Explanation Augmented Feedback in Human-in-the-Loop Reinforcement Learning

Lin Guan [*1] **Mudit Verma** [*1] **Subbarao Kambhampati** [1]

Abstract

Human-in-the-loop Reinforcement Learning (HRL) aims to integrate human guidance with Reinforcement Learning (RL) algorithms to improve sample efficiency and performance. The usual human guidance in HRL is binary evaluative "good" or "bad" signal for queried states and actions. However, this suffers from the problems of weak supervision and poor efficiency in leveraging human feedback. To address this, we present EXPAND (Explanation Augmented Feedback) which allows for explanatory information to be given as saliency maps from the human in addition to the binary feedback. EXPAND employs a state perturbation approach based on the state salient information to augment the feedback, reducing the number of human feedback signals required. We choose two domains to evaluate this approach, *Taxi* and *Atari-Pong*. We demonstrate the effectiveness of our method on three metrics, environment sample efficiency, human feedback sample efficiency, and agent gaze. We show that our method outperforms our baselines. Finally, we present an ablation study to confirm our hypothesis that augmenting binary feedback with state salient information gives a boost in performance.

1. Introduction

Deep Reinforcement Learning has seen a lot of success in learning complex behaviors through high dimensional data. However, in many situations, the current state-of-the-art is yet to outperform human experts. Moreover, it is known to be highly sample-inefficient. For real-world domains, sample efficiency is even more crucial as it is impractical to collect millions of training samples (Knox & Stone, 2009). One of the ways to curb this problem is to leverage human knowledge to help the RL agent to learn complex behaviors faster and better than before. It is often difficult for humans to encode their knowledge in terms of reward functions for desired behaviors (Littman et al., 2017). However, it has been found that if the human can provide guidance signals, i.e. follow the paradigm of Human-in-the-Loop Reinforcement Learning (HRL), the agent can achieve better performance as well as sample efficiency.

One of the most popular forms of human guidance in HRL has been binary evaluative signal on actions. This means that humans in the loop are expected to provide a "good" or "bad" judgement for an action taken in some state of the environment. Although this allows non-expert humans to provide their guidance, it still suffers from poor feedback sample efficiency. Further, binary action evaluations do not hint at what the agent should do instead. If the human can explain, which makes the agent understand the "why" behind their evaluative decision, then it is possible to alleviate the agent's confusion about which parts of its perception are important to decide whether to take an action. One of the ways to do this can be to point out to, say, a driving agent that 'STOP' sign is an essential signal for the right action "apply-break". An ideal way of conveying this information can be through natural language, but this imposes a stronger assumption of having a system that understands the 'STOP' sign concept, which can be a hard task itself. A more straightforward method to deal with this issue can be to allow humans to point to the salient parts of the (visual) environment state. Saliency maps have already been shown to be a means of assessing agent's internal representation by humans in the Explainable Reinforcement Learning research (Wang et al., 2015; Zahavy et al., 2016; Sorokin et al., 2015; Greydanus et al., 2017; Mott et al., 2019; Puri et al., 2019). We extend this idea to use a saliency map as a communication channel between RL agents and humans. This way, human trainers can inform the RL agent about which regions it should focus on to accomplish the given task.

We note that the requirement for human trainers to provide an explanation on how to act desirably does not require them to be more adept than in the case of providing only binary

[*]Equal contribution [1]School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, Arizona, USA. Correspondence to: Lin Guan <lguan9@asu.edu>.

2020 ICML Workshop on Human in the Loop Learning (HILL 2020).

feedback. Like prior approaches utilizing binary evaluations, we assume the human trainer has a high-level understanding of achieving the task. In our driving agent example, the human trainer may not be able to tell the optimal angle of the steering wheel, however, we can expect them to consider it as an essential factor.

In this paper, we present EXPAND - EXPlanation AugmeNted feeDback. EXPAND expects explanations from humans regarding what the agent should focus on in the image observation to achieve the given task and augments this feedback with the conventional binary evaluations on actions. Here we are primarily concerned with improving the environment sample efficiency and human feedback sample efficiency by using saliency guidance. To learn with human saliency information, we propose a novel approach that applies multiple perturbations to irrelevant regions, thereby supplementing each saliency feedback with a set of constructed perturbed states. The idea is to differentiate between relevant and irrelevant regions of image observation to update the agent's Q-values. Ideally, if there are indeed relevant and irrelevant regions, only the relevant parts of the image should be responsible for decision making regardless of perturbations to the other areas. Making multiple perturbations on the irrelevant regions can essentially provide our algorithm with more feedback samples to work with and hence, reduce the feedback sample complexity. To our knowledge, this is the first work that incorporates human explanatory information as saliency maps in the setting of learning from human evaluative feedback by applying multiple perturbations on "irrelevant" regions.

Figure 1 shows the train-interaction loop where the RL agent interacts with the environment to collect transition experiences and query users for binary evaluations as well as explanations in the form of saliency feedback. Sections 3 and 4 explains the our method in detail. Section 5 presents our experiments on the *Taxi* domain and *Atari-Pong*. The experiments show that our approach significantly improves both environment sample efficiency and human feedback sample efficiency. We go on to verify that EXPAND is indeed "looking" at the important regions as pointed out to by the human trainer through the agent-gaze metric. Finally, we also experiment with a different number of perturbations to check whether additional perturbations are helpful or not.

