GANterfactual-RL: Understanding Reinforcement Learning Agents' Strategies through Visual Counterfactual Explanations

Tobias Huber
University of Augsburg
Augsburg, Germany
tobias.huber@uni-a.de

Maximilian Demmler
University of Augsburg
Augsburg, Germany
maxdemmler@googlemail.com

Silvan Mertes
University of Augsburg
Augsburg, Germany
silvan.mertes@uni-a.de

Matthew L. Olson
Oregon State University
Corvallis, OR, United States
olsomatt@oregonstate.edu

Elisabeth André
University of Augsburg
Augsburg, Germany
andre@informatik.uni-augsburg.de

ABSTRACT

Counterfactual explanations are a common tool to explain artificial intelligence models. For Reinforcement Learning (RL) agents, they answer "Why not?" or "What if?" questions by illustrating what minimal change to a state is needed such that an agent chooses a different action. Generating counterfactual explanations for RL agents with visual input is especially challenging because of their large state spaces and because their decisions are part of an overarching policy, which includes long-term decision-making. However, research focusing on counterfactual explanations, specifically for RL agents with visual input, is scarce and does not go beyond identifying defective agents. It is unclear whether counterfactual explanations are still helpful for more complex tasks like analyzing the learned strategies of different agents or choosing a fitting agent for a specific task. We propose a novel but simple method to generate counterfactual explanations for RL agents by formulating the problem as a domain transfer problem which allows the use of adversarial learning techniques like StarGAN. Our method is fully model-agnostic and we demonstrate that it outperforms the only previous method in several computational metrics. Furthermore, we show in a user study that our method performs best when analyzing which strategies different agents pursue.

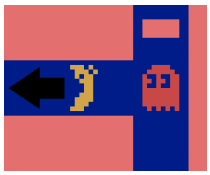
KEYWORDS

Explainable Deep Reinforcement Learning; Explainable Artificial Intelligence; Interpretable Machine Learning

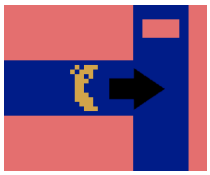
ACM Reference Format:
Tobias Huber, Maximilian Demmler, Silvan Mertes, Matthew L. Olson, and Elisabeth André. 2023. GANterfactual-RL: Understanding Reinforcement Learning Agents' Strategies through Visual Counterfactual Explanations . In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 10 pages.

1 INTRODUCTION

Modern Reinforcement Learning (RL) agents use increasingly complex state spaces and deep learning algorithms, making the decisions and strategies of such agents hard to understand [13]. At the same time, these deep RL agents are being deployed into increasingly high-risk domains like healthcare, autonomous driving, and robotic navigation [11, 21, 43]. In such domains, it is crucial to be able to understand the agents to enable appropriate use of them and to facilitate human-agent cooperation [37]. One prominent paradigm to make the decisions of intelligent agents transparent and comprehensible are so-called *Counterfactual Explanations*. By providing an alternative reality where the agent would have made a different decision, these explanations follow a rather human way of describing decisions [4, 27]. For example, if a person would have to explain why a warehouse robot took a detour instead of directly moving to its desired target, they would probably give an explanation similar to *If there was no production worker in the way, the robot would have moved straight to its target* - and, by doing so, give a counterfactual explanation of the warehouse robot's behavior. Figure 1 shows a similar situation from the Atari game Pacman.



(a) Original state where Pacman goes left.



(b) Counterfactual state where Pacman goes right.

Figure 1: Example for a counterfactual explanation: In the original situation (a), the agent does not take the fastest path to the pill in the top right corner. It is unclear if the agent is afraid of the ghost or does not recognize the shortest path. The counterfactual state (b) shows that the agent would have taken the fastest path to the pill if the ghost was not there. This indicates that the agent is afraid of the ghost.

In other machine learning domains, such as image classification, counterfactual explanations are already frequently used. However, this is not the case for RL, as several factors make explaining the decisions of RL agents more challenging. For one, RL agents are used for sequential decision-making tasks: their actions are not isolated. These actions are part of a long-term strategy that might be influenced by delayed rewards. Secondly, RL agents are not trained on a given ground truth strategy. The reward function only indirectly specifies the agent's goals [10]. The emerging strategies

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

might not be what humans would expect, even if the strategy is optimal for the reward function. Finally, for RL agents, there is no direct counterpart to the training datasets used by supervised models. Therefore, counterfactual explanation approaches for supervised models that utilize the training data cannot be applied to RL agents without adjustment [41].

Due to the difficulties mentioned above, there is only one approach that focuses on creating counterfactual explanations for deep RL agents with visual input [32]. This approach utilizes a complex combination of models where the final generator is only indirectly trained to change the action. Olson et al. [32] show that their approach can be applied to a variety of RL environments and helps users identify a flawed agent. With the help of their counterfactual explanations, users were able to differentiate between a normal RL agent for the Atari game Space Invaders and a flawed agent that did not see a specific in-game object. For this task, it is sufficient for the counterfactual explanation to not change the particular object at all while other objects frequently change. This clearly communicates that the unchanged object is irrelevant and ignored by the agent, implying that it is not seen at all.

But for counterfactual explanations to be employed more widely, they also have to be useful for more complex tasks. According to Hoffman et al. [14], one of the main goals of a good explanation is to refine the user's mental model of the agent. For RL agents, this includes understanding what strategy and intentions an agent pursues. Another critical goal for explanations is that they should help users to calibrate their trust in different agents [14]. For RL agents, this entails that users should be able to choose fitting agents for specific problems, which is more complex than simply identifying defective agents. The two aforementioned challenges require counterfactual explanations to not only convey *what* objects need to change but also *how* the objects need to be altered to change the agents' policy.

To tackle these challenges, this paper proposes a novel method for generating counterfactual explanations for RL agents with visual input. We do so by formulating the generation problem as a domain transfer problem where the domains are represented by sets of states that lead the agent to different actions. Our approach is fully model-agnostic, easier to train than the approach presented by Olson et al., and includes the counterfactual actions more directly into the training routine by solving an action-to-action domain transfer problem. We evaluate our approach with computational metrics (e.g., how often do the counterfactuals change the agent's decision) and a user study using the Atari Learning Environment (ALE) [3], a common benchmark for RL agents with visual input. In our user study, we present participants with different kinds of counterfactual explanations and investigate whether this helps them to understand the strategies of Pacman agents. Furthermore, we investigate if the counterfactuals help them to calibrate their trust, so they can choose fitting agents for specific tasks (surviving or receiving points).

