# Why the Agent Made that Decision: Contrastive Explanation Learning for Reinforcement Learning

**Rui Zuo**[1] , **Simon Khan**[2] , **Zifan Wang**[1] , **Garrett Ethan Katz**[1] and **Qinru Qiu**[1]
[1]Syracuse University
[2]Air Force Research Laboratory
{rzuo02, zwang345, gkatz01, qiqiu}@syr.com, simon.khan@us.af.mil

## Abstract

Reinforcement learning (RL) has demonstrated remarkable success in solving complex decision-making problems, yet its adoption in critical domains is hindered by the lack of interpretability in its decision-making processes. Existing explainable AI (xAI) approaches often fail to provide meaningful explanations for RL agents, particularly because they overlook the contrastive nature of human reasoning—answering "why this action instead of that one?" To address this gap, we propose a novel framework of contrastive learning to explain RL selected actions, named **VisionMask**. VisionMask is trained to generate explanations by explicitly contrasting the agent's chosen action with alternative actions in a given state using a self-supervised manner. We demonstrate the efficacy of our method through experiments across diverse RL environments, evaluating it in terms of faithfulness, robustness and complexity. Our results show that VisionMask significantly improves human understanding of agent behavior while maintaining accuracy and fidelity. Furthermore, we present examples illustrating how VisionMask can be used for counterfactual analysis. This work bridges the gap between RL and xAI, paving the way for safer and more interpretable RL systems.

## 1 Introduction

Deep Reinforcement Learning (DRL) is a powerful technology in machine intelligence, widely used for many applications [Sutton *et al.*, 1999]. However, understanding a DRL agent's decision-making process is challenging, due to the inherent lack of explainability in the high-dimensional, non-linear structure of its underlying Deep Neural Network (DNN) [Heuillet *et al.*, 2021; Hickling *et al.*, 2023]. The lack of transparency undermines users' trust, driving the development of Explainable AI (xAI).

Various methods in computer vision have been proposed to enhance the transparency of AI systems [Ribeiro *et al.*, 2016; Selvaraju *et al.*, 2017; Bach *et al.*, 2015; Shrikumar *et al.*, 2017; Lundberg and Lee, 2017]. At the core, they share a common foundation: **attributing** the classifier's outputs to more interpretable features and using a saliency map to visualize these attributions. Their only differences are how these attributions are calculated. A high-quality attribution-based explanation should meet several key criteria. First, it should demonstrate *faithfulness*, meaning that including features with high attribution should lead the model to the target output, and excluding them should prevent it. Second, it should exhibit *specificity*, ensuring that only critical features receive high attribution. Sometime this is also referred to as *sparseness*. Finally, it should be *robust*, meaning the explanation should remain consistent and not change significantly with minor variations in the input.

Attribution-based explanation has been studied for DRL models. [Greydanus *et al.*, 2018] and [Puri *et al.*, 2020] utilized policy distributional shifts as the basis for attribution in RL. Specifically, they calculate attribution of a feature as the difference in Q/V values or action distributions between the original and perturbed states. For example, given agent policy $\pi$, the attribution of a feature is proportional to $E_{s'}(|\pi(s) - \pi(s')|_2)$ where $s$ stands for the original state and $s'$ represents perturbed states generated for this feature. By calculating the attributions for all features, a saliency map $m$ can be created. However, the perturbation-based explanations lack faithfulness. Since each perturbation focuses only on local features while ignoring the joint impact of feature combinations, overlaying the saliency map with the original state, $(m \odot s)$, does not result in a feature combination that leads the agent to the target action distribution $\pi(s)$.

A better approach to enhance faithfulness is to **learn** a model to predict the saliency map $m$ that minimizes the difference between $\pi(m \odot s)$ and $\pi(s)$. *Explainer* [Stalder *et al.*, 2022] leveraged this idea by training an explanation model for an image classifier. However, Explainer categorizes class labels into target and non-target for each training sample and focus on learning saliency map (or mask) only for the target label while treat all non-target labels as a single group. Unlike a (well trained) image classifier, where predictions for non-target labels are typically close to 0, DRL agent in many scenarios, does not exhibit a clear preference for the actions. Non-target actions may sometimes have probabilities only slightly lower than those of target actions. Analyzing how masking the feature may affect the non-target action probability provides additional information that can be used to train the explanation model more effectively.

The above analysis motivates us to design a trainable saliency map generator for attribution-based explanations and train it using two channels of contrastive information:**(i) Action-wise contrast:** We believe that environment states contain features that motivate the DRL agent to select both target action and non-target actions. However, the target action is ultimately chosen because it corresponds to higher reward or has a stronger presence. For each action, a saliency map can be generated as an explanation. Choosing features according to the saliency map for a non-target action should push the agent away from the target action, and vice versa. This inspired us to treat the saliency map of the non-target action $m_{a'}$ and the target action $m_a$ as a negative pair, which can be leveraged for contrastive learning [Chopra *et al.*, 2005; Schroff *et al.*, 2015; Gutmann and Hyvärinen, 2010]. **(ii) Feature-wise contrast:** To exclude irrelevant features (e.g. background) from the saliency map, explanations also need to be discriminative in filtering out such information. When only irrelevant features are accessible to the agent, the resulting action distribution should be as uniform as possible. Therefore, the target action's saliency map ($m$) and its inverted counterpart ($\tilde{m} = 1 - m$) form another negative pair for contrastive learning.

In this work, we present VisionMask as an RL explainer that is contrastively trained to generate saliency maps to explain agent's actions. We specifically focus on agents that maps images to actions and consider each pixel value as the interpretable input feature, although the similar technique could be extended to other type of features. We carefully design the objective function to enable self-supervised contrastive learning of explanations from both action-wise and feature-wise perspectives, fostering the generation of more faithful explanations. We conduct evaluation on six RL environments with five baselines based on faithfulness, robustness, and sparseness. Quantitatively, VisionMask outperforms the baselines in terms of faithfulness, while exhibiting strong robustness and high sparseness. Qualitatively, we compared VisionMask with the baselines in two settings: visual comparison and human studies. In the visual comparison, VisionMask provides sharper explanations that align more closely with human interpretations, as demonstrated by counterfactual examples. In the human studies, VisionMask's explanations help users better understand the agent's decisions and calibrate appropriate trust.

