assumptions are preferred by humans. Therefore we refrain from an in-depth comparison of different both approaches.

In contrast to TAMER’s reward-shaping approach, a different approach to HRL is policy-shaping [12]. Policy-shaping uses the feedback directly as labels for the corresponding policy instead of interpreting it as a reward signal. A prominent variant of this is the work of Celemin *et al.* [4] for their COACH framework (a different COACH than the above mentioned), which can learn a policy from corrections. The concept of corrections bears similarities to our counterfactual actions. We refrain from a direct evaluation of counterfactuals in this framework due to the incompatibilities in the action spaces (their correction are on continuous actions, while ours are currently on discrete actions only).

A different avenue for feedback in HRL has been preference-based methods, [13] where humans can give preference for one whole trajectory over another. This type of feedback requires the generation of trajectories, which can limit the application for inherently safe applications or simulated environments. Nevertheless, while our work is not directly applicable, the concepts should be easily adaptable for preference-based methods.

Action Advising [14] is a popular feedback method for HRL, in which the human advises the agent on which actions it should perform next. Conceptually, advising is similar to our counterfactual actions, with the difference that advice is given beforehand instead of afterward. Moreover, a critical difference between Action Advising and our counterfactual actions is the contrastive nature of counterfactuals (see Section III-E). Since our work is motivated by allowing more flexible feedback from humans rather than benchmarking different interfaces for feedback in HRL, we refrain from a technical comparison and refer to [15].

Guan *et al.* [16] 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. In contrast, our current proposed solution is primarily independent of the input shape and the approximator’s chosen architecture.

While counterfactuals have not been used in HRL, other work has successfully used counterfactuals in different areas. Lu *et al.* [17] have shown that an integration of counterfactuals, through the usage of Structural Causal Models (SCM), in Reinforcement Learning (without human feedback) can be beneficial to label efficiency. Work by Jin *et al.* [18], [19] used counterfactuals in partial observable settings to increase the learning speed, and robustness of their Reinforcement

Learning approaches. Foerster *et al.* [20] use a counterfactual loss in a multi-agent reinforcement setting to increase their performance. Menglin *et al.* [21] 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 formulate a general method for the inclusion of counterfactuals in HRL.

III. BACKGROUND AND METHODS

This Section explains the relevant background and highlights our extension of the existing method. First, we will explain TAMER, introduce counterfactuals, and investigate the convergence of TAMER from a Reinforcement Learning perspective. Afterward we use this perspective as motivation to introduce counterfactuals in TAMER.

A. The TAMER Framework

Human-in-the-loop Reinforcement Learning (HRL) allows the agent to learn from human feedback directly and incrementally, even in environments with no other reward signal. A major difference of human feedback is the potential inconsistency and sub-optimality of human feedback. Humans can give feedback at various stages throughout the training, but the agent cannot depend on a fixed and reliable reward (unlike the standard environmental reward). HRL is also advantageous in very complex environments where a specification of a solid reward function is challenging since even slight misspecifications of the reward function can lead to unexpected side-effects in actual use cases [22].

In this work, we use TAMER, a framework for evaluative feedback. In this scenario, after the agents performed an action a in a specific state s , the human has the opportunity to give his evaluative feedback f towards the agent. No additional reward function is given and the agent has to learn a policy from the human reward. In most work the range of the feedback is restricted towards discrete feedback of -1 and $+1$.

TAMER [3] allows to learn a policy with human evaluative feedback by assuming that the human already considers the long-term solution in his feedback. This assumption lets us greedily learn a function that maps states to their expected feedback: ($H : S \times A \rightarrow F$). In TAMER, mapping H is learned immediately with a single *state/action* pair when feedback is given. This learned function of the human reward (H) is used to greedily select actions with the highest expected human reward at each time-step, i.e., $\arg\max_a (H(s, a))$. Therefore, TAMER is conceptually similar to Q-Learning with the crucial difference that it assumes long-term feedback and therefore does not have the credit assignment problem. TAMER does not use any environmental reward, methodical exploration, or longer-term planning mechanism.