As such, the contributions of this paper are as follows: We formulate a novel, model-agnostic approach for generating counterfactual explanations for RL agents. We demonstrate that our approach outperforms the previous method in several computational metrics. Furthermore, we conduct a user study that shows, for the first time, that counterfactual explanations can help to understand the strategies of RL agents. This user study also identifies current deficiencies of counterfactual explanations for RL agents that point the way for future work.

2 RELATED WORK

Our work deals with post-hoc explanations that are generated for fully trained black-box agents. Recent years saw a plethora of work on such explanations for (deep) RL agents. The literature often divides them by scope into global and local explanations. Global explanations try to explain the agent's overall strategy. This can be done by picking a subset of important state-action pairs that summarize the agent's strategy [1, 15] or by distilling the agent's policy into a simpler model like a finite state machine [8] or a soft decision tree [7]. In this paper, we focus on local explanations that explain a specific decision of an agent. The most common approach to local explanations for RL agents are Feature Attribution or Saliency Map methods [18, 34, 44]. These methods try to identify the most important input features for a specific decision and highlight them, for example in a heatmap. However, recent work questioned whether one can rely on post-hoc feature attribution to faithfully represent the agent's internal reasoning [2, 17]. Furthermore, previous studies showed that saliency maps for visual RL agents are hard to understand for end-users [19]. Counterfactual explanations are another type of local explanation. Since they follow the human thinking paradigm of counterfactual reasoning, it is often argued that they are easier to interpret than feature attribution methods [4, 27].

For classification models, there is a growing body of work on counterfactual explanations. In 2017, Wachter et al. [39] were the first to introduce counterfactual explanations into the XAI domain by defining them as an optimization problem. Since then, various approaches to generate such counterfactuals were proposed, e.g., van Looveren and Klaise [23], and [12].

As various research has observed that generating counterfactual explanations is, at its core, a generative problem, the use of generative models like Generative Adversarial Networks (GANs) quickly became prevailing in state-of-the-art counterfactual explanation generation algorithms. E.g., Nemirovsky et al. [31] proposed CounterGAN, a framework to build highly realistic and actionable counterfactual explanations. Zhao et al. [45] propose an approach for generating counterfactual image explanations by using text descriptions of relevant features of an image to be explained. Furthermore, various specialized GAN-based algorithms were introduced to generate counterfactual explanations in the medical domain [24, 25, 38]. More recent frameworks for counterfactual explanation generation make use of the StyleGAN architecture, which implicitly models style-related aspects of an image, which makes it perfectly suitable for a whole range of image classification tasks [22, 35]. As for a broad range of use cases, it is essential to be able to provide explanations for multiple counter-classes, various approaches have focused on that particular capability by using architectures based on StarGAN, an adversarial framework that was specifically designed for image translation between multiple domains [42, 46]. One drawback of the aforementioned approaches for supervised learning is that their GANs are trained to transfer between domains given by the labeled classes from the classifier's training dataset.

Then they add additional measures (e.g., loss functions [25, 46]), to ensure that the generated counterfactuals are actually classified as the desired class by the classifier that is to be explained. This is not possible for RL agents that do not have a training set. Furthermore, the additional measures are often not model-agnostic.

RL is often used to create counterfactual explanations for other models (for example in [5]). However, to the best of the authors' knowledge, there is only one previous work on generating visual counterfactual explanations for RL agents [32]. Olson et al. [32] train an encoder E that creates an action-invariant latent representation of the agent's latent space. This is achieved by adversarially training E in tandem with a discriminator D, where D tries to predict the agent's action and E aims to make the decision of D as uncertain as possible. In addition, they train a generative model G to replicate states s based on the action-invariant latent representation $E(s)$ and the agent's action probability distribution $\pi(s)$ for this state. By providing G with a counterfactual action distribution $\pi(s)'$, they obtain a state that is similar to s but brings the agent's action distribution closer to the desired counterfactual distribution. However, Olson et al. argue that an arbitrary counterfactual action distribution does not represent a realistic agent output and thus leads to unrealistic counterfactual states. To avoid this, they train an additional Wasserstein Auto Encoder and use it to perform gradient descent in the latent space of the agent towards an agent output that resembles the desired counterfactual action. Olson et al. refer to their approach as Counterfactual State Explanations (CSE), therefore we will also refer to it as CSE in this paper.

The CSE approach is fairly complex and requires extensive access to the agent's inner workings. Furthermore, as Olson et al. mention themselves, the loss function of the generator G does not directly force the resulting state $G(E(s), \pi(s)')$ to be classified as the counterfactual action distribution $\pi(s)'$. This is only learned indirectly by replicating states based on the action-invariant latent space and the desired action distribution $\pi(s)'$. As we show in Section 4, this does not seem to be enough to change the agent's decision correctly. To solve those problems, we formulate a simpler counterfactual generation method that uses the counterfactual actions in a more direct way.

3 APPROACH

3.1 GANterfactual-RL

RL agents are usually employed in a Markov Decision Process (MDP) which consists of states $s \in S$, actions $a \in A$, and rewards r. Given a state s, the goal of an RL agent $\pi : S \rightarrow A$ is to choose an action $\pi(s)$ that maximizes its cumulative future rewards. To explain such an agent, the objective of a counterfactual explanation approach for RL agents is defined as follows. Given an original state s and a desired counterfactual action a', we want a counterfactual state s' that makes the agent choose the desired action $\pi(s') = a'$. Hereby, the original state s should be altered as little as possible. On an abstract level, the action $\pi(s)$ that the agent chooses for a state s can be seen as a top-level feature that describes a combination of several underlying features which the agent considers to be relevant for its decision. Thus, the counterfactual state s' should only change the features that are relevant to the agent's decision, while maintaining all other features not relevant to the decision.

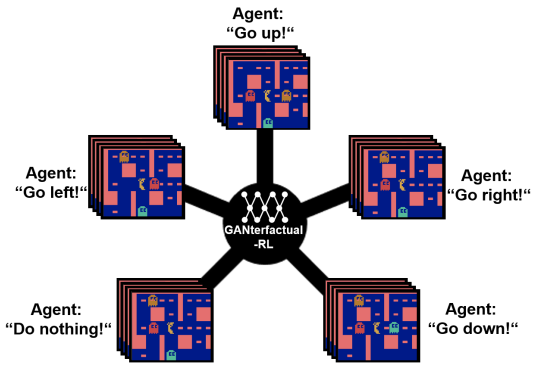


Figure 2: Schematic of our counterfactual generation approach. We formulate the problem as domain transfer where each domain represents an action. States are assigned to domains based on the action that the agent chooses for them.

