Causal State Distillation for Explainable Reinforcement Learning

Wenhao Lu WENHAO.LU@UNI-HAMBURG.DE
Xufeng Zhao XUFENG.ZHAO@UNI-HAMBURG.DE
Thilo Fryen THILO.FRYEN@UNI-HAMBURG.DE
Jae Hee Lee JAE.HEE.LEE@UNI-HAMBURG.DE
Mengdi Li MENGDI.LI@STUDIUM.UNI-HAMBURG.DE
Sven Magg SVEN.MAGG@UNI-HAMBURG.DE
Stefan Wermter STEFAN.WERMTER@UNI-HAMBURG.DE
University of Hamburg

Editors: Francesco Locatello and Vanessa Didelez

Abstract

Reinforcement learning (RL) is a powerful technique for training intelligent agents, but understanding why these agents make specific decisions can be quite challenging. This lack of transparency in RL models has been a long-standing problem, making it difficult for users to grasp the reasons behind an agent's behaviour. Various approaches have been explored to address this problem, with one promising avenue being reward decomposition (RD). RD is appealing as it sidesteps some of the concerns associated with other methods that attempt to rationalize an agent's behaviour in a post-hoc manner. RD works by exposing various facets of the rewards that contribute to the agent's objectives during training. However, RD alone has limitations as it primarily offers insights based on sub-rewards and does not delve into the intricate cause-and-effect relationships that occur within an RL agent's neural model. In this paper, we present an extension of RD that goes beyond sub-rewards to provide more informative explanations. Our approach is centred on a causal learning framework that leverages information-theoretic measures for explanation objectives that encourage three crucial properties of causal factors: *causal sufficiency*, *sparseness*, and *orthogonality*. These properties help us distill the cause-and-effect relationships between the agent's states and actions or rewards, allowing for a deeper understanding of its decision-making processes. Our framework is designed to generate local explanations and can be applied to a wide range of RL tasks with multiple reward channels. Through a series of experiments, we demonstrate that our approach offers more meaningful and insightful explanations for the agent's action selections.

Keywords: Explainable RL; Causality; Reward Decomposition

1. Introduction

Many efforts have been made to adapt *post-hoc* saliency approaches from the field of explainable machine learning (Selvaraju et al., 2016; Ribeiro et al., 2016; Shrikumar et al., 2017; Sundararajan et al., 2017) to understand the behaviour of reinforcement learning (RL) agents. These approaches usually aim to provide visual explanations by highlighting salient state features that influence an agent's action choices (Greydanus et al., 2018; Iyer et al., 2018).

However, we identify two key issues in applying these approaches to RL. First, there is a general concern about using saliency maps to explain RL agent behaviour as post-hoc explanations are not grounded in the agent's learning process (Milani et al., 2022). The work by Atrey et al. (2020) emphasizes that saliency might convey misleading, non-causal interpretations of agent actions. For example, in Breakout, the saliency pattern and intensity around a tunnel vanish when a reflection

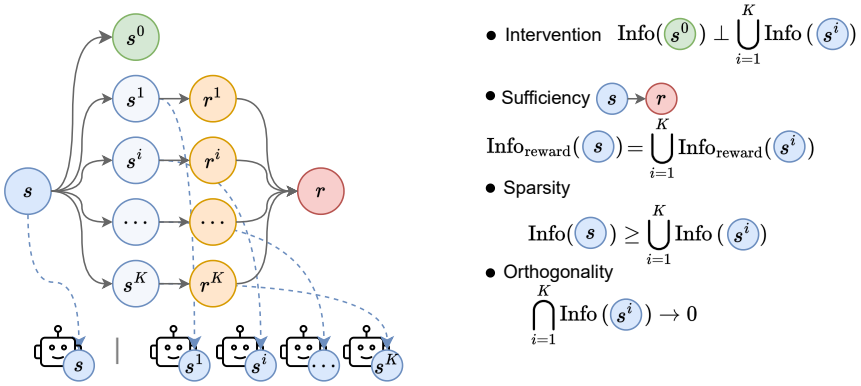


Figure 1: The disentanglement of state representations and resulting sub-agents when uncovering the cause-effect relationships with *causal state distillation* (action omitted for brevity). Here, s^0 denotes the distilled non-causal components of state s, while s^i captures the causal elements, each linked to a distinct reward aspect r^i. Sub-agents focus on a singular causal component for policy learning. The distillation process, consisting of multiple learning steps, is elaborated in Sec. 3.3.

intervention is applied to bricks near the tunnel, refuting the hypothesis that agents learn to aim at tunnels (Atrey et al., 2020).

Second, saliency-based approaches often overlook RL-specific aspects, limiting their effectiveness in generating meaningful saliency maps. They are developed for supervised tasks, which typically address non-temporal reasoning and are focused on model behaviours concerning specific objectives, such as classifying an image into a specific category. Explanations for RL agents must go beyond this and provide additional insights into the agent's interaction data, encompassing the rewards it has received, the states it has transitioned between, and the diverse goals it strives to achieve. This contextual information, which exists during the learning process, is vital for refining our understanding of the agent's decision-making. Unfortunately, saliency maps fall short in this regard, as their generation does not rely on any interaction data.

In this research paper, we thus take a new route and investigate a way to allow RL agents to *intrinsically* attend to *causal* but *distinguishable* state components, predictive of the agent's action and reward obtained during its learning. An appropriate candidate we consider here is *Reward Decomposition* (RD) (Juozapaitis et al., 2019; Septon et al., 2023; Lu et al., 2023) which discerns the contribution of each sub-reward to the agent's decision-making. However, RD has its limitations, as it does not unveil which specific state components are being utilized or attended to by each decision-making policy induced by various sub-rewards. Our primary focus is on RL tasks where there are multiple reward channels (i.e., sub-rewards) sourcing from different environmental factors, for example, both bonking the gopher or filling holes contribute to the achievement of the goal in the Gopher game (Bellemare et al., 2012).

To ensure that we attain various sorts of attention from the agent that faithfully explains its decision-making process, a powerful approach is to use the language of causality. In this paper, we introduce a structural causal model (Pearl, 2009) to formalize the problem of how different state components contribute to diverse reward aspects or, as a consequence, Q-values (see Fig. 1 for an overall visualization). Concretely, we aim to separate the latent factors (or *state components*) that

are causally relevant to the agent's decision-making from those that are not. Besides, we introduce three desired properties of causal factors, i.e., *sufficiency*, *sparsity* and *orthogonality*, to constrain the information flow during the learning process. An inherent advantage of our explanatory framework is that the learned causal factors can serve as a rich vocabulary for explicating an agent's action. These causal factors improve over saliency maps in both expressiveness and diversity. Each latent factor, in isolation, unravels intricate patterns (events) in the agent's interactions. Moreover, this ensemble of diverse factors offers a multifaceted perspective on the agent's attention to each of them, thereby unveiling the rationale behind its actions.

Our contributions can be summarized as follows:

- We investigate RL explanations from a causal perspective and propose a novel framework for generating explanations in the form of causal factors, driven by three essential desiderata.
- We present two paradigms (R-Mask and Q-Mask) of distilling causal factors, in which the factorization is ensured by imposing causal sufficiency of reward and Q-value respectively.
- We establish reasonable evaluation metrics to quantify the explanatory quality.
- We conduct an analysis of this framework in a toy task for intuitive understanding and an extended evaluation applied to explaining agents involved in complex visual tasks.

2. Related Work

In line with the taxonomy of eXplainable Artificial Intelligence (XAI) approaches, XRL approaches can be naturally categorized into two scopes: *local* and *global*. Local approaches refer to explaining a single decision for a single situation. In contrast, global approaches aim to explain the long-term behaviour of a learned RL model (i.e., on policy or trajectory level) (Milani et al., 2022; Qing et al., 2022). Our explanation framework globally learns to discover which state components (latent factors) are beneficial for local explanations.

Local Feature Importance. Most local explanation techniques for RL extend from those in XAI, explaining the prediction for a specific data instance (Selvaraju et al., 2016; Ribeiro et al., 2016; Shrikumar et al., 2017; Sundararajan et al., 2017). Those local explanations provide action-oriented explanations for RL agents' behaviour through post-hoc *rationalization*. Post-hoc interpretability refers to generating action explanations for a non-interpretable RL model, by the forms of saliency maps (Greydanus et al., 2018; Iyer et al., 2018; Gupta et al., 2019). The work of Greydanus et al. (2018) derives saliency maps by observing the changes in the policy after adding Gaussian blur to different parts of input images. However, the saliency map can highlight regions of the input that are not relevant to the action taken by the agent. Complementary saliency work by Gupta et al. (2019) mitigates this issue. Nevertheless, the saliency map used in practice as evidence of explanations for RL agents might be highly subjective and not falsifiable (Atrey et al., 2020). That is, *ad hoc* claims to the agent's behaviour are proposed after the presented saliency is interpreted.

MDP-aware Explanation. Another important category of explanations is those which expose the impact of parts of the MDP (e.g., reward $\mathcal{R}$ and dynamics model $\mathcal{P}$) (Puterman, 2014) on the agent's behaviour. Those techniques generally require additional information for training. For example, the line of work in *reward decomposition* (Juozapaitis et al., 2019; Septon et al., 2023; Lu et al., 2023) needs to know the existing reward structure prior to the agent's learning. The resulting explanation

artefacts clarify the contribution of each reward component to the agent's decision (i.e., Q-values). However, despite their potential, these explanations rely on scalar Q-values and do not disclose which state aspect impacts the estimation of diverse Q-values, limiting their actionable value.

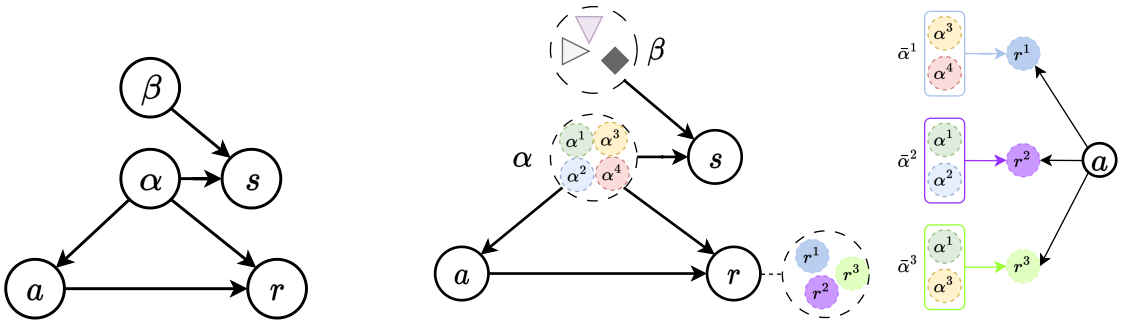
Causality in Explanations. The language of cause and effect has gathered increasing attention in generating explanations (Moraffah et al., 2020). An earlier work for causal explainability is Madumal et al. (2019), which explains "why" and "why not" questions with a learned causal model. However, it relies on known abstract state variables for explanation generation, restricting it to discrete setups. On the contrary, our approach extends to continuous settings, accommodating learned factors, associated with various reward facets, for explanations. The work of Bica et al. (2020) aims to achieve a parameterizable interpretation of the expert's behaviour in the batch Inverse Reinforcement Learning (IRL) setting by employing a counterfactual-based reward function. However, this method is limited to linear reward functions based on data features. Recent work has quantified state and temporal importance to action selection by leveraging learned structural equations for a known causal structure (Wang et al., 2023), but its application is limited to unusual cases with abstract states. Unlike it, some aim to find explanatory input (e.g., graph or image data) for model prediction by measuring information flow (which can be seen as the causal counterpart of mutual information) (Ay and Polani, 2008; O'Shaughnessy et al., 2020; Lin et al., 2022), or by causal interventions (Lv et al., 2022; Wu et al., 2023). However, they provide merely post-hoc causal explanations within the realm of supervised settings. In contrast, our research delves into the realm of *inherent* RL explanations, a more intricate problem, approached from a causal perspective. Though our proposed causal RL explanation framework draws upon similar notions of causality as found in non-RL post-hoc explanation works by O'Shaughnessy et al. (2020); Lin et al. (2022) for constructing explanations, we emphasize that ours is unique in that the framework can generate latent factor-based explanation associated to various reward facets, all the while coevolving with the agent's policy learning.

Our explanation method can be categorized into causal RL (Zeng et al., 2023), an emerging subfield of RL that harnesses the power of causal inference. Different from approaches Zhu et al. (2022); Guo et al. (2022) which utilize causality for learning representation that benefits the generalizability and sample efficiency of RL agents, this work leverages causality to learn an intrinsically interpretable RL policy.

3. Methodology

Our goal is to locally explain an agent's action at a state from the causal view with a structural causal model (SCM) (Pearl, 2009) that globally describes how *factors* or components of states $\langle \alpha, \beta \rangle$ *causally* affect agent's actions and rewards it received. The effect is causal in the sense of changing the causal factors α produces changes in the agent's behaviour and the consequence, while non-causal factors β should not.

To formalize the explanations, we need to define (i) a causal graph that relates state factors, agent's actions (a), and its rewards (r); (ii) an approach to disentangling causal factors from non-causal ones; (iii) a metric to measure the causal influence of α on a and r; and (iv) a learning framework that learns α while ensuring the success of the policy learning of corresponding RL tasks. Here, we focus on RL tasks with multiple reward channels which may be unknown during training.



(*a*) The SCM for RL explanations where α represents causal factors and β for non-causal ones.

