Artificial Intelligence 348 (2025) 104403

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Algebras of actions in an agent's representations of the world

Alexander Dean, Eduardo Alonso *, Esther Mondragón

Artificial Intelligence Research Centre (CitAI), Department of Computer Science, City St George's, University of London, Northampton Square, EC1V 0HB, London, UK

ARTICLE INFO

Keywords:
Representation learning
Agents
Disentanglement
Symmetries
Algebra

ABSTRACT

Learning efficient representations allows robust processing of data, data that can then be generalised across different tasks and domains, and it is thus paramount in various areas of Artificial Intelligence, including computer vision, natural language processing and reinforcement learning, among others. Within the context of reinforcement learning, we propose in this paper a mathematical framework to learn representations by extracting the algebra of the transformations of worlds from the perspective of an agent. As a starting point, we use our framework to reproduce representations from the symmetry-based disentangled representation learning (SBDRL) formalism proposed by [1] and prove that, although useful, they are restricted to transformations that respond to the properties of algebraic groups. We then generalise two important results of SBDRL –the equivariance condition and the disentangling definition– from only working with group-based symmetry representations to working with representations capturing the transformation properties of worlds for any algebra, using examples common in reinforcement learning and generated by an algorithm that computes their corresponding Cayley tables. Finally, we combine our generalised equivariance condition and our generalised disentangling definition to show that disentangled sub-algebras can each have their own individual equivariance conditions, which can be treated independently, using category theory. In so doing, our framework offers a rich formal tool to represent different types of symmetry transformations in reinforcement learning, extending the scope of previous proposals and providing Artificial Intelligence developers with a sound foundation to implement efficient applications.

1. Introduction

Artificial intelligence (AI) has progressed significantly in recent years due to massive increases in available computational power that facilitates the development and training of data-intensive deep learning algorithms [2,3]. However, the best-performing learning algorithms often suffer from poor data efficiency and lack the levels of robustness and generalisation that are characteristic of nature-based intelligence [4–10]. Contrarily, the brain appears to solve complex tasks by learning high-level, low-dimensional *representations* that focus on aspects of the environment that are relevant [11–16].

In this paper, we propose a formal framework that aims to address how "good" representations can be learned within the context of reinforcement learning. In reinforcement learning, an agent learns an optimal policy by interacting with the environment, the world, by trial an error [17–20]. As a result of the agent's actions, the state of the environment changes, and the agent receives a numerical signal, a reward. The task of the agent is to learn the sequence of actions that maximises the expected cumulative reward

* Corresponding author.
E-mail addresses: alexander.dean@city.ac.uk (A. Dean), e.alonso@city.ac.uk (E. Alonso), e.mondragon@city.ac.uk (E. Mondragón).

https://doi.org/10.1016/j.artint.2025.104403
Received 2 October 2023; Received in revised form 28 July 2025; Accepted 11 August 2025

Available online 20 August 2025

by exploring the environment and exploiting the knowledge so acquired. For this process to be efficient, it is thus crucial that the agent learns "good" representations that guide it through the state-action space.

But what makes a "good" state representation? Higgins et al. ([1]) argue that the *symmetries* of the world are important structures that should be present in the representation of that world. The study of symmetries shifts the centre of attention from studying objects directly to studying the transformations of those objects and using the information about these transformations to discover properties about the objects themselves [21]. The exploitation of symmetries has led to many successful deep-learning architectures. Examples include convolutional layers [22], which utilise transitional symmetries to outperform humans in image recognition tasks [23], and graph neural networks [24], which utilise the group of permutations. Not only can symmetries provide a useful indicator of what an agent has learned, but incorporating symmetries into learning algorithms regularly reduces the size of the problem space, leading to greater learning efficiency and improved generalisation [21]. In fact, it has been shown that a large majority of neural network architectures can be described as stacking layers that deal with different symmetries [25]. The main methods used to integrate symmetries into a representation are to build symmetries into the architecture of learning algorithms [26,27], use data augmentation that encourages the model to learn symmetries [28,29], or to adjust the model's learning objective to encourage the representation to exhibit certain symmetries [30,31].

There are two main types of symmetries that are used in AI: *invariant symmetries*, where a representation does not change when certain transformations are applied to it, and *equivariant symmetries*, where the representation reflects the symmetries of the world. Historically, the learning of representations that are invariant to certain transformations has been a fruitful line of research [32–35]. In building these invariant representations, the agent effectively learns to ignore them since such representations remain unaffected by the transformation. It has been suggested that this approach can lead to more narrow intelligence, where an agent becomes good at solving a small set of tasks but struggles with data efficiency and generalisation when tasked with new learning problems [36,37]. Instead of ignoring certain transformations, the equivariant approach attempts to preserve symmetry transformations in the agent's representations in such a way that they match the symmetry transformations of the world. It has been hypothesised that the equivariant approach is likely to produce representations that can be reused to solve a more diverse range of tasks because no transformations are discarded [21]. Equivariant symmetry approaches are commonly linked with *disentangling representations* [38], in which the agent's representation is separated into subspaces that are invariant to different transformations. Disentangled representation learning, which aims to produce representations that separate the underlying structure of the world into disjoint parts, has been shown to improve the data efficiency of learning [39,40].

Inspired by their use in physics, symmetry-based disentangled representations (SBDRs) were proposed as a formal mathematical definition of disentangled representations [1]. SBDRs are built on the assumption that the symmetries of a state of the world display important aspects of that world that need to be preserved in an agent's internal representation (i.e., the symmetries that are present in the world state should also be present in the agent's internal representation state). Higgins et al. describe symmetries of the world state as *"transformations that change only some properties of the underlying world state, while leaving all other properties invariant"* [1, page 1]. For example, the y-coordinate of an agent moving parallel to the x-axis on a 2D Euclidean plane does not change. Symmetry-based disentangled representation learning (SBDRL) has gained traction in AI in recent years [41–50]. However, SBDRL only considers actions that form groups and so cannot take into account, for example, irreversible actions [1]. Besides, [51] showed that a symmetry-based representation cannot be learned using only a training set composed of the observations of world states; instead, a training set composed of transitions between world states, as well as the observations of the world states, is required. In other words, SBDRL requires the agent to interact with the world.

We agree with [1] that symmetry transformations are important structures to include in an agent's representation, but take their work one step further: along with [51] we posit that the relationships of transformations of the world due to the actions of the agent should be included in the agent's representations of the world, but, unlike [51], we do not constraint such transformations to symmetries that rely on actions to form algebraic groups exclusively. In this paper, we show that an agent's representation of a world would lose important information if only transformations of the actions of an agent that form groups are included, and demonstrate that there exist features that cause transformations that do not form group structures. We believe that considering these transformations in the agent's representation of the world has the potential to build powerful learning mechanisms.