This is similar to image-to-image translation, where features that are relevant for a certain image domain should be transformed into features leading to another image domain, while all other features have to be maintained (e.g., the background should remain constant when transforming horses to zebras). Taken together, we can formulate the generation of counterfactual states for RL agents as a domain transfer problem similar to image-to-image translation: The agent's action space A defines the different domains $A_i = \{s \in S | \pi(s) = a_i\}$, where each state belongs to the domain that corresponds to the action that the agent chooses for this state (see Figure 2).

To solve the reformulated domain transfer problem, we base our system on the StarGAN architecture [6], since RL agents usually use more than two actions. The StarGAN architecture incorporates multiple loss components that can be reformulated to be applicable to the RL domain. The first component, the so-called adversarial loss, leads the network to produce highly realistic states that look like states from the original environment. Reformulated for the task of generating RL states, we define it as follows (following Choi et al. [6] we use a Wasserstein objective with gradient penalty):

$$\mathcal{L}_{adv} = \mathbb{E}_s \left[D_{src}(s)\right] - \mathbb{E}_{s,a'} \left[D_{src}(G(s, a'))\right] - \lambda_{gp} \mathbb{E}_{\hat{s}} \left[(||\nabla_{\hat{s}} D_{src}(\hat{s})||_2 - 1)^2\right],$$

where D_{src} is the StarGAN's *discriminator* network and G its *generator* network. The second loss component, which is specific to the StarGAN architecture, guides the generator network to produce states that lead to the desired counterfactual actions. It consists of two sub-objectives, one that is applied while the network is fed with original (*real*) states from the training set (Eq. 1), and the other while the network is generating counterfactual states (Eq. 2):

$$\mathcal{L}^{a}_{cls} = \mathbb{E}_{s,a}[-\log D_{cls}(a|s)], \tag{1}$$

$$\mathcal{L}^{a'}_{cls} = \mathbb{E}_{s,a'}[-\log D_{cls}(a'|G(s, a'))], \tag{2}$$

where D_{cls} refers to the StarGAN discriminator's classification output, which learns to approximate the action that the agent is performing in a particular state. Further, as counterfactual states should be as close to the original states as possible, a *Reconstruction*

Loss is used. This loss forces the network to only change features that are relevant to the agent's choice of action:

$$\mathcal{L}_{rec} = \mathbb{E}_{s,a,a'}[||s - G(G(s,a'),a)||_1]$$

Taken together, the whole loss of the StarGAN architecture, reformulated for RL counterfactual explanations, is defined as follows:

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^{a}_{cls},$$
$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^{a'}_{cls} + \lambda_{rec}\,\mathcal{L}_{rec},$$

where λ_{cls} and λ_{rec} are weights controlling the corresponding loss component's relevance. Since our approach utilizes a GAN architecture to generate counterfactuals for RL agents, we refer to it as GANterfactual-RL.

3.2 Dataset Generation

As described above, our GANterfactual-RL approach relies on training data in the form of state-action pairs. Olson et al. [32] train their CSE approach on state-action pairs generated by concurrently running an MDP with a trained agent. This strategy is simple but allows for little control over the training data, which can lead to the following complications:

- Frames extracted from a running MDP contain a temporal pattern since consecutive states typically have a high correlation. Such correlations and patterns can lead to bias and sub-optimal convergence during training.
- For episodic MDPs, there is a high probability of reaching the same state throughout several episodes. This is amplified by the fact that RL agents often learn to execute only a few optimal trajectories. This results in duplicate samples that are effectively over-sampled during training.
- RL agents generally do not execute each action equally frequently, since most environments contain actions that are useful more often than others. This leads to an imbalanced amount of training samples per domain.

To mitigate the aforementioned issues, we propose to generate datasets as follows: Data is gathered by running a trained agent in an MDP. Each state corresponds to one dataset sample and is labeled with the action that the agent chooses to execute in this state. An ϵ-greedy policy (ϵ=0.2 in our case) is used to increase the diversity of states reached over multiple episodes. State-action pairs with an explored (randomly chosen) action are not added to the dataset. After the data is gathered, duplicates are removed. Then, a class balancing technique (under-sampling in our case) is used to account for over- or underrepresented actions. Finally, the dataset is split into a training set, a test set, and potentially a validation set.

Most of these techniques are commonly used in other application domains of machine learning. However, to our best knowledge, this is the first work to generate and preprocess datasets for generating counterfactual explanations for RL agents.

3.3 Application to Atari Domain

Environment. The environments we use for our experiments are the Atari 2600 games MsPacman (henceforth referred to as Pacman) and Space Invaders, included in the Arcade Learning Environment (ALE) [3]. The ALE states are based on the raw pixel values of the game. Each input frame is cropped so that only the actual playing field remains. This removes components such as the score and life indicators which would allow participants to easily see which agent receives higher scores. After that, we use the same preprocessing as Mnih et al. [29]. Two steps from this preprocessing are particularly important for us. First, the frames are gray-scaled and downsized. Second, in addition to the current frame, the agent receives the last three preprocessed frames as input. This allows the agent to detect temporal relations. The ALE actions normally correspond to the meaningful actions achieved with an Atari 2600 controller (e.g. six actions for Space Invaders). Since we wanted to use our Pacman agents in a user study we removed 4 redundant actions (e.g., *Up & Right*) whose effect differs between situations and is therefore hard to convey to participants. This left us with 5 actions for Pacman (*Do nothing, Up, Down, Left, Right*).

Agent Training. To evaluate participants' ability to differentiate between alternative agents and analyze their strategies, we modified the reward function of three Pacman agents. This is a more natural method of obtaining different agents compared to withholding information from the agent as Olson et al. [32] did. Furthermore, it results in agents that behave qualitatively differently. Therefore participants have to actually analyze the agents' strategies instead of simply looking for objects that the agents ignore.

- ***Blue-Ghost Agent***: This agent was trained using the default reward function of the ALE, where blue ghosts get the highest reward.
- ***Power Pill Agent***: This agent only received positive rewards for eating power pills.
- ***Fear-Ghost Agent***: This agent got a small positive reward of 1 for every step in which it did not die to ghosts.