(*b*) The extended SCM for RL explanations with an example illustration of factors and sub-rewards (right). For instance, both α^1 and α^2 determine sub-reward r^2. For brevity, the edges from all subsets of factors ($\bar{\alpha}^i$) to action a are omitted (rightmost part).

Figure 2: The causal graph for one-step RL explanations.

3.1. A causal view on explanations

Our explanations for agent's behaviours take the form of a set of causal factors. That is, by construction, the functional relationship defining the causal connection $\pi : \alpha \to a$ uses only the factors of a state s that are causal. Based on this observation, we then adopt an SCM as depicted in Fig. 2(*a*) to describe the causal structure between α, a, and r. In tandem with α, non-causal factors β contribute to representing states the agent observed but would not causally influence the agent's actions and rewards. Stated differently, any interventions on α and β cause changes in s, but only interventions on α cause changes in a and r. Besides, any alternations to β would not have an impact on the causal factors α as well. Importantly, we do not *assume* α is given a priori as Datta et al. (2016); Shrikumar et al. (2017) do, but we intentionally *learn* to separate α from β. A formalization of the RL problem in SCM can be found in Appendix C.1.

Since causal factors are generally not observable (Arjovsky et al., 2019) and their extraction relies on the availability of specific supervision signals and interventions (Schölkopf et al., 2021), we seek to learn them in a way that each factor in α corresponds to a different aspect of the environmental state and a subset of causal factors has a sizeable causal influence on a reward component (sub-reward) r^i and the action chosen a. To this end, we expand the SCM in Fig. 2(*a*) to explicitly illustrate the relationship among causal factors, action, and sub-rewards, as depicted in Fig. 2(*b*).

3.2. Notions and desiderata for explanations

Notions. We assume a factorization of $\alpha = \{\alpha^1, \alpha^2, \ldots, \alpha^N\}$ and the additivity of reward $r = \sum_{i=1}^{K} r^i$, where $N, K \in \mathbb{N}$. Notably, N and K may differ. We further denote $\bar{\alpha}^i$ as a subset of causal factors corresponding to a sub-reward r^i and the actual values of sub-rewards may be unknown a priori. As for retrieving causal factors $\bar{\alpha}^i$, we extract them from the raw state s_t or a learned representation of it, i.e., $\alpha = \psi(s_t)$ by using a neural network-based masker $m^i(\cdot)$, i.e., $\bar{\alpha}^i = m^i(s_t) * \psi(s_t) = m^i(s_t) * \alpha$.

To ground the learning of causal factors α functioning as described in Sec. 3.1, we further highlight several desiderata for explanations that these learned factors are expected to fulfill. In

the next sections, we discuss how to approach these desiderata from the standpoint of information theory and by using do-operator $do(\cdot)$ (Pearl et al., 2016).

- The causal factors α should be independent of non-causal factors β, i.e., $\alpha \perp \beta$. Thus, intervening on β does not change α and the learned $\pi : \alpha \to a$ as well.
- The causal factors α (or $\bar{\alpha}^i$) are desired to be causally sufficient for rewards $\alpha \to r$ (or sub-rewards $\bar{\alpha}^i \to r^i$) and action $\alpha \to a$, i.e., to contain all information required to predict r (or r^i) and explain the causal dependency between α and a.
- Given any two subsets of causal factors $\bar{\alpha}^i$, $\bar{\alpha}^j$ corresponding to sub-rewards r^i and r^j respectively, $\bar{\alpha}^i$ (or $\bar{\alpha}^j$) needs to contain less or no information about determining r^j (or r^i). Besides, we expect $\bar{\alpha}^i$ (or $\bar{\alpha}^j$) to be minimally sufficient, i.e., containing the least amount of (sufficient) information for predicting r^i (or r^j).

3.3. The learning framework

Recall that the first criterion indeed amounts to performing the causal intervention (Pearl, 2009) on non-causal factors β, i.e., $P(\alpha|do(\beta))$, the second requires a metric for the causal influence of α on a and r using the SCM in Fig. 2(*b*), and the last needs a measure of independence between any subsets over α. Together, a learning framework is developed to unify these desiderata.

3.3.1. METRIC FOR CAUSAL INTERVENTION

In general, causal and non-causal factors coexist in the agent's interaction with the environment. We aim to separate causal factors α from non-causal ones β by causal intervention, ensuring that α remains invariant when β undergoes interventions ($do(\beta)$). Notably, non-causal factors may not always be directly observable but can be accessed through domain knowledge. For instance, in Atari games, the displayed scores on the scoreboard can be considered a non-causal factor. As Piotrowski and Campbell (1982) noted, the Fourier transformation preserves high-level semantics in the phase component while encoding low-level statistics in the amplitude component. Therefore, in line with Lv et al. (2022), we intervene on β by perturbing the amplitude component while maintaining the phase. Starting with the original state s and a state s' devoid of non-causal factors, we perform the intervention, resulting in an intervened state s^{inter} (i.e., $s \backslash \beta$, where β associated parts are removed). Details on the intervention procedure are available in Appendix C.2. Then, we optimize the encoder ψ by maximizing the following correlation to maintain the invariance of α following the aforementioned intervention upon β:

$$\max \sum_i cos(\psi(s), \psi(s^{inter})), \tag{1}$$

where we leverage cosine similarity $cos(\cdot, \cdot)$ to measure the correlation between causal factors before and after intervening on β.

3.3.2. METRIC FOR CAUSAL SUFFICIENCY

Causal sufficiency for reward. A distilling masker $m^i(\cdot)$ is regarded as causally sufficient if the information transition to the reward is sufficient such that the causality between the (sub-)event trigger and its environmental feedback holds clearly, i.e., $\mathbb{E} \log \hat{p}(r^i|\bar{\alpha}^i_t) = \mathbb{E} \log p(r^i|s_t)$ and

$\mathbb{E}\log\hat{p}(r|\bigcup_{i=1}^{K}\bar{\alpha}_t^i) = \mathbb{E}\log p(r|s_t)$. The sufficiency of $\bar{\alpha}^i$ to deduce r^i can be achieved by maximizing their mutual information $\mathcal{I}(\bar{\alpha}^i; r^i)$ or fitting a reward model $\mathcal{R}_\theta$ such that $r^i = \mathcal{R}_\theta(\bar{\alpha}^i, a)$. The total information regarding the environmental causality thus can be persisted via the regression $r = \sum_i^K r^i = \sum_i^K \mathcal{R}_\theta(\bar{\alpha}^i, a)$, i.e., by minimizing the L_2-norm fidelity loss

$$\min \mathbb{E}\| \sum_i R_\theta(\bar{\alpha}^i, a) - r\|_2, \tag{2}$$

towards reward information persistence (omitted when raw sub-rewards are given in advance).

Causal sufficiency for action. Though, by disentangling state representation with the above objective we can obtain causal factors that are sufficient in terms of determining sub-reward r^i, it is equally crucial to get the impact of causal factors timely involved in action selection, i.e., whether the distilled factors are sufficient or even beneficial for learning an optimal policy. The joint learning process of decomposing state and fitting a policy may fall into an unstable or even vicious loop — insufficient factors exert challenges to policy learning, while non-informative trajectories unrolled by an under-optimized policy, in turn, hinder the causality distillation (Li et al., 2023). We thus report the findings of (masked) Q-learning with causal factors under the setting that sub-rewards are *known* from the environment, leaving the more challenging one, where the reward decomposition has to be jointly learned, for future work.

To assess the impact of causality distillation on Q-agent learning in RD, we contrast two controlled Q-learning variants with and without access to the full state. That is, the Q-agent consumes and updates according to the sub-state (that is sufficient and concise to reveal the i-th causal aspect of the state): $Q^i(\bar{\alpha}_t^i, a_t) \leftarrow (1-\alpha)Q^i(\bar{\alpha}_t^i, a_t) + \alpha[r_t^i + \gamma Q^i(\bar{\alpha}_{t+1}^i, a_t^*)]$, or to the full state (that contains richer yet potentially distracting information): $Q^i(s_t, a_t) \leftarrow (1-\alpha)Q^i(s_t, a_t) + \alpha[r_t^i + \gamma Q^i(s_{t+1}, a_t^*)]$. Here, α and γ are hyper-parameters for Q-learning, while a_t^* denotes the global optimal action. Further details, findings, and discussions are presented in the experiment section.

3.3.3. Metric for Sparsity and Orthogonality

Sparsity. We consider the information shunt to be *sufficient* in terms of reward recognition while being concise, such that any irrelevancy or redundancy information is masked out, resulting in a *sparse* information flow. This property can be described as the maximization of *information loss* after a state transformation $s_t \to \bar{\alpha}_t^i$. That is, deducing the full state from the partial knowledge from a sub-state becomes more difficult as the information loss increases. Following the definition from Geiger and Kubin (2011), the objective of maximizing the information loss for the i-th flow (i.e., transformation) is defined as

$$\max \sum_i \mathcal{L}(s_t \to \bar{\alpha}_t^i) \triangleq \max \sum_i \lim_{\hat{s}_t \to s_t} [\mathcal{I}(\hat{s}_t; s_t) - \mathcal{I}(\hat{s}_t; \bar{\alpha}_t^i)] = \max \sum_i \mathcal{H}(s_t|\bar{\alpha}_t^i), \tag{3}$$

where $\mathcal{H}(s_t|\bar{\alpha}_t^i)$ is the conditional entropy indicating the uncertainty to deduce s_t given $\bar{\alpha}_t^i$.

Orthogonality. To achieve that $\bar{\alpha}^i$ (or $\bar{\alpha}^j$) contains less or no information about determining r^j (or r^i) (cf. Sec. 3.2), we approximately regard this as the information orthogonality describing the independence between inter-states $\bar{\alpha}^i$ and $\bar{\alpha}^j$, which can be achieved by minimizing their mutual information, i.e.,

$$\min \sum_{i \neq j} \mathcal{I}(\bar{\alpha}_t^i; \bar{\alpha}_t^j). \tag{4}$$

Note that the component reward r^i can be given in advance (i.e., a known reward decomposition (Juozapaitis et al., 2019)) or be derived dynamically according to the distillation criteria (Lin et al., 2020). In the latter case for learning $\mathcal{R}$, explicit incentives for the consistency of s^i and r^i should be applied to avoid trivial solutions such as projecting all $K-1$ states to 0 but leaving only one to r. For example, an objective of $\mathcal{I}(\bar{\alpha}^i; r^i)$ to maximize or $\mathcal{I}(\bar{\alpha}^i; r^j)$ to minimize when taking into account the orthogonality and the fact the $\bar{\alpha}^j$ should be aligned with r^j, but not r^i.

3.4. Optimization procedure

The overall optimization objective is a balanced combination of Eq. 1, Eq. 2, Eq. 3 and Eq. 4, which involves neural estimation of entropy and mutual information (Belghazi et al., 2018; van den Oord et al., 2018; Lin et al., 2020; Cheng et al., 2020; Radford et al., 2021). For the estimation of mutual information, we individually approximate the entropy components[1] and follow previous work by Lin et al. (2020) for the entropy approximation. Future work will involve exploring the success of InfoNCE loss in contrastive learning (van den Oord et al., 2018) for better estimation.

In practice, considering the fact that $\bar{\alpha}^i$ is a subset of s, the knowledge of s leads to the knowledge of $\bar{\alpha}^i$, such that $\max \sum_i \mathcal{H}(s_t|\bar{\alpha}_t^i) = \max \sum_i [\mathcal{H}(\bar{\alpha}_t^i|s_t) + \mathcal{H}(s_t)] - \mathcal{H}(\bar{\alpha}_t^i) \approx \max \sum_i -\mathcal{H}(\bar{\alpha}_t^i)$, which leads to an efficient estimation by, approximately, minimizing $\mathcal{H}(\bar{\alpha}^i)$. This approximation reduces to the objective applied in previous works which can be optimized by minimizing one of its upper bounds in proportion to $\sum_i \log |m^i(s)|$ (Geiger and Kubin, 2011; Lin et al., 2020). We additionally optimize it with an L_1 penalty $\sum_i |m^i(s)|$ for the sake of sparse weights and stable information transition[2] (Li et al., 2023), and experimentally demonstrate its effectiveness.

We refer to the technique that employs the objectives (in Eq. 1, Eq. 2, Eq. 3, Eq. 4) to distil $\bar{\alpha}^i$ of a state, which in turn dictates the reward component r^i, as *R-Mask*. In the masked Q-learning, the reward components are known a priori, and the acquisition of $\bar{\alpha}^i$ is synergized with the RL objective, alongside the objectives (in Eq. 1, Eq.3 and Eq.4) governing mask updates. This technique for mask learning is referred to as *Q-Mask*. As an ablation, their counterparts without sparsity and orthogonality losses (Eq. 3 and Eq. 4) are denoted as *R-Mask Lite* and *Q-Mask Lite*, respectively.

4. Experiments

The following research questions outline the progressive evaluation of our explanation framework through extensive experiments:

- Q1. In comparison to vanilla RD (Juozapaitis et al., 2019), how does the auxiliary task of decomposing reward (i.e., predicting $r^i(s, a)$) influence the generation of explanation artefacts?
- Q2. Following reward prediction in Q1, what insights can be gained about the role of causal sufficiency of reward components (i.e., estimating $r^i(\bar{\alpha}^i, a)$ in Sec. 3.3.2) in learning causal factors?
- Q3. Compared to the causal sufficiency of reward components above, how does the causal sufficiency concerning action (Sec. 3.3.2) impact the learning of causal factors *uniquely*?