DeepTAMER [6] extends TAMER by using Deep Neural Networks as the approximator for the H function. The network architecture consists of shared base layers and individual fully-connected heads for each action. Additionally, a replay buffer (which stores past feedback/states) is used to stabilize

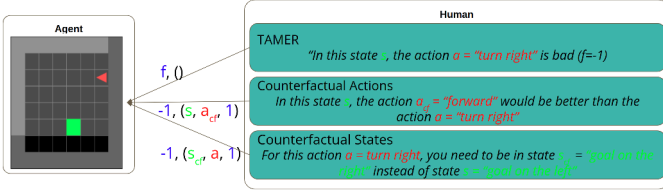


Fig. 1. Left: Visualization of the MiniGrid environment with the goal as a green square. The highlighted area indicates the current viewpoint of the agent. Right: Verbalization of the various feedback mechanism in natural language.

training. In our work, we will roughly follow the architecture of DeepTAMER, with minor modifications due to different state space and action space sizes.

B. 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” [23]. Counterfactuals thus refer to an event A that actually has happened and caused some outcome *fact* X , and to the *contrast* B , which would have resulted in a different outcome (the *foil*) Y [24]. It is important to note that a counterfactual does not only provide a second data point—the *contrast-foil pair*—, but also the *contrast relationship* between the fact and the foil is an equally important property. We will make use of both these properties, that is, we treat the foil as an additional example and exploit the contrastive relationship between fact and foil to achieve faster learning.

As multiple studies [25], [26] show, 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 [23]. We therefore 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_{cf} rather than in state s , then my feedback would have been positive rather than negative” and “If action a_{cf} had been performed s rather than action a , then my feedback would have been positive rather than negative.” This allows the agent to learn how to get from a state with bad feedback into a state with good feedback.

C. The convergence of Reinforcement Learning algorithms

In HRL, learning speed is particularly essential because humans are asked to give feedback to the learner. This is a tedious task we naturally want to minimize. An agent which learns too slowly or shows no progress can limit the motivation of the human to give further feedback [27]. 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 [28]):

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

The optimal policy is denoted by π^* , $\pi^*(s)$ is the visited state distribution of π^* , and $\pi^*(a | s)$ is the action distribution of the optimal policy. Accordingly, 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 an ample 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 the expert would reach, but doing that is essentially the goal of learning.

The previously introduced concept of positively-directed counterfactuals comes directly from the expert distribution since it indicates which part of the current $\langle state, action \rangle$ pair should have been different. This should allow the agent to reduce the difference $V^{\pi^*} - V^{\pi}$, which leads to faster learning. 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 faster learning. 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.

D. Using counterfactuals in TAMER

To introduce counterfactuals into TAMER, 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*. In the following we will denote counterfactual actions or states with the subscript s_{cf} and a_{cf} . When asking the human for feedback with $gather_feedback(s, a)$ the subscripts s_{t-1} , a_{t-1} indicate the last state or action performed (since the human gives their feedback after an action is performed, so the agent is already in a new state).

A counterfactual state provides a state where the action would be a good one. The feedback $gather_feedback(s_{t-1}, a_{t-1}) = (-1, \langle 1, s_{cf}, a \rangle)$ can be read as “Action a would be good if you were in state s_{cf} instead of s ”. In case of counterfactual actions, $gather_feedback(s_{t-1}, a_{t-1}) = (-1, \langle 1, s, a_{cf} \rangle)$ can be described in natural language with: “In state s , action a_{cf} would be better than action a ”. Both types of counterfactuals are exemplified in Figure 1.

To include these counterfactuals in TAMER, we extend the human feedback function to return an optional triple $\langle f_{cf}, s_{cf}, a_{cf} \rangle$. This triple contains the counterfactual feedback f_{cf} and either s_{cf} or a_{cf} contains a counterfactual version of the original input, depending on the type of counterfactual feedback provided. When provided by the human, we use the triple for an additional update of H , combined with the