In short, we aim to help answer the question of which features should be present in a 'good' representation by, as suggested by [1], looking at the transformation properties of worlds. However, while [1] only considered a limited type of symmetry transformations, those that are formalised as groups, we aim to go further and consider the full algebra of world transitions. We propose a mathematical framework to describe the transformations of the world, thereby formally characterising the features we expect to find in the representations of an artificial agent. More specifically, our contributions are as follows:

1. We propose a general mathematical framework for describing the transformation of worlds due to the actions of an agent that can interact with the world.
2. We derive the SBDRs proposed by [1] from our framework and, in doing so, identify the limitations of SBDRs in their current form.
3. We use our framework to explore the structure of the transformations of worlds for classes of worlds containing features found in common reinforcement learning scenarios, and that go beyond symmetry groups and SBDRL. We also present the algorithm used to generate the algebra of the transformations of the world due to the actions of an agent.
4. We generalise the equivariance condition and the definition of disentangling given by [1] to worlds that do not satisfy the conditions for SBDRs. This generalisation is performed using category theory.

It should be emphasised that our framework is about learning representations of an agent interacting with their environment, independently of the reinforcement learning algorithm used in the process (say Q-learning, SARSA, PPO or any deep reinforcement learning algorithm). Also, the paper is formal in that it establishes a mathematical framework for representation learning. It responds to the need to formulate sound formalisms from which specific applications may follow. That is, our work is foundational.

The rest of the paper is structured as follows: In Section 2, we define our framework and then describe how it deals with generalised worlds, which consist of distinct world states connected by transitions that represent the dynamics of the world as a result of the actions of an agent. In Section 3, our framework is used to reproduce SBDRL. This is achieved by defining an equivalence relation that makes the actions of an agent equivalent if the actions produce the same outcome if performed while the world is in any given state. In Section 4, we apply our framework to worlds exhibiting common reinforcement learning scenarios that cannot be described fully using SBDRs and study the algebraic structures exhibited by the dynamics of these worlds. In Section 5, we generalise, using category theory, two important results of [1] –the equivariance condition and the disentangling definition– to worlds with transformations whose algebras do not fit into the SBDRL paradigm. We shall finish with conclusions and a discussion in Section 6.

2. A mathematical framework for an agent in the world

In this section, we introduce our general mathematical framework for formally describing the transformations of a world. First, we define world states and transitions. We then consider how agents are understood within our framework and how their actions are formalised as labelled transitions.

2.1. Model of the world

For the sake of simplicity, we consider a fully observable world consisting of a set of discrete states, which are distinguishable in some way, and a set of transitions between those states; these transitions convey the world dynamics (i.e., how the world can transform from one world state to another). This world can be represented by a directed multigraph, where the world states are the vertices of the graph and the transitions between states are arrows between the vertices. We will use this framework to reproduce the group action structure of the evolution of a world as an agent's representation of it, as described by [1]. In so doing, we uncover the requirements for this group action structure to be present in the world.

2.1.1. World states and world state transitions

We believe that defining a world as a discrete set of world states with world state transitions between them is the most general definition of a world. Therefore, we take it as our starting point to define the algebra of the actions of an agent.

Transitions We consider a directed multigraph $\mathscr{W} = (W, \hat{D}, s, t)$ where W is a set of *world states*, $\hat{D}$ is a set of *minimum world state transitions*, and $s, t : \hat{D} \to W$; s is called the *source map* and t is called the *target map*. For the remainder of the paper, we fix such a $(W, \hat{D}, s, t)$. $\mathscr{W}$ is called a *world*.

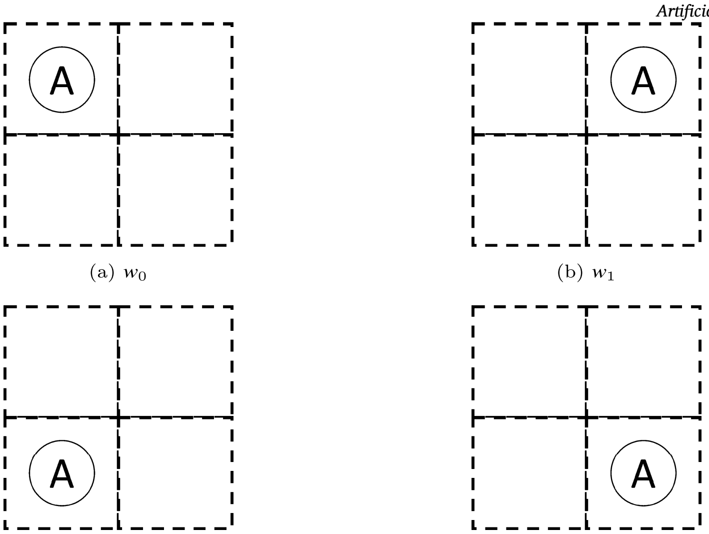
Minimum world state transitions are extended into a set D of paths called *world state transitions*: a path is a sequence of minimum world state transitions $d = \hat{d}_n \circ \hat{d}_{n-1} \circ \ldots \circ \hat{d}_1$ such that $t(\hat{d}_i) = s(\hat{d}_{i+1})$ for $i = 1, \ldots, n-1$. We extend s, t to D as $s(d) = s(\hat{d}_1)$, and $t(d) = t(\hat{d}_n)$. We also extend the composition operator $\circ$ to D such that $d_n \circ d_{n-1} \circ \ldots \circ d_1$ is defined if $t(d_i) = s(d_{i+1})$ for $i = 1, \ldots, n-1$. For $d \in D$ with $s(d) = w$ and $t(d) = w'$, we will often denote d by $d : w \to w'$.

For the rest of the paper, we assume that, for each world state $w \in W$, there is a unique trivial world state transition $1_w \in \hat{D}$ with $s(1_w) = t(1_w)$; the trivial transition 1_w is associated with the world being in state w and then no change occurring due to the transition 1_w.

Connected and disconnected worlds We now introduce connected and disconnected worlds. Simply, a world A is connected to a world B if there is a transition from a world state in world A to a world state in world B. The concepts of connected and disconnected worlds are necessary for generality; we are only interested in the perspective of the agent and so only care about the world states and transitions that the agent can come into contact with. Connected and disconnected worlds give us the language to describe and then disregard the parts of worlds that the agent will never explore and therefore are not relevant to the agent's representation. For example, if an agent is in a maze and a section of the maze is inaccessible from the position that the agent is in, then that section of the maze would be disconnected from the section of the maze that the agent is in; if we want to study how the agent's representation evolves as it learns, it makes sense to disregard the disconnected section of the maze since the agent never comes into contact with it and so the disconnected section of the maze will not affect the agent's representation.

Formally, we first define a *sub-world* W' of a world W as a subset $W' \subseteq W$ along with $D' = \{d \in D \mid s(d) \in W' \text{ and } t(d) \in W'\}$. Note that a sub-world is a world. A sub-world W is *connected to* a sub-world W' if there exists a transition $d : w \to w'$ where $w \in W$ and $w' \in W'$; if no such transition exists, then W is *disconnected from* W'. Similarly, a world state w is *connected to* a sub-world W if there exists a transition $d : w \to w'$ where $w' \in W'$; if no such transition exists, then w is *disconnected from* W'.

Effect of transitions on world states We define $*$ as a partial function $D \times W \to W$ by $d * w = w'$ where $d : w \to w'$ and undefined otherwise.



