Policy Presentation
Policy Summarization and Interactive Learning

Fabian Damken (fabian.damken@stud.tu-darmstadt.de)

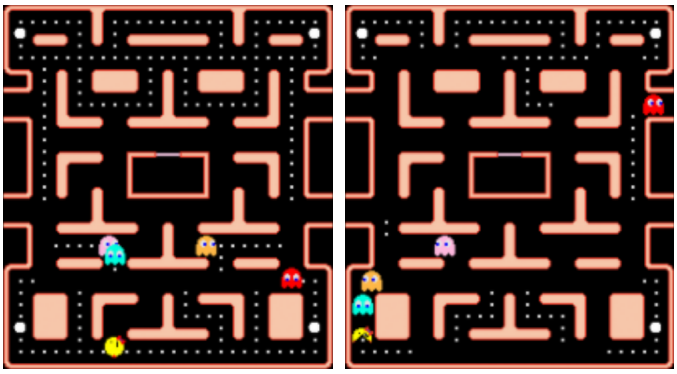
***Abstract*—Although intelligent systems outperform human experts in a variety of tasks, deployment in high-stakes domains such as medicine or finance are still a great risk due to their inherent inexplainability, and lack of trust. This is especially prevalent for deep and stochastic approaches. Additionally, while deep approaches are extremely powerful, their training is time-consuming. The first part of this report therefore explores techniques for explaining a policy to a human by providing a meaningful *summary* of the agent's behavior. Subsequently, methods for *interactive learning* are studied in detail which tackle the problem of long training times by incorporating human feedback. This literature review revealed that these two areas have developed independently. However, combining both may be advantageous and provide substantial insights for development in both fields.**

I. Introduction

While it has been shown that Intelligent Agents (IAs) outperform humans in various tasks, human collaboration and trust are hardly achievable with agents that do not explain the rationales behind their actions [1]. Even though humans can become accustomed to a robot's actions after some time, explanations of them can drastically speed up this process [2]. This is especially important in high-stakes disciplines like finance, medicine, and autonomous driving. However, generating such explanations is extremely difficult, especially in the era of (Deep) Reinforcement Learning ((D)RL) and stochastic policy representations and remains an active area of research. The first part of this report deals with special kinds of Explainable AI (XAI) frameworks, namely policy summaries. Instead of explaining immediate actions taken, policy summarization attempts to give a global overview of the policy. This enables the human collaborator to judge whether the policy is pertinent for a task at hand and whether it is trustworthy. This framework and various realizations are discussed in section III.

Another challenging problem arising when using data-driven methods for building control policies is the enormous amount of interaction with the environment needed to learn useful control laws. As humans usually hold great knowledge about the world and about the task a robot has to learn to solve, this raises the question of whether it would be helpful to incorporate their knowledge into the training process. This idea lead to the development of a family of methods for imitating policies from (human) demonstrations, for example Behavioral Cloning (BC) [3] and Inverse Reinforcement Learning (InvRL) [4], [5]. However, this requires an optimal execution of the task. As this is not always possible, it is desirable to interactively teach an agent *while it is learning* without having to provide trajectories of an optimal policy beforehand. Different methods for interactive learning are explored in the second part of this report, section IV.

Throughout the report, connections are drawn between the two fields. Additionally, the literature review revealed some research gaps that are also highlighted. Finally, the findings and the aforementioned research gaps are summarized and discussed in section V.



(a) Low Importance (b) High Importance

Figure 1: Illustration of the importance metric used by HIGHLIGHTS [6] in the game Mrs. Pacman [7]. (a) is a scene of low importance as all applicable actions (going up, left, or right) lead to states of similar value. (b) is instead of high importance as choosing the "up"-action would be severe. Taken from Amir and Amir, "HIGHLIGHTS: Summarizing Agent Behavior to People." [6]

II. Related Work

Previous surveys have been conducted that include policy summarization and interactive learning [8]–[11]. Cruz and Igarashi [8] published a survey solely on Interactive Reinforcement Learning (IntRL) methods. They identify and describe guidelines for designing interactions to provide hints for approaching future research. Also, they highlight open challenges in IntRL that should be tackled in the future. Najar and Chetouani [9] conducted a survey on RL with human advice, too. They provide an overview over the taxonomy used in IntRL and highlight several existing methodologies to interpret the advice provided by a human. Both surveys give a general overview of the field while this report focuses on accelerating training of an IA using human reinforcement signals. Puiutta and Veith [12] focus on Explainable RL (XRL) in general and give an overview of current methods. On the other hand, Wells and Bednarz [11] give a thorough overview over a multitude of fields in XRL, featuring policy summarization and human collaboration in particular. These two surveys are close to this work. However, this report focuses merely on policy summaries and interactive learning and not XRL in general.

III. Policy Summarization

In order to deploy IAs in high-stakes domains such as autonomous driving, it is necessary to understand the rationale behind their actions. However, explanations of single actions (*local* explanations) are not sufficient for grasping the policy as a whole. While such explanations are helpful in domains where actions are taken at decent pace (picking university courses, for example [13]), real-time tasks require a more general view that can be studied before deployment. This shifts the view towards policy summarization techniques which aim to provide global explanations.

This section primarily explores techniques based on *trajectory extraction* (subsection III-A). Trajectory extraction methods summarize the policy by providing *meaningful* examples of trajectories that the policy outputs. These can, for example, be trajectories executed during training or trajectories generated only for the sake of using them in a summary.

Other methods (not explored in detail here) are *textual summaries* [13]–[15] which, while being more flexible and diverse, mostly focus on generating local explanations rather than providing a global policy summary. While this allows flexible knowledge acquisition, a user has to think about situations that might occur to query the actions that will be taken beforehand. This issue can be circumvented by pre-computing all possible queries to generate a global summary, but this approach is only feasible if the number of actions is manageable.