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
     $f, \langle f_{cf}, s_{cf}, a_{cf} \rangle \leftarrow$  gather feedback ( $s_{t-1}, a_{t-1}$ )
    if ( $f_{cf}$ ) then
      update  $H$  with  $L(s_{t-1}, a_{t-1}, f, s_{cf}, a_{cf}, f_{cf})$ 
    else
      update  $H$  with  $\langle f, s_{t-1}, a_{t-1} \rangle$ 
    end if
     $a_t \leftarrow \operatorname{argmax}_a (H(s, a))$ 
    perform action  $a_t$ 
  end for
end for
```

standard TAMER update $\langle f, s, a \rangle$. The full procedure with all steps is also described in Algorithm 1.

E. Using the contrastive nature of counterfactuals

Besides using counterfactuals as additional data points, we will also leverage another essential property of counterfactuals: their natural contrast between the *fact* and the *foil*. This contrast allows us to introduce an additional term with contrastive constraints in the loss function to distance the activations of each state-action pair from each other.

In Equation 2, our contrastive loss is shown, where $E(s, a)$ is the output (or embedding) of the last-hidden layer, which is common to the counterfactual and feedback pair (depending on the respective architecture for the H-Model). This loss term should speed up the learning process due to the additional information contained in the gradients.

$$L_{cos}(s_{t-1}, a_{t-1}, s_{cf}, a_{cf}) = \max(0, \cos(E(s_{t-1}, a_{t-1}), E(s_{cf}, a_{cf}))) \quad (2)$$

With this contrastive term, the complete loss (see Equation 3) can be built as a simple combination of all three loss terms. The L_{normal} and the L_{cf} could be any loss classes, in our case, they both refer to a L2-loss.

$$L(s_{t-1}, a_{t-1}, f, s_{cf}, a_{cf}, f_{cf}) = L_{normal}(s_{t-1}, a_{t-1}, f) + L_{cf}(s_{cf}, a_{cf}, f_{cf}) + L_{cos}(s_{t-1}, a_{t-1}, s_{cf}, a_{cf}) \quad (3)$$

IV. EVALUATION

For our evaluation, we will use TAMER as the baseline, and compare it to the proposed additions of counterfactuals states (TAMER+CFS) and counterfactual actions (TAMER+CFA). First, we evaluate the performance of each variant in different environments and show that our changes lead to faster learning. Afterwards, we introduce changes in the feedback frequency and in the quality of the feedback to show that the

improvement with counterfactual feedback also holds under more realistic feedback conditions. Finally, we perform some simple ablations to evaluate our choices regarding the direction of counterfactuals and their benefit over random samples.

A. Environment Setup & Measurements

We run the evaluations in multiple environments (CartPole, LunarLander, Minigrid¹). We changed the MiniGrid environment, so that the goal position is random (instead of a static goal at the bottom-right corner.) We selected an architecture similar to the one used in DeepTAMER [6].

To evaluate the performance of the learning agent, we use the environmental reward as a measure of success. 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, at fixed steps, we run 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 learning of an agent throughout its training. Since the learning of HRL can be highly unstable, we perform multiple training runs per algorithm/environment with different random seeds to initialize the agent and the environment. To allow for easier comparisons, we normalize the rewards at the end of each episode so that a (perfect) solution results in the total reward of 1, while random agents get a 0 reward (on average). Since we want to enable faster learning and therefore want to look at the whole training process (instead of the performance after the training), we selected the optimality gap (the area between the current performance and a perfect performance) as a metric for the evaluation. This metric enables us to evaluate the performance of our variants in a statistically sound way, without the pitfalls of showing large amounts of metrics in large tables or interpreting graphs, as discussed by Agarwal *et al.* [29].

With our scores already normalized, an utterly random agent would receive an optimality gap of 1, while a perfect agent would produce a 0. Additionally, we compute 95% confidence intervals through bootstrapping (as proposed by [29]) in each plot, indicated by the shaded area.

B. Human-Feedback Oracle