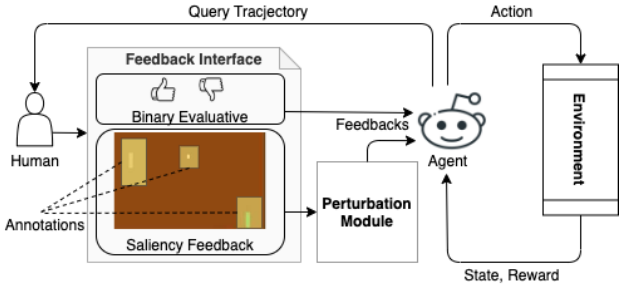

Figure 1. Overall Flow of EXPAND. The agent queries the human in the loop with a sampled trajectory. Then the human responds with a binary evaluation on the action and a saliency annotation on the state. The perturbation Module supplements the saliency explanation by perturbing irrelevant regions. The feedback is consumed by the agent for updating the policy parameters. This loop continues as the RL agent is trained, with feedback being queried every N_f episodes.

2. Related Work

Leveraging human guidance to speed up reinforcement learning has been extensively investigated in different literatures, which include imitation learning (Ross et al., 2011; Ho & Ermon, 2016), learning from demonstration (Schaal, 1997; Hester et al., 2018), inverse reinforcement learning (Ng et al., 2000; Abbeel & Ng, 2004), reward shaping (Ng et al., 1999) and learning from human preference (Christiano et al., 2017; Ibarz et al., 2018). Surveys on these topics are provided by (Zhang et al., 2019a; Wirth et al., 2017)

Compared to the approaches mentioned above, learning from human evaluative feedback has the advantage of placing minimum demand of both the human's expertise and the trainer's ability to provide guidance (e.g. the requirements of complex and expensive equipment setup). Representative works include the TAMER framework (Knox & Stone, 2009; Warnell et al., 2018), and the COACH framework (MacGlashan et al., 2017; Arumugam et al., 2019). The TAMER+RL framework extends TAMER by learning from both human evaluative feedback and environment reward signal (Knox & Stone, 2010; 2012). DQN-TAMER further augments TAMER+RL by utilizing the deep neural network to learn in high dimensional state space (Arakawa et al., 2018).

A variety of approaches have been proposed to increase the information gathered from human feedback, which takes the complexities in human feedback-providing behavior into account. (Loftin et al., 2014; 2016) speed up learning by adapting to different feedback-providing strategies; the Advice framework (Griffith et al., 2013; Cederborg et al., 2015) treats human feedback as direct policy labels and uses a probabilistic model to learn from inconsistent feedback. Some other works also consider the action execution speed (Peng et al., 2016), the confidence in predicting human feedback (Xiao et al., 2020), and different levels of satisfaction indicated by human voice (Tenorio-Gonzalez et al., 2010). Although these approaches better utilize human feedback with improved modeling of human behaviors, they do not address the lack of informativeness of human feedback which is a fundamental problem of evaluative feedback.

Human explanatory information is exploited in some prior works. The main challenge of using human explanation is to translate human's high-level linguistic representation to low-level agent-understandable language. As an early attempt, (Thomaz et al., 2006) allows humans to give anticipatory guidance rewards and tell the object tied to the reward. However, they assume the availability of an object-oriented representation of the state. (Krening et al., 2016) resorts to human explanatory advice in natural language. Still, they assume a natural language processing model and a recognition model that can understand concepts in human explanation and recognize objects in image observation. In this work, we bridge the vocabulary gap between humans and the agent by taking human explanations in the form of saliency maps.

Works like (Zhang et al., 2018; 2019b; Kim et al., 2020) use human gaze data as human saliency information, collected with the help of sophisticated eye-tracking equipment, to help agents in decision making in an imitation learning setting. Moreover, we refrain from a comparison with these works since they involve humans in an offline manner. In contrast, in this work, human trainers are required to be more actively involved during the agent training similar to (Arakawa et al., 2018; Xiao et al., 2020). (Yeh et al., 2018) has tried extracting human saliency mask from template-based human natural-language advice, thereby assuming a mapping from templates to object-level masks available to the agent.

Algorithm 1 Train - Interaction Loop

Result: trained DQN agent M
Input: DQN agent M with randomly-initialized weights, replay buffer $\mathcal{D}$, human feedback buffer $\mathcal{D}_h$, feedback frequency N_f, total episodes N_e, update interval b
Begin
for $i = 1$ **to** N_e **do**
  for $t = 1$ **to** T **do**
    Observe state s
    Sample action from current DQN policy π with ϵ-greedy, observe reward r and next state s$'$ and store (s, a, r, s$'$) in $\mathcal{D}$
    if t mod $b == 0$ **then**
      Sample a mini-batch of transitions from $\mathcal{D}$ with prioritization
      Compute the DQN loss L_{DQN} and update θ over L_{DQN}
      Sample a mini-batch of human feedback from $\mathcal{D}_h$
      Compute the feedback loss L_F and update θ over L_F
    end if
  end for
  if i mod $N_f == 0$ **then**
    Obtain the last trajectory τ from $\mathcal{D}$
    Query τ to obtain feedback $\mathcal{H}_i$
    Append $\mathcal{H}_i$ to buffer $\mathcal{D}_h$
  end if
end for
End

3. Problem Setup

The hypothesis we intend to verify is whether the use of state saliency information such as human saliency maps, along with binary evaluative feedback, can boost the performance of a reinforcement learning agent. For this, we will first formalize our setup in this section.

Similar to prior works, we have an agent M which interacts in an environment $\mathcal{E}$ (like an *Atari* Emulator) in a sequence of actions, image observations and rewards. Following standard practice in reinforcement learning, k ($k = 4$) preprocessed consecutive image observations are stacked together to form our state as $s_t = [x_{t-(k-1)}, ..., x_{t-1}, x_t]$. At each time-step t the agent can select a legal action from the set of all possible actions $\mathcal{A} = \{1, ..., K\}$ in the state $s_t \in \mathcal{S}$. Then the agent receives a reward $r_t \in \mathcal{R}$ from the environment $\mathcal{E}$. This sequence of interaction ends when the agent is able to achieve its goal or when the time-step budget allocated to the agent is exhausted. This formalism follows finite Markov Decision Process tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}, \mathcal{A}, \mathcal{R}$ are as defined before, $\mathcal{P}$ is the transition probability function which tells that an action a_t in state s_t will lead to state s_{t+1} and γ is the discount factor for calculating the return $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_t$ from current time t to the time at which the game terminates T. The goal of the agent is to learn a policy function $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximises expected return.