Additionally, a variety of other approaches are examined in subsection III-B, some of which are a combination of the aforementioned methods.

A. Trajectory Extraction

The most challenging task in extracting meaningful trajectories from a collection of trajectories is to decide what *meaningful* means. This definition is the principal distinctive feature between the presented approaches. In the following, three approaches to policy summarization via trajectory extraction are presented: the first [16] relies on hand-crafted metrics to assess how interesting a trajectory is and the second and third [17], [18] leverage Imitation Learning (IL) for finding trajectories that optimally support learning.

1) Min-Max Span of Q-Function: The *HIGHLIGHTS* algorithm [6] uses the notion of *state importance* [19] defined as the difference between the maximum and minimum value of the Q-function:

$$I^\pi(s) := \max_{a \in \mathcal{A}} Q^\pi(s,a) - \min_{a \in \mathcal{A}} Q^\pi(s,a).$$

An advantage of this metric is that it is easy to grasp: if some state s is of high importance, i.e., the min-max span is wide, it is of high relevance which action is chosen since it potentially has drastic consequences. An illustration of this metric is given in Figure 1.

The importance metric is then leveraged as follows: instead of just extracting states that exceed some importance threshold I_s, a fixed number of states before and after the critical state are extracted, forming a trajectory. This process is repeated until a fixed number of simulations have been run and the most important trajectories are kept (the trajectory generation takes place *after* training and non-important states can be discarded early making the algorithm less costly in terms of memory usage).

Amir and Amir [6] conducted a user study comparing the summaries generated by HIGHLIGHTS against two baselines: one which presents the first k trajectories that take place and one where k random trajectories from the agent's buffer are shown. The participants were asked to a) select which agent they would pick to play on their behalf and rate their confidence and b) rate their opinions on the helpfulness of the summaries. They found that the accuracy of selecting the better agent as well as the confidence of doing so was significantly elevated using HIGHLIGHTS summaries. Similar results where

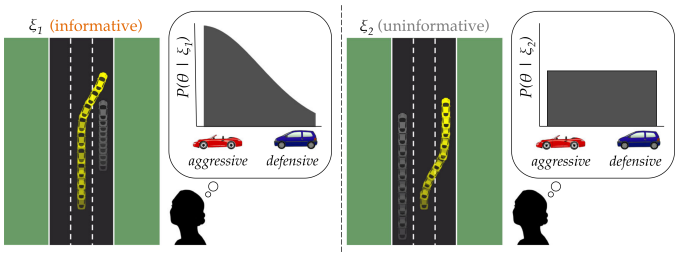


Figure 2: Autonomous driving environment used for the user study in [17]. The autonomous agent is depicted in yellow. It is expected that the right trajectory is less meaningful as both aggressive and defense driving styles would execute such a policy. Conversely, the left trajectory conveys more information and hence should be preferred for a meaningful policy summary. Taken from Huang et al., "Enabling Robots to Communicate Their Objectives." [17]

found regarding the preference for the HIGHLIGHTS summaries.

One of the major advantages of HIGHLIGHTS compared to the approaches presented below is its inherent simplicity: the amount of work to be done and code to be written is rather minimal allowing rapid deployment.

2) Inverse Reinforcement Learning: A different approach for selecting the summarization trajectories was proposed by Huang et al. [17]. Building on the hypothesis that the objective function itself conveys the most information (as an optimal policy constantly aims to maximize the objective), an *algorithmic teaching* approach is used to extract the trajectories. That is, a model of human inference is leveraged to assess the helpfulness of a trajectory.

To build this model, Huang et al. [17] first assume that the reward function is a linear function of known features governed by some parameters θ which shall be inferred[1]. Second, a Bayesian inference model over the parameters θ and the set of example trajectories[2] $\mathcal{T}$ is employed to model a human's belief update:

$$P(\theta \mid \mathcal{T}) \propto P(\theta)\, P(\mathcal{T} \mid \theta) = P(\theta) \prod_{\tau \in \mathcal{T}} P(\tau \mid \theta).$$

[1]Note that, although it is framed as the user inferring the parameters itself, the user is not required to name concrete numbers but just has to have a basic understanding of the weighting.

[2]Note that the trajectories are assumed to be optimally executed in the respective circumstances. Hence, it can, as done by Huang et al. [17], also be stated that the problem is to find a set of environments $\mathcal{E}$ such that executing the optimal policy on each of the environments would yield $\mathcal{T}$. This report sticks to the wording of selecting optimal trajectories for consistency.

To find the trajectories best suited for inferring the objective, the posterior over the actual parameters θ^* is maximized w.r.t. the set of trajectories $\mathcal{T}$.

Modeling the likelihood $P(\tau \mid \theta)$ can be done in a straightforward fashion using exact-inference InvRL [4], [17]. However, while machines are capable of performing exact-inference, humans likely perform approximate-inference [17]. Hence, Huang et al. [17] tested six approximate-inference models "by manipulating two factors in a 2-by-3 factorial design" [17] in addition to InvRL.