While HRL aims to incorporate human feedback into the training loop, it can be costly to collect human feedback. This cost can hinder the development and evaluation of HRL algorithms since evaluations are often performed with only a few humans, leading to false interpretations due to statistical randomness. 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 (like other prior work, e.g., by Christiano *et al.* [30]). To achieve this, we trained an agent for the oracle with proximal policy optimization (PPO) [31] algorithm on every environment for

¹<https://github.com/maximecb/gym-minigrid>

sufficiently many steps to solve the tasks. We then use this policy to give dynamically calculated (counterfactual) feedback for the actual learning algorithm.

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. With the “feedback frequency” parameter, we control how often the oracle gives feedback towards the agent to archive inconsistent feedback. With the “feedback quality” parameter, we allow the oracle to give random feedback, which results in lower-quality feedback.

C. Results

As shown in Figure 2, TAMER extended with counterfactuals leads to faster learning than the original TAMER. The main advantage is at the early stages of training, where learning is significantly faster than the original TAMER. At later stages of the training, the difference shrinks when all variants (vanilla TAMER and our extensions) reach the maximum reward plateau (with enough training steps).

To highlight the faster learning, we measure the absolute performance of each variant at regular intervals for each environment and calculate the optimality gap with these measurements. The results can be seen in Figure 3, which shows that both of our methods learn faster than vanilla TAMER. Counterfactual actions perform better than counterfactual states, since counterfactual actions outperform vanilla TAMER, while counterfactual states struggle in some environments.

Reduced feedback frequency. Perfect feedback rates are not a realistic assumption for HRL. Hence, we investigate our extensions under more realistic conditions: we reduce the frequency in which the oracle gives feedback. This mimics one essential characteristics of HRL, namely that the feedback is not deterministic and can be inconsistent and sporadic [27]. To achieve this, we change the feedback frequency of our oracle and evaluate our extension under this changed condition. As shown in Figure 3, the results show that even with reduced feedback rates, the general pattern of an increase in learning with the inclusion of counterfactuals is similar to the results with a perfect feedback rate. We noted that some environments suffer more from the reduced rate of feedback.

Reduced feedback quality. Another essential characteristic of human feedback is that it does not have to be optimal at every point in time. Next, we evaluate the learning behavior for lower-quality feedback. We try to approximate these with the synthetic oracle (see Section IV-B), for which set the “feedback quality” parameter to 0.75. The results show that counterfactual actions (CFA) still work with lower quality and outperform the baseline. Naturally, the learning is slower with less optimal feedback since the “signal strength” is lower. It can be noted that counterfactual states struggle to perform on a same level than the baseline.

Downward Counterfactuals The usage of upward counterfactuals was motivated by the performance differences lemma (see section III-C and the preference of humans. We include

a downward counterfactual condition in the evaluation to evaluate whether the upward property is essential.

The results can be seen in Figure 4, for which we calculate on each comparison the probability that one variant leads to a better optimality gap than the other. Using downward counterfactuals still improves the learning compared to vanilla TAMER, but upward-directed counterfactuals lead to faster learning than downward counterfactuals.

Random Samples Since we treat counterfactual explanations as additional samples, one could argue that, by counterfactual explanations, users just submit more samples to the learning agent, and therefore the learning process converges faster. To control for this possibility, we evaluate a Random condition. In this condition, the oracle provides a random state/action tuple, which is correct but totally unrelated to the current state. In this case we disable the contrastive loss, since there is no guarantee for a contrast between the current and the random tuple. Figure 4 shows that indeed the counterfactual samples perform superior to the random samples.