(a) w_0 (b) w_1

(c) w_2 (d) w_3

Fig. 1. The world states of a cyclical 2×2 grid world W_c, where changes to the world are due to an agent moving either up, down, left, or right. The position of the agent in the world is represented by the position of the circled A.

Table 1
Each entry in this table shows the outcome state of the agent performing the action given in the column label when in the world state given by the row label.

	1	U	D	L	R
w_0	w_0	w_2	w_2	w_1	w_1
w_1	w_1	w_3	w_3	w_0	w_0
w_2	w_2	w_0	w_0	w_3	w_3
w_3	w_3	w_1	w_1	w_2	w_2

2.1.2. Example

We consider a cyclical 2×2 grid world, denoted by $\mathscr{W}_c$, containing an agent as shown in Fig. 1. The transformations of $\mathscr{W}_c$ are due to an agent moving either up (U), down (D), left (L), right (R), or doing nothing (1). The possible world states of $\mathscr{W}_c$ are shown in Fig. 1. $\mathscr{W}_c$, and variations of it, are used as a running example to illustrate the concepts presented in this paper.

We say the world being cyclical means that if the agent performs the same action enough times, then the agent will return to its starting position; for example, for the world $\mathscr{W}_c$ if the agent performs the action U twice when the world is in state w_0 in Fig. 1 then the world will transition into the state w_0 (i.e., $U^2 * w_0 = w_0$). The transition due to performing each action in each state can be found in Table 1.

The transitions shown in Table 1 can be represented as the transition diagram given in Fig. 2. It should be noted that, since the structure of the diagram is wholly dependent on the arrows between the world states, the positioning of the world states is an arbitrary choice.

2.2. Agents

We consider worlds containing an embodied agent that is able to interact with the environment by performing actions. The end goal of the agent's learning process is to map the useful aspects of the structure of the world to the structure of its representations; the useful aspects are those that enable the agent to complete whatever task it is programmed to achieve (e.g., in reinforcement learning, finding an optimal policy).

We use the treatment of agents adopted by [1]. The agent has an unspecified number of sensors that allow it to make observations of the state of the environment. Information about the world state that the agent is currently in is delivered to the internal state of the agent, its representation of the (state of the) world. Mathematically, the process of information propagating through the sensory states is a mapping $b : W \to O$ (the 'observation process'), which produces a set of observations o_1 containing a single observation for each sensory state. These observations are then used by an inference process $h : O \to Z$ to produce an internal representation. The agent then uses some internal mechanism to select an action to perform.

It is important to note that the agent's state representation only reflects the observations the agent makes with its sensors; in other words, the agent's internal state is built using the information about aspects of the world state propagated through its sensory states, and not directly from the world state.

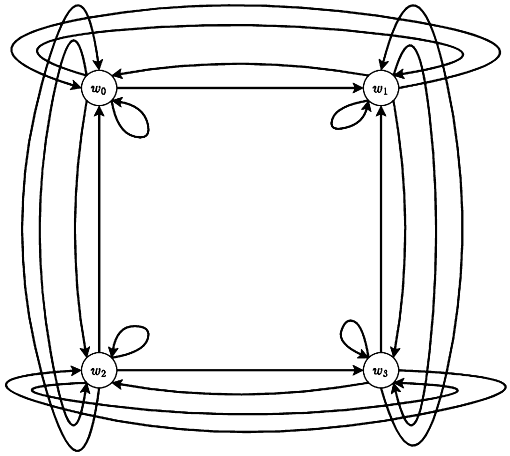


Fig. 2. A transition diagram for the transitions shown in Table 1.

2.2.1. Actions of an agent as labelled transitions

Consider a set $\hat{A}$ called the set of *minimum actions*. Let the set A be the set of all finite sequences formed from the elements of the set $\hat{A}$; we call A the set of *actions*. Consider a set $\hat{D}_A \subset D$, where $1_w \in \hat{D}_A$ for all $w \in W$; we call $\hat{D}_A$ the set of *minimum action transitions*. We consider a labelling map $\hat{l} : \hat{D}_A \to \hat{A}$ such that:

1. Any two distinct transitions leaving the same world state are labelled with different actions.

   Action condition 1. *For any $d, d' \in \hat{D}_A$ with $d \neq d'$ and $s(d) = s(d')$, $\hat{l}(d) \neq \hat{l}(d')$.*

2. There is an identity action that leaves any world state unchanged.

   Action condition 2. *There exists an action $1 \in \hat{A}$ such that $\hat{l}(1_w) = 1$ for all $w \in W$. We call 1 the identity action.*

Given $\hat{D}_A$ as defined above and satisfying action condition 1 and action condition 2, we define $Q_{D_A} = (W, \hat{D}_A, s_A, t_A)$, where s_A, t_A are the restrictions of s, t to the set $\hat{D}_A$. We now define a set D_A, the set of *action transitions*, which is the set of all paths of Q_{D_A}.

We extend the map $\hat{l}$ to a map $l : D_A \to A$ such that if $d = \hat{d}_n \circ ... \circ \hat{d}_1$ then $l(d) = \hat{l}(\hat{d}_n)...\hat{l}(\hat{d}_1)$. For $d \in D_A$ with $s(d) = w$, $t(d) = w'$ and $l(d) = a$, we will often denote d by $d : w \xrightarrow{a} w'$.

If an action $a \in A$ is expressed in terms of its minimum actions as $a = \hat{a}_n \circ ... \circ \hat{a}_1$, then $a = l(d) = l(\hat{d}_n \circ ... \circ \hat{d}_1) = \hat{l}(\hat{d}_n) \circ ... \circ \hat{l}(\hat{d}_1) = \hat{a}_n \circ ... \circ \hat{a}_1$, where the $\hat{a}_i$ are called *minimum actions*.

Remark 2.2.1. *For a given $w \in W$, we can label transitions in D_A with an appropriate element of A through the following: for each $d \in D_A$ with $s(d) = w$, express d in terms of its minimum transitions in D_A as $d = d_n \circ ... \circ d_2 \circ d_1$; if $\hat{l}(d_i) = a_i$ then d is labelled with $a_n ... a_2 a_1 \in A$. We denote the map that performs this labelling by $l : D_A \to A$.*

Fig. 3 shows how transitions are labelled with actions in our 2×2 cyclical world example. We only show the minimum actions for simplicity but there are actually infinite action transitions between each pair of world states; for example, the action transitions from w_0 to w_1 include those labelled by: $D \circ R$, $D \circ R \circ 1^n$ ($n \in \mathbb{N}$), $1^n \circ D \circ R$ ($n \in \mathbb{N}$), $D \circ R \circ (L \circ R)^n$ ($n \in \mathbb{N}$) etc...