1. Recall that $\mathcal{I}(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y)$.
2. The objective without "log" can be derived from the perspective of f-mutual information but the choice of proper information measure, e.g., Kullback–Leibler or Jensen–Shannon divergence, remains undetermined for sophisticated learning systems.

For a comprehensive list of methods and their distinctions studied in the experiments addressing the research questions, please refer to Table 6. An illustration of training flows of the R-Mask and the Q-Mask can be found in Fig. 17 and Fig. 18 in the Appendix, respectively. Neural network architecture details can be found in Appendix C.12, and pseudocode in Appendix D.

4.1. Experimental setup

To validate our causal attention principles in agent learning and answer the research questions, we conduct experiments on tasks of varying complexity and scale. We use two Atari 2600 (Bellemare et al., 2012) tasks from OpenAI Gym (Brockman et al., 2016), including Gopher and MsPacman.

Environments. In the **Gopher** game ($K = 2$), a farmer (i.e., the agent) protects carrots from a gopher. The agent receives a reward of 0.8 for bonking the gopher as it emerges from the holes or anywhere above ground, and a reward of 0.15 for filling those holes before the gopher tunnels out and eats carrots. In the **MsPacman** game ($K = 3$), Pacman walks through a maze populated with various items (e.g., enemies and dots) and its object is to score as many as possible by eating them. The multiple-reward structure in the game is as follows: the agent receives a reward of 0.25 when it gobbles a Dot up and a reward of 1 for eating an Energy Pill. When the agent gulps down one Energy Pill, the ghosts turn blue and Pacman can eat them. It earns a reward of 5 for each ghost (maximum 4 ghosts, i.e., 20) gobbled up. Note that we also introduce a **MiniGrid** toy task when addressing research question Q3.

Performance. This metric represents the maximum score attained by the RL agent in a task.

Critical State. Given the human interest in understanding agent decisions relative to expectations, not all encountered states hold equal explanatory value. Critical states, characterized by significant utility gaps between optimal and second-best actions, are of particular interest. We evaluate and explain states as *critical* based on the utility gap: $C(s) = \max_a Q(s,a) - \text{second-highest}_a Q(s,a)$, as specified in Amir and Amir (2018); Septon et al. (2023). We also consider states where the agent receives positive rewards (only non-negative rewards in the Atari tasks we considered).

4.2. Analysis of research questions

Q1. How does the task of reward prediction influence the explanation generation?

This question is raised under the hypothesis that our understanding of an agent's behaviour may benefit from probing other aspects (e.g., reward) of the agent's interaction data. Hence, on top of RD[3], we introduce an auxiliary task where the agent learns to predict reward components $r^i(s,a)$, each supervised by a ground truth sub-reward signal. We denote this variant as *RD-pred*. Compared with RD in Table 1, the performance drop is only considered moderate (-7.62%). However, we illustrate that the reward prediction task helps interpretability of agent behaviour.

To visually differentiate the resulting explanations[4], we adopt the GradCAM technique to generate post-hoc saliency maps for each component Q-value and reward r^i concerning a state s. Fig. 3 shows that Q-value saliency associated with the ground reward erroneously focuses on the scoreboard, leading to a causal fallacy of putting the effect before the cause. In contrast, R saliency attends to temporarily relevant, yet not precise areas (e.g., leftmost ground and avatar body). This

3. Refer to Appendix C.5 for a concise description of how RD functions.
4. In vanilla RD, explanations typically involve sub-Q-value trade-offs or their differences.

can be attributed to the fact that predicted rewards reflect the value of transitioning to the next step from the current state, while Q-values reveal the expected long-term gain that may result in distortion of the causal structure because of this information compression along the time-axis. This finding indicates that reward saliency is more informative in terms of interpreting the agent's temporary behaviour than Q-value saliency.

In the following section, we will introduce further learning objectives to explore causal structures (cf. Sec. 3.2).

Table 1: Evaluation results for RD, RD-pred, RD-pred-u.

Evaluation Metric		Performance
Gopher	RD	$\mathbf{15.62 \pm 1.58}$
	RD-pred	14.43 ± 0.41
	RD-pred-u	13.78 ± 0.21

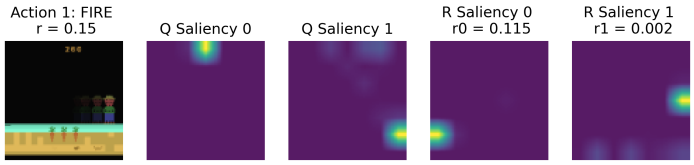


Figure 3: Comparison of saliency maps (associated with ground and gopher rewards) of RD with RD-pred in a state where the agent filled the hole and attained reward 0.15. Q saliency refers to the generated saliency of Q-value; R saliency pertains to the generated saliency of reward.

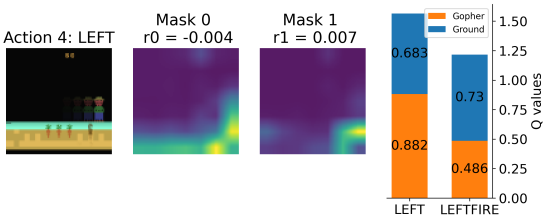
Q2. WHAT IS THE GAINED INSIGHT INTO THE ROLE OF CAUSAL SUFFICIENCY OF REWARD COMPONENTS IN LEARNING CAUSAL FACTORS IN THE R-MASK APPROACH?

The RD-pred approach, a variant of RD with reward prediction, does not encourage the information transition to be sufficient as a full state (i.e., all environmental aspects) is used to deduce r^i, thus complicating the disentanglement between reward components. The R-Mask approach constrains this information flow by employing the aforementioned objectives (Sec. 3.3) to distil disentangled components of a state. Its effectiveness can be seen in Fig. 4 where causal factors (represented as attention masks) precisely identify relevant areas, enhancing our understanding of the agent's attention. For a fair comparison, we introduce a modified RD-pred with *unknown* sub-rewards, denoted by *RD-pred-u*, which only uses full reward supervision (for reward prediction) similar to R-Mask (see Table 6). Masks generated by R-Mask emphasize more relevant objects, such as the avatar and gopher in Fig. 4, while RD-pred-u focuses on irrelevant objects, like a flying bird, or loses focus entirely (as observed in Fig. 6(*a*) and Fig. 6(*b*) in the Appendix). This underscores the necessity of explicit signals (like those in Sec. 3.3 relied upon by R-Mask) to establish the correspondence between environmental aspects and sub-rewards. Interestingly, despite the performance drop in RD-pred-u (Table 1), R-Mask achieves a relatively higher task return, albeit slightly lower than the baseline RD performance.

Q3. HOW DOES THE CAUSAL SUFFICIENCY CONCERNING ACTION IMPACT THE LEARNING OF CAUSAL FACTORS UNIQUELY?

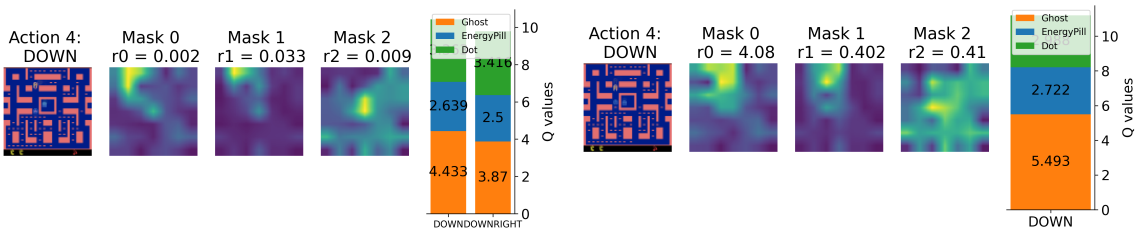
The information sufficiency of determining the rewards and optimal decisions for an agent are highly correlated but not necessarily equivalent. The agent from the Q-Mask consumes the distilled state

5. Note, in Fig. 3, reward r0 signifies the ground reward as task specified. Here, r0 denotes the gopher reward, which we manually verify post-hoc after decomposition has been learned.



(*a*) R-Mask masks for a state with reward $r = 0$. (*b*) R-Mask masks for the **next** state with reward $r = 0.95$.

Figure 4: R-Mask attention masks[5] from Gopher and their interpretation along with Q-value bars.



(*a*) R-Mask masks for a state with reward $r = 0$. (*b*) R-Mask masks for the **next** state with reward $r = 5$.

Figure 5: R-Mask attention masks from MsPacman and their interpretation. (a) The masks (Mask 0 attends to ghosts, Mask 1 to energy pills, and Mask 2 to dots) and bar plots are for a critical but non-rewarding scenario. For a full description of the scene, please refer to Fig. 12 in Appendex B.2.

(i.e., factors) and insufficient factors may exert a challenge in optimizing a policy, which may stem from many factors such as unstable Q-agent update. Thus, finding an appropriate disentanglement is deemed not straightforward in this case. The lower task return in Table 2 evidence our first observation.

We further propose more tractable and human-intuitive evaluation metrics to quantitatively gauge the attainment of the desired behaviour of masks. *Fidelity* computes as $\frac{\#(a^*=\hat{a}^*)}{\#(a^*)}$, measuring the consistency of decision a^* made with full state and the decision $\hat{a}^*$ with distilled state. *Sparsity* roughly measures the decrease of the information capacity (the lower the better) when the state is masked, computed as $\frac{|\bar{\alpha}^i|}{|s|}$. Finally, to approximately measure state inter-independency, we count the overlap of masks regarding *orthogonality*. (See Appendix C.6 for derivation and explanation.)

Comparing R-Mask masks in Fig. 4 and Q-Mask masks in Fig. 10 (rightmost two columns), though both deliver us a visual intuition that R-Mask attention masks are more distinct, more orthogonal, and void of spurious objects (Kulkarni et al., 2019; Wu et al., 2021). One explanation is that top-down attention (e.g., Q-Mask) is guided explicitly by the RL objective. As a result, the mask shaping becomes heavily tied to this objective, potentially causing the agent to link its rewarding behaviour with changes in displayed scores. This, in turn, can inevitably introduce bias in the causal relationship between state representation and chosen actions.

On the hypothesis that challenging tasks, especially the ones with a high-dimensional state, usually lead to unstable training and thus difficulties of distillation of causal factors, we further conduct experiments on a toy task, Monster-Treasure (Chevalier-Boisvert et al., 2018), where the ground

Table 2: Evaluations on Atari tasks. Metrics include fidelity (higher is better), sparsity (lower indicates sparser as desired), orthogonality (higher for better factor disentanglement), and task return performance.

Evaluation Metrics	Gopher				MsPacman			
	Q-Mask	Q-Mask Lite	R-Mask	R-Mask Lite	Q-Mask	Q-Mask Lite	R-Mask	R-Mask Lite
Fidelity	—	—	$\mathbf{84.58 \pm 0.64}\%$	$79.92 \pm 0.95\%$	—	—	$65.75 \pm 0.85\%$	$\mathbf{88.16 \pm 0.09}\%$
Sparsity	0.782	0.468	0.106	0.488	3.4e-4	0.826	0.435	0.932
Orthogonality	-0.24	5.63	9.43	2.8	27.42	41.06	-8.449	32.74
Performance	13.56 ± 2.58	12.17 ± 3.06	$\mathbf{14.54 \pm 2.04}$	12.48 ± 0.83	19.75 ± 0.11	$\mathbf{29.94 \pm 0.16}$	27.86 ± 0.59	29.65 ± 0.16

truth of causal factors are accessible and manageable for analysis. It turns out that, on tasks with low-dimensional states and easily disentangable causals, the Q-Mask shows better alignment with the ground truth than the R-Mask, which indicates that feeding the agent with distilled states helps both, reward prediction and state disentanglement. See Appendix C.7 for detailed case analysis.

Notably, while no definitive benchmarks exist for optimal orthogonality and sparseness, lower values are preferable, i.e., disentangled and sparse factors are favoured. In the process of learning masks, there exists a trade-off between sparsity and orthogonality. When contrasting evaluation results in Table 2 within the Gopher context, a notable trend emerges: a lower level of sparsity tends to correlate with a heightened degree of orthogonality. However, for the Pacman task, we observe the opposite pattern. Not surprisingly, depending on the specific RL task, the optimal balance of those desiderata may vary. Nevertheless, these indicators generally align with our perception of the generated explanations and prevent trivial and irrelevant causal factors from being learned. A comprehensive description of how the proposed desiderata contribute to our understanding of the agent's behaviour can be found in Appendix C.8. See case studies for details in Sec. 4.3 below.

4.3. Case studies

R-Mask Attention Masks on Gopher. We showcase attention masks learned by R-Mask in a critical scenario (Fig. 4). The agent's preference for the "LEFT" move over "LEFTFIRE" in a critical scenario is explained by a larger Q-value difference under the gopher reward component (see computation in Appendix C.13). This indicates that the agent is aiming for double rewards by moving left before executing a "UPFIRE" action when the gopher emerges, as supported by the analysis of attention masks provided by R-Mask (e.g., as the agent nears the object, Mask 0 and Mask 1 follow and contract). Note that attention masks adeptly capture subtle nuances in the two visually similar scenarios, which is crucial for understanding the agent's one-step action. Furthermore, the R-Mask method accurately predicts reward components in the scene, bolstering our confidence in explaining the agent's preference for "LEFTFIRE" through R-Mask's attention masks. For an in-depth case study, please refer to the Appendix C.9.