The first factor is a deterministic versus probabilistic effect on selecting θ: in InvRL, a value of θ is either kept in the set of possible values or thrown out, depending on whether the example trajectories are optimal w.r.t. the respective value. In the deterministic setting, this is kept as is, however, the decision on whether a trajectory is optimal is relaxed. In the probabilistic setting, no θ is eliminated. Instead, a lower probability is assigned to it when the trajectories are far away from the modeled optimum. The distribution $P(\tau \mid \theta)$ can, in both cases, be formulated to depend on some distance measure $d(\tau_{\theta^*}, \tau_\theta)$ between an optimal trajectory τ_{θ^*} and an example trajectory τ. Three different distance measures are studied throughout the user trial, constituting the second factor:

1) *Reward-Based:* difference of the total rewards
2) *Euclidean-Based:* average Euclidean distance of all states experienced in the trajectory
3) *Strategy-Based:* all trajectories are clustered into strategies (aggressive vs. defensive, for example) and the distance is defined as 0 if two trajectories are in the same cluster and as ∞ otherwise[3]

The authors conducted a user study which focuses on a simulated autonomous driving environment, depicted in Figure 2. First, the participants were presented the generated summaries and subsequently four random trajectories, polling which is the autonomous car along with the confidence of that decision. They found that, as expected, approximate-inference models outperform exact-inference with statistical significance. However, it is extremely relevant for the outcome which approximate-inference model is used: the deterministic Euclidean-based model outperformed all other models by far.

The results found are promising towards machine teaching and successful policy summarization using a

[3]Note that with the strategy-based distance, the deterministic and probabilistic models become equivalent.

principled approach of choosing example trajectories instead of hand-crafting a metric.

3) Imitation Learning: The approach by Lage et al. [18] is analogous to Huang et al. [17] but leverages IL using a Gaussian random field model and active learning [20] instead of InvRL. They used a random grid world environment [21], a Pacman environment, and an HIV simulator [22] for evaluating the performance. However, they did not conduct a user study but instead cross-matched the different models and evaluated how well some model can infer the policy while the algorithmic teaching assumes another model. They found that the reconstruction accuracy is best for the same model if the teaching algorithm assumes the correct model and is poor if the models mismatch.

This result is in line with the results found by Huang et al. [17] where the exact-inference model that mismatches human's inference exhibits poor performance. It also highlights that further studying of human inference is essential for effective policy summarization and algorithmic teaching [18].

B. Further Approaches

The previous two sections covered trajectory extraction and textual summaries. In this section, further approaches that do not fall into these categories are explored and connections to the previous methods are drawn.

1) Counterfactual Explanations: Olson et al. [23] proposed another scheme for summarizing policies and presenting interesting states, focusing on giving counterfactual explanations (that is, answering "why not?"-questions). They do so by illustrating the minimal change in a state needed such that the agent chooses a different action, effectively answering why the agent chose an action. Hence, it is not required to store all visited trajectories or letting the agent run until interesting trajectories occur, but they are generated on demand. To generate states, an Adversarial Auto-Encoder (AAE) [24] is used: an encoder, $E(s) \to z$, maps a state s to a latent representation z. The discriminator $D(z) \to a$ then aims to extract the action from the latent, forcing the generator $G(z, a) \to s$ to incorporate the actions while reconstructing the states from the latents such that the discriminator is not able to recover the action. This setup allows generating states (which are images in the studied task) while maintaining a latent representation and incorporating the actions themselves into the generation process [25]. Additionally, to find counterfactual states, it is assumed that the agent itself holds an encoder $A(s) \to z_a$ to a latent space and a decoder $a \sim \pi(\cdot \mid z_a)$ to select an action[4]. However, the representations generated by a regular Auto-Encoder (AE) usually have gaps and generating images results in unrealistic counterfactuals [23], [26]. Hence, Olson et al. [23] propose to use a Wasserstein Auto-Encoder (WAN) [27] to find a well-behaving manifold z_w in which the agent's latent z_a can be embedded (with the en- and decoder $E_w(z_a) \to z_w$ and $D_w(z_w) \to z_a$). To find a minimal counterfactual state for a state s and a given action pair (a, a') (where a is the action that was taken in state s and one wants to know why a' was not taken), the distance in the state's WAN embedding is used. This idea is summarized in the following optimization problem:

$$\begin{aligned} &\min_{z_w} \left\| E_w(A(s)) - z_w \right\|_2^2 \\ &\text{s.t.} \quad a' = \arg\max_a \ \pi\big(D_w(z_w), a\big) \end{aligned}$$

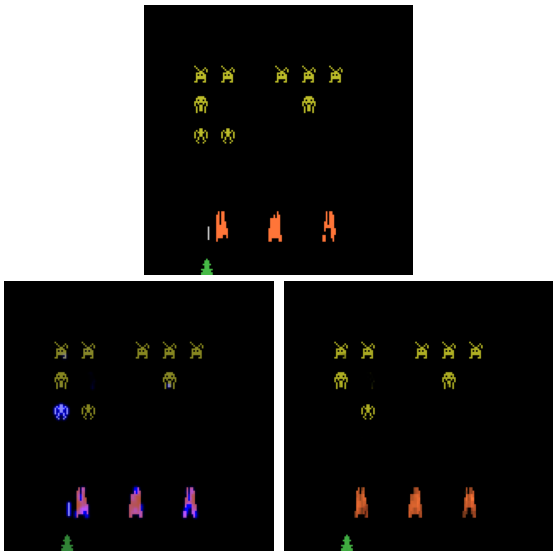
Using z_w^* and a', the counterfactual state s' can be generated using the generator, i.e., $s' = G(D_w(z_w^*), a')$.

Olson et al. [23] conducted two user studies to evaluate the fidelity of the generated states and whether a flawed agent can be detected using counterfactual explanations.