The contribution of our work can be summarized as the following:

- We present VisionMask, a novel attribution method for explainable visual reinforcement learning that generates action-specific saliency maps. To the best of our knowledge, this is the first work that learns saliency map in a self-supervised contrastive learning manner, yielding highly faithful explanations across various RL environments.
- VisionMask is model-agnostic in that it works with any vision-based DRL agent and requires only the agent's input state and output action during inference time. VisionMask does not make any modification on the agent, however, each VisionMask model is trained specifically for a given agent by including the fixed agent in the back-propagation path for gradient descent.
- We conducted extensive experiments, both qualitatively and quantitatively, to demonstrate the improvements of the saliency map in terms of faithfulness, robustness, and sparseness, as well as a comprehensive ablation study for each component.

## 2 Background and Related Work

The nested structure and non-linear operation of DNN make it challenging for humans to understand how the outputs are derived from inputs. Some existing works address this challenge by explaining model outputs through input attributions [Petsiuk *et al.*, 2018; Ribeiro *et al.*, 2016; Fong and Vedaldi, 2017]. For example, Randomized Input Sampling for Explanation of Black-box Models (RISE) [Petsiuk *et al.*, 2018] is a perturbation-based approach that explains neural image classifiers by applying randomly generated masks to the input image and assessing their impact. Local Interpretable Model-agnostic Explanations (LIME)[Ribeiro *et al.*, 2016] tries to explain local instances by approximating them within a nearby vicinity using a linear decision model, where the explainable elements are superpixels (i.e., small image regions of similar pixels).

Built on top of the DNNs, the lack of transparency of the DRL agents undermines humans trust and hinders their adoption. Several studies have addressed this challenging problems. According to [Qing *et al.*, 2022], existing efforts can be categorized into four main approaches: model explaining, reward explaining, task explaining, and state explaining.

Model explaining relies on inherently interpretable model architectures or auxiliary reasoning mechanisms to generate explanations [Topin *et al.*, 2021]. However, these self-explanatory or symbolic models often suffer from decreased performance compared to state-of-the-art neural network based RL policies and may lack the representational power needed to learn more complex policies. Reward explanation typically involves using explicitly designed explainable reward functions to generate explanations [Ashwood *et al.*, 2022]. Task explanation considers a policy as a composition of sub-task and action sequences, explaining the behavior in terms of relationships among sub-tasks [Shu *et al.*, 2017]. These approaches often assume the existence of a reward decomposition scheme or a predefined sub-task partition mechanism, which may not hold true in all RL environments.

State explanations are based on agent observations from the environment. This approach determines the significance of *explainable elements* within a state observation in relation to action selection or reward determination. An explainable element in the state could be a small region in the visual input or semantic features in the environment state. The proposed *VisionMask* falls into this category.

State explaining methods can further be divided into three categories: attention-based, Shapley value-based and perturbation-based mechanisms, which are detailed below. Attention is a common approach used for explainable RL in several existing works [Annasamy and Sycara, 2019]. However, similar to i-DQN [Annasamy and Sycara, 2019], these

methods cannot explain a given DRL model because their attention model must be trained alongside the agent model. Shapley value [Shapley and others, 1953], a concept from game theory, has also been introduced into XRL. SVERL [Beechey *et al.*, 2023] provides a theoretical analysis on how Shapley values can be applied to explain value and policy network in DRL. However, this approach is still in its early stages and assumes prior knowledge of the transition model of the environment, which is not realistic for realistic applications.

Perturbation-based methods [Greydanus *et al.*, 2018; Iyer *et al.*, 2018; Puri *et al.*, 2020] compute saliency maps of the input features by comparing the action probability or value function before and after perturbation. [Greydanus *et al.*, 2018] perturbed the state with Gaussian blur at fixed intervals, while [Iyer *et al.*, 2018] identified objects and perturbed them with a constant gray value. [Puri *et al.*, 2020] improved on [Greydanus *et al.*, 2018] by removing the effect of perturbations from other irrelevant actions. As discussed in section 1, the performance of these techniques is constrained by predefined perturbation rules and the lack of knowledge accumulation. Furthermore, there is no guarantee that the perturbed input remains physically meaningful.

# 3 VisionMask

In this section, we present our VisionMask architecture. The primary goal is to generate action-wise saliency maps that attribute the most relevant features in the state to each action. For agents that map images to actions, the features and states correspond to pixels and images.

## 3.1 Problem Formulation

Formally, we define the environment as a Markov Decision Process (MDP) $\{\mathcal{S}, \mathcal{A}, P, R, \gamma\}$, where $\mathcal{S}$ represents the state space; $\mathcal{A}$ denotes the action space with $|\mathcal{A}| = K$; the state transition function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ depicts the transition between states based on actions, where $\Delta(\mathcal{S})$ represents the set of probability distributions over states; the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ provides the immediate reward for state-action pairs; $\gamma \in [0, 1]$ is discount factor and $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ represent policy. Return $G$ is defined as $G = \sum_{k=0}^{\infty} \gamma^k R_{k+1}$, and the expected cumulative reward of a policy $\pi$ is $\mathbb{E}_\pi[G] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{k+1}]$, where the expectation is taken with respect to the initial state distribution, transition probabilities, and action probabilities determined by $\pi$. VisionMask operates on a given trained expert policy $\pi_E$ such that $\pi_E \approx \pi^* = \arg\max_\pi \mathbb{E}_\pi[G]$, where $\pi^*$ is the optimal policy. We can obtain a dataset of expert demonstrations $\mathcal{D}_E = \{(s_i, a_i \sim \pi_E(s_i))\}_{i=1}^N$ consisting of $N$ state-action pairs, from trajectories sampled while executing $\pi_E$ in the environment. Our goal is to learn an explainer $f_{\theta^*}$ that minimizes the loss $\theta^* = \arg\min_\theta \sum_{(s,a)\in\mathcal{D}_E} \mathcal{L}(a, s, \theta)$ where $\mathcal{L}$ is the training loss function to be discussed in section 3.3. The explainer function $f_\theta : \mathcal{S} \to [0, 1]^{K \times d_s}$, where $d_s$ represents the feature size of the state $s \in \mathcal{S}$ and $K$ denotes the number of candidate actions, predicts the attributions of each action to each feature in the state $s$. The value of the output is bounded within the range $[0, 1]$, with the $(i, j)$th element indicating the $j$th feature's attribution to the $i$th action.