R-Mask Attention Masks on MsPacman. To further validate the ability of the proposed methods to mine the cause-effect relationships for more challenging environments when the reward causes are actually *interdependent*, we test R-Mask on the MsPacman environment (Q-Mask results are in Appendix B.2). The results in Fig. 5 indicate that the method can reveal the agent's decision-making rationale in challenging scenarios, but there are challenges when rewards are interdependent, affecting the accuracy of reward prediction. A more detailed explanation of this can be found in Appendix C.9.

5. Discussion

In this paper, we present a novel approach to unravelling the complex relationships between model predictions, the reasoning mechanism, and explanations in reinforcement learning. On top of the non-post-hoc RD approach, we introduce a causal model that identifies explanatory factors contributing to an agent's decisions, which differs from traditional saliency-based methods. The proposed framework provides a diverse perspective on the agent's interactions and can be integrated with policy-level explanations, such as that by Guo et al. (2021), to identify critical time steps and localize features for a deeper understanding of the agent's attention history.

Limitations. Our approach assumes the existence of multiple channels, which might not always hold. Challenges may arise when rewards are interdependent or tasks involve numerous reward components, potentially affecting computational efficiency. Achieving full invariance of factors through intervention to irrelevant task components may not always be feasible, particularly in complex tasks.

Outlook. Although we focus on the use of learned causal factors to generate explanations by visualizing factors, represented by various masks, the learned factors can also be used to generate counterfactual explanations — minimal perturbations of causal factors that change the agent's behaviour (Olson et al., 2021). Another promising but challenging future direction is relaxing the assumption of multiple rewards, i.e., exploring a more general setting without sub-rewards. This introduces a more complex expression for information flow, i.e., how various causal factors contribute to a *single* reward, but the same guiding desiderata would apply with some adjustments. The main challenge is assigning nontrivial meanings to factors when there exists one reward facet. However, learning causal factors could be enhanced through the auxiliary task of modelling dynamics, i.e., by utilizing environmental changes as extra supervision, learned factors may be more interpretable. Finally, techniques like LLMs, which can convey the aspects controlled by each factor to humans in language, would further improve explanation quality.

Acknowledgments

This research was funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) under the Federal Aviation Research Programme (LuFO), Projekt VeriKAS (20X1905)

References

Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *Adaptive Agents and Multi-Agent Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:21755369.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. 2019. URL https://arxiv.org/abs/1907.02893.

Akanksha Atrey, Kaleigh Clary, and David Jensen. Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep RL. In *International Conference on Learning Representations (ICLR)*, 2020.

Nihat Ay and Daniel Polani. Information flows in causal networks. *Advances in Complex Systems*, 11(01):17–41, 2008.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. *CoRR*, abs/2003.13350, 2020. URL https://arxiv.org/abs/2003.13350.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR, July 2018.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012. URL http://arxiv.org/abs/1207.4708.

Ioana Bica, Daniel Jarrett, Alihan Hüyük, and Mihaela van der Schaar. Batch inverse reinforcement learning using counterfactuals for understanding decision making. *CoRR*, abs/2007.13531, 2020. URL https://arxiv.org/abs/2007.13531.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL http://arxiv.org/abs/1606.01540.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning (ICML)*, pages 1779–1788. PMLR, 2020.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 598–617, 2016. doi: 10.1109/SP.2016.42.

DF Elliott and KR Rao. Fast fourier transform and convolution algorithms, 1982.

Bernhard C. Geiger and Gernot Kubin. On the information loss in memoryless systems: The multivariate case. *CoRR*, abs/1109.4856, 2011. URL http://arxiv.org/abs/1109.4856.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning (ICML)*, pages 1792–1801. PMLR, 2018.

Jixian Guo, Mingming Gong, and Dacheng Tao. A relational intervention approach for unsupervised dynamics generalization in model-based reinforcement learning. 2022. URL https://arxiv.org/abs/2206.04551.

Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. Edge: Explaining deep reinforcement learning policies. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 12222–12236. Curran Associates, Inc., 2021.

Piyush Gupta, Nikaash Puri, Sukriti Verma, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using focused feature saliency. *CoRR*, abs/1912.12191, 2019. URL http://arxiv.org/abs/1912.12191.

Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia P. Sycara. Transparency and explanation in deep reinforcement learning neural networks. *CoRR*, abs/1809.06061, 2018. URL http://arxiv.org/abs/1809.06061.

Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable Reinforcement learning via Reward Decomposition. In *IJCAI/ECAI Workshop on Explainable Artificial Intelligence*, 2019.

Tejas D. Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *CoRR*, abs/1906.11883, 2019. URL http://arxiv.org/abs/1906.11883.

Mengdi Li, Xufeng Zhao, Jae Hee Lee, Cornelius Weber, and Stefan Wermter. Internally rewarded reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 20556–20574. PMLR, 2023.

Wanyu Lin, Hao Lan, Hao Wang, and Baochun Li. Orphicx: A causality-inspired latent variable model for interpreting graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13729–13738, 2022.

Zichuan Lin, Derek Yang, Li Zhao, Tao Qin, Guangwen Yang, and Tie-Yan Liu. Rd2: Reward decomposition with representation disentanglement. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*. ACM, December 2020.

Wenhao Lu, Xufeng Zhao, Sven Magg, Martin Gromniak, Mengdi Li, and Stefan Wermterl. A closer look at reward decomposition for high-level robotic explanations. In *2023 IEEE International Conference on Development and Learning (ICDL)*, pages 429–436. IEEE, 2023.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8046–8056, 2022.

Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. *CoRR*, abs/1905.10958, 2019. URL http://arxiv.org/abs/1905.10958.

Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. A survey of explainable reinforcement learning. 2022. URL https://arxiv.org/abs/2202.08434.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. Causal interpretability for machine learning - problems, methods and evaluation. *SIGKDD Explor. Newsl.*, 22(1):18–33, may 2020. ISSN 1931-0145. doi: 10.1145/3400051.3400058. URL https://doi.org/10.1145/3400051.3400058.

Matthew L. Olson, Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong. Counterfactual state explanations for reinforcement learning agents via generative deep learning. *CoRR*, abs/2101.12446, 2021. URL https://arxiv.org/abs/2101.12446.

Matthew R. O'Shaughnessy, Gregory Canal, Marissa Connor, Mark A. Davenport, and Christopher Rozell. Generative causal explanations of black-box classifiers. *CoRR*, abs/2006.13913, 2020. URL https://arxiv.org/abs/2006.13913.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Judea Pearl, M Maria Glymour, and Nicholas P. Jewell. Causal inference in statistics: A primer. 2016. URL https://api.semanticscholar.org/CorpusID:148322624.

Leon N Piotrowski and Fergus W Campbell. A demonstration of the visual importance and flexibility of spatial-frequency amplitude and phase. *Perception*, 11(3):337–346, 1982.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Yunpeng Qing, Shunyu Liu, Jie Song, and Mingli Song. A survey on explainable reinforcement learning: Concepts, algorithms, challenges. 2022. URL https://arxiv.org/abs/2211.06665.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *CoRR*, abs/2102.11107, 2021. URL https://arxiv.org/abs/2102.11107.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

Yael Septon, Tobias Huber, Elisabeth André, and Ofra Amir. Integrating policy summaries with reward decomposition for explaining reinforcement learning agents. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 320–332. Springer, 2023.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning (ICML)*, pages 3145–3153. PMLR, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328. PMLR, 2017.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461, 2015. URL http://arxiv.org/abs/1509.06461.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461, 2015.

Xiaoxiao Wang, Fanyu Meng, Xin Liu, Zhaodan Kong, and Xin Chen. Causal explanation for reinforcement learning: Quantifying state and temporal importance. *Applied Intelligence*, pages 1–19, 2023.

C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Oxford, 1989.

Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. A causality inspired framework for model interpretation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2731–2741, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599240. URL https://doi.org/10.1145/3580305.3599240.

Haiping Wu, Khimya Khetarpal, and Doina Precup. Self-supervised attention-aware reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021.

Yan Zeng, Ruichu Cai, Fuchun Sun, Libo Huang, and Zhifeng Hao. A Survey on Causal Reinforcement Learning, June 2023. URL https://arxiv.org/abs/2302.05209.

Zheng-Mao Zhu, Shengyi Jiang, Yu-Ren Liu, Yang Yu, and Kun Zhang. Invariant action effect model for reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):9260–9268, Jun. 2022. doi: 10.1609/aaai.v36i8.20913. URL https://ojs.aaai.org/index.php/AAAI/article/view/20913.

Appendix A. Additional Results in Monster-Treasure Environment

A.1. Reward Estimate

Table 3 is referenced in the Experiments section (Sec. 4.2). The table documents the reward estimate corresponding to the state depicted in Fig. 16.

Table 3: Reward Predictions with the R-Mask

	Right	Down	Left	Up
r^0	2.288	0.28	0.287	0.312
r^1	-0.29	-0.262	-2.189	-0.295
sum	1.998	0.018	-1.902	0.017

A.2. Mask Scores

Given our knowledge of the ground truth masks in this environment, we depart from the metrics detailed in the Evaluation section (Sec. C.6). Instead, we capture the environment-specific mask score in Table 4. This score quantifies deviation from the ideal masks for this setting: one concealing monster information (i.e., coordinates) and another hiding treasure details. A lower score indicates better masks, with scores below 1 signifying effective masks.

Table 4: Mask Scores for Monster-Treasure Environment

	Mean	Standard Deviation
Q-Mask	0.507	0.302
R-Mask	1.913	1.133

A.3. Performance and Mask Accuracy Trade-off

The average return for Q-Mask stands at 1.97, contrasting with R-Mask's value of 2. Despite Q-Mask's precise mask generation, its performance has slightly declined compared to R-Mask. This observation can be attributed to the fact that all Q-agents within Q-Mask are exposed solely to a partial environmental view generated by learnable mask networks (e.g., updated by Eq. 3 and Eq. 4). Consequently, during the initial stages of mask learning, Q-agents might grapple with acquiring task-solving skills. This struggle could inadvertently lead to the erroneous filtering of both irrelevant and relevant information, possibly affecting task performance.

Appendix B. Additional Results in Atari Environments

B.1. Additional Results in Gopher

Fig. 6 is discussed in Sec. 4.2. We present examples that compare R-Mask (Q-Mask) with its lite variant in Figures 7, 8, 9, and 10. Additionally, Fig. 11 presents an in-depth illustration of Q-Mask attention masks for the Gopher environment.

(*a*)

(*b*)

Figure 6: Comparison of saliency maps (associated with ground and gopher rewards) of RD with RD-pred-u in a state where the agent filled the hole and attained reward 0.15.

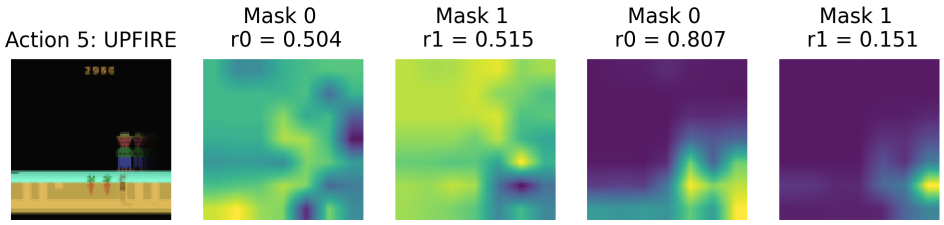


Figure 7: R-Mask Lite masks vs. R-Mask masks for a state with reward $r = 0.95$.

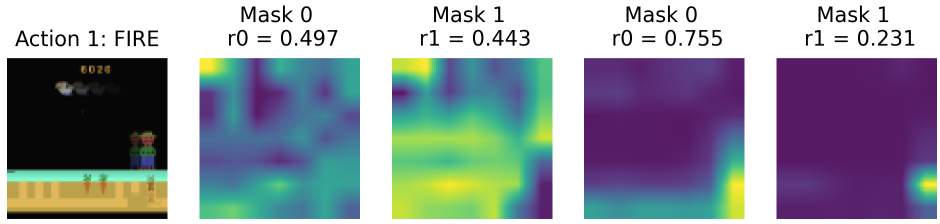


Figure 8: R-Mask Lite masks vs. R-Mask masks for a state with reward $r = 0.95$ (another example state where a flying bird recently passed by).

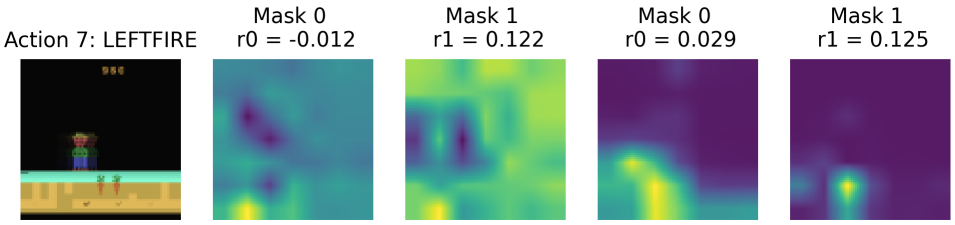


Figure 9: R-Mask Lite vs. R-Mask for a state with reward $r = 0.15$.