For training the first two Pacman agents, we use the DQN algorithm [29]. Each agent was trained for 5 Million steps. The *fear-ghosts agent* was trained using the ACER algorithm [40] for 10M steps. At the end of the training period, the best-performing policy is restored. For all three agents, we build upon the OpenAI baselines [9] repository. For Space Invaders, we used the two Asynchronous Advantage Actor-Critic (A3C) agents trained by Olson et al. [32]. For training details, we refer to their paper. One agent is trained normally, while the other agent is flawed and does not see the laser cannon at the bottom of the screen.

GANterfactual-RL on Atari. To generate human-understandable counterfactual explanations for our Atari agents, the generated counterfactual states should represent the frames that humans see during gameplay. That means we cannot train our GANterfactual-RL model on the preprocessed and stacked frames that the Atari agents use. Instead, we train it on the cropped RGB frames before preprocessing. The only preprocessing we still use on those frames is a countermeasure against flickering objects in Atari games, which was proposed by Mnih et al. [29]. While generating the dataset, we only save the most recent of the four stacked frames for each state s. This frame generally influences the agent's decision the most. For feeding the counterfactual frame back into the agent (e.g., to evaluate the approach), we stack it four times and then apply preprocessing.

Implementation details are described in the appendix (to be found in the authors' version [16]). The full code is available online.[1]

[1] https://github.com/hcmlab/GANterfactual-RL

4 COMPUTATIONAL EVALUATION

4.1 Used Metrics

We evaluate our approach using the metrics *validity* (or *success rate*), *proximity* (or *cost*), *sparsity*, and *generation time*. We consider these metrics to be the most suitable and widely used metrics for image-based counterfactual explanations [5, 20, 30, 33].

Validity captures the rate of CounterFactuals (CFs) that actually evoke the targeted action when fed to the agent. With N_T being true CFs (correctly changing the agent's action), N_F being false CFs, and N the total amount of evaluated CFs, this metric is defined as:

$$Validity = \frac{N_T}{N_T + N_F} = \frac{N_T}{N}$$

Proximity measures the similarity between an original state image and its CF via the $L1$-norm. We normalize the metric to measure the proximity in the range $[0, 1]$.

$$Proximity(s, G) = 1 - \frac{1}{255 \cdot \mathcal{S}} ||s - G(s, a)||_1$$

where s is the original state image, $G(s, a)$ is the generated CF for an arbitrary target action domain a and $\mathcal{S}$ is the domain of color values of s ($\mathcal{S} = 3 \cdot Width \cdot Height$ for RGB-encoded images). The normalization with $255 \cdot \mathcal{S}$ assumes an 8-bit color encoding with color values in range $[0, 255]$. High proximity values are desirable since they indicate small adjustments to the original state.

Sparsity quantifies the number of unmodified pixel values between an original state image and its CF via the $L0$-norm (a pseudo-norm that counts the number of non-zero entries of a vector/matrix). The sparsity is normalized to the range $[0, 1]$ as well.

$$Sparsity(s, G) = 1 - \frac{1}{\mathcal{S}} ||s - G(s, a)||_0$$

A completely altered image has a sparsity of 0, an unmodified image has a sparsity of 1. High sparsity values are thus desirable.

Generation Time determines the time it takes to generate one CF with a trained generator, not including pre- or post-processing.

4.2 Computational Results

The computational results for the three Pacman agents are shown in Table 1 and the results for the two Space Invaders agents in Table 2. For the Pacman agents, we generated fully cleaned datasets (Section 3.2) and sampled 10% of each action for the evaluation test set. To show the contribution of our proposed dataset generation, we additionally trained a GANterfactual-RL model for the *blue-ghost agent* without the steps proposed in Section 3.2 and evaluated it on the test set from the clean dataset. This dropped the validity to 0.45 and sparsity to 0.50 ± 0.01 while the other values stayed comparable. To be more comparable to the results by Olson et al. [32], we do not remove duplicates from the Space Invaders datasets and do not apply class balancing. Here we create the test set by sampling 500 states for each action and removing all duplicates of these states from the training set. Our GANterfactual-RL approach outperforms the CSE counterfactuals in every single metric.

Figure 3 shows example counterfactuals generated for the Pacman *fear-ghosts agent* and the two Space Invaders agents. Additional examples for all our agents can be seen in the appendix [16].

Table 1: Computational evaluation results for the Pacman agents. *Proximity*, *sparsity* and *generation time* are specified by *mean* ± *standard deviation*.

Approach	*Validity* (↑)	*Proximity* (↑)	*Sparsity* (↑)	*Gen. Time [s]* (↓)
Blue-Ghost Agent				
Ours	0.59	0.997 ± 0.001	0.73 ± 0.02	0.011 ± 0.012
CSE	0.28	0.992 ± 0.002	0.33 ± 0.03	0.085 ± 0.021
Power-Pill Agent				
Ours	0.49	0.997 ± 0.001	0.70 ± 0.02	0.011 ± 0.008
CSE	0.20	0.993 ± 0.002	0.32 ± 0.02	0.566 ± 0.731
Fear-Ghost Agent				
Ours	0.46	0.995 ± 0.001	0.45 ± 0.01	0.013 ± 0.014
CSE	0.20	0.992 ± 0.002	0.32 ± 0.04	0.020 ± 0.017

Table 2: Computational evaluation results for the Space Invaders agents. *Proximity*, *sparsity* and *generation time* are specified by *mean* ± *standard deviation*.