Additionally, we assume a human trainer providing binary evaluative feedback $\mathcal{H} = (h_1, h_2, ...h_n)$ that conveys their assessment of the queried state-action pairs in trajectory sampled by the agent. In this work, we define the feedback as $h_t = (x_t^h, b_t^h, x_t, a_t, s_t)$, where $x_t^h = \{Box_1, ..., Box_1\}$ is saliency map given by the human trainer for the image observation x_t in state s_t and $b_t^h \in \{-1, 1\}$ is a binary "bad" or "good" feedback given for the action a_t. Box_i is a tuple (x, y, w, h) for top left eucledian coordinates x and y, the width (w), and the height (h) of the rectangular region annotated on the observation image x_t.

4. Our Method

In our setting, the agent learns from both the human feedback and the environment reward simultaneously. To learn from environment reward, we use an off-policy Reinforcement Learning algorithm, the Deep Q-Networks (DQN)

(Mnih et al., 2015). To learn from human feedback, we first interpret binary feedback as the label on the optimality of the performed actions, which is similar to the interpretation used by (Griffith et al., 2013; Cederborg et al., 2015). Considering that we are using DQN as our policy network, modeling binary feedback as labels on action optimality allows us to link human feedback with the advantage value (Baird, 1995) of an action directly. The way we learn from human evaluative feedback will be discussed in detail in Section 4.3. To learn with explanation augmented feedback, we amplify saliency feedback by perturbing irrelevant regions in the state and expect the Q-value approximator to be indifferent to these perturbations when computing the action values. The ideas manifest in the form of various loss terms to update the DQN network. Algorithm 1 presents the train-interaction loop of EXPAND. Within an episode loop, the agent interacts with the environment and stores transition experiences. Every few episodes, it collects human feedback queried on a trajectory sampled from the most recent policy. The DQN weights are updated twice, with the usual DQN loss and then with the proposed feedback loss, in a single train step. This section covers the proposed loss terms and how we obtain specialized inputs (perturbed states) for these losses.

4.1. Learning with Environment Reward

We use the Deep Q-Learning algorithm to approximate the optimal policy in this work. Following (Mnih et al., 2015), a set of tricks is applied to stabilize training, such as experience replay, reward clipping, and soft-target network updates. The learned policy π is defined as the actions that maximize the Q-values. Since π is a deterministic policy here, according to the Bellman equation, the state value function can be defined as $V^\pi(s) = Q^\pi(s, \pi(s))$.

To ensure sufficient exploration, we also use ϵ-greedy action selection mechanism, in which the probability ϵ of taking a random action is annealed down episodically.

Finally, the loss function in DQN that updates the Q-value function weights θ is as follows:

$$L_{DQN} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(r + \gamma \max_{a'} Q(s', a'; \bar{\theta}) - Q(s, a; \theta))^2] \quad (1)$$

where $\bar{\theta}$ is the target network parameters and $\mathcal{D}$ is the replay buffer which stores experience transition tuples (s, a, r, s') such that action a taken in the state s brings immediate reward r and gets the agent to state s'.

4.2. Learning with Binary Evaluative Feedback

Like the idea behind DQN's replay buffer, we maintain a feedback replay buffer $\mathcal{D}_h$, which stores the observed human feedback. Feedback signals in $\mathcal{D}_h$ are sampled uniformly later on and used to compute the feedback losses during training.

We use the advantage value to formulate our *advantage loss* that learns from human binary feedback. Advantage value is essentially the difference between the Q-value of the action upon which the feedback was given and the Q-value of the current optimal action by the policy network. Given state s and action a for which the feedback was given, when the current agent's policy is π, the advantage value is defined as,

$$\begin{aligned} A^\pi(s, a) &= Q^\pi(s, a) - V^\pi(s) \\ &= Q^\pi(s, a) - Q^\pi(s, \pi(s)) \end{aligned}$$

Hence, advantage value quantifies the possible advantage the agent would have if some other action were chosen instead of the current-best. It can be viewed as the agent's judgment on the optimality of an action. Accordingly, positive feedback means the human trainer expects the advantage value of the annotated action to be zero, or negative, vice versa. Therefore, we define a loss function, i.e., the *advantage loss*, which forces the network to have the same judgment on the optimality of action as the human trainer. Intuitively, we penalize the policy-approximator when a marked "good" action is not chosen as the best action, or, when a marked "bad" action is chosen as the best action.

For a feedback $h = (x^h, b^h, x, a, s)$, when the label is "good", i.e., $b^h = 1$, we expect the network to output a target value $\hat{A}(s, a) = 0$, so the loss can be defined as $|\hat{A}(s, a) - A^\pi(s, a)| = Q^\pi(s, \pi(s)) - Q^\pi(s, a)$. When the label is "bad", i.e., $b^h = -1$, we expect the network to output an advantage value $A^\pi(s, a) < 0$. Since in this case we do not have a specific target value for $A^\pi(s, a)$, we resort to the idea of large margin classification loss (Piot et al., 2014; Hester et al., 2018), which forces $Q^\pi(s, a)$ to be at least a margin l_m lower than the Q-value of the second best action, i.e., $\max_{a' \neq a} Q^\pi(s, a')$. One advantage of such an interpretation of human feedback is that it allows us to make use of the feedback to directly affect the Q-values, which avoids the need to use a separate set of parameters to model human feedback.