Effect of actions on world states We define the effect of the action $a \in A$ on world state $w \in W$ as the following: if there exists $d \in D_A$ such that $s(d) = w$ and $l(d) = a$, then $a * w = t(d)$; if there does not exist $d \in D_A$ such that $s(d) = w$ and $l(d) = a$, then we say that $a * w$ is *undefined*. The effect of actions on world states is well-defined due to action condition 1. We can apply the minimum actions that make up an action to world states individually: if $a * w$ is defined and $a = \hat{a}_k ... \hat{a}_1$ then $a * w = (\hat{a}_k ... \hat{a}_1) * w = \hat{a}_k ... \hat{a}_2 * (\hat{a}_1 * w)$. Physically, the identity action $1 \in A$ corresponds to the no-op action (*i.e.*, the world state does not change due to this action).

Actions as (partial) functions Consider all the transitions that are labelled by a particular action $a \in A$. Together these transitions form a partial function $f_a : W \to W$ because for any $w \in W$ either $a * w$ is undefined or $a * w$ is defined and there is a unique world state $w' \in W$ for which $a * w = w'$ (due to condition 1). f_a is not generally surjective because for a given $w \in W$ there is not necessarily a transition $d \in D$ with $l(d) = a$ and $t(d) = w$. f_a is not generally injective because it is possible to have an environment where $f_a(w) = f_a(w')$ for some $w \in W$ different from $w' \in W$. We can also reproduce these functions using the formalism given by

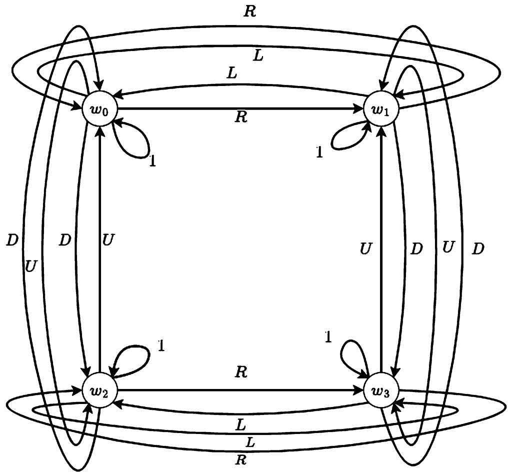


Fig. 3. Labelling the transitions in Fig. 2 with the relevant actions in A.

[51], which describes the dynamics of the world in terms of a multivariate function $f : A \times W \to W$. If we let $f : A \times W \to W$ be the dynamics of the environment then the transition caused by an action $a \in A$ on a world state $w \in W$ (where $a * w$ is defined) is given by $(a, w) \mapsto f(a, w) = a * w$. Mathematically, we curry the function $f : A \times W \to W$ to give a collection $\{f_a\}$ of partial functions with a partial function $\hat{f}(a) = f_a : W \to W$ for each action $a \in A$ as *Curry* $: (f : A \times W \to W) \to (f_a : W \to W)$.

Once we have introduced our framework to formalise word states and their transitions through the agent's intervention, we are investigating next its representational power by showing that it can generate the same transitions as SBDR, that is, those that form symmetry groups over disentangled representations, in Section 3, and other types of representations, that do not fit in the constraints imposed by SBDR, and that are standard in reinforcement learning scenarios in Section 4.

3. Reproducing SBDRL

We now use the framework set out in the previous section to reproduce SBDRL and illustrate it using worlds that are similar to those given by [1] and [51]. We choose to begin by reproducing symmetry-based representations because (1) symmetry-based representations describe transformations of the world that form relatively simple and well-understood algebraic structures (groups), (2) groups, and the symmetries they describe, are gaining increasing prominence in Artificial Intelligence research, (3) it shows how our framework encompasses previous work in formalising the structure of transformations of a world, and (4) it provides a more rigorous description of SBDRL, which should aid future analysis and development of the concept.

Section 3.1 provides a description of SBDRL, and Section 3.2 shows how to obtain SBDRL using an equivalence relation on the actions of the agent. Section 3.3 details the algorithmic exploration of world structures performed on example worlds and goes through a worked example. Finally, Section 3.4 shows the conditions of the world that are required for the actions of an agent to be fully described by SBDRs.

3.1. Symmetry-based disentangled representation learning

We proceed to present a more detailed description of the SBDRL formalism, how to extract representations from world states, and how to build symmetry-based and symmetry-based disentangled representations, before stating their limitations and thus, of SBDRL.

From world states to representation states The world state is an element of a set W of all possible world states. The observations of a particular world state made by the agent's sensors are elements of the set O of all possible observations. The agent's internal state representation of the world state is an element of a set Z of all possible internal state representations. There exists a composite mapping $f = h \circ b : W \to Z$ that maps world states to states of the agent's representation ($w \mapsto z$); this composite mapping is made up of the mapping of an observation process $b : W \to O$ that maps world states to observations ($w \mapsto o$) and the mapping of an inference process $h : O \to Z$ that maps observations to the agent's internal state representation ($o \mapsto z$) (see Fig. 4).

Given that the symmetries represented by SBDRL form algebraic groups, we first define the notions of groups and group action formally, as follows:

Groups and symmetries

Definition 3.1 *(Group). A group G is a set with a binary operation $G \times G \to G$, $(g, g') \mapsto g \circ g'$ called the composition of group elements that satisfies the following properties:*

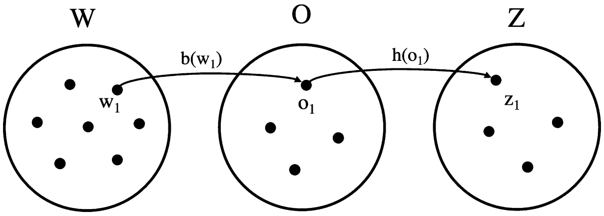


Fig. 4. The composite mapping from the set W of world states to the set Z of state representations via the set O of observations.

1. *Closure.* $g \circ g'$ *is defined for all* $g, g' \in G$.
2. *Associative.* $(g \circ g') \circ g'' = g \circ (g' \circ g'')$ *for all* $g, g', g'' \in G$.
3. *Identity. There exists a unique identity element* $1 \in G$ *such that* $1 \circ g = g \circ 1 = g$ *for all* $g \in G$.
4. *Inverse. For any* $g \in G$*, there exists* $g^{-1} \in G$ *such that* $g \circ g^{-1} = g^{-1} \circ g = 1$.

Applying symmetries to objects is mathematically defined as a *group action*.

Definition 3.2 *(Group action). Given a group* G *and a set* X*, a group action of* G *on* X *is a map* $G \times X \to X$*,* $(g, x) \mapsto g * x$ *that satisfies the following properties:*

1. *Compatibility with composition. The composition of group elements and the group action are compatible:* $g' \circ (g * x) = (g' \circ g) * x$ *for* $g, g' \in G$ *and* $x \in X$.
2. *Identity. The group identity* $1 \in G$ *leaves the elements of* X *unchanged:* $1 * x = x$ *for all* $x \in X$.

Another important property of groups is commutation. Two elements of a group *commute* if the order they are composed does not matter: $g \circ g' = g' \circ g$. If all elements in a group commute with each other then the group is called *commutative*. Subgroups of a group might commute with each other.