Approach	*Validity* (↑)	*Proximity* (↑)	*Sparsity* (↑)	*Gen. Time [s]* (↓)
Normal Agent				
Ours	0.70	0.998 ± 0.002	0.97 ± 0.02	0.011 ± 0.013
CSE	0.18	0.995 ± 0.003	0.89 ± 0.05	6.180 ± 9.727
Flawed Agent				
Ours	0.53	0.998 ± 0.002	0.96 ± 0.01	0.011 ± 0.015
CSE	0.17	0.995 ± 0.004	0.94 ± 0.01	0.020 ± 0.035

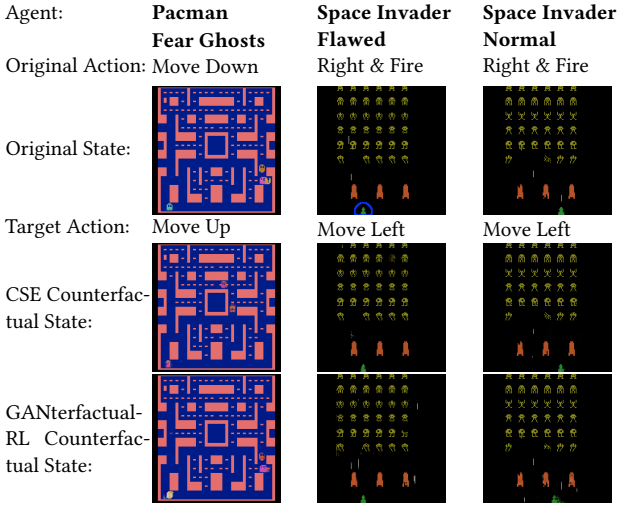


Figure 3: Example counterfactual states. Our approach does not change the *Laser Cannon* (marked in blue) for the flawed agent, who does not see it, but changes it for the normal agent.

5 USER STUDY

5.1 Study Design

5.1.1 Research Question and Hypothesis. The research question for our study was which counterfactual explanations help users to understand the strategies of RL agents and help them to choose fitting

agents for a specific task. We hypothesized that our GANterfactual-RL method is more useful than the CSE method and is more useful than a presentation of the original states without counterfactuals. Further, we thought that the counterfactuals generated by the CSE approach might mislead participants due to the low validity of the generated counterfactual explanations (see Section 4). Therefore, we hypothesized that only providing the original states is more useful than adding CSE counterfactuals.

5.1.2 *Dependent Variables and Main Tasks.*

Agent Understanding Task. To measure whether participants understand the strategies of different agents and build a correct mental model of them, we used an agent understanding task inspired by Hoffman et al. [14] and Huber et al. [19]. Here, participants were presented with five states and the actions that the agent chooses in these states. This was done for each of the three Pacman agents described in Section 3.3 (one agent at a time). The states were selected by the HIGHLIGHTS-Div algorithm [1]. To this end, we let each trained agent play for additional 50 episodes and chose the most important states according to HIGHLIGHTS-Div. The resulting states show gameplay that is typical for the agent, without the need to manually select states that might be biased toward our approach. Based on these states (and additional explanations depending on the condition), participants had to select up to two in-game objects that were most important for the agent's strategy from a list of objects (Pacman, normal pills, power pills, ghosts, blue ghosts, or cherries). As described in Section 3.3, each agent, strongly focuses on a different single in-game object depending on their reward function (e.g., the *fear-ghosts agent* focuses on normal ghosts). If the participants select this object and none of the other objects, they receive a point. The only exception is Pacman. Every agent heavily relies on the position of Pacman as a source of information. Therefore, participants receive the point whether they select Pacman or not.

Agent Comparison Task. To measure how well the participants' trust is calibrated, we used an agent comparison task inspired by Amir and Amir [1] and Miller [28]. Here, we implicitly measure if the participants' trust is appropriate by asking them, for each possible pair of the three Pacman agents, which agent they would like to play on their behalf to obtain certain goals. Since a single agent can be good for one goal but bad for another, this requires a deeper analysis than the distinction between a normal and a defective agent. For each pair, the participants are shown their own descriptions of each agent from the agent understanding task and the same states and explanations that they saw during the agent understanding task. Then they have to decide which agent should play on their behalf to achieve more points and which agent should play on their behalf to survive longer. We know the ground truth for this by measuring the agents' average score and amount of steps for the 50 episodes used to find the HIGHLIGHTS states. The amount of steps that the *blue-ghost agent* and the *power pill agent* survive is so close that we do not include this specific comparison in the evaluation.

Explanation Satisfaction. To measure the participant's subjective satisfaction, we use statements adapted from the Explanation Satisfaction Scale by Hoffman et al. [14]. Participants have to rate their agreement with each statement on a 5-point Likert scale. Participants' final rating was averaged over all those ratings, reversing the rating of negative statements. We do this once after the agent understanding task and once after the agent comparison task in case there are satisfaction differences between the tasks.

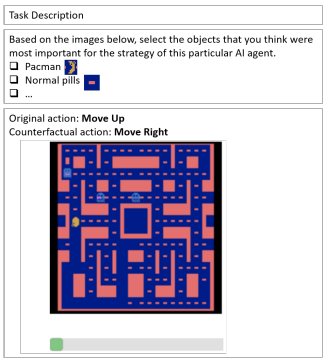


Figure 4: A simplified scheme of the beginning of our agent understanding task with a single example state.

5.1.3 *Conditions and Explanation Presentation.*

We used three independent conditions, one *Control* condition without explanations and two conditions where the states during the agent understanding task and the agent comparison task are accompanied by counterfactual explanations. In the *CSE* condition, the counterfactuals are generated by the approach from Olson et al. [32], and in the *GANterfactual-RL* condition the counterfactuals are generated by our proposed method. The presentation of the counterfactual explanations is designed as follows. For each state, we generate a single counterfactual state. We were concerned that too many counterfactual states would cause too much cognitive load. The way that MsPacman is implemented, actions that do nothing or move directly into a wall are ignored. To generate meaningful counterfactual states, we limited the counterfactual action to turning around in a corridor and randomly selecting a new direction at an intersection (do not turn around). The counterfactual states are presented by a slider under each state. Moving the slider from left to right linearly interpolates the original state to the counterfactual state (*per-pixel interpolation*). The original and counterfactual actions are written above the state. Figure 4 shows a simplified version of the beginning of our agent understanding task.