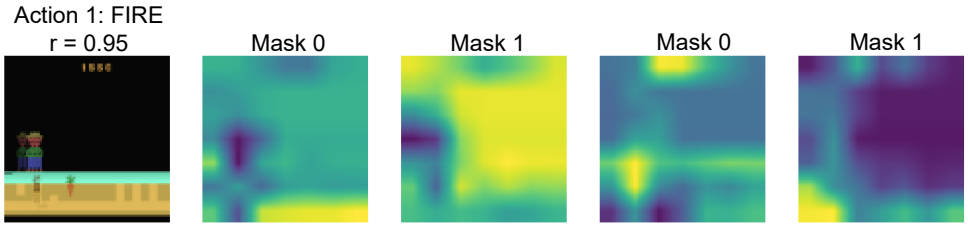
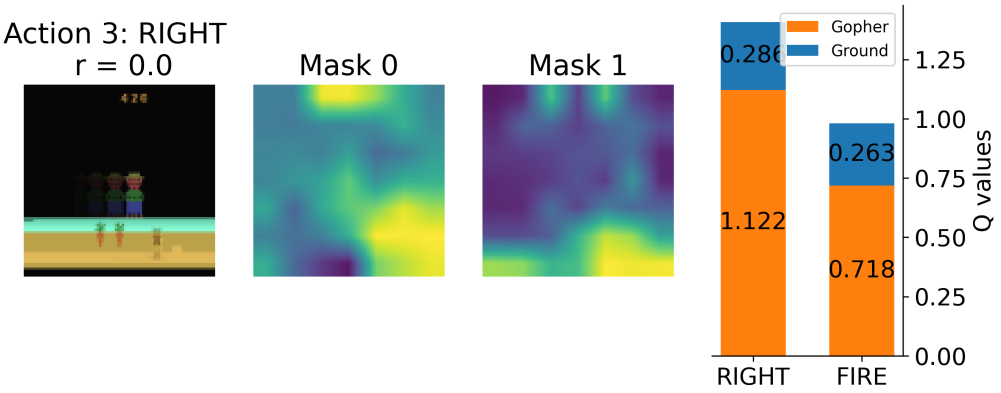
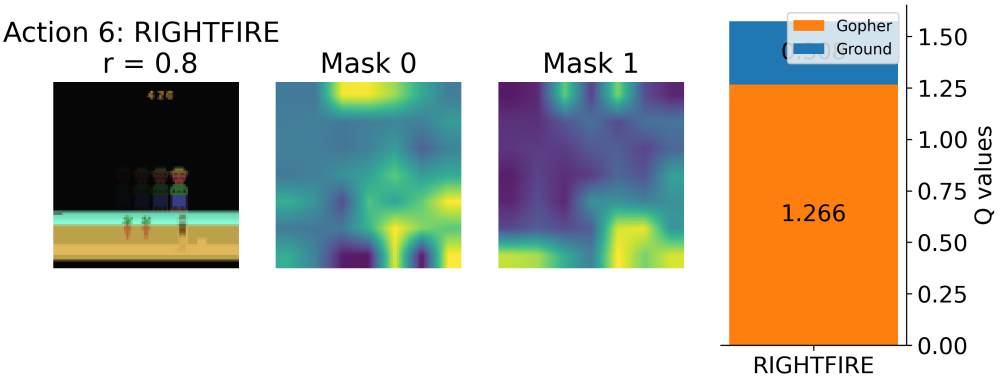


Figure 10: This figure depicts a rewarding state ($r = 0.95$), along with masks from Q-Mask Lite (first two) and Q-Mask (last two).



(*a*) Q-Mask masks for a state

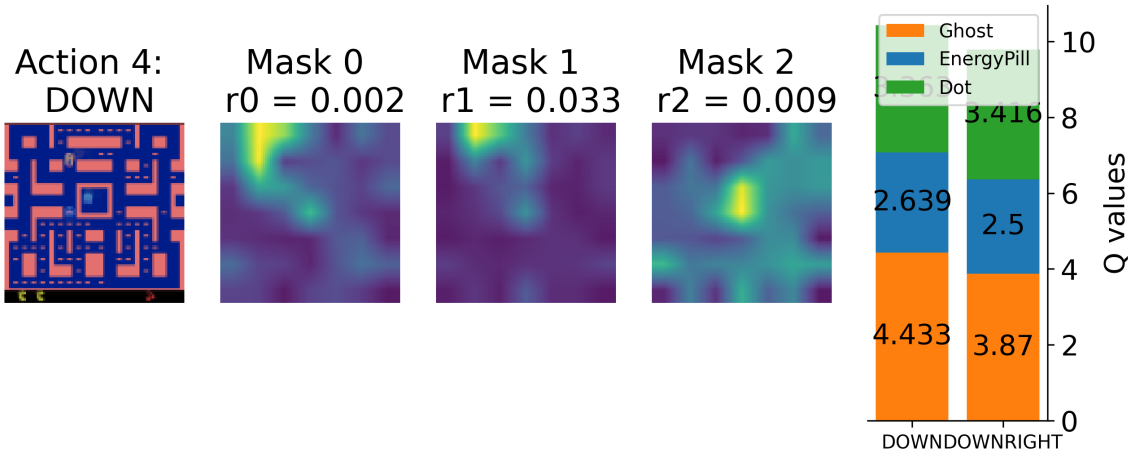


(*b*) Q-Mask masks for the **next** state

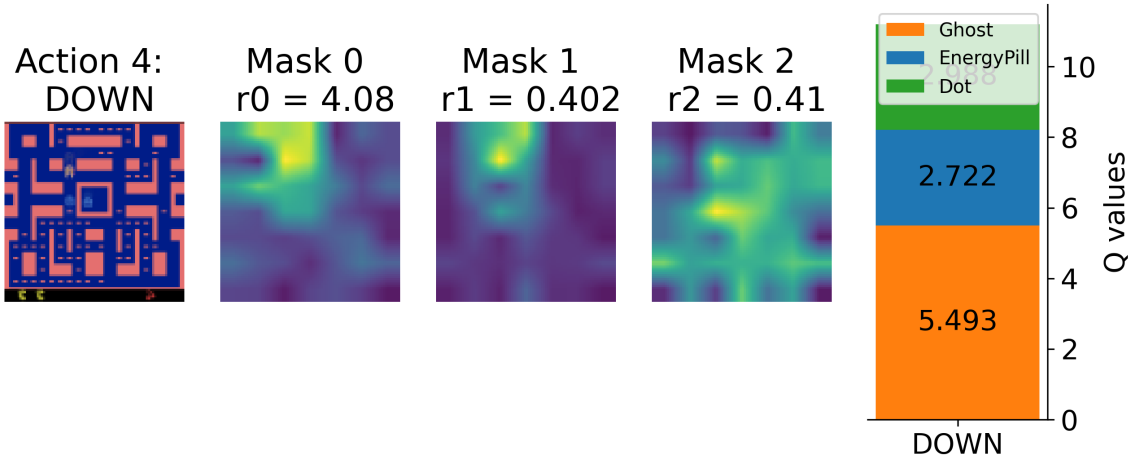
Figure 11: Q-Mask attention masks from Gopher and their interpretation. (a) The masks (Mask 0 represents attention to the gopher while Mask 1 to ground) and bar plots are for a scenario (critical state with no reward), where there is a large Q-value gap between a chosen "RIGHT" move and a second-best "FIRE" action. The agent's choice to opt for a "RIGHT" move rather than a "FIRE" action as the gopher emerges from its hole is visually unclear. However, a closer examination of the following state (11(*b*)) and the contracting attention masks (particularly areas at the bottom-right) exposes the gopher's strategy. It plans to "RIGHTFIRE" after moving right, intentionally aiming for a collision and a reward.

B.2. Additional Results in MsPacman

Fig. 12 is referenced in the Evaluation section (Sec. 4.3) when introducing R-Mask attention masks in the MsPacman environment. Furthermore, Fig. 13 presents another illustrative example of R-Mask masks designed for the MsPacman environment. In addition to these, we showcase an instance of Q-Mask masks in Fig. 14.

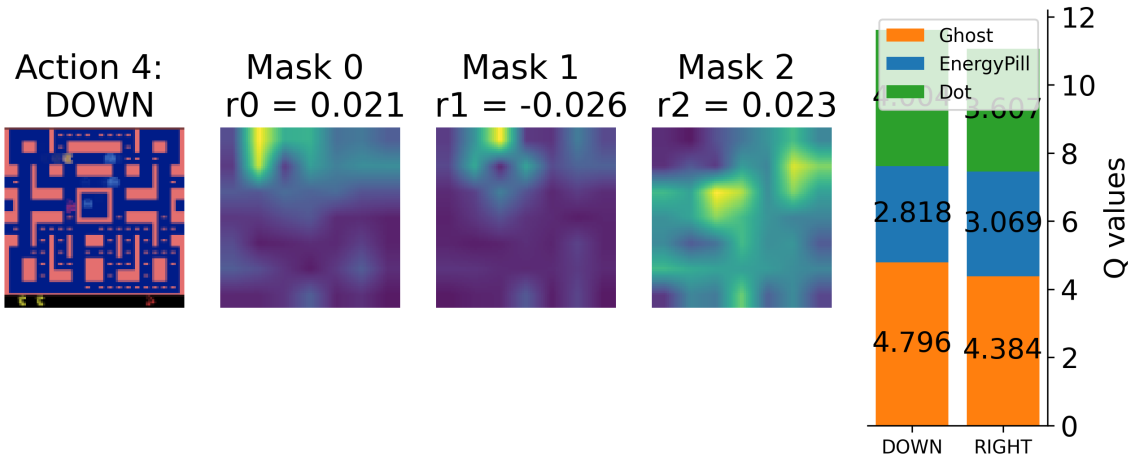


(*a*) R-Mask masks for a state with reward $r = 0$.

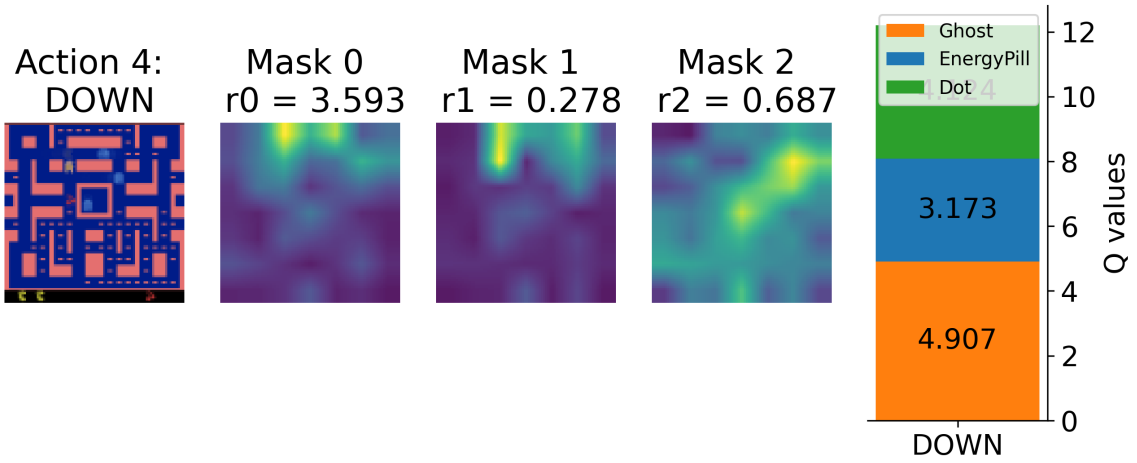


(*b*) R-Mask masks for the **next** state with reward $r = 5$.

Figure 12: R-Mask attention masks from MsPacman and their interpretation. (a) The masks (Mask 0 attends to ghosts, Mask 1 to energy pills, and Mask 2 to dots) and bar plots are for a critical but non-rewarding scenario. Positioned at the top-left crossroad of the maze, the Pacman faces an imminent encounter with a ghost. In this state (Fig. 12(*a*)), the agent can select a "DOWN" move instead of a risky "DOWNRIGHT" action, evading the ghost. By examining the subsequent state and attention masks (12(*b*)), particularly the upper-left region, the Pacman's intention becomes evident. Detecting the ghost, the Pacman executes a "DOWN" move, causing a collision and thereby yielding a reward.

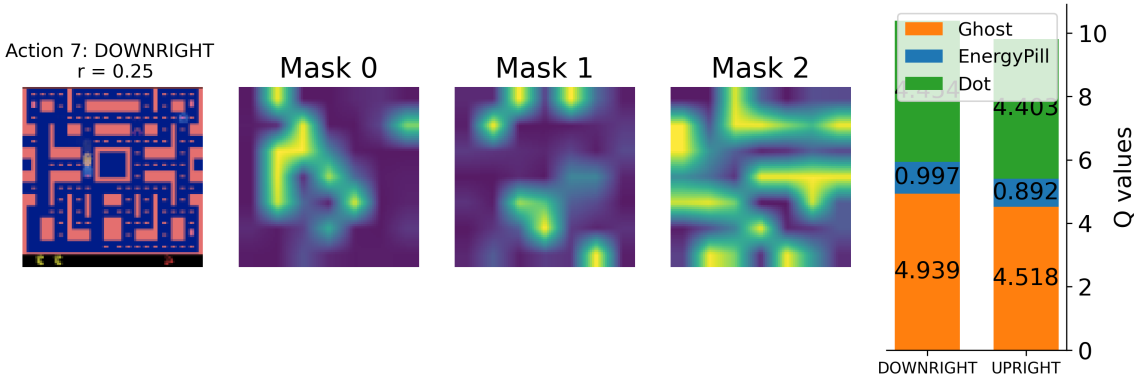


(*a*) R-Mask masks for a state with reward $r = 0$.