In the case of visual DRL, $s$ is the visual input of the agent and $d_s$ is the number of pixels in $s$, i.e., $d_s = W \times H$, with $W$ and $H$ representing the width and height of the visual input. The output of the explainer can be viewed as $K$ saliency maps, $M = \{m_0, m_1, ...., m_{K-1}\}$, each corresponds to an action. A saliency value $m_i[x, y]$ near 1 indicates that the pixel $(x, y)$ has significant contributions to action $i$, whereas a value near 0 denotes irrelevance. Overlaying the $i$th saliency map with the state $s$ will highlight the input features that lead the agent to the $i$th action.

## 3.2 Architecture

As shown in Figure 1, we first collect the expert dataset $\mathcal{D}_E$ using the expert policy $\pi_E$. From this dataset $\mathcal{D}_E$, state-action pairs $(s_i, a_i)$ are sampled and fed to VisionMas $f_\theta$ to generate the set of saliency maps $M$.

Generating the saliency map from given visual input is a dense prediction task that shares similarities with image segmentation, where each pixel is assigned a value to indicate whether it belongs to an object or background. Hence, we structure the explainer $f_\theta$ akin to the widely used image segmentation model, DeepLabv3 [Chen *et al.*, 2017], however, retrain it using self-supervised contrastive learning. To make sure that the output saliency value are bounded to the range $[0, 1]$, a sigmoid function is applied at the output of $f_\theta$. For each $m_i \in M$, we also calculate a complement map $\tilde{m}_i = \mathbf{1} - m_i$ highlighting the irrelevant regions for the action $i$. Then the masks $m_i$ and $\tilde{m}_i$ are overlaid onto the original state $s$ to generate two masked states $s_i$ and $\tilde{s}_i$ using the following overlay function:

$$s_i = s \odot m_i + r \odot (\tilde{m}_i), \tilde{s}_i = s \odot \tilde{m}_i + r \odot (m_i) \quad (1)$$

where $\odot$ is Hadamard Product and $r$ is a reference value. Numerous options exist for the reference value, such as setting the pixel to zero, assigning a constant value, blurring the pixel, or cropping it. Empirical study shows that setting the reference to the background gives the best results. More details can be found in the ablation study 4.4. See Appendix F for the background values for different environments.

To generate self-supervised contrastive loss to train the model, we query the agent to obtain the corresponding logits $z_i = z(s_i)$ and $\tilde{z}_i = z(\tilde{s}_i)$, where $z_i, \tilde{z}_i \in \mathbb{R}^K$ , and the action probability distributions $p_i = \text{softmax}(z_i)$ and $\tilde{p}_i = \text{softmax}(\tilde{z}_i)$, where $p, p_i \in [0, 1]^K$, $0 \le i < K$. By concatenating each $p_i$ and $\tilde{p}_i$, we have the the action probability distributions of each mask $\mathbf{p}, \tilde{\mathbf{p}} \in \mathbb{R}^{K \times K}$,

$$\mathbf{p} = [p_1, p_2, \ldots, p_k]^T \quad \tilde{\mathbf{p}} = [\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_k]^T$$

## 3.3 Training Loss

To enable the agent contrastively learn the saliency map $m_a$, we carefully designed the training loss function $\mathcal{L}$ as follows:

$$\mathcal{L}(\mathbf{s}, \mathbf{a}, \theta) = \mathcal{L}_a(\mathbf{s}, \mathbf{a}) + \lambda_{ne}\mathcal{L}_{ne}(\mathbf{s}) + \lambda_{area}\mathcal{L}_{area}(\mathbf{s}, \mathbf{a})$$

where $\mathcal{L}_a(\mathbf{s}, \mathbf{a})$ is action-wise contrastive action loss, $\mathcal{L}_{ne}(\mathbf{s}, \mathbf{a})$ is feature-wise loss, $\mathcal{L}_{area}(\mathbf{s}, \mathbf{a}, \mathbf{n})$ is the area size loss and $\lambda_{ne}, \lambda_{area}$ are regularization hyper-parameters.

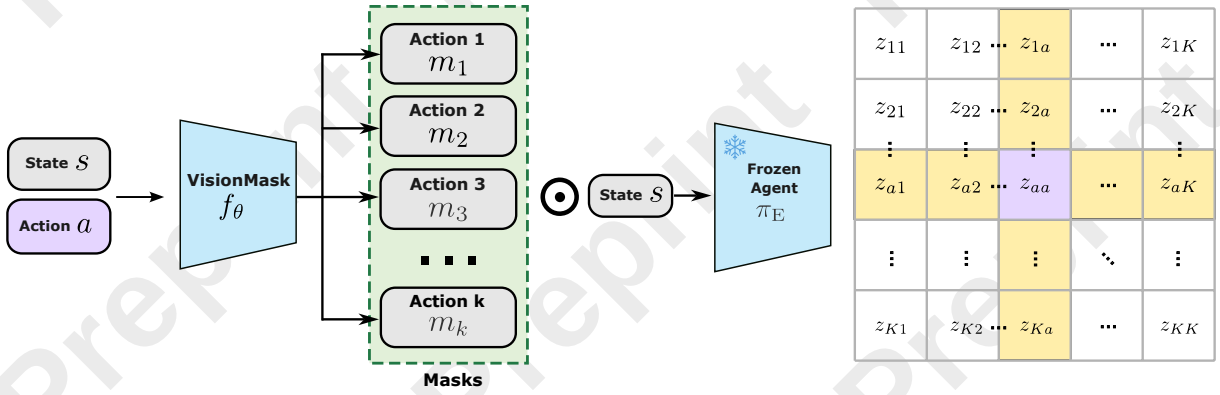


Figure 1: **Architecture of VisionMask.** State-action pairs $(s, a)$ are sampled from $\mathcal{D}_E$.