5.1.4 *Procedure and Compensation.*

After completing a consent form, participants were asked to answer demographic questions (age and gender) and questions regarding their experience with Pacman and their views on AI. Then, they were shown a tutorial explaining the rules of the game Pacman and were asked to play the game to familiarize themselves with it. To verify that participants understood the rules, they were asked to complete a quiz and were only allowed to proceed with the survey after answering all questions correctly. Afterward, participants in the counterfactual conditions received additional information and another quiz regarding the counterfactual explanations. Then, they proceeded to the agent understanding task which was repeated three times, once for each agent. The order of the agents was randomized. After that,

participants filled the explanation satisfaction scale and continued to the agent comparison task. Again, this task was repeated three times, once for each possible agent pair, and the order was randomized. Finally, participants had to complete another satisfaction scale for the agent comparison task. Participants got a compensation of 5$ for participating in the study. As an incentive to do the tasks properly, they received a bonus payment of 10 cents for each point they get in the agent understanding task and 5 cents for each point in the agent comparison task. The complete questionnaire can be seen in the appendix [16]. We preregistered our study online.[2]

5.1.5 Participants. We recruited participants through Amazon Mechanical Turk. Participation was limited to Mechanical Turk Masters from the US, UK, or Canada (to ensure a sufficient English level) with a task approval rate greater than 95% and without color vision impairment. We conducted a power analysis with an estimated medium effect size of 0.7 based on previous similar experiments [19, 25, 26]. This determined that we need 28 participants per condition to achieve a power of 0.8. and a significance level of 0.05. To account for participant exclusions, we recruited 30 participants per condition. Participants were excluded if they did not look at any of the counterfactual explanations for any of the agents during the agent understanding task, if their textual answers were nonsensical or if they took considerably less time than the average. This left us with 30 participants in the Control condition, 28 participants in the CSE condition, and 23 in the GANterfactual-RL condition.

The distribution of age, AI experience, and Pacman experience was similar between the conditions (see the appendix Huber et al. [16]). There was a difference in the gender distribution and the attitude towards AI between the conditions. The Control condition had 40% female participants, the CSE condition had 32% and the GANterfactual-RL condition had 26%. The mean attitude towards AI was the highest in the GANterfactual-RL condition and the lowest in the Control condition (see the appendix [16]).

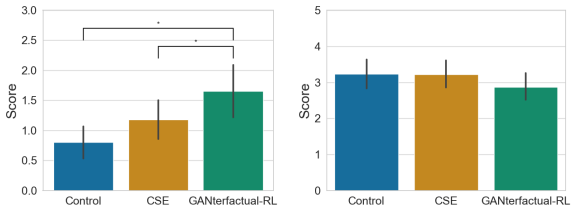
5.2 Results

The results for the participants' scores during the main tasks can be seen in Figure 5, while their explanation satisfaction values are shown in Figure 6. In the following, we will summarize the results of our main hypotheses, which we analyzed using non-parametric one-tailed Mann-Whitney U tests.

Counterfactuals helped participants to understand the agents' strategies. In the agent understanding task, there was a significant difference between the Control condition (M=0.8) and the GANterfactual-RL condition (M=1.65), U=181, p=0.001, r=0.477.[3] Contrary to our hypothesis, the Control condition got lower scores than the CSE condition (M=0.8 vs M=1.18), p=0.953.

Our GANterfactual-RL explanations were significantly more useful than the CSE approach for understanding the agents' strategies. In the agent understanding task, the CSE condition got a mean score of 1.18, while the GANterfactual-RL condition got a mean score of 1.65 (U=232, p=0.038, r=0.2795).

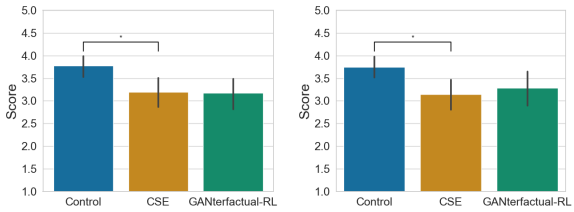
The increased understanding of the agents' strategies did not result in a more calibrated trust. Contrary to our hypothesis, there were no significant differences in the trust task (Control vs CSE: p=0.536, Control vs. GANterfactual-RL: p=0.852, CSE vs GANterfactual-RL: p=0.876).



(a) Total score (summed over all three agents) for the agent understanding task.

(b) Number of correct agent selections in the agent comparison task (Out of five).

Figure 5: Comparison of participants' average performance in each task, by condition. Error bars show the 95% CI.



(a) Agent Understanding Task.

(b) Agent Comparison Task.

Figure 6: Comparison of participants' average explanation satisfaction in each task, by condition.

Counterfactuals did not increase explanation satisfaction. Even though participants objectively had a better understanding of the agents' strategies, they did not feel more satisfied with them. Participants in the Control condition were significantly more satisfied than participants in the CSE condition in both the agent understanding task (Control: M=3.77, CSE: M=3.20; U=249, p=0.004, r=0.4071) and the agent comparison task (Control: M=3.75, CSE: M=3.14; U=267, p=0.008, r=0.3643). Contrary to our expectations, the participants in the GANterfactual-RL condition were not more satisfied than the participants in the Control condition or the CSE condition in both the agent understanding task (Control vs. GANterfactual-RL: p=0.996, CSE vs GANterfactual-RL: p=0.546) or the agent comparison task (Control vs. GANterfactual-RL: p=0.967, CSE vs GANterfactual-RL: p=0.334).

6 DISCUSSION

6.1 Computational Evaluation

Our computational evaluation shows that our proposed approach is correctly changing the agents' actions in 46% to 70% of the cases depending on the agent. While this is not perfect, one has to consider that this is not a binary task but that the agents have 5 or 6 different actions. Furthermore, CSE [32], the only previous method that focuses on generating counterfactual explanations for RL agents, only successfully changed the agent's decision in 17% to 28% of the cases. We can think of two reasons for the low validity values for the CSE approach. First, they only incorporate the agent's action in their

[2] https://aspredicted.org/m9fi5.pdf
[3] M is the mean and r is rank biserial correlation.

loss functions related to the latent space (where their discriminator and WAE were trained). The generation of the final pixels did not include constraints to faithfully ensure that a specific action was taken by the agent. Second, their loss functions for the latent space focus on creating action-invariant states. Olson et al. [32] showed that their CSE approach was useful for differentiating between a normal agent and a flawed agent. We think this is due to the fact that CSE is good at generating action-invariant states. This can help to identify the object that the flawed agent did not see since irrelevant objects are not changed for action-invariant states. We found that our approach also does not change the irrelevant object for the flawed agent (illustrated in Figure 3). This demonstrates that the counterfactuals generated by our approach are similarly effective for identifying the flawed agent. Looking at the distance between the original and the counterfactual states in pixel-space, we see that counterfactual states generated by our GANterfactual-RL approach on average have less distance to the original states and change fewer pixel values compared to the counterfactuals generated by the previous CSE method by Olson et al. [32]. This indicates that our GANterfactual-RL method is better at achieving the goal of finding the smallest possible modification of the original state to change the agent's decision. Since our method only requires a single forward pass to generate a counterfactual state, it is faster than the CSE method, which relies on potentially time-consuming gradient descent for the counterfactual generation.