Formally, for any human feedback $h = (x^h, b^h, x, a, s)$ and the corresponding advantage value $A_{s,a} = A^\pi(s, a)$, the *advantage loss* is described as follows:

$$L_A(s, a, h) = L_A^{Good}(s, a, h) + L_A^{Bad}(s, a, h) \quad (2)$$

where

$$L_A^{Good}(s, a, h; b^h = 1) = \begin{cases} 0 & ; A_{s,a} = 0 \\ Q^\pi(s, \pi(s)) - Q^\pi(s, a) & ; \text{otherwise} \end{cases}$$

and,

$$L_A^{Bad}(s,a,h;b^h=-1) = \begin{cases} 0 & ;A_{s,a}<0 \\ Q^\pi(s,a)-(\max_{a'\neq a} Q^\pi(s,a')-l_m) & ;A_{s,a}=0 \end{cases}$$

4.3. Learning with Human Saliency Information

State saliency must inform, in some way, about which parts of the state affect or help in achieving the goal. In our case, we depend upon saliency maps to obtain such information. Of course, these "parts" of the state can be specific regions and even more abstract annotated objects like wall and enemy. The intuition is that once the agent knows where the relevant regions are, it can much faster figure out the optimal actions to take. In this work, the saliency maps are simply a number of bounding boxes over image observations, marking those regions as being important for the agent to focus at in order to achieve the give task. We utilize this saliency information in a manner that directly affects the policy-approximation network.

It should be noted that the previous section only made use of the binary evaluations via the *advantage loss*, however, in this section we utilize both the saliency feedback along with the binary feedback. This section is organized as follows. Section 4.3.1 covers how multiple perturbations have been applied to the image observation. The remainder of this section introduces the loss terms EXPAND uses to train the policy approximation function. In addition to the *advantage loss* that learns from human binary feedback, we propose other two loss terms namely, the *policy invariant loss* and the *value invariant loss* that make use of the human explanation along with the binary evaluation.

4.3.1. Perturbing Observations For Separating Relevant and Irrelevant Regions

As we will see in later sections when we discuss about our loss terms, we use the idea that Q-values of states with perturbed irrelevant regions should ideally be the same as the Q-values of original states. An ideal perturbation would be at an object-level, where we can manipulate, say, the position of other passengers in *Taxi* or the background color in *Atari-Pong*. These manipulations to irrelevant regions should not affect the agent's policy. However, it can be very difficult to do even in simulated domains. A workaround can be with the use of Gaussian perturbations over these irrelevant regions, essentially blurring them out and motivating the agent to focus more on the clear relevant regions.

Consider a feedback $h=(x^h,b^h,x,a,s)$. We need to convert state s into states with perturbations on relevant regions s_t^r and states with perturbations on irrelevant regions $\tilde{s}_t^r$. We follow the Gaussian perturbation mentioned in (Greydanus et al., 2017), which is as follows, let $\mathbb{M}(x,i,j)$ denote a mask over relevant regions given in the feedback for image x. If $x(i,j)$ denotes the pixel at index (i,j) for image x, then,

$$\mathbb{M}(x,i,j) = \begin{cases} 1 & \text{if (i, j) lies in Box, } \exists \text{ Box} \in b^h \\ 0 & \text{otherwise} \end{cases}$$

and, we can perturb pixel (i,j) in image observation x according to mask $\mathbb{M}$ using a function ϕ defined as follows,

$$\phi(x,M,i,j) = x\odot(1-\mathbb{M}(x,i,j))+G(x,\sigma_G)\odot\mathbb{M}(x,i,j)$$

where $\odot$ is the Hadamard Product and function $G(x,\sigma_G)$ is the Gaussian blur of the observation x. Hence, we can get image with perturbed relevant regions (x^r) with mask $\mathbb{M}$ and perturbed irrelevant regions ($\tilde{x}^r$) with mask $\neg\mathbb{M}$.

4.3.2. Policy Invariant Loss

The intuition behind the *policy invariance loss* is, under a set of perturbations over irrelevant regions in human explanation, the action marked "good" is always good and the action marked "bad" is always bad. This means the agent's judgement on the optimality of an action should not change under perturbations over irrelevant regions in human saliency map. Thus, this loss is essentially the *advantage loss* calculated over states with perturbed irrelevant regions. As defined in Section 4.3.1, we use Gaussian perturbations of various filter sizes and variance [1] to obtain a number of states with perturbed irrelevant regions. The g^{th} perturbation is denoted by $\tilde{s}^{rg}$ with the feedback $h=(x^h,b^h,x,a,s)$. Formally, if g number of perturbations were produced from a single state s in feedback h, the loss is given as,

$$L_P = \frac{1}{g}(\sum_g L_A(\tilde{s}^{rg},a,h)) \tag{3}$$

4.3.3. Value Invariant Loss

In previous loss function definitions, we utilized the difference between the agent's and the human's judgment on the optimality of action. Apart from that, we also note that, to the human, these states with perturbations on irrelevant regions are not "seen" differently than the original state. Thus, the agent is expected to learn an internal representation that only captures the relevant regions of the image observation state and should be indifferent towards irrelevant regions. Therefore, the Q-values of the original state should be similar to the Q-values of states with perturbed irrelevant regions. Hence we define a mean squared error loss term over the two Q-values as our *value invariant loss* L_V as

$$L_V = \frac{1}{g}\sum_{i=1}^{g}\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}(Q^\pi(s,a)-Q^\pi(\tilde{s}^{ri},a)) \tag{4}$$

[1] The settings of filter size and variance are listed in the appendix at https://bit.ly/icml20-hill-expand

where $\mathcal{A}$ is a set of actions annotated by the human trainer and g is the number of perturbations.