**Action-wise contrastive loss** $\mathcal{L}_a$. Let $a$ denote the target action chosen by the agent. Our primary goal is to learn explanations faithful to the $a$, making $(a, p_a)$ the only positive pair. Furthermore, the explanation must be discriminative, meaning it should clearly distinguish the target action $a$ from all other possible actions. As a result, every other pair of $(a, p_{i\neq a})$ is treated as a negative pair. Then, we compute the cross-entropy loss between these pairs,

$$\mathcal{L}_{pos}(\mathbf{s}, \mathbf{a}) = -\frac{1}{K}\sum_{k=1}^{K}[\![k = \mathbf{a}]\!]\log(\frac{\exp(z_{aa})}{\sum_{i=1}^{K}\exp(z_{ai})}) \quad (2)$$

$$\mathcal{L}_{neg}(\mathbf{s}, \mathbf{a}) = -\frac{1}{K}\sum_{k=1}^{K}[\![k = \mathbf{a}]\!]\log(\frac{\exp(z_{aa})}{\sum_{i=1}^{K}\exp(z_{ia})}) \quad (3)$$

where $z_{ij}$ is the logit for action $j$ when querying the policy with the masked state $s_i$. The contrastive action loss $\mathcal{L}_a = \mathcal{L}_{pos} + \mathcal{L}_{neg}$. Here $[\![k = \mathbf{a}]\!]$ denotes the indicator function which returns 1 if $k$ is the same as label $\mathbf{a}$, and 0 otherwise. Note that we do not compute the loss with $\mathbf{p}_{k,i\neq a}$, as ensuring the faithfulness of the target action $a$ is our primary objective here.

**Feature-wise loss** $\mathcal{L}_{ne}$. To ensure that the visual input regions selected by $m_i$ is necessary for the agent to make decisions, we also need to make sure that the unselected region, i.e., $s_{\tilde{m}_i}$, does not provide useful information for action selection, hence the action distribution $\tilde{\mathbf{p}}$ should follow a uniform distribution. Motivated by this rationale, we define negative entropy loss regarding $\tilde{\mathbf{m}}$ as the following:

$$\mathcal{L}_{ne}(\mathbf{s}) = \frac{1}{K^2}\sum_{ij}\tilde{\mathbf{p}}_{ij}\log\tilde{\mathbf{p}}_{ij}. \quad (4)$$

**Area size loss** $\mathcal{L}_{area}$. A low effort way to minimize $\mathcal{L}_a$ and $\mathcal{L}_{ne}$ is to include all pixels in the mask, $m_i$, and no pixel in the complement mask, $\tilde{m}_i$, which obviously is not a valid solution. We need to ensure that each importance mask only consists a small number of crucial pixels. Thus, we define $\mathcal{L}_{area}$ using L1 norm as follows:

$$\mathcal{L}_{area}(\mathbf{s}) = \frac{1}{K}\sum_{k}(|\frac{1}{Z}\sum_{i,j}m_k[i,j] - a_{max}|) \quad (5)$$

where $Z$ it the number of pixels in state.

## 4 Experiments

In this section, we begin by outlining the experimental setup. We then present quantitative and qualitative analyses to evaluate our approach. Additionally, we provide counterfactual explanations to demo VisionMask's faithfulness and sensitivity. Finally, we perform an ablation study to assess the contribution of each component.

### 4.1 Experimental Setup

**Environment Selection**. We conduct experiments across three types of environments: *Super Mario Bros (SMB)* [Kauten, 2018], *Enduro*, *Seaquest* and *MsPacman* [Bellemare *et al.*, 2013; Machado *et al.*, 2018] for 2D game, *Highway-env* [Leurent, 2018] for autonomous driving simulation and *VizDoom* [Wydmuch *et al.*, 2018] for 3D game.

**Baseline Selection**. We mainly compare our model with perturbation-based baselines for black-box RL such as Greydanus [Greydanus *et al.*, 2018] and *SARFA* [Puri *et al.*, 2020]. In addition, we also compared with three techniques originally designed to explain image classifiers, including a learning-based method, *Explainer* [Stalder *et al.*, 2022], and two perturbation-based methods, LIME [Ribeiro *et al.*, 2016] and RISE [Petsiuk *et al.*, 2018]. Although these methods focus on image classification, their main concept is similar to ours: to attribute a given label (action) to a subset of visual features. We use the public implementation from torchray [Fong *et al.*, 2019] for RISE and the original published implementations for other baselines. Among all the baselines, the Explainer is the most similar to VisionMask, as both are learning-based approaches. However, VisionMask employs action-contrastive learning and is trained using different regularization. In the experimental results, we demonstrate that these differences significantly enhances VisionMask's performance.