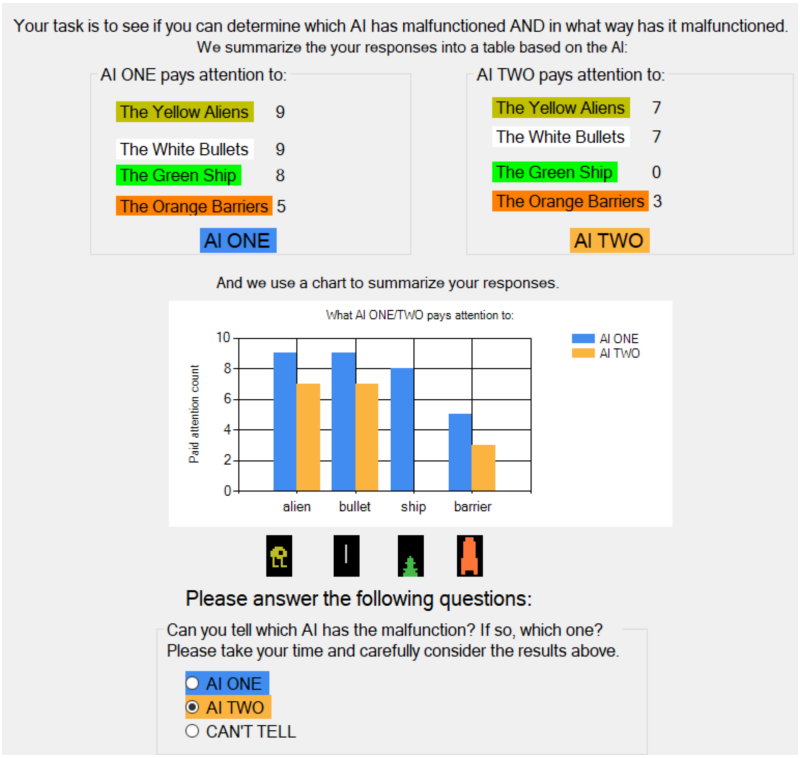
The former experiment was evaluated against an ablated version of the proposed algorithm by removing the encoder, discriminator, and WAN (effectively generating states from just the actions and searching for counterfactual states by maximizing $\pi(a' \mid z_a)$ w.r.t. z_a). To evaluate whether the generated images are persuasive, ten images were generated from the actual environment (Space Invaders [28]), the counterfactual explanation model, and the ablated version, asking the participants which images were generated and which are real. While the difference in selection accuracy between the ablated version and actual images was found to be statistically significant, the difference between the counterfactual explanations and real images was not. Hence, it was concluded that the model generates persuasive states, even though the generated images are not perfect.

In the second experiment (to evaluate whether the explanations actually help), the participants were shown two agents, each with an original image, a counterfactual explanation for some action, and an explanation highlighting the areas of the input image that had the most influence on the agent's decision. The participants

[4]When using a neural network policy, this assumption makes sense as every intermediate representation between layers can be interpreted as a latent representation.



(a) Interface of the first part of the second experiment. The top image is the original image and the bottom images are, from left to right, the highlighted and counterfactual state. Participants had to select what component of the state the agent pays the most attention to.



(b) Interface of the second part of the second experiment. Participants had to select which of the two AI agents is flawed given the identified components the agent pays attention to.

Figure 3: User interface of the user study conducted in [23]. Both taken from Olson et al., "Counterfactual State Explanations for Reinforcement Learning Agents via Generative Deep Learning." [23]

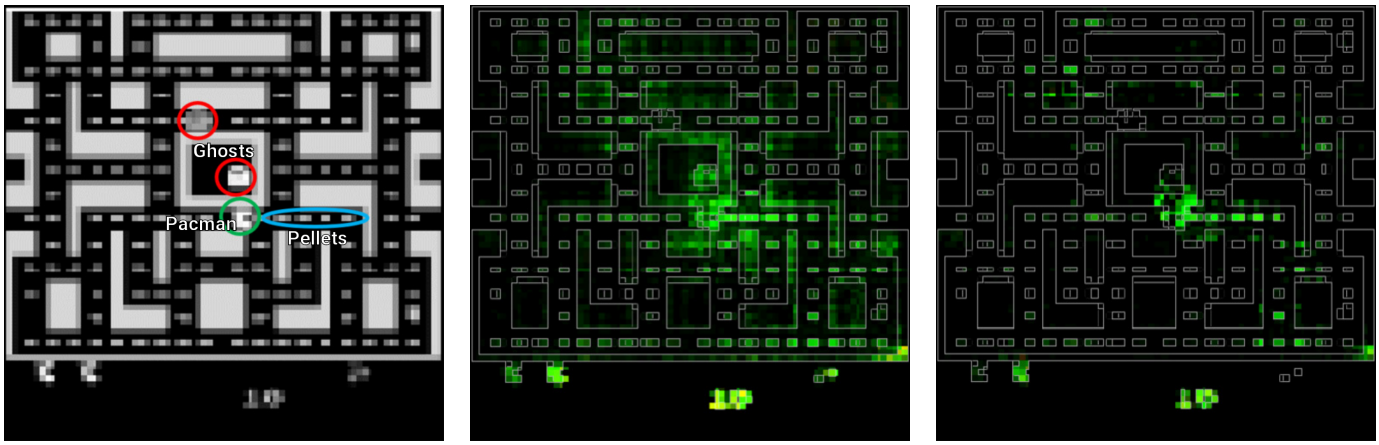
were then asked to select the object the agent pays most attention to and the explanation that helped the most (highlights vs. counterfactual explanation). After going through all trajectories, the results were summarized in a diagram and the participants were asked to select which IA has a malfunction. The user interface for the study shown in Figure 3. Olson et al. [23] found, with statistical significance, that the participants' accuracy increased when explanations were present. Also, without explanations, none of the participants were able to identify the exact flaw of the agent while with counterfactual explanations, a majority of the participants could identify the flaw (with statistical significance). The reported reason for selecting the object the agent pays attention to was, however, often the highlighting of specific regions rather than the counterfactual explanation. Nevertheless, most participants reported that they used both (the highlighting for finding the regions that possibly changed in the counterfactual state and the counterfactual explanation to identify the real reason).

Overall, counterfactual explanations seem like an effective way to teach a human how an agent behaves in various situations. Especially when it is combined with other methods such as highlighting important areas, it provides an advantage over just exploring various trajectories.