4.4. Combining Feedback Losses

We combine the three losses in a straightforward manner, i.e., weighted addition of the terms. The feedback loss is given as,

$$L_F = \lambda_A L_A + \lambda_P L_P + \lambda_V L_V \tag{5}$$

where λ_A, λ_P and λ_V are the weights of *advantage loss*, *policy invariant loss* and *value invariant loss* respectively.

The DQN is then trained with L_{DQN} as well as L_F. As an extreme case, the *value invariant loss* can go down to zero when the policy network assigns an identical Q-value to actions in all the states, which will hurt the learning performance. To prevent this we give a smaller weight to *value invariant loss* (λ_V=0.1, λ_P=1.0, λ_A=1.0). It should also be noted that these three losses help with human feedback sample efficiency and environment sample efficiency in different ways, and the ablation shows that only their combination achieves the best performance. *Value invariant loss* imposes a stricter latent space similarity, whereas *policy invariant loss* helps differentiate between different actions. Both these losses use the simple idea that states with perturbed irrelevant regions shall be viewed similarly to the original states, giving us a window of opportunity to augment the given original state with various perturbed states.

5. Experimental Evaluation

To show the effectiveness of EXPAND, we conducted experiments on two domains, namely, *Taxi* domain and *Atari-Pong*. Section 5.2 introduces the domains and tasks, followed by Section 5.3, which presents the different metrics we use to compare EXPAND with our baselines. The baselines we use include one HRL algorithm DQN-TAMER (Arakawa et al., 2018) that simultaneously learns from environment reward and human binary feedback. We also compare to three variants of EXPAND with one loss term at a time: EXPAND-Advantage (λ_A=1.0, λ_P=0, λ_V=0), EXPAND-Value-Invariant (λ_A=1.0, λ_P=0, λ_V=0.1) and EXPAND-Policy-Invariant (λ_A=1.0, λ_P=1.0, λ_V=0). The goal of experimenting with different variants of EXPAND is to determine each loss term's effect on the learning performance.

5.1. Experimental Settings

In all our experiments, our algorithm and the baselines employ the same DQN network architecture identical to the one in (Mnih et al., 2015), which has three convolutional layers following by one fully-connected layer. The same set of hyperparameters is used to train the models from scratch. The details of architecture and hyperparameter can be found in the appendix. We also use the prioritized experience replay mechanism (Schaul et al., 2016) in both our algorithm and the baselines. Following the standard preprocessing procedure, each frame is converted from RGB format to grayscale and is resized to shape 84×84. The input to the networks is normalized to the range of [0, 1].

During training, we start with an ϵ-greedy policy ($\epsilon = 1.0$) and reduce ϵ by a factor of λ_ϵ at the end of each episode until it reaches 0.01. The reported results are on five runs of each algorithm.

5.2. Domains

The *Taxi* domain is a self-devised domain; however, similar domains have been extensively used to evaluate RL algorithms (Dietterich, 2000). The *Taxi* environment is a grid world setup where the agent is the taxi (which occupies one grid cell at a time), and there are passengers in the world (denoted by different colored dots occupying separate grid cells). Our taxi agent's goal is to pick up the correct passenger and reach the destination cell [2]. To let the agent figure out the passenger instead of simply remembering the "location", we randomize the passengers' positions at the beginning of each episode. A reward in this domain is given only when the taxi drops the correct passenger to the destination cell. *Atari-Pong*, on the other hand, is a more complex domain, yet it is sparse in the sense that any reward is given when pong-rally finishes.

5.3. Evaluation Metrics

We evaluate our work on three fronts. First, the principal role of humans in the reinforcement learning process is to improve the environment sample efficiency; hence this serves as our primary indicator of success. Second, one of the major claims of this work is that the augmentation of human explanation in the form of a saliency map to binary feedback improves the human feedback sample efficiency; hence this is our second metric. Third, since we are using saliency maps as the means to let the human communicate with the agent the "important" regions to focus on, it becomes interesting to see whether EXPAND focuses on the correct regions as against to our baseline. Finally, this work presents an augmentation to the human input; hence we perform an ablation to verify that the performance gains we see are there because of this augmentation. We also perform a small experiment to analyze whether supplementing saliency feedback with more number of perturbations would help.

To verify how much the "important" objects like the taxi

[2] An example of image observation in *Taxi* can be found in the appendix

agent and the target passenger are affecting the agent's decisions, we compute the SARFA score (Puri et al., 2019), on those regions. SARFA is an apt choice for two reasons. First, it computes action-specific saliency, i.e., it will only give a high score when the agent finds the region to be "relevant" when taking one action and "irrelevant" when taking any other action. Second, SARFA can be efficiently computed from Q-values and perturbed observations, thus fitting perfectly to our setting. We are interested in seeing whether the agent has relatively lower SARFA scores for human-annotated relevant regions with actions labeled as "bad" (S_b^r, $\mathcal{A}_b$) and, conversely, relatively higher SARFA scores with actions labeled as "good" for these regions (S_a^r, $\mathcal{A}_a$). Hence the saliency score S_F is defined as,

$$S_F = \frac{1}{|\mathcal{A}_g|} \sum_{s_g^r \in S_g^r, a_g \in \mathcal{A}_g} SARFA(s_g^r, a_g) - \frac{1}{|\mathcal{A}_b|} \sum_{s_b^r \in S_b^r, a_b \in \mathcal{A}_b} SARFA(s_b^r, a_b)$$

5.4. Obtaining Feedback

Algorithm 1 mentions about sampling a trajectory to query to the user. In every n ($n = 4$) episodes, we sample one trajectory and query it for binary evaluative feedback and human saliency maps.