**Expert Policy**. We use the open-source pre-trained PPO [Schulman *et al.*, 2017] agents from [Nguyen, 2020] for *SMB* environment and from stable-baselines3 [Raffin *et al.*, 2021] for *Enduro*, *Seaquest* and *MsPacman* environments. We train the DQN [Mnih *et al.*, 2013] agents from scratch for the *Viz-*

| | *SMB* | | | | | *Enduro* | | | | | *Seaquest* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ |
| LIME | 78.7 | 27.0 | 30.2 | 69.4 | **98.3** | 88.1 | 36.3 | 38.0 | 47.9 | **90.2** | 90.1 | 14.3 | 21.4 | 17.3 | **97.7** |
| RISE | 80.5 | **18.9** | 38.0 | **1.9** | 2.0 | 90.3 | 37.6 | 39.1 | **0.01** | 0.5 | 92.0 | 8.1 | 29.0 | **0.02** | 1.4 |
| Greydanus | 16.9 | 56.4 | 20.2 | 20.8 | 63.3 | 27.4 | 34.6 | 33.3 | 0.25 | 82.9 | 44.6 | 14.4 | 7.88 | 0.12 | 75.1 |
| SARFA | 16.9 | 55.6 | 21.4 | 68.7 | 65.5 | 27.9 | 33.3 | 33.4 | 0.26 | 77.0 | 44.8 | 7.6 | 8.7 | 0.13 | 75.2 |
| *Explainer* | 92.2 | 23.6 | 49.0 | 38.1 | 81.2 | 90.2 | 34.2 | 33.1 | 0.22 | 81.2 | 95.3 | 13.4 | 30.0 | 0.9 | 97.5 |
| VisionMask | **95.9** | 20.4 | **67.6** | 38.0 | 82.3 | **98.7** | **32.9** | **41.2** | 0.5 | 80.0 | **99.6** | **6.4** | **34.3** | 0.9 | 97.6 |
| | *MsPacman* | | | | | *VizDoom* | | | | | *Highway* | | | | |
| Method | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ | Acc. | Del. ↓ | Ins. ↑ | LLE ↓ | Sp. ↑ |
| LIME | 84.5 | 32.7 | 37.7 | 9.1 | **96.6** | 72.1 | 35.1 | 11.2 | 22.1 | **96.4** | 77.8 | 20.0 | 23.64 | 52.7 | 80.5 |
| RISE | 92.6 | 26.2 | 45.7 | **0.01** | 0.59 | 83.8 | 14.6 | 14.5 | **0.02** | 0.5 | 82.5 | 20.0 | 22.1 | **0.05** | 0.6 |
| Greydanus | 46.1 | 39.1 | 14.2 | 0.17 | 57.4 | 75.2 | 20.7 | 17.2 | 0.12 | 62.1 | 92.7 | 20.9 | 24.7 | 0.93 | 83.8 |
| SARFA | 42.4 | 18.8 | 15.6 | 0.13 | 66.5 | 76.1 | 16.7 | 17.4 | 0.55 | 66.6 | 97.6 | 20.4 | 25.6 | 0.43 | 83.0 |
| *Explainer* | 97.6 | 19.6 | 59.2 | 0.22 | 81.2 | 84.2 | 14.4 | 17.6 | 0.2 | 65.5 | 95.0 | 20.37 | 24.4 | 1.5 | 83.9 |
| VisionMask | **98.7** | **17.0** | **62.8** | 0.2 | 89.5 | **87.8** | **14.2** | **18.1** | 3.89 | 47.8 | **98.1** | **19.8** | **25.8** | 1.14 | **84.8** |

Table 1: **Quantitative results** on *SMB*, *Enduro*, *Seaquest*, *MsPacman*, *VizDoom* and *Highway* of VisionMask against 5 baselines. Five metrics are compared. The faithfulness is measured by Accuracy(Acc.), Deletion(Del.) and Insertion(Ins.) metrics(%); Robustness is measured by Local Lipschitz Estimate (LLE)(%); And Complexity is measured by Sparseness(Sp.)(%). Blue represents second best results.

*Doom* and *Highway-env* environments. See Appendix A for agents details.

**Dataset.** We collect state-action pairs for each environment and split the data into 80% for training, 10% for validation, and 10% for testing. All results in this section are reported using the test split. We make this dataset publicly available; see Appendix B for details.

## 4.2 Quantitative Analysis

**Metrics.** Since there is no ground truth explanations [Adebayo *et al.*, 2020], it is crucial to select appropriate metrics to evaluate the trustworthiness of explanations. Designing robust and consistent metrics for XAI remains an unresolved challenge [Hedström *et al.*, 2023]. To alleviate the inconsistency between metrics and avoid selection bias, we closely follow the LATEC [Klein *et al.*, 2024] benchmark to evaluate performance across three dimensions: *Faithfulness*, *Robustness*, and *Complexity*.

Faithfulness measures the correlation between the agent's action and the masked visual input. In this context, we consider three metrics: Accuracy, Deletion and Insertion performance [Petsiuk *et al.*, 2018]. The accuracy gives the percentage of time that masked input and the original input lead to the same action according to the expert agent. It is defined as the following: Accuracy $= \frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{i=1}^{|\mathcal{D}_{\text{test}}|}[\![\pi_E(s_i) = \pi_E(s_i')]\!]$, where $s_i' = s_i \odot m_i + r \odot \tilde{m}_i$, and $[\![\cdot]\!]$ again denotes the indicator function.

Insertion and deletion measure the impact on the target action probability by progressively adding or removing pixels from the original input based on the descending order of their attribution scores, with the most important pixels being inserted/deleted first. The inserted/deleted input image will be processed by the expert agent again to obtain the probability of the target action. By plotting the probability against the number of pixels added or removed, we obtain an insertion and a deletion curve. The insertion/deletion performance is measured by the area under the curve (AUC) of the insertion/deletion curves. A larger (smaller) AUC for the insertion (deletion) curve means that including (removing) the important pixels identified by the explainer can effectively increase (reduce) the probability of the target action. Hence a larger (smaller) AUC of insertion (deletion) curve indicates more accurate attribution prediction. The pseudo-code of the detailed information of deletion and insertion could be found in Appedix E.