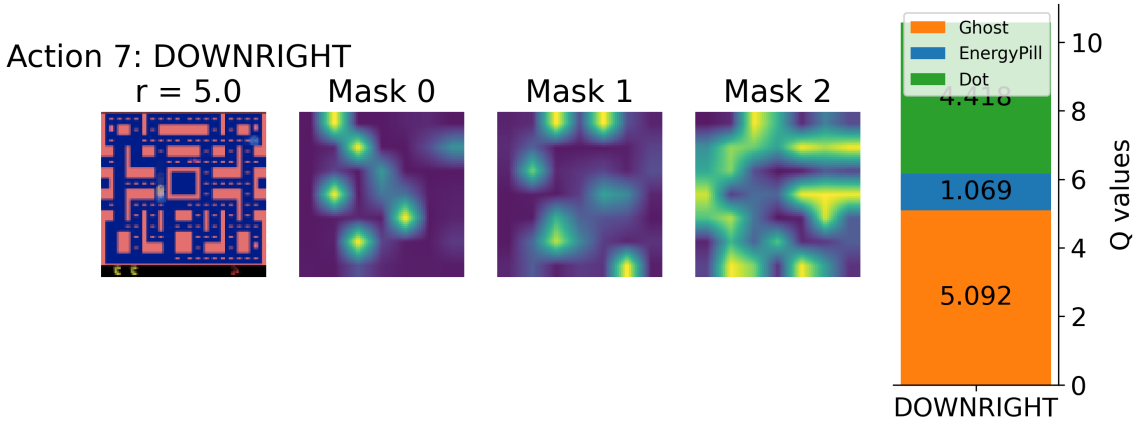


(*b*) R-Mask masks for the **next** state with reward $r = 5$.

Figure 13: Another R-Mask mask in MsPacman environment.



(*a*) Q-Mask masks for a state.



(*b*) Q-Mask masks for the **next** state.

Figure 14: Q-Mask attention masks from MsPacman and their interpretation. (a) The masks (Mask 0 attends to ghosts, Mask 1 to energy pills, and Mask 2 to dots) and bar plots are for a critical and rewarding scenario. As Pacman progresses downward within the middle-left maze area, it consumes a dot while simultaneously encountering a ghost. In this situation (depicted in Fig. 14(*a*)), the agent selects a "DOWNRIGHT" move over an "UPRIGHT" action, which would involve passing the ghost. An analysis of the subsequent state and attention masks (14(*b*)) exposes the Pacman's strategy. Recognizing the ghost, the Pacman continues its downward movement, resulting in a collision with the ghost and the subsequent reward.

B.3. Convergence in Agent's Learning

As a reference for convergence (of episodic scores/returns for Atari environments), we put the statistics of scores (after 10M environment steps) in Table 5, in which we compare human baseline, common DQN approach, vanilla RD, and our approaches (extensions of RD) for games Gopher, MsPacman. Note this table may only serve as a rough comparison as some methods are evaluated under different conditions, e.g., different neural network architectures, hyperparameters, and learning steps. Note, DDQN* and RD* represent our implementations of the DDQN and RD algorithms, respectively.

Table 5: Scores at Convergence in Different Atari Games

Games	**Average Human** Badia et al. (2020)	**DDQN** van Hasselt et al. (2015)	**DDQN***	**RD***	**R-Mask (our)**	**Q-Mask (our)**
Gopher	2412.5	8742.8	8338	7881.3	8671.3	8078.8
MsPacman	6951.6	1401.8	6132	5810	6961.8	5818.4

Appendix C. Full Reference to Main Text

C.1. Formalization of RL Problem with SCM

We formalize the RL problem with the following structural causal model (SCM):

$$\mathcal{S} := f_{\mathcal{S}}(\alpha, \beta, U_{\mathcal{S}}),\ \mathcal{A} := f_{\mathcal{A}}(\alpha, U_{\mathcal{A}}),\ \mathcal{R} := f_{\mathcal{R}}(\alpha, \mathcal{A}, U_{\mathcal{R}}), \tag{5}$$

where noise variables are jointly independent: $U_{\mathcal{S}} \perp U_{\mathcal{A}} \perp U_{\mathcal{R}}$. As for $f_{\mathcal{S}}, f_{\mathcal{A}}, f_{\mathcal{R}}$, they are unknown structural functions; $f_{\mathcal{A}}$ can be regarded as the policy to be learned and causal factors α can be obtained by a masker $m(\cdot)$ which we will detail in the main text.

C.2. Computing Causal Intervention

Formally, given an environment state s, its Fourier transformation is expressed in $\mathcal{F}(s) = A(s) \times \exp^{-j \times P(s)}$, where $A(s)$, $P(s)$ denote the amplitude and phase components, respectively. The Fourier transformation $\mathcal{F}(\cdot)$ and its inverse $\mathcal{F}^{-1}(\cdot)$ can be calculated with the FFT algorithm (Elliott and Rao, 1982) effectively. Following the practice in Lv et al. (2022): we intervene the amplitude by linearly interpolating between the amplitude of the original state s and a state s' sampled randomly from a set which contains states where the non-causal factors have been removed (For Atari games, it is the displayed scored removed):

$$\hat{A}(s) = (1 - \lambda) * A(s) + \lambda * A(s'), \tag{6}$$

where $\lambda \sim U(0, \epsilon)$ and ϵ adjusts the magnitude of intervention. Then we combine the perturbed amplitude with the original phase component to generate the intervened state s^{inter} by inverse Fourier transformation: $\mathcal{F}(s^{inter}) = \hat{A}(s) \times \exp^{-j \times P(s)}$, $s^{inter} = \mathcal{F}^{-1}(\mathcal{F}(s^{inter}))$.

C.3. Clarifications: Causal Factors and Multi-task RL

C.3.1. RELATIONSHIPS BETWEEN SUBSETS OF CAUSAL FACTORS

As is depicted in Fig. 2, we expect causal factors to be as independent as possible. However, overlapping (between $\bar{\alpha}^i$ and $\bar{\alpha}^j$ subjecting to similar sub-tasks) is inevitable in many cases. For example, in the Monster-Treasure toy case that we study (see details in Appendix C.7), the agent receives a reward for reaching the treasure but incurs a penalty for landing on the monster; the agent (ego) becomes the overlapping part. Indeed, we use orthogonality and derive an objective function to encourage the subsets of causal factors ($\bar{\alpha}$) to be independent (if possible).

C.3.2. CONNECTION OF RD TO MULTI-TASK RL

Similarity:

- Both RL with RD and multi-task RL contain the setting where there are multiple rewards (functions) from which corresponding policies can be learned.

Dissimilarity:

- RL with RD assumes the additivity of reward ($r = \sum_i r_i$), hence, we learn a global policy which is the summation of component policies (i.e., Q-function associated with each reward component). However, there is no such constraint toward the relation of reward functions designed for each task in the multi-task setting.

- In RL with RD setting, we learn all component policies in parallel, however, in multi-task RL, a single policy is generally sequentially updated across a sequence of tasks one by one.

Regarding the point of utilizing multiple reward channels to learn a multi-task RL agent, we believe it is a direction worth exploring. In RL with RD, the rollout is made by a global action which is derived from all component policies. However, when adapting multi-task RL to the setting of multiple reward channels, it may raise a further discussion about which rollout mechanism to employ (to collect trajectories) as it lacks a "global task" or "global policy" as we have in the RD setting. The intuitive way, for example, is to randomly choose the i-th task policy ($\pi(\cdot|s, z_i)$) to do the trajectory collection, and then use these trajectories to update other task policies, i.e., $\pi(\cdot|s, z_j)$, $j \neq i$ with offline RL techniques.

C.4. A Full List of Methods Used in Experiments

Table 6 lists all methods used in the experiments. In Q1, we compare RD with RD-pred to assess the impact of the auxiliary task of reward decomposition on the generation of explanation artefacts. Q2 involves a comparison between RD-pred-u and R-Mask, exploring the value of causal sufficiency of reward components. Q3 delves into the role of causal sufficiency concerning actions, comparing Q-Mask with R-Mask and RD-pred. Lastly, in Q4, we contrast R-Mask and Q-Mask with their Lite versions to elucidate the role of our proposed explanation criteria in learning disentangled, sparse causal factors.

Table 6: The list of methods studied in experiments with varying learning features, encompassing aspects such as *decomposing reward* (with full state or masked state factors), *Q-agent learning* (with full state or masked state factors), *knowledge of sub-reward values* in reward prediction (if applicable) and Q-learning, and *the use of proposed desiderata* in factor learning. For example, the RD-pred method involves reward prediction and Q-agent learning with full state factors, and known sub-rewards, but it does not incorporate desiderata. RD, on the other hand, differs from RD-pred by not including reward prediction.

Method	reward prediction r^i		Q-value estimate Q^i		known sub-rewards	desiderata losses
	full state	sub-state	full state	sub-state		
RD	—	—	✓	✗	✓	✗
RD-pred	✓	✗	✓	✗	✓	✗
RD-pred-u	✓	✗	✓	✗	✗	✗
Q-Mask	—	—	✗	✓	✓	✓
Q-Mask Lite	—	—	✗	✓	✓	✗
R-Mask	✗	✓	✓	✗	✗	✓
R-Mask Lite	✗	✓	✓	✗	✗	✗

C.5. Deep Q-learning and Reward Decomposition

One of the fundamental approaches to learning the policy π for an MDP involves initially acquiring knowledge about an action-value function Watkins (1989). This function encapsulates the anticipated cumulative discounted reward when the agent executes action a_t within state s_t and subsequently adheres to policy π in the future. Formally, it can be expressed as $Q(s_t, a_t) = \mathbb{E}_\pi[\, r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$, where γ denotes the discount factor. By determining the maximum value within the action-value function, an estimation of the optimal policy can be derived as $\hat{\pi}^* = \arg\max_{a_t} Q(s_t, a_t)$. Building upon the framework of deep Q-learning Mnih et al. (2015), we approximate the value function Q_ϕ using a neural network-based function approximator that is parameterized by ϕ. These parameters ϕ are iteratively refined by minimizing the loss function

$$J(\phi) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}}[(r_t + \gamma Q_{\phi'}(s_{t+1}, \arg\max_{a_{t+1}} Q_\phi(s_{t+1}, a_{t+1})) - Q_\phi(s_t, a_t))^2].$$

In this context, $Q_{\phi'}$ denotes a target network, periodically synchronized with the main network Q_ϕ to stabilize learning van Hasselt et al. (2015).

When there are multiple reward components, we adopt a collection of $K \in \mathbb{N}$ Q-functions, each guided by an individual component r^i. The optimal (global) action a_t^* corresponding to a state s_t is identified as the one with the highest Q-value obtained by aggregating the Q-functions from all K components Q_{ϕ^i}, expressed as $a_t^* = \arg\max_{a'_t} \sum_{i=1}^K Q_{\phi^i}(s_t, a'_t)$.

C.6. Evaluation Metrics for Explanations

Fidelity. To assess the faithfulness of explanations objectively, we calculate the *fidelity* of the causal information transferred into the Q-agent, measured by the approximate information loss (see Sec. 3.3) $\mathcal{L}[Q(a_t|s_t) \to Q(a_t|\bar{\alpha}_t^i)] = \mathcal{H}[Q(a_t|s_t)|Q(a_t|\bar{\alpha}_t^i)]$, i.e., the ability to make *consistent* decisions when depending on the masked state (causal factor). The information loss (upper

bound) can be measured as $\mathbb{E}\log p(a_t^*|\hat{a}_t^*) \leq \log \mathbb{E}p(a_t^*|\hat{a}_t^*) \approx \log \frac{\#(a^*=\hat{a}^*)}{\#(a^*)}$, which is the accuracy of directly estimating the full state decision $a^* = \arg\max_a \sum_i Q^i(a|s)$ with a distilled state $\hat{a}^* = \arg\max_a \sum_i Q^i(a|\bar{\alpha}^i)$, computed by counting (#) the consistency.

Sparsity. As the attention mask acts as an explanation artefact, it must be sufficiently obvious that users can appreciate it. Thus, sparse but distinct masks are preferred over dense ones (i.e., masks of value 1) for explanation purposes. For the evaluation of *sparsity*, it involves a measure of information loss (the higher the better for sparsity) and information independence of sub-states. The information loss can be approximately measured as the decrease of the information capacity (the lower the better) when the state is masked, i.e., $\mathcal{L}(s \to \bar{\alpha}^i) \approx \mathcal{H}(\bar{\alpha}^i) \approx \mathbb{E}\frac{|\bar{\alpha}^i|}{|s|}$.

Orthogonality. For the benefit of interpretability, it is expected to obtain diverse attention masks each associated with a reward component, instead of all attention masks collapsing into a single mask. For the *orthogonality* among states, we roughly evaluate their inter-independency as $I(\bar{\alpha}^i;\bar{\alpha}^j) = \mathcal{H}(\bar{\alpha}^i)+\mathcal{H}(\bar{\alpha}^j)-\mathcal{H}(\bar{\alpha}^i;\bar{\alpha}^j) \approx \frac{1}{|s|}\mathbb{E}(|\bar{\alpha}^i|+|\bar{\alpha}^j|-|\bar{\alpha}^i \cap \bar{\alpha}^j|)$, i.e., the overlap of masks.