Symmetry-based representations The set W of world states has a set of symmetries that are described by the group G. This group G acts on the set W of world states via a group action $\cdot_W : G \times W \to W$. For the agent's representations $z_i \in Z$ to be symmetry-based representations, a corresponding group action $\cdot_Z : G \times Z \to Z$ must be found so that the symmetries of the agent's representations reflect the symmetries of the world states. The mathematical condition for this is that, for all $w \in W$ and all $g \in G$, applying the action $g \cdot_W$ to w and then applying the mapping f gives the same result as first applying the mapping f to w to give $f(w)$ and then applying the action $g \cdot_Z$ to $f(w)$. Mathematically, this is $f(g \cdot_W w) = g \cdot_Z f(w)$. If this condition is satisfied, then f is called a *group-equivariant map*.

Symmetry-based disentangled representations To go from symmetry-based representations to symmetry-based disentangled representations, suppose the group of symmetries G of the set W of world states decomposes as a direct product $G = G_1 \times \ldots \times G_i \times \ldots \times G_n$. The group action $\cdot_Z : G \times Z \to Z$ and the set Z are disentangled with respect to the decomposition of G, if there is a decomposition $Z = Z_1 \times \ldots \times Z_i \times \ldots \times Z_n$ and actions $\cdot_{Z_i} : G_i \times Z_i \to Z_i, i \in \{1, \ldots, n\}$ such that $(g_{G_1}, g_{G_2}) \cdot_Z (z_{Z_1}, z_{Z_2}) = (g_{G_1} \cdot_{Z_1} z_{Z_1}, g_{G_2} \cdot_{Z_2} z_{Z_2})$, where $g_{G_i} \in G_i$ and $z_{Z_i} \in Z_i$. In other words, each subspace Z_i is invariant to the action of all the $G_{j \neq i}$ and only affected by G_i.

The representations in Z are symmetry-based disentangled with respect to the decomposition $G = G_1 \times \ldots G_i \times \ldots \times G_n$, where each G_i acts on a disjoint part of Z, if:

1. There exists a group action $\cdot_W : G \times W \to W$ and a corresponding group action $\cdot_Z : G \times Z \to Z$;
2. The map $f : W \to Z$ is group-equivariant between the group actions on W and Z: $g \cdot_Z f(w) = f(g \cdot_W w)$. In other words, the diagram

$$\begin{tikzcd} w \arrow[r, "g \cdot_W"] \arrow[d, "f"] & g \cdot_W w \arrow[d, "f"] \\ f(w) \arrow[r, "g \cdot_Z"] & g \cdot_Z f(w) = f(g \cdot_W w) \end{tikzcd}$$

   commutes.
3. There exists a decomposition of the representation $Z = Z_1 \times \ldots \times Z_n$ such that each subspace Z_i is unaffected by the action for all $G_{j \neq i}$ and is only affected by G_i.

Limitations of SBDRL Both [1] and [51] suggest that these group actions can be used to describe some types of real-world actions. However, it is important to note that they do not believe that all actions can be described by their formalism: *"It is important to*

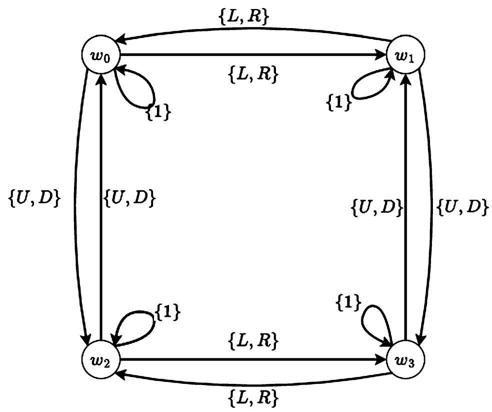


Fig. 5. Action equivalence classes in $A/\sim$ for the actions show in Fig. 3.

mention that not all actions are symmetries, for instance, the action of eating a collectible item in the environment is not part of any group of symmetries of the environment because it might be irreversible." [51, page 4].

3.2. SBDRL through equivalence

For the algebra of the actions of our agent to form a group, we need some sense of actions being the same so that the algebra can satisfy the group properties (*e.g.*, for the identity property we need an element 1 in the algebra A such that $1a = a1 = a$ for any $a \in A$). We define an equivalence relation on the elements of A that says two actions are equivalent (our sense of the actions being the same) if they lead to the same end world state when performed in any initial world state. This equivalence relation is based on our mathematical interpretation of the implication given by [1] that transformations of the world are the same if they have the same effect, which is used to achieve the group structure for SBDRL. Our use of an equivalence relation was inspired by [52], which uses a similar equivalence relation to equate action sequences that cause the same final observation state after each action sequence is performed from an initial observation state. We then derive some properties of the equivalence classes created by $\sim$ that will be used to show that the actions of an agent form the group action described by [1] under the equivalence relations we define and for worlds satisfying certain conditions.

Definition 3.3 *(Equivalence of actions under $\sim$). Given two actions $a, a' \in A$, we denote $a \sim a'$ if $a * w = a' * w$ for all $w \in W$.*

Remark 3.2.1. *If $a \sim a'$, then either for each $w \in W$ (1) there exists transitions $d : w \xrightarrow{a} t(d)$ and $d' : w \xrightarrow{a'} t(d)$ or (2) there exists no transitions $d : w \xrightarrow{a} t(d)$ or $d' : w \xrightarrow{a'} t(d)$.*

Proposition 3.1. *$\sim$ is an equivalence relation.*

Proof. We demonstrate the three properties of an equivalence relation, namely, reflexive, transitive, and symmetric.

Reflexive. If $a \sim a'$ then $a * w = a' * w$ for all $w \in W$.

Transitive. If $a \sim a'$ and $a' \sim a''$, then $a * w = a' * w$ for all $w \in W$ and $a' * w = a'' * w$ for all $w \in W$. Therefore, $a * w = a'' * w$ for all $w \in W$ and so $a \sim a''$.

Symmetric. If $a \sim a'$, then $a * w = a' * w$ for all $w \in W$. Therefore $a' * w = a * w$ for all $w \in W$, and so $a' \sim a$. □

Fig. 5 shows the effect of applying the equivalence relations to our 2×2 cyclical example world $\mathscr{W}_c$.

We define the canonical projection map $\pi_A : A \to A/\sim$ that sends actions in A to their equivalence classes under $\sim$ in the set $A/\sim$. We denote the equivalence class of a by $[a]_\sim$. Sometimes we will drop the $[a]_\sim$ in favour of $a \in A/\sim$ for ease.

Composition of actions We define the composition of elements in $A/\sim$ as $\circ : (A/\sim) \times (A/\sim) \to (A/\sim)$ such that $[a']_\sim \circ [a]_\sim = [a' \circ a]_\sim$ for $a, a' \in A$.