For robustness, we report the Local Lipschitz Estimate (LLE) scores [Alvarez Melis and Jaakkola, 2018], which quantify the local smoothness of explanations by estimating the Lipschitz constant within a specific neighborhood. The Lipschitz constant measures the maximum rate of change of the function, ensuring explanation does not vary too rapidly within the state's neighborhood. Given state $s_i$ and neighborhood size $\epsilon$, LLE defined as

$$\hat{L}(s_i) = \underset{s_j \in \mathcal{N}_\epsilon(s_i)}{\text{argmax}} \frac{\|f_\theta(s_i) - f_\theta(s_j)\|_2}{\|s_i - s_j\|_2}$$

where $\mathcal{N}_\epsilon(s) = \{s' \in X \mid \|s - s'\| \leq \epsilon\}$.

We evaluate complexity with Sparseness [Chalasani *et al.*, 2020], which uses the Gini index on the vector of absolute attribution values sorted in non-descending order. Sparseness ensures that features genuinely influencing the output have substantial contributions, while insignificant or only slightly relevant features should have minimal contributions. A higher Sparseness indicates more contrastive attribution values and hence more understandable explanations [Chalasani *et al.*, 2020].

**Results.** In Table 1, our VisionMask achieves the best performance in terms of faithfulness (i.e., Acc, Del, Ins) in all testing environment except *SMB*, where its deletion score is slightly lower than RISE. However, RISE has significantly lower insertion score and accuracy in this environment. Hence, our explanations are more aligned with the agent's decision-making process. Moreover, compared to, *Explainer* [Stalder *et al.*, 2022], VisionMask exhibits much higher faithfulness, which suggests the effectiveness of action contrastive learning. Overall, learning-based model performs

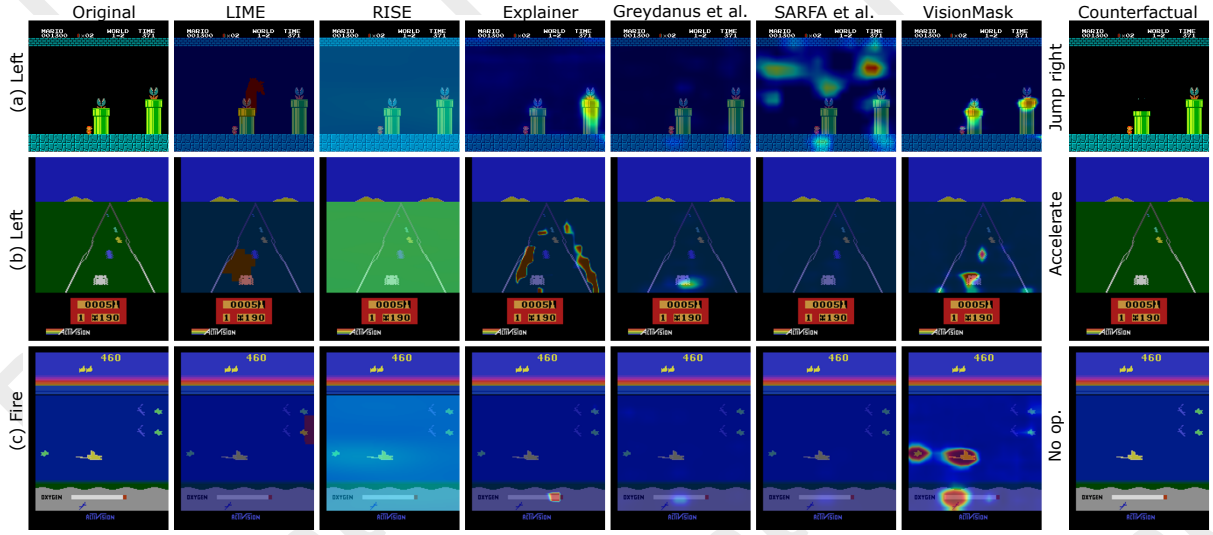


Figure 2: **Qualitative examples** of VisionMask and five baselines across three environments. (a)–(c) show the saliency map overlaid on the input image and counterfactuals where regions are removed based on VisionMask's map. (a) Human: "Mario moves left to avoid the Piranha Plant." VisionMask correctly highlights the Plant; removing it changes the action from 'Left' to 'Jump right'. (b) Human: "The agent moves left at constant speed due to the front car." VisionMask identifies both; removing the car allows acceleration instead of 'Right + accelerate'. (c) Human: "The agent fires because a shark follows." VisionMask detects the shark and oxygen bar; removing the shark stops firing. Additional examples in Appendix H.

better compared to perturbation based approach.

In terms of robustness, RISE achieves the best performance in terms of LLE score across all settings. This is because, as a perturbation-based method, RISE uses the same type of perturbation that we used to generate the neighbor image for LLE score calculation. Hence the perturbation almost has no impact to it. On the other hand, RISE has the lowest sparsity, which means the attribution predicted by RISE are evenly distributed and contains little information. For Complexity, benefiting from the binary mask of LIME which is much more sparse compared to other baselines, LIME has the highest sparseness score. However, it has the worst performance in terms of LLE score. Overall, VisionMask achieves the best balance between the robustness and sparseness. See Appendix for the Radar map.

### 4.3 Qualitative Analysis

In this section, we conduct two qualitative analyses across two settings, visual comparision and human studies.

**Visual Comparision**. We present example explanations from three environments, *SMB*, *Enduro*, and *Seaquest*, in Figure 2, along with some counterfactual analysis generated from the explanations provided by VisionMask. The examples show that both LIME and RISE fail to generate interpretable explanations. LIME's superpixels are too large to capture the specific regions, while RISE's explanations include almost all pixels. *Explainer* generates more accurate and interpretable explanations compared to LIME and RISE. This suggests that purely perturbation-based approaches may fail in RL due to the high dynamics of environments and the lack of learning ability. [Greydanus *et al.*, 2018] and SARFA generate better explanations than LIME and RISE but often focus on irrelavent objects or background. In contrast, VisionMask accurately highlights the relevant regions, providing sharp explanations that are both accurate and interpretable. By removing some of the regions highlighted by the VisionMask, counterfactual analysis could be performed to answer questions like: "Why this action instead of that one?"