2) Combination of Local and Global Explanations: Huber et al. [29] took a step towards integrating local and global explanations into a single policy summary. Specifically, they combine Layer-Wise Relevance Propagation (LRP) and (a variant of) HIGHLIGHTS [6].

LRP is a class of methods for explaining the decisions of a neural network, usually with visual inputs. Its goal is to generate a *saliency map* highlighting the regions in the input image that were most relevant for the decision. Huber et al. [29] proposed a novel approach for LRP based on the z^+-rule for LRP, called *argmax*. An illustration of the components of LRP and saliency maps is given in Figure 4.

The second aspect for the combined local and global explanation is a modified version of HIGHLIGHTS named HIGHLIGHTS-DIV which was introduced along-



(a) Input (b) z^+-Rule (c) *argmax*-Rule [29]

Figure 4: Illustration of the saliency maps used in [29]. From left to right, the input (with Pacman, pellets, and ghosts depicted in green, blue, and red, respectively) and saliency maps generated by the z^+-rule and *argmax*-rule are shown. It can be seen that the right map focuses more on the vicinity of the player while the z^+-rule highlights huge areas of the field. Adopted from Huber et al., "Local and Global Explanations of Agent Behavior: Integrating Strategy Summaries With Saliency Maps." [29]

side [6]. The -DIV variant considers state diversity within the example trajectories. That is, only trajectories which are decently different are kept for presentation, reducing repetitive explanations and increasing one's attention [6].

Finally, the salience maps are integrated into HIGHLIGHTS-DIV by adding the salience values to the green channel of the generated videos, supporting the local interpretation of the agent's policy.

Huber et al. [29] conducted multiple user studies with four different models: HIGHLIGHTS-DIV and a likelihood-based method (showing most probable states), each with and without incorporating saliency maps. Among other questions which exceed the scope of this report, the participants were asked to select an agent to play on their behalf (given multiple summaries of two agents) and rate their confidence in the selection. Additionally, the participants were shown summaries of a single agent and were asked to describe the strategy and justify their description. It was found that in the agent selection task, HIGHLIGHTS-DIV greatly outperformed the likelihood-based approach, but the addition of saliency maps to the summaries did not greatly influence the result. Similar results were found for the description of the agent's policy: while HIGHLIGHTS-DIV outperforms the likelihood-based method, adding saliency maps does not drastically improve one's ability to grasp the policy. The difference is, however, noticeable compared with the difference in the agent selection task, although it is not statistically significant.

While the combination of local and global explanations seems promising at first, the results show that the quality of a summary primarily depends on the quality of the global summary. They also show that saliency maps might not be the most suitable tool for communicating an agent's rationale to laypersons as one has to learn how to use them [29].

IV. Interactive Learning

So far, methods for policy summarization have been explored. While these methods enable trust in autonomous agents, a user is not able to correct unwanted behavior that may be revealed by such a summary: the only option would be to re-design or re-train the agent which is tedious. To remedy this, various methods for incorporating human feedback into training can be applied, enabling a user to actively influence the policy learned by an autonomous agent. Such methods are therefore explored in this section.

The prevalent methods used for interactive learning are *reward-* and *policy-shaping* (both approaches are illustrated in Figure 5).

In reward-shaping, the environmental reward is augmented or replaced by a human teacher to reflect their goals [30]. While this has the advantage of removing the need to specify a reward function (which can be a tedious task), it has the disadvantage that the environmental reward might not be used at all. Hence, the

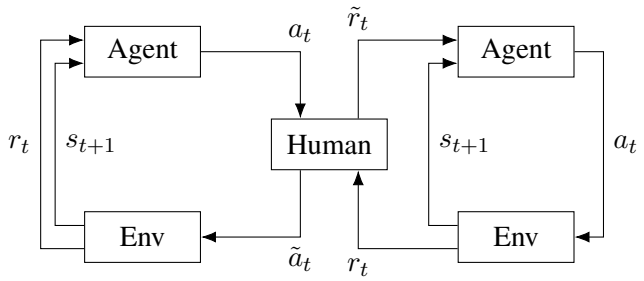


Figure 5: Illustration of the policy- and reward-shaping cycle on the left and right hand side, respectively. States are depicted s_t and s_{t+1}, corresponding to the current and next state. Actions are depicted a_t and $\tilde{a}_t$, where the latter was augmented by the human teacher. Rewards are depicted r_t and $\tilde{r}_t$, where the latter was augmented by the human teacher. Adopted from Cruz et al., "Training Agents With Interactive Reinforcement Learning and Contextual Affordances." [30]

agent only learns from human judgment which may be sub-optimal. However, research has been conducted to combine human reinforcement and environmental reward in reward-shaping settings which is further explored in the following sections.

In policy-shaping, the reward is not augmented but actions are judged directly before applying them to the environment [30]. That is, the human feedback is treated as direct information about the policy [31]. Although it has been shown that this method of feedback is more effective [31], the following sections focus on reward shaping as it is, to the best of the author's knowledge, the more common technique.

A. TAMER

The *TAMER*-framework (with "TAMER" standing for "Training an Agent Manually via Evaluative Reinforcement") is a general framework for interactively training agents introduced by Knox and Stone [32]. The basis of TAMER is formed by a model $\hat{H}$ that predicts the human feedback H based on the current state-action pair. This model should exhibit some important properties: first, it should generalize to unseen state-action pairs to yield best results. Second, it should be robust towards a moving target. That is, a human might not give the same feedback every time a state-action pair is encountered and the feedback may also depend on the current policy's performance. While this problem is primarily ignored by Knox and Stone [33], it was later addressed by the COACH-framework [34], [35] that is discussed in the next section. Training $\hat{H}$ is the principal goal of TAMER as the policy is chosen greedily w.r.t. the action, i.e., to maximize the instantaneous human feedback. The key insight leading to this idea is that humans already consider the long-term performance of an agent when giving the reinforcement signal. Hence, the problem of incorporating future rewards into the current action selection vanishes.