D. Discussion

Faster learning. We extended TAMER to allow the human to give additional counterfactual feedback and evaluated our extension in various environments and under multiple conditions. The results indicate that the inclusion of counterfactuals increases the learning in every condition. Thus, counterfactual feedback is handled more label efficient than simple binary feedback or random explanations. This behavior confirms the expectation stated in section III-C: In the early stages of training, a large difference between the agent and the expert (human) policy exists. Therefore, introducing a single example from the expert policy can have a significant 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 learning is that the agent sees less and less upward counterfactual in the learning process since we only give counterfactual feedback in negative cases, which occur fewer in later stages of the training.

Less and lower quality feedback. In cases where the feedback is of lower quality (which also makes the counterfactual feedback of lower quality), our counterfactual actions still proves to increase the rate of learning. Our counterfactual states struggle with reduced feedback quality. This is due to our construction of the counterfactual feedback. Usually, the state space is much larger than the action space. Since our lower quality feedback for counterfactuals randomly samples the respective state or action space (for either counterfactual states or actions), this leads to a lower signal strength in counterfactual states. Further research could be done to determine acceptable lower bounds for the quality of human feedback.

Counterfactual states vs. actions. Both extensions, counterfactual actions and counterfactual states, increase the rate of learning, with counterfactual actions (CFA) providing better results than counterfactual states (CFS). We reason this is because counterfactual actions provide more immediate useful feedback than counterfactual states feedback. This, in turn,

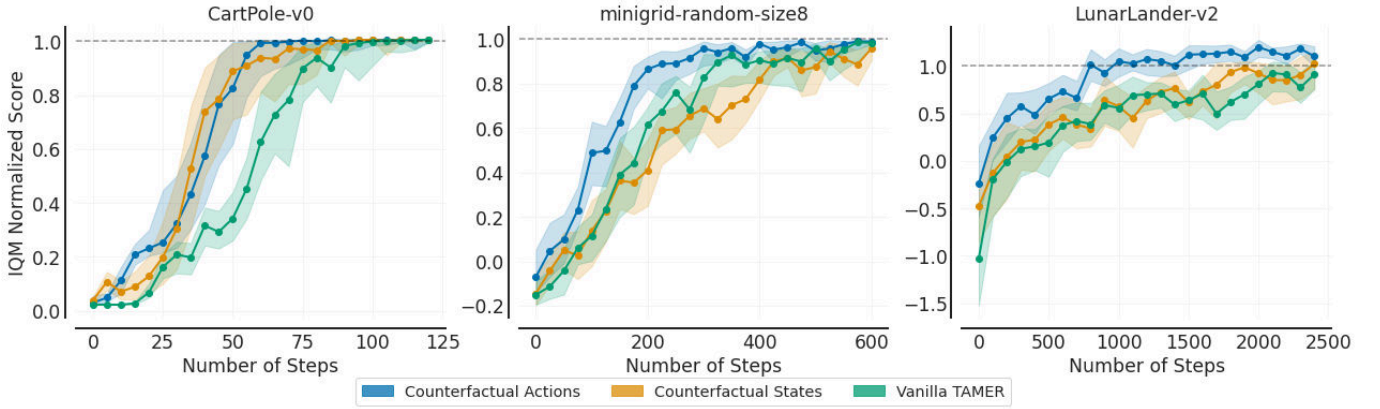


Fig. 2. Interquartile mean reward of each feedback method on the evaluation set throughout training. Bold line is the IQM reward of random seeds ($n=25$), area is the 95% confidence interval.

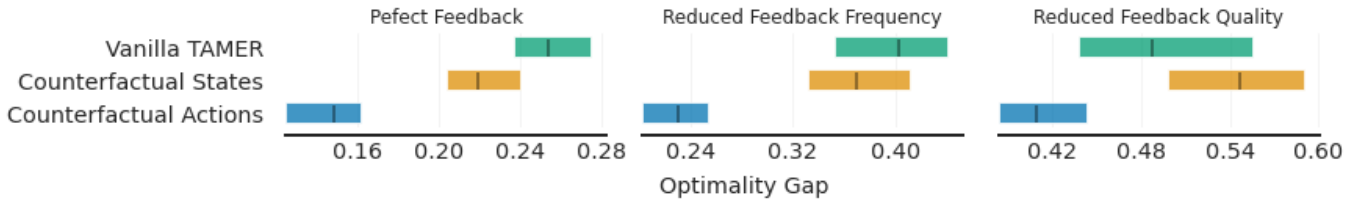


Fig. 3. Optimality Gap with confidence intervals on all tasks. (Lower means better)

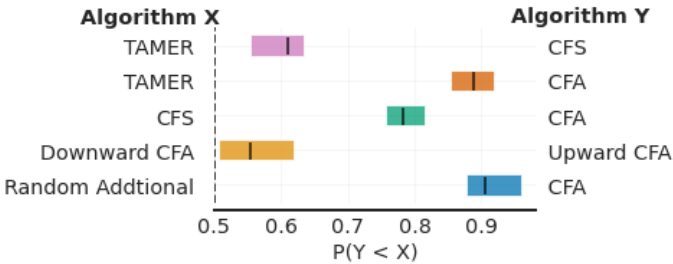


Fig. 4. Placeholder Probability of Improvement, calculated on the optimality gap, so a improvements means a lower score $P(Y < X)$, for various ablations

can be explained since with counterfactual actions, the agent already knows how to reach state s , and it can use the counterfactual action a_{cf} the next time the agent reaches it.

Upward Counterfactuals & Random Feedback Our main focus in this work was on upward counterfactuals. Since they come directly from the expert policy (see III-C), we expected them to perform better than downward directed counterfactuals. Additionally, as discussed, upward counterfactuals are preferred by humans [25], [26]. Nevertheless, we evaluated the impact of downward counterfactuals. To no surprise, they perform not as well as upward counterfactuals but still benefit the learning. Similarly, we evaluated the case when additional random feedback samples are given, which performed even worse. This highlights that using explanation concepts can provide more natural interfaces towards the human and provide

more benefits than just simple samples. In this work we have shown that counterfactuals, especially upward-directed ones, provide a clear benefit from a learning perspective while giving the human more freedom. We think these results can be seen as motivation to investigate more explanations concepts.

Realistic oracles. With our implementation of a synthetic oracle, we took a step to make the synthetic oracle more realistic due to the randomness in feedback frequency and the reduced optimality of the feedback. This is far from a perfect simulation of human feedback but the oracle also allowed us to run experiments with a large sample size in multiple environments with ablations. We believe that our reduced frequency and optimality are realistic enough to ensure that the results of our extensions will still hold in practice.