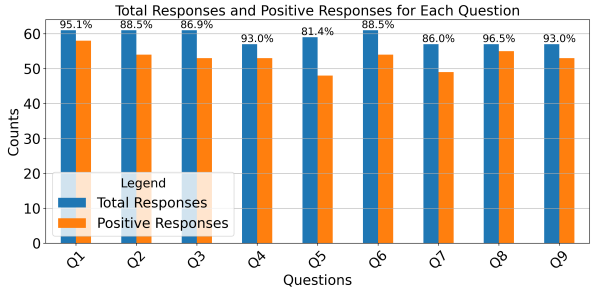


Figure 3: **Human studies**. Comparison of total responses and positive responses across questions.

**Human Studies**. To assess whether the saliency maps improve humans' understanding of the agent's decisions, we conduct human studies for *SMB*, *Enduro* and *Seaquest*. We present 63 participants with 9 state-action pairs accompanied by the saliency maps generated by VisionMask. Participants are then asked whether the saliency maps help them better understand the decision-making process of the RL agents. We record the total number of responses as well as the number of positive responses, as shown in Figure 3. Over 89% of participants find the saliency maps generated by VisionMask helpful in understanding the agents' decisions. Details about

| Metrics | Stalder | L1 |
|---|---|---|
| Acc. | 95.6 | **95.9** |
| Ins. ↑ | 64.3 | **67.6** |
| Del. ↓ | 21.1 | **20.4** |
| LLE ↓ | **56.6** | 62.6 |
| Sp. ↑ | **73.8** | 62.3 |
| Hours | 0.72 | **0.68** |
| Speedup | - | 1.1× |
| Sp./LLE | **1.3** | 1.0 |

(a) **Area size regularization**. L1 is better for faithfulness and faster. Time is estimated by training 10 epochs.

| Metrics | FCN | UNet | DeepLabv3 |
|---|---|---|---|
| Acc. | 94.2 | 94.7 | **96.0** |
| Ins. ↑ | 56.5 | 56.8 | **67.5** |
| Del. ↓ | **19.6** | 19.7 | 20.4 |
| LLE ↓ | 100.0 | 86.3 | **62.6** |
| Sp. ↑ | 58.0 | 19.8 | **62.3** |
| Sp./LLE | 0.58 | 0.23 | **0.99** |

(b) $f_\theta$ **model**. DeepLabv3 achieves better results.

| Metrics | 0.1 | 0.2 | 0.4 |
|---|---|---|---|
| Acc. | **95.9** | 94.5 | 95.8 |
| Ins. ↑ | **57.0** | 56.9 | **57.0** |
| Del. ↓ | 19.5 | 19.5 | 19.5 |
| LLE ↓ | 96.0 | **93.8** | 99.5 |
| Sp. ↑ | **79.0** | 58.6 | 48.1 |
| Sp./LLE | **0.82** | 0.62 | 0.48 |

(c) **Max area size** $a_{\max}$. Maximum area size of 0.1 produces favorable results.

| Metrics | background | black | mean | blur |
|---|---|---|---|---|
| Acc. | **97.8** | 95.9 | 97.6 | 94.4 |
| Ins. ↑ | **67.5** | 51.7 | 54.5 | 56.9 |
| Del. ↓ | 20.4 | **16.8** | 18.2 | 25.0 |
| LLE ↓ | 62.6 | **54.4** | 92.6 | 99.6 |
| Sp. ↑ | **62.3** | 28.5 | 45.5 | 56.6 |
| Sp./LLE | **1.0** | 0.52 | 0.49 | 0.57 |

(d) **Reference value**. Using background produce better faithfulness values.

| Comp. | C. | +NE | +NE+L1 | +NE+L1+TV |
|---|---|---|---|---|
| Acc. | **96.7** | 96.5 | 95.9 | 95.6 |
| Ins. ↑ | 56.7 | 56.5 | **67.6** | 64.6 |
| Del. ↓ | 23.7 | 23.8 | **20.4** | **20.7** |
| LLE ↓ | 99.3 | 96.0 | 62.6 | 68.8 |
| Sp. ↑ | 24.0 | 23.8 | 62.3 | 60.5 |
| Sp./LLE | 0.24 | 0.25 | **0.99** | 0.88 |

(e) **Component**. Negative Entropy (NE) with L1 area regularization performs better in terms of faithfulness. Contrastive-only (C.), Stalder Area Loss (S.), and Total Variation (TV.)

Table 2: **VisionMask ablation experiments** on *SMB*. Default settings are marked in gray.

this study can be found in the Appendix D.

### 4.4 Ablation Studies

Table 2 shows the results for ablation studies. In all tables, the column marked in gray represents the default setting of VisionMask used to generate results in previous sections.

**Area size regularization**. In this experiment, we replace the default L1 area regularization with the *Min-Max Area Loss* from *Explainer* [Stalder *et al.*, 2022]. Without this constraint, *Explainer* tends to produce masks where all values are one in order to achieve the best performance. *Explainer* first vectorizes and sorts the mask elements, and then applies penalties to pixels falling outside the range $[a, b]$, where the hyperparameters $a$ and $b$ specify the minimum and maximum allowable area sizes. Instead, VisionMask uses L1 area regularization to constrain the maximal area, as we do not wish to impose constraints on the minimum area size. To implement action contrastive learning, we need to generate masks for actions that are not selected, and these masks should be allowed to be all zeros if necessary. As shown in Table 2a, using L1 area regularization improves the faithfulness of the explanation and increases training speed.

**Max area size**. We further vary the maximum permitted area size $a_{\max}$ to evaluate its impact. As shown in Table 2c, reducing $a_{\max}$ significantly increases the sparseness as it constrains the model to focus on the most discriminative regions without compromising other performance metrics. The 0.1 setting achieves the best results, as expected.