A major problem with directly using human reinforcement becomes apparent in high-frequency domains: a human is not able to provide feedback as fast as actions are executed [32]. Knox and Stone [32] approach this problem by assuming a model that can be trained using gradient descent and assigning each state-action pair that was generated since the last reinforcement signal arrived a *credit* c_t, i.e., a weighting factor. The credit then trades off the influence of the human feedback onto each state-action pair and the problem boils down to computing the credits. In practice, older older state-action pairs are pruned and the credit for all pairs that occurred less than $0.2\,\mathrm{s}$ ago is set to zero to account for reaction time.

Knox and Stone [32] compared the TAMER algorithm using a linear regression model with Radial Basis Function (RBF) features and gradient descent updates. The investigated environments were Gym's Tetris and Mountain Car environments [28]. They showed that in the Tetris environment, TAMER was able to learn a policy within three games while RRL-KBR [36] needed 120 games. Policy iteration [37], a genetic algorithm [38], and noisy cross-entropy [39] did not solve the environment at all. For Mountain Car, the results were not as extreme, but TAMER still outperformed the baselines (SARSA-3 and SARSA-20 [40]) in terms of sampling efficiency. While TAMER agents are more efficient and successful in the short term, completely autonomous learners performed more consistent after training. This suggests that the combination of both, using TAMER to train fast and using RL to fine-tune, might be beneficial.

However, a major caveat of TAMER is that it solely depends on human reinforcement and is not able to take environmental reward into account even when it is available. Knox and Stone addressed this by proposing *TAMER+RL* to combine manual with environmental reward [33], allowing autonomous fine-tuning after learning with TAMER. While fine-tuning with just using the environmental reward is straightforward, the underlying idea of [33] is to leverage the learned model $\hat{H}$ of the human reinforcement signals. They evaluated eight different variants for combining the learned human reinforcement signal with the actual environmental

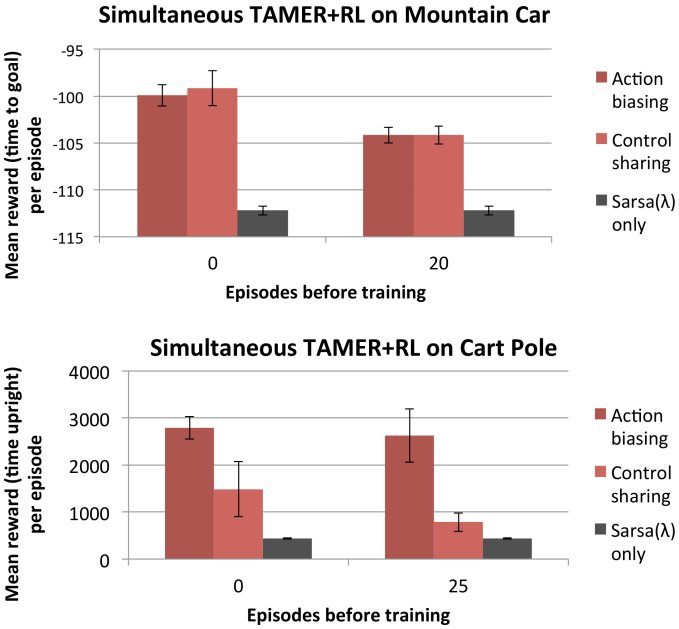


Figure 6: Results of Simultaneous TAMER+RL on the Mountain Car and Cart-Pole environments [28] (top and bottom, respectively). It can be seen that both TAMER variants (action biasing and control sharing) outperform plain SARSA in both settings (delayed and non-delayed training). Taken from Knox and Stone, "Reinforcement Learning From Simultaneous Human and MDP Reward." [41]

reward. By direct experimentation on the Mountain Car environment [28], they found that from all variants, the following two performed best (all with a hyper-parameter β):

- *Action Biasing:* Augment Q-function for action selection only
- *Control Sharing:* Choose an action greedily from $\hat{H}$ with probability $\min\{\beta, 1\}$ and otherwise use the base agent's mechanism

These methods outperformed plain SARSA(λ) both in terms of long-term performance and sample efficiency (with action biasing being the best). While these results are impressive, TAMER+RL still has the disadvantage that TAMER and the RL algorithm are executed sequentially, i.e., TAMER has to be run first and it is not possible to simultaneously learn from human reinforcement and environmental reward.

This gap is addressed again by Knox and Stone [41] by introducing *Simultaneous TAMER+RL*. The primary challenge is to trade off the influence of $\hat{H}$ and the environmental reward R, boiling down to selecting the parameter β for all the aforementioned integration methods. To determine β, Simultaneous TAMER+RL used a notion of *eligibility traces* [40]. However, instead of updating the traces in every step, they are only updated during training (i.e., when human feedback is given) and decayed when not training (i.e., when the RL algorithm learns from environmental reward).

Knox and Stone [41] conducted experiments using the Mountain Car and Cart-Pole environments [28] with SARSA(λ) as a baseline. For each method (action biasing and control shaping), two separate settings are evaluated: in the first, training begins in the zeroth episode (i.e., from the beginning) and in the second, training begins after 20 episodes for the Mountain Car and 25 episodes for the Cart-Pole. While the sample size was too low to report statistical significance, they showed that Simultaneous TAMER+RL outperforms plain SARSA in both settings (with zero and 25 episodes before training) and both environments. The results are summarized in Figure 6. While the agents that first learned solely from the Markov Decision Process (MDP) perform worse in the initial episodes, they found that when excluding these episodes from the calculation of the average reward, learning first outperforms the agents that were trained from the beginning. This suggests that learning prior to training via human reinforcement improves policies.