Following (Griffith et al., 2013) and (Arakawa et al., 2018), a synthetic oracle is used to provide simulated human feedback in all experiments. Specifically, on *Atari Pong*, we use the well-trained model from the *Atari Zoo* framework (Such et al., 2019); and on *Taxi*, we use a DQN model trained by ourself. The oracle must also annotate regions on the image observation to signal "relevant" regions, which is done by highlighting "objects" in the domain that are important for the agent to focus on. In our experiments, we use hard-coded models to highlight the taxi-cell, the destination-cell, and the target-passenger in *Taxi* domain, and the two paddles and the pong-ball in *Atari-Pong*.

Using synthetic oracle enables us to give consistent feedback across different runs, which allows us to fairly and systematically compare the effectiveness of different approaches. Additionally, it gives us flexibility in conducting our experiments and report more robust metric values.

5.5. Results

Feedback Sample Efficiency & Performance: Figure 2 compares feedback sample efficiency as well as performance of EXPAND (in red) with our baselines EXPAND-Advantage (in brown) and DQN-TAMER (in blue) on the *Taxi* domain. EXPAND learns a near-optimal policy in just 130k environment samples, whereas our baselines

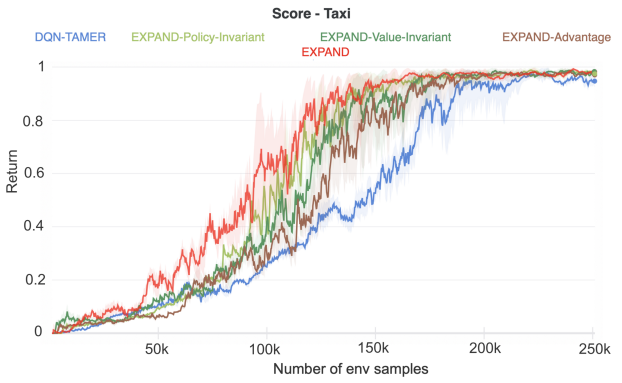

Figure 2. Score on *Taxi* domain, a running average over last 20 rollouts. The solid lines show the mean score over 5 random seeds. The shaded area represents the minimum and maximum scores in the 5 runs.

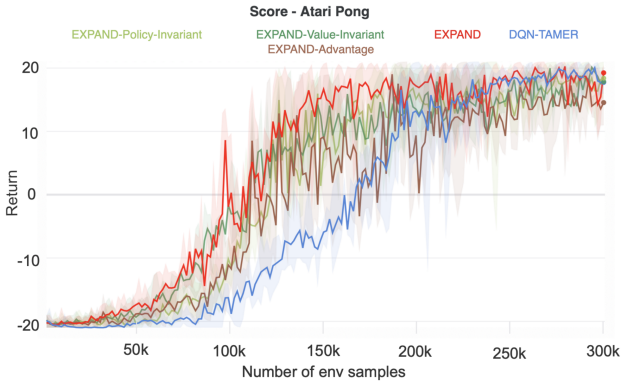

Figure 3. Score on *Atari-Pong* domain. The solid lines show the mean score over 5 random seeds. The shaded area represents the minimum and maximum scores in the 5 runs.

EXPAND-Advantage and DQN-TAMER take around 170k and 180k samples respectively to reach a similar performance. This is a 25% improvement in environment sample efficiency. On *Atari-Pong* (Figure 3), EXPAND learns a high-quality policy with the use of 150k environment samples, whereas the baselines DQN-TAMER and EXPAND-Advantage take around 230k and more than 250k respectively to reach similar performance. Regarding human feedback efficiency, our results show that, on the *Taxi* domain, EXPAND collects around 30k binary and saliency feedback pairs. In contrast, the baselines EXPAND-Advantage and DQN-TAMER collect over 40k binary feedback, a 25% improvement. On *Atari-Pong*, EXPAND (less than 40k feedback) achieves a near 30% improvement in human feedback efficiency over EXPAND-Advantage (over 60k feedback) and DQN-TAMER (over 55k feedback).

Agent Saliency Score: We keep track of the saliency score during training to ascertain how much the agent relies on the

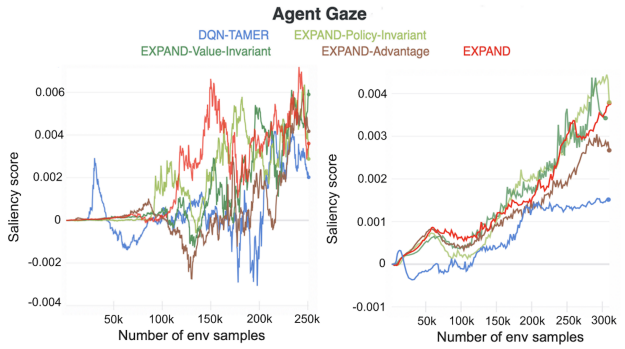

Figure 4. The left plot shows the saliency score on regions "destination", "passenger to pick up" and "player" in *Taxi* domain. The right plot shows the saliency score on regions "player-paddle", "ball" and "opponent-paddle" in *Atari-Pong*.

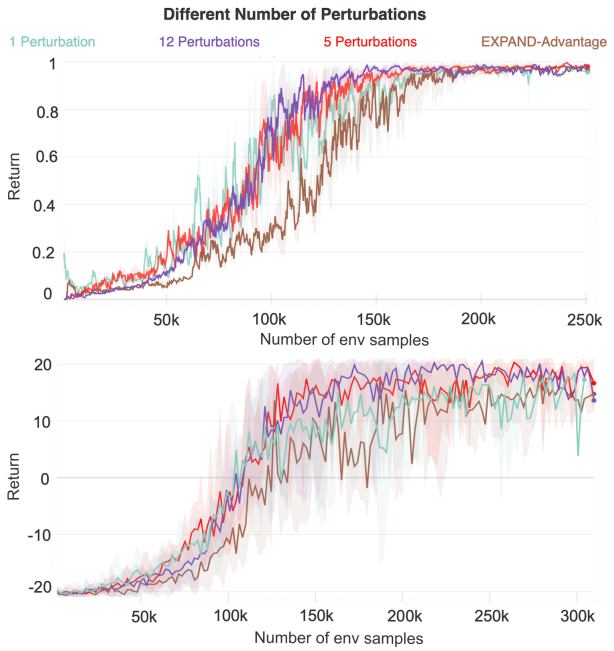

Figure 5. The learning curves of the variants of EXPAND with different number of perturbations on *Taxi* (top) and *Atari-Pong* (bottom).