Proposition 3.2. *$[a']_\sim \circ [a]_\sim = [a' \circ a]_\sim$ is well-defined for all $a, a' \in A$.*

Proof. We need to show that the choice of a, a' doesn't matter: if $a_1 \sim a_2$ and $a_3 \sim a_4$ for $a_1, a_2, a_3, a_4 \in A$, then $[a_3 \circ a_1]_\sim = [a_4 \circ a_2]_\sim$. $a_1 \sim a_2$ means there exists $d_i : s(d_1) \xrightarrow{a_i} t(d_1)$ for $i = 1, 2$. Since actions are unrestricted in W, for any world state and any action there is a transition with a source at that world state that is labelled by that action. $a_3 \sim a_4$ means there exists $d_j : s(d_3) \xrightarrow{a_j} t(d_3)$ for $j = 3, 4$, and so there exists $d_j : t(d_1) \xrightarrow{a_j} t(d_3)$ for $j = 3,4$. $\Longrightarrow$ there exists $(d_3 \circ d_1) : s(d_1) \xrightarrow{a_3 \circ a_1} t(d_3)$ and $(d_4 \circ d_2) : s(d_1) \xrightarrow{a_4 \circ a_2} t(d_3)$. $\Longrightarrow$ $s(d_3 \circ d_1) = s(d_4 \circ d_2)$ and $t(d_3 \circ d_1) = s(d_4 \circ d_2)$. $\Longrightarrow$ $(a_3 \circ a_1) \sim (a_4 \circ a_2)$. $\Longrightarrow$ $[a_3 \circ a_1]_\sim = [a_4 \circ a_2]_\sim$. $\square$

Effect of equivalent actions on world states We define the effect of an element of $A/\sim$ on world states as $*: (A/\sim) \times W \to W$ such that $[a]_\sim * w = a * w$. Note that this is only defined if there exists $d : w \xrightarrow{a} t(d)$ for $d \in D_A$.; if not, then $[a]_\sim * w$ is called *undefined*.

Proposition 3.3. *$[a]_\sim * w$ is well-defined for all $a \in A$ and for all $w \in W$.*

Proof. We need to show that $a_1 * w = a_2 * w$ if $[a_1]_\sim = [a_2]_\sim$ for $a_1, a_2 \in A$ and $w \in W$. If $[a_1]_\sim = [a_2]_\sim$, then $a_1 \sim a_2$. Since actions are unrestricted in W, for any world state and any action there is a transition with a source at that world state labelled by that action. $\Longrightarrow$ there exists $d_i : w \xrightarrow{a_i} t(d_1)$ for $i = 1,2$. $\Longrightarrow$ $a_1 * w = d_1 * w = t(d_1)$ and $a_2 * w = d_2 * w = t(d_1)$. $\square$

Reversible actions An action $a \in A$ is called *reversible* in a given state $w \in W$ if $a \in A_{Rw}$ where $A_{Rw} = \{a \in A \mid$ *there exists* $a' \in A$ *such that* $a' \circ a * w = w\}$. An action $a \in A$ is called *reversible* if it is reversible in every $w \in W$. An action that is not reversible is called *irreversible*.

Properties of the quotient set $A/\sim$

Proposition 3.4. *$(A/\sim, \circ)$ has an identity element.*

Proof. To show that $(A/\sim, \circ)$ has an identity element we can show that there is an element $e \in A$ which satisfies (a) $[a]_\sim \circ [e]_\sim = [a]_\sim$ and (b) $[e]_\sim \circ [a]_\sim = [a]_\sim$ for all $a \in A$. We will prove that the identity action $1 \in A$ satisfies the above condition. Consider any transition $d : s(d) \xrightarrow{a} t(d)$ labelled by any action $a \in A$.

(a) There exists a transition $1_{s(d)} : s(d) \xrightarrow{1} s(d)$ due to action condition 2. $t(1_{s(d)}) = s(d) \Longrightarrow a \circ 1$ is defined for d. $s(a \circ 1) = s(1) = s(d) = s(a)$ and $t(a \circ 1) = t(a)$. $\Longrightarrow$ $a \circ 1 \sim a$. $\Longrightarrow$ $[a \circ 1]_\sim = [a]_\sim$. $\Longrightarrow$ $[a]_\sim \circ [1]_\sim = [a]_\sim$.

(b) There exists a transition $1_{t(d)} : t(d) \xrightarrow{1} t(d)$ due to action condition 2. $t(1_{t(d)}) = t(d) \Longrightarrow 1 \circ a$ is defined for d. $s(1 \circ a) = s(a)$ and $t(1 \circ a) = t(1) = t(a)$. $\Longrightarrow$ $1 \circ a \sim a$. $\Longrightarrow$ $[1 \circ a]_\sim = [a]_\sim$. $\Longrightarrow$ $[1]_\sim \circ [a]_\sim = [a]_\sim$. Therefore $1 \in A$ satisfies the conditions for $[1]_\sim$ being an identity element in $(A/\sim, \circ)$. $\square$

Proposition 3.5. *$\circ$ is associative with respect to $(A/\sim, \circ)$.*

Proof. For $\circ$ to be associative we need $[a_1]_\sim \circ ([a_2]_\sim \circ [a_3]_\sim) = ([a_1]_\sim \circ [a_2]_\sim) \circ [a_3]_\sim$ for any $a_1, a_2, a_3 \in A$. We have $a_1 \circ (a_2 \circ a_3) = (a_1 \circ a_2) \circ a_3$ from the associativity of $\circ$ with respect to $(A, \circ)$, and $[a']_\sim \circ [a]_\sim = [a' \circ a]_\sim$ for any $a, a' \in A$ by definition of $\circ$ on $A/\sim$. $\Longrightarrow$ $[a_1]_\sim \circ ([a_2]_\sim \circ [a_3]_\sim) = [a_1 \circ (a_2 \circ a_3)]_\sim = [(a_1 \circ a_2) \circ a_3]_\sim = [(a_1 \circ a_2)]_\sim \circ [a_3]_\sim = ([a_1]_\sim \circ [a_2]_\sim) \circ [a_3]_\sim$. $\square$

In summary, we have $(A, \circ, *)$, which is a set A along with two operators $\circ : A \times A \to A$ and $*: A \times W \to W$, and we have $(A/\sim, \circ, *)$, which is a set $A/\sim$ along with two operators $\circ : (A/\sim) \times (A/\sim) \to (A/\sim)$ and $*: (A/\sim) \times W \to W$. We have shown that $\circ$ is associative with respect to $(A/\sim, \circ)$, and that $(A/\sim, \circ)$ has an identity element by action condition 2.

3.3. Algorithmic exploration of world structures

To gain an intuition of the structure of different worlds and to illustrate our theoretical work with examples, we developed an algorithm that uses an agent's minimum actions to generate the algebraic structure of the transformations of a world. We display this structure as a generalised Cayley table (a multiplication table for the distinct elements of the algebra).

The main algorithm generates what we call a state Cayley table (Algorithm 1). The elements of this state Cayley table are the world states reached when the row action and then the column action are performed in succession from an initial world state w (*i.e.*, $column\_label * (row\_label * w)$). Once the state Cayley table has been generated, we can use it to generate the action Cayley table, in which the elements of the table are the equivalent elements in the algebra if the agent performs the row action followed by the column action (Algorithm 2). For the sake of readability, Algorithms 3 to 6, which are internal loops of Algorithm 1, can be found in the Appendix. The code for their implementation of all the algorithms is available at github.com/awjdean/CayleyTableGeneration.