C.7. Monster-Treasure Toy-case

This simple 2D mini-grid environment (Fig. 16), initially introduced by Chevalier-Boisvert et al. (2018), features a 4×4 grid hosting an agent with four possible movement directions, alongside a randomly spawned monster and treasure in each episode. The agent receives a reward $r^0 = 2$ for reaching the treasure's grid cell (goal) but incurs a $r^1 = -2$ penalty for landing on the monster's cell (i.e., $K = 2$). The state includes the x- and y-coordinates of the agent, monster, and treasure, while the action space is going up, down, left and right.

To gain further insight into why R-Mask outperforms Q-Mask in generating high-quality masks (quantitatively and qualitatively) and determine whether this observation is coincidental, we evaluate them in a simplified scenario where we have complete access to ground truth causal factors for each sub-reward.

We depict the mask results learned by both Q-Mask and R-Mask methods in Fig. 15. It can be observed that mask values in Q-Mask gradually converge to optimal values, where the optimal monster mask is $\{1, 1, 1, 1, 0, 0\}$, i.e., estimated sub-state s^{monster} = { agent_x, agent_y, monster_x, monster_y } under reward r^1, and the optimal treasure mask is $\{1, 1, 0, 0, 1, 1\}$ for r^0. However, R-Mask has difficulty distilling accurate sub-states, e.g., non-zero mask values for monster coordinates in the treasure mask 0.

In the depicted state (Fig. 16), under Q-Mask, moving right yields the highest full Q-value (blue and light blue bars) for both rewards, while moving left (colliding with the monster) results in the lowest values. Each Q-agent in Q-Mask correctly focuses on its sub-state estimate when the agent chooses to move right toward the treasure cell. Instinctively, the agent's decision in that state is deemed trustworthy. In R-Mask, although imprecise masks are learned, when presenting the reward component estimates for a state under various actions in Table 3 in Appendix A, we observed the agent accurately estimating rewards. For instance, an estimate near 2 for a right move and close to -2 for a left move indicates trustworthy decision-making, favouring a right action.

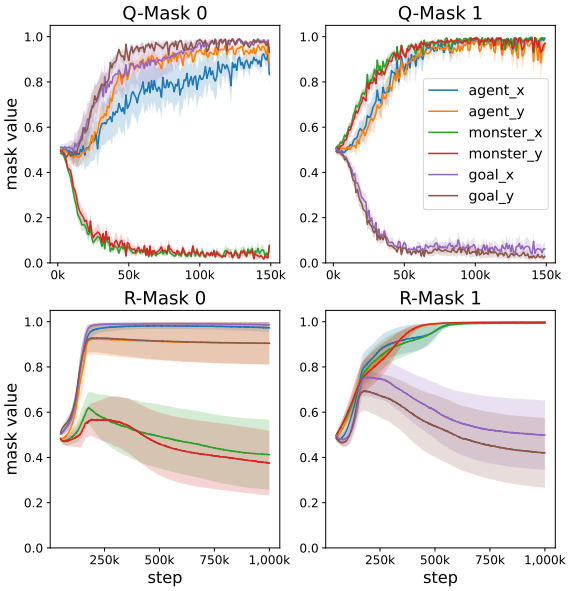


Figure 15: Masks for the Monster-Treasure environment generated by Q-Mask and R-Mask. The plot shows the mean and standard error of ten runs. For R-Mask, the masks have been manually ordered so that mask 0 attends more to the treasure and mask 1 more to the monster.

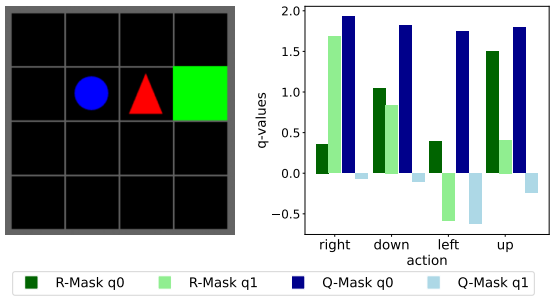


Figure 16: Example state for the Monster-Treasure environment with corresponding Q-values and reward predictions (Table 3). The agent (red arrow) is between the monster (blue circle) and the treasure (green square), and its choice is to move right. The component Q-values and component rewards add up to the full Q-values and the anticipated reward for each action.

C.8. Comparing R-Mask and Q-Mask with Their Lite Versions

R-Mask (Q-Mask) distinguishes itself from its lite version by incorporating explicit desiderata for exploring causal factors. The proposed indicators typically align with our perception of the generated explanations. Judging by the attention mask quality (e.g., Fig. 7, Fig. 8, Fig. 9, Fig. 10 in Appendix B.1), it becomes evident that R-Mask (Q-Mask) achieves a more favourable balance between these desiderata when contrasted with masks generated by their Lite versions, without the use of additional desiderata losses. For instance, in Fig. 10, the efficacy of Q-Mask's mask creation is evident: Q-Mask's Mask 0 highlights the agent's interaction with the gopher, while Mask 0 in Q-Mask Lite misses this. Similarly, Mask 1 in Q-Mask avoids irrelevant areas, such as the sky, unlike that in Q-Mask Lite which is less interpretable. Thus we cannot reliably trust them for explanation purposes. Another illustrative instance arises when comparing R-Mask Lite to R-Mask. Despite R-Mask Lite exhibiting superior fidelity scores compared to R-Mask, it generates masks that are dense and closely resembling one another (resulting in a high sparsity score of 0.932 and a substantial orthogonality score of 32.74).

Masks created by R-Mask for Gopher environment exhibit a relatively high fidelity score[6] and low sparsity score, indicating that ample but sparse information about the state is retained. This information proves predictive of both the agent's subsequent reward and its choice of action. However, in the MsPacman environment, R-Mask demonstrates lower fidelity. Given the intricate dynamics

6. Achieving a fidelity score of 100% can be readily demonstrated by setting masks to 1, yet it fails to be sparse.

within MsPacman, including multiple moving characters (such as enemies) with which the agent must interact, as well as more reward sources ($K = 3$), the process of rendering masks interpretable in MsPacman may encounter challenges.

C.9. Case Studies

Two case studies are presented to demonstrate how diverse causal factors (attention masks) enhance our understanding of the agent's behaviour. We acknowledge that some conclusions are drawn from our subjective assessment of the generated explanations, and further refinement through a user study is a future consideration. Nonetheless, we leverage these case studies to illustrate how attention masks align with our expectations regarding the rationale behind the agent's actions.

For each scenario, we depict two examples of masks, juxtaposed for comparison. To understand the scenario the agent experienced and the masks correspond to, we overlay 4 consecutive (RGB) states by plotting each state with low transparency over one another. Thus, it is clear to see what each scenario represents. The first scenario adheres to the critical state criterion, while the subsequent one illustrates the following state.

R-Mask Attention Masks on Gopher. We showcase attention masks learned by R-Mask in a critical scenario (Fig. 4). To elaborate on why the agent prefers the "LEFT" move over the action "LEFTFIRE" at the scene, we first adopt reward difference explanation (RDX) as in Juozapaitis et al. (2019) to gain insight into the Q-value difference between the two actions under reward components gopher and ground, based on the bar plot of Q-values (rightmost in Fig. 4; detailed computations in Appendix C.13). RDX indicates moving left is preferable to the "LEFTFIRE" action due to a larger Q-value difference under the gopher reward component. This underscores the association between moving left and the presence of the gopher. Though it gives us the *plain* reason, the diverse attention masks provided by R-Mask visually complement it and a broad look at Mask 0 (to the gopher and the agent) and Mask 1 (to the ground) gives us a visual intuition of what's going on. Mask 0 stays focused on the gopher and agent jointly, and as the agent nears the object, Mask 0 and Mask 1 follow and contract, as depicted in Fig. 4(*b*). This supports our hypothesis: the agent aims for double rewards through a sequence of actions: sprinting to the left before a "UPFIRE" action[7].

Notice the visual similarity between the two consecutive scenarios in Fig. 4, with negligible pixel changes. Despite this, attention masks for each component adeptly capture and visually reflect subtle nuances, which is essential for understanding the agent's one-step actions. This property holds for Q-Mask as well (see examples in Fig. 11).

Beyond attentive masks, the R-Mask method accurately predicts reward components r_i in the Gopher environment. For instance, in Fig. 4(*b*), Mask 0 attends to the gopher and agent, predicting 0.827 (close to 0.8 actual value), while Mask 1 focuses on the ground, predicting 0.199 (close to 0.15 actual value). This reliability enables explaining the agent's preference for "LEFTFIRE" using R-Mask's attention masks.

R-Mask Attention Masks on MsPacman. To further validate the ability of the proposed methods to mine the cause-effect relationships for more challenging environments when the reward causes are actually *interdependent*, we test R-Mask on the MsPacman environment. Examining a critical scenario as depicted in Fig. 12, Mask 0 significantly highlights Pacman and the blue ghost underneath, expanding as they converge. Hence, it visually reveals the rationale for the agent's

7. As the gopher prepares to emerge from its hole, and the agent is above, executing a "UPFIRE" or "FIRE", creating a chance for a double reward.

downward movement choice. Notably, in experiments, Mask 0 and Mask 1 often exhibit similarity, possibly due to the interplay between "Ghost" and "EnergyPill" rewards, where "Ghost" activation (i.e., is received) follows "EnergyPill" activation. This inter-dependency between causal factors violates our assumption of additivity, making it challenging to decouple them from current learning objectives. However, the other causal components are still able to be extracted by the method.

Overall, we noticed a relatively low accuracy in predicting the reward for eating dots, possibly due to their significant magnitude difference (e.g., 0.25 vs. 5). Sparse and compact masks for this component were also rare, likely because of the dispersed dot distribution across the maze, making distinct masks less likely to appear (e.g., Mask 2 in Fig. 12(*b*)).

C.10. Adapting Causal Learning Across RL Domains: Text, Sound, and Tabular Data

The adaptation process should be straightforward when dealing with reinforcement learning applications featuring multiple reward channels. It involves two key steps: 1. learning a causal representation (i.e., factors) of the raw state, separated from non-causal factors accessible through domain knowledge, and 2. developing maskers to selectively retrieve subsets of learned causal factors associated with each reward component.

As an illustration, consider applications involving auditory data. Initially, raw auditory data can be transformed into a spectrogram using a Short-Time Fourier Transform (STFT). Next, a neural network (NN) can be employed to extract a latent representation from it. Subsequently, our explanation approach can be applied.

C.11. Details in the Implementation of Evaluation Metrics

C.11.1. The Choice of Critical State.

The selection of the critical state hinges on the criterion that the highest Q-value surpasses the second-highest Q-value by either 10% or 15%.

C.12. Details of Neural Network Architecture and Hyperparameters

C.12.1. Training Flow.

The training flow for R-Mask is illustrated in Fig. 17, while the training flow for Q-Mask is illustrated in Fig. 18.

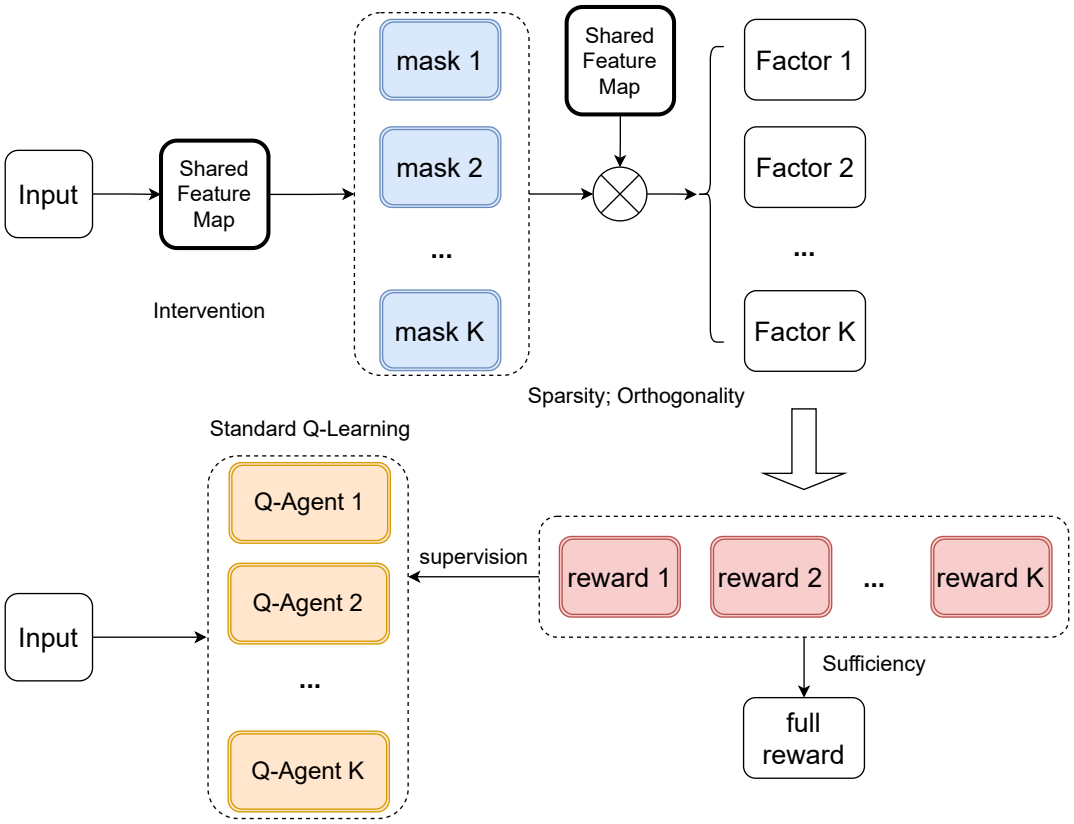


Figure 17: Training Flow in R-Mask: The neural modules (mask, Q-agent, and reward) to be learned are depicted by double-rounded rectangles, while the shared feature map (to be learned) upon which mask modules are constructed is represented by a bold-rounded rectangle. Input is channelled through all K mask modules, resulting in decomposed states. Subsequently, each reward module processes a decomposed state, generating a corresponding reward estimate. This yields a total of K reward estimates, denoted as r_{θ^i}. These estimates then serve as supervision signals, facilitating the update of each Q-function within the Q-agent module.