$\mathbf{f}_\theta$ **model**. In Table 2b, we compare the performance of three different versions of VisionMask with three different segmentation models: our default DeepLabv3 [Chen *et al.*, 2017], Fully Convolutional Network (FCN)[Long *et al.*, 2015], and UNet[Ronneberger *et al.*, 2015]. The results show that DeepLabv3 achieves a better performance.

**Reference Value.** We evaluate the impact of hyperparameter $r$ in Equation 1 by setting the reference value to *background*, *black*, *mean*, and *blur*. The "black" reference sets $r$ to 0, the "mean" reference sets $r$ to the RGB mean value of the entire dataset, and the "blur" reference applies Gaussian blur (kernel is 39, $\sigma = 15$) to the image and use the blurred pixel value as $r$. Table 2d compares the performance of VisionMask with these four different reference values. We choose the background as reference value as it provides a more balanced performance across all metrics. Details about the background reference value are provided in Appendix F.

**Loss Function Components**. In Table 2e, we evaluate the impact of different components in the loss function on the performance of VisionMask. The results show that, although adding negative entropy and L1 area regularization slightly reduces accuracy, it improves the insertion and deletion scores and significantly boosts the Sp./LLE ratio. In contrast, adding total variation to produce smoother masks has minimal impact on overall performance.

## 5 Conclusion

We presented *VisionMask*, an agent-agnostic DRL explanation model trained in self-supervised contrastive learning. *VisionMask* generates explanations with higher fidelity and better effectiveness compared to existing attribute-based methods. It is our future plan to extend this approach to multimodality input and couple the visual explanation generated by the VisionMask with other information such as agent's long-term goals and future rewards.

## Acknowledgments

This research is partially supported by the Air Force Office of Scientific Research (AFOSR), under contract FA9550-24-1-0078, and NSF award CNS-2148253. The paper was received and approved for public release by Air Force Research Laboratory (AFRL) on May 28th 2024, case number AFRL-2024-2908 . Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

## References

[Adebayo *et al.*, 2020] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 700–712. Curran Associates, Inc., 2020.

[Alvarez Melis and Jaakkola, 2018] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[Annasamy and Sycara, 2019] Raghuram Mandyam Annasamy and Katia P. Sycara. Towards better interpretability in deep q-networks. pages 4561–4569. AAAI Press, 2019.

[Ashwood *et al.*, 2022] Zoe Ashwood, Aditi Jha, and Jonathan W Pillow. Dynamic inverse reinforcement learning for characterizing animal behavior. *Advances in Neural Information Processing Systems*, 35:29663–29676, 2022.

[Bach *et al.*, 2015] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[Beechey *et al.*, 2023] Daniel Beechey, Thomas MS Smith, and Özgür Şimşek. Explaining reinforcement learning with shapley values. In *International Conference on Machine Learning*, pages 2003–2014. PMLR, 2023.

[Bellemare *et al.*, 2013] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[Chalasani *et al.*, 2020] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1383–1391. PMLR, 13–18 Jul 2020.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[Chopra *et al.*, 2005] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005.

[Fong and Vedaldi, 2017] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3449–3457. IEEE Computer Society, 2017.

[Fong *et al.*, 2019] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019.

[Greydanus *et al.*, 2018] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1787–1796. PMLR, 2018.

[Gutmann and Hyvärinen, 2010] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[Hedström *et al.*, 2023] Anna Hedström, Philine Lou Bommer, Kristoffer Knutsen Wickstrøm, Wojciech Samek, Sebastian Lapuschkin, and Marina MC Höhne. The meta-evaluation problem in explainable AI: Identifying reliable estimators with metaquantus. *Transactions on Machine Learning Research*, 2023.

[Heuillet *et al.*, 2021] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, 2021.

[Hickling *et al.*, 2023] Thomas Hickling, Abdelhafid Zenati, Nabil Aouf, and Phillippa Spencer. Explainability in deep reinforcement learning: A review into current methods and applications. *ACM Comput. Surv.*, 56(5), December 2023.

[Iyer *et al.*, 2018] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia P. Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 144–150. ACM, 2018.

[Kauten, 2018] Christian Kauten. Super Mario Bros for OpenAI Gym. GitHub, 2018.

[Kirillov *et al.*, 2023] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura

Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[Klein *et al.*, 2024] Lukas Klein, Carsten T. Lüth, Udo Schlegel, Till J. Bungert, Mennatallah El-Assady, and Paul F Jaeger. Navigating the maze of explainable AI: A systematic approach to evaluating methods and metrics. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[Leurent, 2018] Edouard Leurent. An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env, 2018.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[Machado *et al.*, 2018] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Nguyen, 2020] Viet Nguyen. Proximal policy optimization (ppo) for playing super mario bros. GitHub, 2020.

[Petsiuk *et al.*, 2018] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 151. BMVA Press, 2018.

[Puri *et al.*, 2020] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad V. Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Qing *et al.*, 2022] Yunpeng Qing, Shunyu Liu, Jie Song, Huiqiong Wang, and Mingli Song. A survey on explainable reinforcement learning: Concepts, algorithms, challenges. *arXiv preprint arXiv:2211.06665*, 2022.

[Raffin *et al.*, 2021] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Shapley and others, 1953] Lloyd S Shapley et al. A value for n-person games. 1953.

[Shrikumar *et al.*, 2017] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017.

[Shu *et al.*, 2017] Tianmin Shu, Caiming Xiong, and Richard Socher. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *arXiv preprint arXiv:1712.07294*, 2017.

[Stalder *et al.*, 2022] Steven Stalder, Nathanaël Perraudin, Radhakrishna Achanta, Fernando Perez-Cruz, and Michele Volpi. What you see is what you classify: Black box attributions. *Advances in Neural Information Processing Systems*, 35:84–94, 2022.

[Sutton *et al.*, 1999] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

[Topin *et al.*, 2021] Nicholay Topin, Stephanie Milani, Fei Fang, and Manuela Veloso. Iterative bounding mdps: Learning interpretable policies via non-interpretable methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9923–9931, 2021.

[Wydmuch *et al.*, 2018] Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 2018.