It is worth noting the usefulness of these algorithms: independently of their use in this paper, to show to generate both SBDRL groups (Section 3) and symmetries that fall beyond SBDRL's scope (Section 4), researchers in representation learning and reinforcement learning are provided with an automatic mechanism to generate transitions of any arbitrary structure.

Algorithm 1 Generate state Cayley table.

```
Require: minimum_actions: a list of minimum actions, w: initial world state.
state_cayley_table ← an empty square matrix with dimensions len(minimum_actions)×len(minimum_actions), with rows and columns labelled by minimum_actions.
for a in minimum_actions do
    Create an equivalence class for a.
    state_cayley_table ← AddElementToStateCayleyTable(state_cayley_table, w, a).    ▷ See Algorithm 3.
end for
for row_label in state_cayley_table do
    equivalents_found ← SearchForEquivalents(state_cayley_table, w, row_label).    ▷ See Algorithm 5.
    if len(equivalents_found) ≠ 0 then
        Merge the equivalence classes of equivalent minimum actions.
        Delete the row and column from state_cayley_table for the minimum actions not labelling the merged equivalence class.
    end if
end for
Initialize an empty list candidate_cayley_table_elements.
candidate_cayley_table_elements ← SearchForNewCandidates(state_cayley_table, w, candidate_cayley_table_elements).    ▷ See Algorithm 4.
while len(candidate_cayley_table_elements) > 0 do
    a_C ← pop an element from candidate_cayley_table_elements.
    equivalents_found ← SearchForEquivalents(state_cayley_table, w, a_C).    ▷ See Algorithm 5.
    if len(equivalents_found) ≠ 0 then
        Add a_C to the relevant equivalence class.
        Continue to the next iteration of the while loop.
    else
        Check if a_C breaks any of the existing equivalence classes.
        broken_equivalence_classes ← SearchForBrokenEquivalenceClasses(state_cayley_table, w, a_C).    ▷ See Algorithm 6.
        if len(broken_equivalence_classes) ≠ 0 then
            for each new equivalence class do
                state_cayley_table ← AddElementToStateCayleyTable(state_cayley_table, w, element labelling new equivalence class).    ▷ See Algorithm 3.
            end for
        end if
        Create new equivalence class for a_C.
        state_cayley_table ← AddElementToStateCayleyTable(state_cayley_table, w, a_C).    ▷ See Algorithm 3.
    end if
    candidate_cayley_table_elements ← SearchForNewCandidates(state_cayley_table, w, candidate_cayley_table_elements).    ▷ See Algorithm 4.
end while
return state_cayley_table
```

Algorithm 2 Generate action Cayley table.

```
Require: state_cayley_table.
action_cayley_table ← an empty square matrix with the dimensions of state_cayley_table, with rows and columns labelled by the rows and columns of state_cayley_table.
for row_label in action_cayley_table do
    for column_label in action_cayley_table do
        a_C ← column_label ∘ row_label.
        ec_label ← label of equivalence class containing a_C.
        state_cayley_table[row_label][column_label] = ec_label.
    end for
end for
return: state_cayley_table.
```

Displaying the algebra We display the algebra in two ways: (1) a *w-state Cayley table*, which shows the resulting state of applying the row element to w followed by the column element (*i.e., w-state Cayley table value = column label $*$ (row label $*$ w)*), and (2) an *action Cayley table*, which shows the resulting element of the algebra when the column element is applied to the left of the row element (*i.e., action Cayley table value = column element $\circ$ row element*).

Algebra properties We also check the following properties of the algebra algorithmically: (1) the presence of identity, including the presence of left and right identity elements separately, (2) the presence of inverses, including the presence of left and right inverses for each element, (3) associativity, (4) commutativity, and (5) the order of each element in the algebra. For our algorithm to successfully generate the algebra of a world, the world must contain a finite number of states, the agent must have a finite number of minimum actions, and all the transformations of the world must be due to the actions of the agent.

3.3.1. *Example*

For our example world $\mathscr{W}_c$, the equivalence classes shown in Fig. 5 - those labelled by 1, R, and U - are the only equivalence classes in $A/\sim$. The w-state Cayley table in Table 2 shows the final world state reached after the following operation: table entry = column element $*$ (row element $*$ w).

The w-action Cayley table in Table 3 shows the equivalent action in $A/\sim$ for the same operation as the w-state Cayley table: [table entry] $* w =$ column element $*$ (row element $* w$) for all $w \in W$.

Table 2
w_0 state Cayley table for $A/\sim$.

$A/\sim$	1	D	L	RU
1	w_0	w_2	w_1	w_3
D	w_2	w_0	w_3	w_1
L	w_1	w_3	w_0	w_2
RU	w_3	w_1	w_2	w_0

Table 3
Action Cayley table for $A/\sim$.

$A/\sim$	1	D	L	RU
1	1	D	L	RU
D	D	1	RU	L
L	L	RU	1	D
RU	RU	L	D	1

Table 4
Action Cayley table equivalence classes.

$\sim$ equivalence class label	$\sim$ equivalence class elements
1	$1, 11, DD, LL, RU\,RU, \ldots$
D	$D, D1, 1D, RU\,L, L\,RU, \ldots$
L	$L, L1, RU\,D, 1L, D\,RU, \ldots$
RU	$RU, RU1, LD, DL, 1RU, \ldots$

Table 5
Abstract action Cayley table for $A/\sim$.

$A/\sim$	1	2	3	4
1	1	2	3	4
2	2	1	4	3
3	3	4	1	2
4	4	3	2	1

Table 6
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	Y
Identity	Y
Inverse	Y
Associative	Y
Commutative	Y

The choice of the equivalence class label in Table 4 is arbitrary; it is better to think of each equivalence class as a distinct element as shown in the Cayley table in Table 5.

There are four elements in the action algebra, therefore, if the agent learns the relations between these four elements, and then it has complete knowledge of the transformations of our example world.

Properties of $A/\sim$ algebra The properties of the $A/\sim$ algebra are displayed in Table 6 and show that $A/\sim$ is a commutative group, where the no-op action is the identity, and all elements are their own inverses. Since the action algebra of our example world is a group, it can be described by SBDRL. The order of each element is given by Table 7.

3.4. Conditions for SBDRL to apply

To simplify the problem, we only consider worlds where the transformations of the world are only due to the actions of an agent for the remainder of this paper unless otherwise stated. Therefore, we will only consider worlds with $D = D_A$.

Table 7
Order of elements in $A/\sim$.

Element	Order
1	1
D	2
L	2
RU	2

To be a group, $A/\sim$ must satisfy the properties of (1) identity, (2) associativity, (3) closure, and (4) inverse. The identity and associative properties are satisfied by Proposition 3.4 and Proposition 3.5 respectively. For the closure property to be satisfied, the following condition is sufficient:

World condition 1 *(Unrestricted actions). For any action $a \in A$ and for any world state $w \in W$, there exists a transition $d \in D_A$ with $d : w \xrightarrow{a} t(d)$. In other words, $a * w$ is defined for all $a \in A$ and all $w \in W$.*

The inverse property of a group (4) is stricter than our current definition of the reversibility of an action. For the structure $A/\sim$ to have the inverse property, each element (action) in $A/\sim$ must not only be reversible from each starting state, but additionally, the inverse of a given element in $A/\sim$ must be the same for each starting state; for example, if a' is the inverse of a from a state $w \in W$ ($a' \circ a * w = w$), then a' must be the inverse of a from all states in W.