6.2 User Study

Our user study showed that counterfactual explanations help users to understand which strategies different agents pursue. In particular, our method was significantly more useful than both the CSE method and not providing counterfactuals. Contrary to our hypothesis, even the counterfactuals generated by the CSE method resulted in a better understanding of the agents than not providing counterfactual explanations. This demonstrates the usefulness of counterfactual explanations for RL agents even in more complex tasks than identifying defective agents. Two recent studies evaluated the usefulness of other explanation techniques for understanding the strategies of RL agents in a similar way to our study. Huber et al. [19] looked at saliency map explanations and found that they did not help more than showing HIGHLIGHTS states without saliency maps. Their participants achieved 37% of the maximum possible score in their agent understanding task, while the participants with our counterfactual explanations obtained 50%. Septon et al. [36] investigated so-called reward decomposition explanations and found that they helped participants to achieve 60% of the maximum score in their agent understanding task. However, reward decomposition is an intrinsic explanation method which the agent and the reward function have to be specifically designed for. Our counterfactual explanations resulted in only 10% less average score even though they are post-hoc explanations that can be generated for already trained black-box agents.

Our agent comparison task showed that the increased understanding of the agent's strategies through both counterfactual explanation methods did not help participants choose fitting agents for specific tasks. For choosing the correct agent for a given problem, it is not enough to identify the strategies of the agents. It also requires enough expertise in the environment (e.g., Pacman) to judge which strategy is better suited for the problem at hand. For example, in Pacman, humans often assume that an agent that survives longer will accumulate more points in the long run. However, this is not necessarily the case since an aggressive agent can better exploit the very high rewards of eating blue ghosts. Our results for this task are in line with the results of the agent comparison task for saliency maps by Huber et al. [19].

Finally, our study showed that participants subjectively were not satisfied with the counterfactual explanations even though they objectively helped them to understand the agents. This might be due to the additional cognitive load of interpreting the explanations. The two aforementioned studies [19, 36] also did not find a significant difference in user satisfaction for their local explanation techniques. Only the choice of states, which does not provide additional information, influenced the satisfaction in [19]. However, our study is the first to see significantly higher satisfaction for the no-explanation condition than one of the two explanation conditions. This indicates that counterfactuals are subjectively less satisfying than saliency maps or reward decomposition. One possible explanation for this is the visual quality of the counterfactuals. Some participants from both counterfactual conditions commented that the counterfactuals had too many artifacts. One participant from the GANterfactual-RL condition for example wrote that *"the counterfactuals were somewhat helpful, but they would have worked better if there were fewer or no artifacts"*. Another possible reason for the low satisfaction is the presentation of the explanation. Because our study primarily aimed at investigating the benefits and drawbacks of our specific counterfactual approach, we did not use a user-friendly explanatory system where different types of explanations are provided according to the requests of the explainee.

7 CONCLUSION AND FUTURE WORK

In this work, we formulated a novel method for generating counterfactual explanations for RL agents. This GANterfactual-RL method is fully model-agnostic, which we demonstrate by applying it to three RL algorithms, two actor-critic methods, and one deep Q-learning method. Using computational metrics, we show that our proposed method is better at correctly changing the agent's decision while modifying less of the original input and taking less time than the only previous method that focuses on generating visual counterfactuals for RL. Furthermore, it significantly improved users' understanding of the strategies of different agents in a user study.

Our user study also identified two remaining deficiencies of counterfactual explanations. First, participants were subjectively not satisfied with the explanations, which might be due to unnatural artifacts in some counterfactuals. Second, the counterfactuals did not help them to calibrate their trust in the agents. Future work should try to improve counterfactual explanations in these directions.

While there is still room for improvement, we can confidently say that our approach can be considered the current state of the art for counterfactual explanations for RL agents with visual input.

ACKNOWLEDGMENTS

This paper was partially funded by the DFG through the Leibniz award of Elisabeth André (AN 559/10-1).

REFERENCES

[1] Dan Amir and Ofra Amir. 2018. HIGHLIGHTS: Summarizing Agent Behavior to People. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 1168–1176.

[2] Akanksha Atrey, Kaleigh Clary, and David D. Jensen. 2020. Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. https://openreview.net/forum?id=rkl3m1BFDB

[3] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *J. Artif. Intell. Res.* 47 (2013), 253–279. https://doi.org/10.1613/jair.3912

[4] Ruth M. J. Byrne. 2019. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 6276–6282. https://doi.org/10.24963/ijcai.2019/876

[5] Ziheng Chen, Fabrizio Silvestri, Gabriele Tolomei, He Zhu, Jia Wang, and Hongshik Ahn. 2021. ReLACE: Reinforcement Learning Agent for Counterfactual Explanations of Arbitrary Predictive Models. *CoRR* abs/2110.11960 (2021).

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8789–8797.

[7] Youri Coppens, Kyriakos Efthymiadis, Tom Lenaerts, Ann Nowé, Tim Miller, Rosina Weber, and Daniele Magazzeni. 2019. Distilling deep reinforcement learning policies in soft decision trees. In *Proceedings of the IJCAI 2019 workshop on explainable artificial intelligence*. 1–6.

[8] Mohamad H. Danesh, Anurag Koul, Alan Fern, and Saeed Khorram. 2021. Re-understanding Finite-State Representations of Recurrent Policy Networks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. 2388–2397. http://proceedings.mlr.press/v139/danesh21a.html

[9] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. 2017. OpenAI Baselines. https://github.com/openai/baselines.

[10] Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. 2022. Goal Misgeneralization in Deep Reinforcement Learning. In *International Conference on Machine Learning*. PMLR, 12004–12019.

[11] Tingxiang Fan, Pinxin Long, Wenxi Liu, and Jia Pan. 2020. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robotics Res.* 39, 7 (2020). https://doi.org/10.1177/0278364920916531

[12] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual Visual Explanations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2376–2384. http://proceedings.mlr.press/v97/goyal19a.html

[13] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz Rodríguez. 2021. Explainability in deep reinforcement learning. *Knowl. Based Syst.* 214 (2021), 106685.

[14] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608* (2018).