So far (Simultaneous) TAMER+RL was applied to linear Q-function approximations only. Arakawa et al. [42] proposed a further extension called *DQN-TAMER* that combined Deep Q-Learning (DQN) with the TAMER framework for supporting deep Q-functions. For brevity, this methods will not be studied in greater detail in this report other than providing a short outline of the results. Extending TAMER to deep RL methods allows application to more complicated environments. They applied their method to a Maze and Taxi environment [28], showing that DQN-TAMER outperforms plain DQN as well as Deep TAMER (which uses a deep model for $\hat{H}$ but does not incorporate an explicit Q-function) [43].

This outlook sums up the TAMER framework. The next section explores another, yet similar, framework that incorporates that as the policy changed, the feedback provided by humans changes as well.

B. COACH

As mentioned before, a major downside of TAMER is that it assumes that human feedback does not change when the policy is changing. This problem is addressed by COACH (where "COACH" stands for "Convergent Actor-Critic by Humans") proposed by MacGlashan et al. [34]. As the name suggests, COACH is based on

actor-critic methods and the insight that the advantage function is a good model for human feedback [34]. Hence, as the human provides the advantage values, no "critic" component is needed. Similar to TAMER, COACH leverages eligibility traces [40] to apply feedback to states visited before giving the feedback. To enable the user to choose how many states a reinforcement signal is applied backwards in time, COACH keeps multiple eligibility traces with different decay rates. The user then selects which trace to use for applying the update. To cover for a human's reaction time, the latest d steps are discarded from the policy update (where d is a hyper-parameter and usually set such that the latest $0.2\,\mathrm{s}$ to $0.8\,\mathrm{s}$ are not covered). As with regular actor-critic methods, a variety of concrete update methods are available.

McGlashan et al. [34] evaluated COACH comparing it to Q-Learning [44] and TAMER [32]. Instead of conducting a user study, the human reinforcement was generated in three different ways: using plain (sparse) environmental reward (favoring Q-Learning), only action-dependent feedback (favoring TAMER), and using the advantage function of the current policy (favoring COACH). These three feedback variants mimic the feedback model assumed by the respective models. As expected, COACH outperforms Q-Learning and TAMER in the latter setting while performing mediocre in the other settings.

An extension of COACH was proposed by Arumugam et al. [35] for deep policies. They evaluate the algorithm in a Minecraft environment where the agent has to walk to a specific target. The policy is represented by a Convolutional Neural Network (CNN) that uses screen images as inputs. To reduce training time, a Convolutional Auto-Encoder (CAE) [45], [46] is trained beforehand to extract parsimonious features. The experiment was executed with two trainers (from the set of the authors) rather than a wider group of participants to assess the theoretical advantage of Deep COACH rather than usability by laypersons. They found that Deep COACH converges to a reasonable performance within a few hundred steps, corresponding to approximately ten minutes of training time. The results were compared with plain COACH and Deep TAMER [43]. While Deep COACH and Deep TAMER performed rather similar, plain (non-deep) COACH did not manage to solve the task. However, no statement on statistical significance was made.

V. Discussion and Conclusion

In this report, various methods for summarizing policies and interactive learning have been explored.

Policy summarization tackles the problem of conveying information about an agent's policy to a human the agent might collaborate with. Establishing trust in an autonomous system is important for effective collaboration [1]. While summaries are helpful to explain a policy, it does not enable the user to actively influence the policy that is being executed. Hence, they are not able to correct unwanted behavior.

This problem is tackled by methods for interactive learning. Instead of just using environmental reward, they leverage human feedback and incorporate it into training. As opposed to policy summarization, the research conducted in this domain did not feature lots of user studies, hence the assessment of the suitability for laypersons is questionable.

A problem that was identified during the literature review is that the two fields (policy summarization and interactive learning) seem to evolve quite separately. However, as shown in the policy summaries, it can be hard to grasp the policy an agent is actually executing and to give meaningful feedback. Approaches for combining the research are, for example, showing the user a counterfactual explanation of why the agent did not choose the action the user would suggest (maybe it has its reason). The human can then still decide whether to overwrite the action or punish the agent, but the consequences will be clearer. One can imagine similar integration methods for policy extraction approaches to punish or reward past actions (when the training takes very long and it is infeasible to supervise the learning process continuously).

References

[1] A. Glass, D. L. McGuinness, and M. Wolverton, "Toward Establishing Trust in Adaptive Agents," in *Proceedings of the 13th International Conference on Intelligent User Interfaces*, ser. IUI '08. New York, NY, USA: Association for Computing Machinery, Jan. 2008, pp. 227–236.

[2] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa, "Effects of Robot Motion on Human-Robot Collaboration," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2015, pp. 51–58.

[3] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum Entropy Inverse Reinforcement Learning," in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[4] A. Y. Ng and S. J. Russell, "Algorithms for Inverse Reinforcement Learning," in *Icml*, vol. 1, 2000, p. 2.

[5] P. Abbeel and A. Y. Ng, "Apprenticeship Learning via Inverse Reinforcement Learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, Jul. 2004, p. 1.

[6] D. Amir and O. Amir, "HIGHLIGHTS: Summarizing Agent Behavior to People," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, vol. 17, 2018, pp. 1168–1176.

[7] P. Rohlfshagen and S. M. Lucas, "Ms. Pac-Man Versus Ghost Team CEC 2011 Competition," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, Jun. 2011, pp. 70–77.

[8] C. A. Cruz and T. Igarashi, "A Survey on Interactive Reinforcement Learning: Design Principles and Open Challenges," in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 1195–1209.