"relevant" regions to make its decisions. A higher saliency score means the agent learns to pay more "attention" on "important" regions when taking "good" actions as well as learns to pay less "attention" on "important" regions when taking "bad" actions. Figure 4 shows the saliency score during the training process. We can observe that, by means of human explanations, EXPAND and its variants are able to attain a higher saliency score in a shorter time. The result validates our expectation that the loss terms we define will help the agent focus more on the "important" regions. Although there is no theoretical guarantee of a strong correlation between higher saliency score and better performance, the saliency score does give us a sense of how human explanations can facilitate agent learning.

Ablation Study: Figure 2 and 3 show that all the four variants (EXPAND, and EXPAND with individual loss terms) follow a similar pattern to convergence (near-optimal scores). We note that EXPAND-Policy-Invariant (in light green) and EXPAND-Value-Invariant (in dark green) perform significantly better than EXPAND-Advantage, highlighting that saliency losses alone provide significant improvements over baseline. Finally, we see combining these losses boosts this performance even further hinting to the fact that utilizing saliency information in addition to binary evaluations is a better approach.

Varying Number of Perturbations: We experimented with the number of perturbations required in each state to get the best performance. Figure 5 shows a comparison when the number of perturbations are varied to be $\{1, 5, 12\}$. The plots suggest that increasing the perturbations may entail only slight performance gains, and therefore setting the number of perturbations to be 5 for EXPAND is apt.

6. Conclusion & Future Work

In this work, we present a novel method to integrate human explanation, in the form of saliency maps on image observation states, to their binary evaluations of agent's actions, in a Human-in-the-Loop paradigm. We show that our proposed method, Explanation Augmented Feedback (EXPAND) outperforms our baseline DQN-TAMER in environment sample efficiency and provides significant improvements in human feedback sample efficiency. We also verify that the intuition of leveraging the information about which parts of the image are relevant can make the agent indeed focus on those regions. Finally, we also verify that supplementing human saliency feedback with perturbed irrelevant regions is helpful when multiple such perturbations are used.

In this work, we assume saliency feedback as human advice; however, the advice in the form of natural language interaction would be an improvement. Additional steps can be taken that use object tracker, extrapolation of given feedback samples to similar states, etc., to alleviate human effort in providing guidance. Moreover, we note that we have restricted "perturbations" to be Gaussian blurs to the state image. In contrast, future work can be to experiment with different types of perturbations (and even state-dependent perturbations that involve object manipulation).

Acknowledgements

Kambhampatis research is supported in part by ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-19-1-2119, AFOSR grant FA9550-18-1-0067, DARPA SAIL-ON grant W911NF-19-2-0006, NSF grants 1936997 (C-ACCEL), 1844325, and a NASA grant NNX17AD06G.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Arakawa, R., Kobayashi, S., Unno, Y., Tsuboi, Y., and Maeda, S.-i. Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748*, 2018.

Arumugam, D., Lee, J. K., Saskin, S., and Littman, M. L. Deep reinforcement learning from policy-dependent human feedback. *arXiv preprint arXiv:1902.04257*, 2019.

Baird, L. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Cederborg, T., Grover, I., Isbell, C. L., and Thomaz, A. L. Policy shaping with human teachers. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

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

Dietterich, T. G. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

Greydanus, S., Koul, A., Dodge, J., and Fern, A. Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*, 2017.

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*, pp. 2625–2633, 2013.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4565–4573. Curran Associates, Inc., 2016.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. In *Advances in neural information processing systems*, pp. 8011–8023, 2018.

Kim, H., Ohmura, Y., and Kuniyoshi, Y. Using human gaze to improve robustness against irrelevant objects in robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2020.

Knox, W. B. and Stone, P. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

Knox, W. B. and Stone, P. 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*, pp. 5–12. Citeseer, 2010.

Knox, W. B. and Stone, P. Reinforcement learning from simultaneous human and mdp reward. In *AAMAS*, pp. 475–482, 2012.

Krening, S., Harrison, B., Feigh, K. M., Isbell, C. L., Riedl, M., and Thomaz, A. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, 2016.

Littman, M. L., Topcu, U., Fu, J., Isbell, C., Wen, M., and MacGlashan, J. Environment-independent task specifications via gltl. *arXiv preprint arXiv:1704.04341*, 2017.

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., and Roberts, D. L. Learning something from nothing: Leveraging implicit human feedback strategies. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 607–612. IEEE, 2014.

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., and Roberts, D. L. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous agents and multi-agent systems*, 30(1):30–59, 2016.

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., and Littman, M. L. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2285–2294. JMLR. org, 2017.

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

Mott, A., Zoran, D., Chrzanowski, M., Wierstra, D., and Rezende, D. J. Towards interpretable reinforcement learning using attention augmented agents. In *Advances in Neural Information Processing Systems*, pp. 12329–12338, 2019.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

Ng, A. Y., Russell, S. J., et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Peng, B., MacGlashan, J., Loftin, R., Littman, M. L., Roberts, D. L., and Taylor, M. E. A need for speed: Adapting agent action speed to improve task learning from non-expert humans. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2016.

Piot, B., Geist, M., and Pietquin, O. Boosted bellman residual minimization handling expert demonstrations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 549–564. Springer, 2014.