[15] Sandy H. Huang, Kush Bhatia, Pieter Abbeel, and Anca D. Dragan. 2018. Establishing Appropriate Trust via Critical States. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3929–3936. https://doi.org/10.1109/IROS.2018.8593649

[16] Tobias Huber, Maximilian Demmler, Silvan Mertes, Matthew L. Olson, and Elisabeth André. 2023. GANterfactual-RL: Understanding Reinforcement Learning Agents' Strategies through Visual Counterfactual Explanations (Appendix). *CoRR* abs/2302.12689 (2023). https://doi.org/10.48550/arXiv.2302.12689 arXiv:2302.12689

[17] Tobias Huber, Benedikt Limmer, and Elisabeth André. 2022. Benchmarking Perturbation-Based Saliency Maps for Explaining Atari Agents. *Frontiers in Artificial Intelligence* 5 (2022). https://doi.org/10.3389/frai.2022.903875

[18] Tobias Huber, Dominik Schiller, and Elisabeth André. 2019. Enhancing Explainability of Deep Reinforcement Learning Through Selective Layer-Wise Relevance Propagation. In *KI 2019: Advances in Artificial Intelligence*, Christoph Benzmüller and Heiner Stuckenschmidt (Eds.). Springer International Publishing, Cham, 188–202.

[19] Tobias Huber, Katharina Weitz, Elisabeth André, and Ofra Amir. 2021. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artif. Intell.* 301 (2021), 103571. https://doi.org/10.1016/j.artint.2021.103571

[20] Mark T. Keane, Eoin M. Kenny, Eoin Delaney, and Barry Smyth. 2021. If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. ijcai.org, 4466–4474. https://doi.org/10.24963/ijcai.2021/609

[21] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. 2022. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems* 23, 6 (2022), 4909–4926. https://doi.org/10.1109/TITS.2021.3054625

[22] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. 2021. Explaining in Style: Training a GAN to explain a classifier in StyleSpace. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 673–682. https://doi.org/10.1109/ICCV48922.2021.00073

[23] Arnaud Van Looveren and Janis Klaise. 2021. Interpretable Counterfactual Explanations Guided by Prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12976)*, Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and José Antonio Lozano (Eds.). Springer, 650–665. https://doi.org/10.1007/978-3-030-86520-7_40

[24] Teppei Matsui, Masato Taki, Trung Quang Pham, Junichi Chikazoe, and Koji Jimura. 2022. Counterfactual explanation of brain activity classifiers using image-to-image transfer by generative adversarial network. *Frontiers in Neuroinformatics* 15 (2022), 79.

[25] Silvan Mertes, Tobias Huber, Katharina Weitz, Alexander Heimerl, and Elisabeth André. 2022. GANterfactual - Counterfactual Explanations for Medical Non-experts Using Generative Adversarial Learning. *Frontiers Artif. Intell.* 5 (2022), 825565. https://doi.org/10.3389/frai.2022.825565

[26] Silvan Mertes, Christina Karle, Tobias Huber, Katharina Weitz, Ruben Schlagowski, and Elisabeth André. 2022. Alterfactual Explanations - The Relevance of Irrelevance for Explaining AI Systems. *CoRR* abs/2207.09374 (2022). https://doi.org/10.48550/arXiv.2207.09374 arXiv:2207.09374

[27] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* 267 (2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[28] Tim Miller. 2022. Are we measuring trust correctly in explainability, interpretability, and transparency research? *CoRR* abs/2209.00651 (2022). https://doi.org/10.48550/arXiv.2209.00651

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[30] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. ACM, 607–617. https://doi.org/10.1145/3351095.3372850

[31] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. 2022. CounteRGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands (Proceedings of Machine Learning Research, Vol. 180)*, James Cussens and Kun Zhang (Eds.). PMLR, 1488–1497. https://proceedings.mlr.press/v180/nemirovsky22a.html

[32] Matthew L. Olson, Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong. 2021. Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artif. Intell.* 295 (2021), 103455. https://doi.org/10.1016/j.artint.2021.103455

[33] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. 2021. CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

[34] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. 2020. Explain Your Move: Understanding Agent Actions Using Specific and Relevant Feature Attribution. In *8th International Conference on Learning Representations, ICLR*. OpenReview.net.

[35] Kathryn Schutte, Olivier Moindrot, Paul Hérent, Jean-Baptiste Schiratti, and Simon Jégou. 2021. Using StyleGAN for Visual Interpretability of Deep Learning Models on Medical Images. *CoRR* abs/2101.07563 (2021).

[36] Yael Septon, Tobias Huber, Elisabeth André, and Ofra Amir. 2022. Integrating Policy Summaries with Reward Decomposition for Explaining Reinforcement Learning Agents. *arXiv preprint arXiv:2210.11825* (2022).

[37] Andrew Silva, Mariah Schrum, Erin Hedlund-Botti, Nakul Gopalan, and Matthew Gombolay. 2022. Explainable Artificial Intelligence: Evaluating the Objective and Subjective Impacts of xAI on Human-Agent Interaction. *International Journal of*

Human–Computer Interaction (2022), 1–15.

[38] Sumedha Singla, Motahhare Eslami, Brian Pollack, Stephen Wallace, and Kayhan Batmanghelich. 2023. Explaining the black-box smoothly - A counterfactual approach. *Medical Image Anal.* 84 (2023), 102721.

[39] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

[40] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224* (2016).

[41] Lindsay Wells and Tomasz Bednarz. 2021. Explainable AI and Reinforcement Learning - A Systematic Review of Current Approaches and Trends. *Frontiers Artif. Intell.* 4 (2021), 550030. https://doi.org/10.3389/frai.2021.550030

[42] Adam White, Kwun Ho Ngan, James Phelan, Saman Sadeghi Afgeh, Kevin Ryan, Constantino Carlos Reyes-Aldasoro, and Artur d'Avila Garcez. 2021. Contrastive Counterfactual Visual Explanations With Overdetermination. *CoRR* abs/2106.14556 (2021). arXiv:2106.14556

[43] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. 2021. Reinforcement Learning in Healthcare: A Survey. 55, 1, Article 5 (2021), 36 pages. https://doi.org/10.1145/3477600

[44] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding DQNs. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 1899–1908. http://proceedings.mlr.press/v48/zahavy16.html

[45] Wenqi Zhao, Satoshi Oyama, and Masahito Kurihara. 2020. Generating Natural Counterfactual Visual Explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 5204–5205. https://doi.org/10.24963/ijcai.2020/742

[46] Yunxia Zhao. 2020. Fast Real-time Counterfactual Explanations. *CoRR* abs/2007.05684 (2020). arXiv:2007.05684