Interfaces for counterfactual feedback. Throughout our work, we have shown that counterfactuals are beneficial for learning. However, it is still an open question how humans would actually give this kind of feedback to the agents. While the handling of more complex feedback is definitely a complex problem, one should note there are many instances when humans naturally want to provide more complex feedback. We argue that, depending on the use case, humans can be enabled to give complex feedback through better interfaces. For example, an NLP-based parsing of utterances could extract counterfactuals from humans. Alternatively, for environments like MiniGrid, one could provide humans with a graphical interface, where they can not only give feedback but also change the state (by dragging objects around) to produce counterfactual states.

Feedback Cost Our results show that the inclusion of counterfactuals leads to faster learning in terms of fewer steps in the environment needed before it is ‘solved’. This also leads to fewer feedback interactions from humans. While this is a preferable result, counterfactual feedback could also lead to additional mental workload by the human. This load was not considered in our evaluation. However, studies suggest that in many situations humans naturally want to enrich their feedback with additional explanations (see [1], [2]), without much thinking. Our method gives humans more flexibility in their feedback and allows them to harness it. We believe that the additional mental load depends on the environment and the feedback interface and has to be evaluated on a case-by-case basis. Additionally, we think it could benefit the user experience if the user could express his feedback differently.

CONCLUSIONS

We have shown, with the help of synthetic oracle, that the inclusion of upward counterfactual explanations in TAMER can significantly improve the speed of learning as compared to simple binary feedback, random feedback, or downward counterfactual feedback. Both approaches, counterfactuals based on states and counterfactuals based on actions, show these improvements. Overall counterfactual actions delivered the lowest optimality gap, speeding up the learning the most, while counterfactual states still provide benefits over TAMER without counterfactuals. We tried to circumvent the limitations of the synthetic oracle with experiments in which the quality of the oracle had been reduced. The results of these experiments showed that the improvements should also hold under more realistic conditions. Nevertheless for definite results further studies with human trainers are needed. Future work will investigate how users actually use these enhanced feedback interfaces and how costly for the human they are.

REFERENCES

- [1] S. Krening and K. M. Feigh, “Interaction algorithm effect on human experience with reinforcement learning,” *J. Hum.-Robot Interact.*, vol. 7, no. 2, 2018.
- [2] S. Stumpf, V. Rajaram, L. Li, M. Burnett, T. Dietterich, E. Sullivan, R. Drummond, and J. Herlocker, “Toward harnessing user feedback for machine learning,” in *Proc. of the 12th Int. Conf. on Intelligent User Interfaces*, 2007, p. 82–91.
- [3] W. B. Knox and P. Stone, “TAMER: Training an agent manually via evaluative reinforcement,” *2008 7th IEEE Int. Conf. on Development and Learning ICDL*, pp. 292 – 297, 2008.
- [4] C. Celemin and J. R. del Solar, “COACH: Learning continuous actions from COrrective Advice Communicated by Humans,” *2015 Int. Conf. on Advanced Robotics (ICAR)*, pp. 581–586, 2015.
- [5] S. Milani, N. Topin, B. Houghton, W. H. Guss, S. P. Mohanty, K. Nakata, O. Vinyals, and N. S. Kuno, “Retrospective analysis of the 2019 MineRL competition on sample efficient reinforcement learning,” in *Proc. of the NeurIPS 2019 Competition and Demonstration Track*, 2019.
- [6] G. Warnell, N. R. Waytowich, V. Lawhern, and P. Stone, “DeepTAMER: interactive agent shaping in high-dimensional state spaces,” in *Proc. of the 32nd AAAI Conference on Artificial Intelligence*, no. 1, 2018.
- [7] W. B. Knox and P. Stone, “Combining manual feedback with subsequent MDP reward signals for reinforcement learning,” in *Proc. of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems AAMAS*, 2010, pp. 5–12.