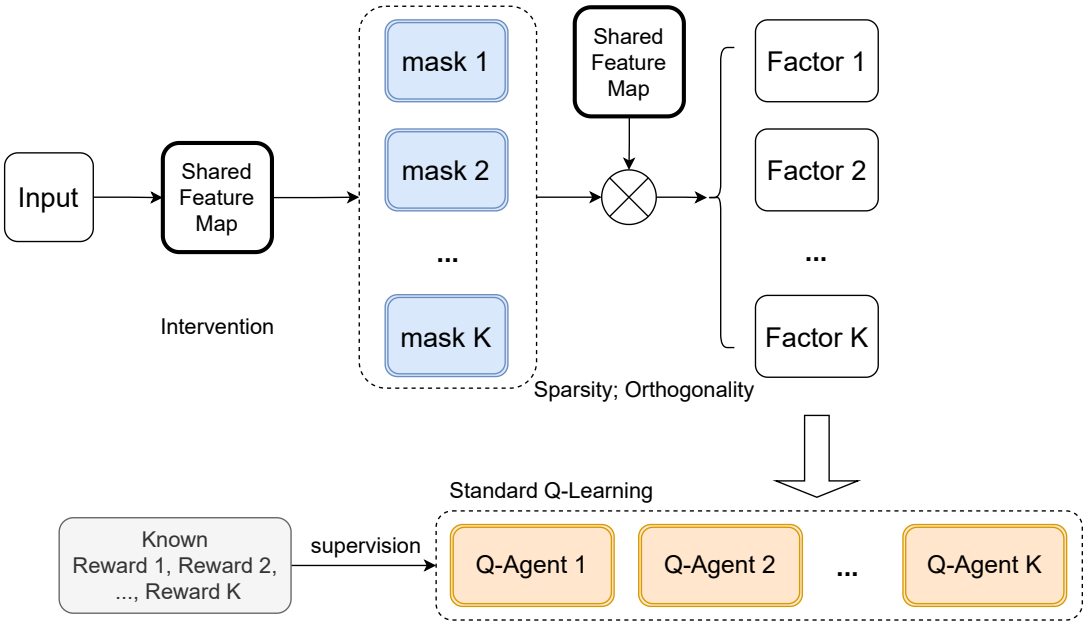


Figure 18: Training Flow in Q-Mask: The neural modules (mask and Q-agent) to be learned are depicted by double-rounded rectangles, while the shared feature map (to be learned) upon which mask modules are constructed is represented by a bold-rounded rectangle. The input is routed through all K mask modules, generating decomposed states. Each Q-agent module then takes a decomposed state as input and is supervised by the corresponding ground truth reward component.

C.12.2. SHARED FEATURE MAP.

Both R-Mask and Q-Mask share a feature map structure depicted in Fig. 19. This structure comprises Conv-ReLU blocks with the following specifications: 1) Stride 4, 8×8 with 32 filters; 2) Stride 2, 4×4 with 64 filters; 3) Stride 1, 3×3 with 64 filters.

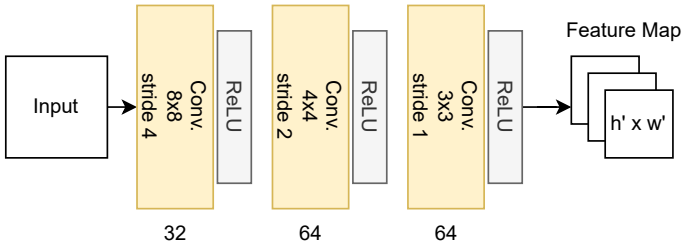


Figure 19: Conv-ReLU blocks in shared feature map (Conv: convolutional layer) in Fig. 17.

C.12.3. MASK MODULE.

Each mask module follows a pattern as demonstrated in Fig. 20. This pattern encompasses Conv-ReLU blocks (the same as in Fig. 19) in conjunction with a 1×1 Conv layer, which produces the attention mask.

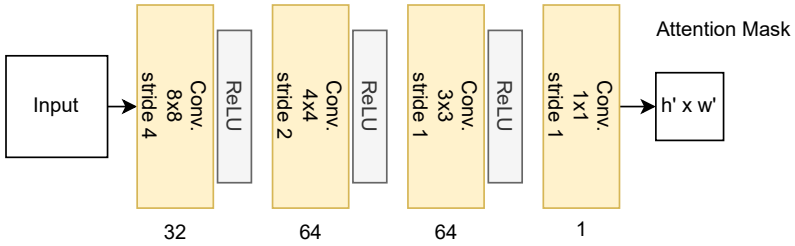


Figure 20: Conv-ReLU blocks in mask module (Conv: convolutional layer) in Fig. 17. A single 1×1 convolutional layer is employed to generate the attention mask output.

C.12.4. HYPERPARAMETERS.

For Monster-Treasure and Atari environments, we choose to use Adam with a learning rate of $6.25e-5$ to update Q-functions, reward prediction networks and mask networks. Table 7 lists the hyperparameters we use across all Atari games. The update frequencies n_1, n_2, n_3, n_4 are referred to in Algorithm 1 and Algorithm 2, with the specific values being: $n_1 = 20$, $n_2 = 100$, $n_3 = 20$, and $n_4 = 20$. For the Monster-Treasure environment, we use $n_1 = 4$, $n_2 = 16$, $n_3 = 4$, and $n_4 = 4$. We run all experiments on a single GPU RTX 2080 Ti.

C.13. Details in Computing Reward Decomposition Explanation (RDX)

Section 4.3 introduces RDX when explaining the agent's preference for the "RIGHT" move over the "FIRE" action in Fig. 11. The computation of RDX is outlined as follows:

Table 7: Preprocessing steps and hyperparameters

Parameter	Values
Image Width	84
Image Height	84
GrayScaling	Yes
Action Repetitions	4
Batch Size	32
Learning Rate	$6.25e-5$
Discount Factor	0.95

For any pair of actions, say a_1 and a_2, the difference in Q-values between the two actions under each component is represented as $\Delta_i(s, a_1, a_2) = Q_{\phi^i}(s, a_1) - Q_{\phi^i}(s, a_2)$. RDX serves as a quantitative measure, indicating the advantage or disadvantage of action a_1 compared to action a_2 under each component.

Considering Fig. 4, we define a_1 as "LEFT" and a_2 as "LEFTFIRE". The Q-values are computed as follows: $Q(s, \text{LEFT}) = Q_{\text{Gopher}}(s, \text{LEFT}) + Q_{\text{Ground}}(s, \text{LEFT}) = 0.882 + 0.683$, and $Q(s, \text{LEFTFIRE}) = Q_{\text{Gopher}}(s, \text{LEFTFIRE}) + Q_{\text{Ground}}(s, \text{LEFTFIRE}) = 0.486 + 0.73$.

Under the Gopher reward component, we find $\Delta_{\text{Gopher}} = Q_{\text{Gopher}}(s, \text{LEFT}) - Q_{\text{Gopher}}(s, \text{LEFTFIRE}) = 0.396$. Under the Ground reward component, $\Delta_{\text{Ground}} = Q_{\text{Ground}}(s, \text{LEFT}) - Q_{\text{Ground}}(s, \text{LEFTFIRE}) = -0.047$. As $\Delta_{\text{Gopher}} \geq \Delta_{\text{Ground}}$, the agent's decision to move left rather than doing leftfire is influenced by the gopher, substantiating this behaviour.

Appendix D. Pseudo-Code

Code is available at https://github.com/LukasWill/causal-xrl.

D.1. Algorithm for R-Mask

Algorithm 1 provides pseudo-code for R-mask on Atari environments which jointly learns component Q-functions and component rewards.

D.2. Algorithm for Q-Mask

Algorithm 2 provides pseudo-code for Q-mask on Atari environments which jointly learns component Q-functions and component rewards.

Algorithm 1: Reinforcement Learning with Masking (R-Mask)

Input: The number of reward components K, encoder parameters ψ, Q-function parameters ϕ^i, parameters of reward prediction network θ^i, parameters of mask network Ψ^i, and an empty replay buffer $\mathcal{D}$, where $i = 1, 2, \cdots, K$.

Set target parameters of Q-agent equal to main parameters $\phi^i_{\text{target}} \leftarrow \phi^i$

for $t \leq$ *Total Steps* **do**

Observe state s_t and select action a_t using ϵ-greedy, $a_t = \arg\max_{a'_t} \sum_{i=1}^{K} Q_{\phi^i}(s_t, a'_t)$

Execute a_t in the environment

Observe the next state s_{t+1}, reward r_t, and terminal signal d

Store $(s_t, a_t, r_t, s_{t+1}, d)$ in the replay buffer $\mathcal{D}$

If s_{t+1} is terminal, reset environment state

if $t \geq$ *Learning Start Steps* **then**

if $t \pmod{n_1} == 0$ **then**

`// Intervention, Sufficiency, Sparsity`

Randomly sample batched transitions $B = \{(s_t, a_t, r_t, s_{t+1}, d)\}$ from $\mathcal{D}$

Update parameters ψ to maximize Eq. 1, update parameters θ^i to minimize Eq. 2 and update parameters Ψ^i to maximize Eq. 3

if $t \pmod{n_2} == 0$ **then**

`// Orthogonality`

Randomly sample batched transitions $B = \{(s_t, a_t, r_t, s_{t+1}, d)\}$ from $\mathcal{D}$

Update parameters Ψ^i to minimize Eq. 4

if $t \pmod{n_3} == 0$ **then**

`// Q-update`

Randomly sample batched transitions $B = \{(s_t, a_t, r_t, s_{t+1}, d)\}$ from $\mathcal{D}$

Perform standard Q-learning using full reward r_t to update each parameter ϕ^i to minimize TD-error δ_1

$$\delta_1 = r_t + \gamma \sum_{i=1}^{K} Q_{\phi^i_{\text{target}}}(s_{t+1}, \arg\max_{a'} \sum_{i=1}^{K} Q_{\phi^i}(s_{t+1}, a')) - \sum_{i=1}^{K} Q_{\phi^i}(s_t, a_t)$$

if $t \pmod{n_4} == 0$ **then**

`// Component Q-update`

Randomly sample batched transitions $B = \{(s_t, a_t, r_t, s_{t+1}, d)\}$ from $\mathcal{D}$

Perform standard Q-learning using each estimate reward r_{θ^i} to update each parameter ϕ^i to minimize TD-error δ_2

$$\delta_2 = r_{\theta^i} + \gamma Q_{\phi^i_{\text{targ}}}(s_{t+1}, a^*) - Q_{\phi^i}(s_t, a_t), \forall i.$$

where $a^* = \arg\max_{a'} \sum_{i=1}^{K} Q_{\phi^i}(s_{t+1}, a')$

end

Algorithm 2: Reinforcement Learning with Masking (Q-Mask)

Input: The number of reward components K, encoder parameters ψ, Q-function parameters ϕ^i, parameters of mask network Ψ^i, and an empty replay buffer $\mathcal{D}$, where $i = 1, 2, \cdots, K$.

Set target parameters of Q-agent equal to main parameters $\phi^i_{\text{target}} \leftarrow \phi^i$

for $t \leq$ *Total Steps* **do**

Observe state s_t and select action a_t using ϵ-greedy: $a_t = \arg\max_{a'_t} \sum_{i=1}^{K} Q_{\phi^i}(s_t, a'_t)$

Execute action a_t in the environment

Observe the next state s_{t+1}, rewards $\{r^i_t\}$, and terminal signal d

Store $(s_t, a_t, \{r^i_t\}, s_{t+1}, d)$ in the replay buffer $\mathcal{D}$

If s_{t+1} is terminal, reset the environment state

if $t \geq$ *Learning Start Steps* **then**

if $t \pmod{n_1} == 0$ **then**

`// Intervention, Sparsity`

Randomly sample batched transitions $B = \{(s_t, a_t, \{r^i_t\}, s_{t+1}, d)\}$ from $\mathcal{D}$

Update parameters ψ to maximize Eq. 1 and update parameters Ψ^i to maximize Eq. 3

if $t \pmod{n_2} == 0$ **then**

`// Orthogonality`

Randomly sample batched transitions $B = \{(s_t, a_t, \{r^i_t\}, s_{t+1}, d)\}$ from $\mathcal{D}$

Update parameters Ψ^i to minimize Eq. 4

if $t \pmod{n_4} == 0$ **then**

`// Component Q-update`

Randomly sample batched transitions $B = \{(s_t, a_t, \{r^i_t\}, s_{t+1}, d)\}$ from $\mathcal{D}$

Perform standard Q-learning using ground truth sub-reward r^i_t to update each parameter ϕ^i and minimize TD-error δ

$$\delta = r^i_t + \gamma Q_{\phi^i_{\text{target}}}(s_{t+1}, a^*) - Q_{\phi^i}(s_t, a_t), \forall i$$

where $a^* = \arg\max_{a'} \sum_{i=1}^{K} Q_{\phi^i}(s_{t+1}, a')$

end