[9] A. Najar and M. Chetouani, "Reinforcement Learning With Human Advice: A Survey," *Frontiers in Robotics and AI*, vol. 8, p. 584075, Jun. 2021.

[10] E. Puiutta and E. M. S. P. Veith, "Explainable Reinforcement Learning: A Survey," in *Machine Learning and Knowledge Extraction*, ser. Lecture Notes in Computer Science, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer International Publishing, 2020, pp. 77–95.

[11] L. Wells and T. Bednarz, "Explainable AI and Reinforcement Learning – A Systematic Review of Current Approaches and Trends," *Frontiers in Artificial Intelligence*, vol. 4, p. 48, 2021.

[12] S. A. Raza, "Computational Reinforcement Learning Using Rewards From Human Feedback," Thesis, University of Technology Sydney, Sydney, 2018.

[13] O. Khan, P. Poupart, and J. Black, "Minimal Sufficient Explanations for Factored Markov Decision Processes," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 19, pp. 194–200, Oct. 2009.

[14] B. Hayes and J. A. Shah, "Improving Robot Controller Transparency Through Autonomous Policy Explanation," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, Mar. 2017, pp. 303–312.

[15] Y. Fukuchi, M. Osawa, H. Yamakawa, and M. Imai, "Autonomous Self-Explanation of Behavior for Interactive Reinforcement Learning Agents," in *Proceedings of the 5th International Conference on Human Agent Interaction*, ser. HAI '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 97–101.

[16] O. Amir, F. Doshi-Velez, and D. Sarne, "Agent Strategy Summarization," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 1203–1207.

[17] S. H. Huang, D. Held, P. Abbeel, and A. D. Dragan, "Enabling Robots to Communicate Their Objectives," *Autonomous Robots*, vol. 43, no. 2, pp. 309–326, Feb. 2019.

[18] I. Lage, D. Lifschitz, F. Doshi-Velez, and O. Amir, "Toward Robust Policy Summarization," *Autonomous agents and multiagent systems*, vol. 2019, pp. 2081–2083, May 2019.

[19] L. Torrey and M. Taylor, "Teaching on a Budget: Agents Advising Agents in Reinforcement Learning," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, 2013, pp. 1053–1060.

[20] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining Active Learning and Semi-supervised Learning Using Gaussian Fields and Harmonic Functions," in *ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, vol. 3, 2003.

[21] D. S. Brown and S. Niekum, "Machine Teaching for Inverse Reinforcement Learning: Algorithms and Applications," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 7749–7758, Jul. 2019.

[22] B. M. Adams, H. T. Banks, M. Davidian, H.-D. Kwon, H. T. Tran, S. N. Wynne, and E. S. Rosenberg, "HIV Dynamics: Modeling, Data Analysis, and Optimal Treatment Protocols," *Journal of Computational and Applied Mathematics*, vol. 184, no. 1, pp. 10–49, Dec. 2005.

[23] M. L. Olson, R. Khanna, L. Neal, F. Li, and W.-K. Wong, "Counterfactual State Explanations for Reinforcement Learning Agents via Generative Deep Learning," *Artificial Intelligence*, vol. 295, p. 103455, Jun. 2021.

[24] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," *arXiv:1511.05644 [cs]*, May 2016.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[27] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein Auto-Encoders," in *International Conference on Learning Representations*, Feb. 2018.

[28] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv:1606.01540 [cs]*, Jun. 2016.

[29] T. Huber, K. Weitz, E. André, and O. Amir, "Local and Global Explanations of Agent Behavior: Integrating Strategy Summaries With Saliency Maps," *Artificial Intelligence*, vol. 301, p. 103571, Dec. 2021.

[30] F. Cruz, S. Magg, C. Weber, and S. Wermter, "Training Agents With Interactive Reinforcement Learning and Contextual Affordances," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 271–284, Dec. 2016.

[31] 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. Curran Associates, Inc., 2013.

[32] W. B. Knox and P. Stone, "Interactively Shaping Agents via Human Reinforcement: The TAMER Framework," in *Proceedings of the Fifth International Conference on Knowledge Capture*, ser. K-CAP '09. New York, NY, USA: Association for Computing Machinery, Sep. 2009, pp. 9–16.

[33] ——, "Combining Manual Feedback With Subsequent MDP Reward Signals for Reinforcement Learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1-Volume 1*. Citeseer, 2010, pp. 5–12.

[34] 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 *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 2285–2294.

[35] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman, "Deep Reinforcement Learning From Policy-Dependent Human Feedback," *arXiv:1902.04257 [cs, stat]*, Feb. 2019.

[36] J. Ramon and K. Driessens, "On the Numeric Stability of Gaussian Processes Regression for Relational Reinforcement

Learning," in *ICML-2004 Workshop on Relational Reinforcement Learning*, Jan. 2004, pp. 10–14.
[37] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
[38] N. Böhm, G. Kókai, and S. Mandl, "Evolving a Heuristic Function for the Game of Tetris," in *LWA*, 2004, pp. 118–122.
[39] I. Szita and A. Lörincz, "Learning Tetris Using the Noisy Cross-Entropy Method," *Neural Computation*, vol. 18, no. 12, pp. 2936–2941, Dec. 2006.
[40] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2020.
[41] W. B. Knox and P. Stone, "Reinforcement Learning From Simultaneous Human and MDP Reward," in *AAMAS*, 2012, pp. 475–482.
[42] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda, "DQN-TAMER: Human-In-The-Loop Reinforcement Learning With Intractable Feedback," *arXiv:1810.11748 [cs]*, Oct. 2018.
[43] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone, "Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018.
[44] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.
[45] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data With Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[46] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, ser. Lecture Notes in Computer Science, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer, 2011, pp. 52–59.