- [8] —, “Reinforcement learning from simultaneous human and MDP reward,” in *Proc. of the 11th Int. Conf. on Autonomous Agents and Multiagent Systems AAMAS*, 2012, p. 475–482.
- [9] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S. Maeda, “DQN-TAMER: Human-in-the-loop reinforcement learning with intractable feedback,” *ArXiv*, vol. abs/1810.11748, 2018.
- [10] N. A. Vien, W. Ertel, and T. C. Chung, “Learning via human feedback in continuous state and action spaces,” *Applied Intelligence*, vol. 39, no. 2, pp. 267–278, 2012.
- [11] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman, “Interactive learning from policy-dependent human feedback,” in *Proc. of the 34th Int. Conf. on Machine Learning, ICML 2017*, 2017, p. 2285–2294.
- [12] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. L. Thomaz, “Policy shaping: Integrating human feedback with reinforcement learning,” in *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [13] R. Akrou, M. Schoenauer, and M. Sebag, “Preference-based policy learning,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011, pp. 12–27.
- [14] L. A. Torrey and M. E. Taylor, “Teaching on a budget: agents advising agents in reinforcement learning,” in *Int. Conf. on Autonomous Agents and Multi-Agent Systems, AAMAS ’13*, 2013, p. 1053–1060.
- [15] P. Koppol, H. Admoni, and R. Simmons, “Interaction considerations in learning from humans,” in *Proc. of the Thirtieth Int. Joint Conf. on Artificial Intelligence, IJCAI-21*, 2021, pp. 283–291.
- [16] L. Guan, M. Verma, and S. Kambhampati, “Explanation augmented feedback in human-in-the-loop reinforcement learning,” in *ICML Workshop on Human in the Loop Learning (HILL)*, 2020.
- [17] C. Lu, B. Huang, K. Wang, J. M. Hernández-Lobato, K. Zhang, and B. Schölkopf, “Sample-efficient reinforcement learning via counterfactual-based data augmentation,” in *Proc. of Neural Information Processing Systems (NeurIPS) Workshop on Offline Reinforcement Learning*, 2020.
- [18] P. H. Jin, K. Keutzer, and S. Levine, “Regret minimization for partially observable deep reinforcement learning,” in *Proc. of the 35th Int. Conf. on Machine Learning, ICML 2018*, 2018, pp. 2342–2351.
- [19] L. Buesing, T. Weber, Y. Zwols, N. Heess, S. Racanière, A. Guez, and J. Lespiau, “Woulda, coulda, shoulda: Counterfactually-guided policy search,” in *7th Int. Conf. on Learning Representations, ICLR 2019*, 2019.
- [20] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” in *Proc. of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [21] L. Menglin, C. Jing, C. Shaofei, and G. Wei, “A new reinforcement learning algorithm based on counterfactual experience replay,” in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 1994–2001.
- [22] G. Dulac-Arnold, D. Mankowitz, and T. Hester, “Challenges of real-world reinforcement learning,” in *Proc. of the 36th Int. Conf. on Machine Learning (ICML)*, 2019.
- [23] R. M. J. Byrne, “Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning,” in *Proc. of the Twenty-Eighth Int. Joint Conf. on Artificial Intelligence, IJCAI-19*, 7 2019, pp. 6276–6282.
- [24] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.
- [25] K. Markman, M. McMullen, and R. Elizaga, “Counterfactual thinking, persistence, and performance: A test of the reflection and evaluation model,” *Journal of Experimental Social Psychology*, pp. 421–428, 2008.
- [26] S. Rim and A. Summerville, “How far to the road not taken? the effect of psychological distance on counterfactual direction,” *Personality & social psychology bulletin*, vol. 40, pp. 391–401, 2013.
- [27] M. Cakmak and A. L. Thomaz, “Optimality of human teachers for robot learners,” in *2010 IEEE 9th Int. Conf. on Development and Learning ICDL*, 2010, pp. 64–69.
- [28] S. M. Kakade, “On the sample complexity of reinforcement learning,” Ph.D. dissertation, Gatsby Computational Neuroscience Unit, 2004.
- [29] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare, “Deep reinforcement learning at the edge of the statistical precipice,” in *35th Conference on Neural Information Processing Systems NeurIPS*, 2021.
- [30] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” in *31st Conference on Neural Information Processing Systems NeurIPS*, 2017.
- [31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.