Puri, N., Verma, S., Gupta, P., Kayastha, D., Deshmukh, S., Krishnamurthy, B., and Singh, S. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *International Conference on Learning Representations*, 2019.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Schaal, S. Learning from demonstration. In *Advances in neural information processing systems*, pp. 1040–1046, 1997.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. In *ICLR 2016 : International Conference on Learning Representations 2016*, 2016.

Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., and Ignateva, A. Deep attention recurrent q-network. *arXiv preprint arXiv:1512.01693*, 2015.

Such, F. P., Madhavan, V., Liu, R., Wang, R., Castro, P. S., Li, Y., Zhi, J., Schubert, L., Bellemare, M. G., Clune, J., and Lehman, J. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 3260–3267, 2019.

Tenorio-Gonzalez, A. C., Morales, E. F., and Villaseñor-Pineda, L. Dynamic reward shaping: training a robot by voice. In *Ibero-American conference on artificial intelligence*, pp. 483–492. Springer, 2010.

Thomaz, A. L., Breazeal, C., et al. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Aaai*, volume 6, pp. 1000–1005. Boston, MA, 2006.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Freitas, N. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

Warnell, G., Waytowich, N., Lawhern, V., and Stone, P. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1): 4945–4990, 2017.

Xiao, B., Lu, Q., Ramasubramanian, B., Clark, A., Bushnell, L., and Poovendran, R. Fresh: Interactive reward shaping in high-dimensional state spaces using human feedback. *arXiv preprint arXiv:2001.06781*, 2020.

Yeh, E., Gervasio, M., Sanchez, D., Crossley, M., and Myers, K. Bridging the gap: Converting human advice into imagined examples. *Advances in Cognitive Systems*, 6: 1168–1176, 2018.

Zahavy, T., Ben-Zrihem, N., and Mannor, S. Graying the black box: Understanding dqns. In *International Conference on Machine Learning*, pp. 1899–1908, 2016.

Zhang, R., Liu, Z., Zhang, L., Whritner, J. A., Muller, K. S., Hayhoe, M. M., and Ballard, D. H. Agil: Learning attention from human for visuomotor tasks. In *Proceedings of the european conference on computer vision (eccv)*, pp. 663–679, 2018.

Zhang, R., Torabi, F., Guan, L., Ballard, D. H., and Stone, P. Leveraging human guidance for deep reinforcement learning tasks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 6339–6346. International Joint Conferences on Artificial Intelligence Organization, 7 2019a. doi: 10.24963/ijcai.2019/884.

Zhang, R., Walshe, C., Liu, Z., Guan, L., Muller, K. S., Whritner, J. A., Zhang, L., Hayhoe, M. M., and Ballard, D. H. Atari-head: Atari human eye-tracking and demonstration dataset. *arXiv preprint arXiv:1903.06754*, 2019b.

Appendix

A. Hyperparameters

- Convolutional channels per layer: [32, 64, 64]
- Convolutional kernel sizes per layer: [8, 4, 3]
- Convolutional strides per layer: [4, 2, 1]
- Convolutional padding per layer: [0, 0, 0]
- Fully connected layer hidden units: [512, number of actions]
- Update interval: 4
- Discount factor: 0.99
- Replay buffer size: 50,000
- Batch size: 64
- Feedback buffer size: 50,000 in *Atari-Pong*, 10,000 in *Taxi* [3]
- Feedback batch size: 64 in *Atari-Pong*, 32 in *Taxi*
- Learning Rate: 0.0001
- Optimizer: Adam
- Prioritized replay exponent $\alpha = 0.6$
- Prioritized replay importance sampling exponent $\beta = 0.4$
- *Advantage loss* margin $l_m = 0.05$
- Rewards: clip to [-1, 1]
- ϵ episodic decay factor λ_ϵ: 0.99 in *Taxi*, 0.9 in *Atari-Pong*

B. Settings of Gaussian Perturbation

- 1 Perturbation:
  - filter size: 5, σ: 5
- 5 Perturbations:
  - filter size: 5, σ: 2
  - filter size: 5, σ: 5
  - filter size: 5, σ: 10
  - filter size: 11, σ: 5
  - filter size: 11, σ: 10
- 12 Perturbations:
  - filter size: 5, σ: 2
  - filter size: 5, σ: 5
  - filter size: 5, σ: 10
  - filter size: 7, σ: 3
  - filter size: 7, σ: 5
  - filter size: 7, σ: 10
  - filter size: 9, σ: 3
  - filter size: 9, σ: 5
  - filter size: 9, σ: 10
  - filter size: 11, σ: 3
  - filter size: 11, σ: 5
  - filter size: 11, σ: 10

[3] We use a smaller feedback buffer size for *Taxi* because the trajectory length in *Taxi* is much smaller than that in *Atari-Pong*, so less human feedback data are collected per query.

C. Examples of Perturbation in *Taxi*

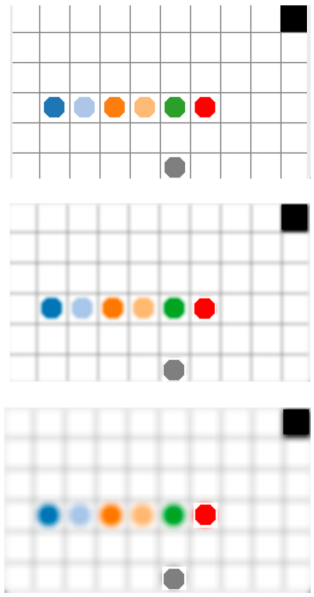

Figure 6. Examples of different perturbations on irrelevant regions in an image observation in the *Taxi* domain. The top one is the original observation. The remainder are observations with different perturbations on irrelevant regions, which are regions excluding the taxi-agent (the gray cell), the passenger to pick up (the red cell) and the destination (the black cell).