World condition 2 *(Inverse actions). For each $a \in A/\sim$, there exists an $a' \in A/\sim$ such that $a' \circ a * w = w$ and $a \circ a' * w = w$ for all $w \in W$.*

Proposition 3.6. *If the world satisfies world conditions 1 and 2 then $(A/\sim, \circ)$ is a group.*

Proof. Totality is given by world condition 1. Associativity is given by Proposition 3.5. Identity element given by Proposition 3.4. Inverse element is given by world condition 2. □

Proposition 3.7. *If the world obeys world conditions 1 and 2, then $*: (A/\sim) \times W \to W$ is a left group action.*

Proof. We have already established that $(A/\sim, \circ)$ is a group (Proposition 3.6). Therefore, to show that $*$ is a left group action we only have to prove the group action conditions of (a) identity and (b) compatibility. Consider an arbitrary world state $w \in W$.

(a) $[1]_\sim * w = 1 * w = 1_w * w = w$.

(b) We need to show that $a' * (a * w) = (a' \circ a) * w$. Because actions are unrestricted in W, for any $w \in W$, there exists the transitions $d_1 : w \xrightarrow{a} t(d_1)$ and $d_2 : t(d_1) \xrightarrow{a} t(d_2)$. $\Longrightarrow$ there exists the transition $(d_2 \circ d_1) : w \xrightarrow{a' \circ a} t(d_2)$. Therefore, $(a' \circ a) * w = t(d_2)$. Using the transitions d_1 and d_2 for the LHS of the condition, $a' * (a * w) = a' * t(d_1) = t(d_2)$. □

From Proposition 3.7, if a world satisfies world conditions 1 and 2, then the transformations of that world can be fully described using SBDRL (*i.e.*, $*: (A/\sim) \times W \to W$ is a group action).

Proposition 3.8. *If $*: (A/\sim) \times W \to W$ is a group action, then world condition 1 is satisfied.*

Proof. Since a group action is a full operation by definition, world condition 1 is satisfied. □

Proposition 3.9. *If $*: (A/\sim) \times W \to W$ is a group action, then world condition 2 is satisfied.*

Proof. If $*$ is a group action, then $A/\sim$ is a group. If $A/\sim$ is a group, then for each $a \in A/\sim$ there is an inverse element a^{-1} such that (1) $a^{-1} \circ a = 1$ and (2) $a \circ a^{-1} = 1$.

For an arbitrary state $w \in W$, $a^{-1} \circ (a * w) = a^{-1} \circ (a * w)$, therefore $a^{-1} \circ (a * w) = (a^{-1} \circ a) * w$ from the group action compatibility condition, therefore $a^{-1} \circ (a * w) = 1 * w$ from (1), therefore (3) $a^{-1} \circ (a * w) = w$ from the group action identity condition.

Similarly, for an arbitrary state $w \in W$, $a \circ (a^{-1} * w) = a \circ (a^{-1} * w)$, therefore $a \circ (a^{-1} * w) = (a \circ a^{-1}) * w$ from the group action compatibility condition, therefore $a \circ (a^{-1} * w) = 1 * w$ from (2), therefore (4) $a \circ (a^{-1} * w) = w$ from the group action identity condition.

(3) and (4) together are world condition 2. □

Proposition 3.10. *If a world does not satisfy world condition 1, then $*$ is not a group action.*

Proof. If a world does not satisfy world condition 1, then there exists some world state $w \in W$ and some action $a \in A$ such that $a * w$ is undefined. Therefore, for $[a] \in A/\sim$, $[a] * w$ is undefined, and so $*: (A/\sim) \times W \to W$ is a partial operation and so not a group action. □

Proposition 3.11. *If a world does not satisfy world condition 2, then $A/\sim$ is not a group and so $*$ is not a group action.*

Proof. If a world does not satisfy world condition 2, then there exists an $a \in A/\sim$ such that there is no $a' \in A/\sim$ for which $a' \circ a * w = w$ or $a \circ a' * w = w$ for all $w \in W$. **Proof by contradiction.** Assume a has an inverse $a'' \in A/\sim$. Therefore, $a'' \circ a \sim 1$ and $a \circ a'' \sim 1$. Since $1 * w = w$ for all $w \in W$, $a'' \circ a * w = w$ and $a \circ a'' * w = w$ for all $w \in W$, which is a contradiction. □

Proposition 3.7 shows that world conditions 1 and 2 are sufficient conditions for $*$ to be a group action, while Propositions 3.10 and 3.11 show that world conditions 1 and 2 are necessary conditions for $*$ to be a group action. Since world conditions 1 and 2 are sufficient and necessary conditions for $*$ to be a group action and therefore $A/\sim$ to be a group, these conditions give a characterisation of the worlds with transformations (due to the actions of an agent) that can be fully described using SBDRL; in other words, if the transformations of a world can be fully described using SBDRL then that world satisfies world conditions 1 and 2, and if a world satisfies world conditions 1 and 2 then the transformations of that world can be fully described using SBDRL.

3.5. Action-homogeneous worlds

Given that the algebraic structure underlying symmetric representations in SBDRL are groups, the (extra) condition of homogeneity directly applies:

World condition 3 *(Action homogeneity). For every pair $(w_1, w_2) \in W^2$, there exists a bijective map $\sigma_{(w_1,w_2)} : W \to W$ such that $\sigma_{(w_1,w_2)}(w_1) = w_2$ and such that:*

1. *for every $d \in D_A$ with $d : s(d) \xrightarrow{a} t(d)$, there exists a $d' \in D_A$ with $d' : \sigma_{(w_1,w_2)}(s(d)) \xrightarrow{a} \sigma_{(w_1,w_2)}(t(d))$;*
2. *for every $d \in D_A$ with $d : s(d) \xrightarrow{a} t(d)$, there exists a $d' \in D_A$ with $d' : \sigma^{-1}_{(w_1,w_2)}(s(d)) \xrightarrow{a} \sigma^{-1}_{(w_1,w_2)}(t(d))$.*

World condition 3 means that action sequences have the same result for any initial world state. Essentially, this means that the world looks the same from any world state with respect to the relationships of actions. We call worlds with world condition 3 *action-homogeneous worlds*.

In this section, we have presented a mathematical formalism for describing transitions due to the actions of an agent between world states and an algorithm that generates them. We have then laid out an example world and shown that, after applying an equivalence relation, the algebra of the agent's actions in this world forms a group. We have also characterised the worlds with transformations that can be fully described by SBDRL by giving world conditions that are sufficient and necessary for the algebra of the transformations of that world to be a group. In the next section, we present transformations relevant to reinforcement learning that do not form groups and are thus beyond the expressive power of SBDL but that can be formulated using our framework.

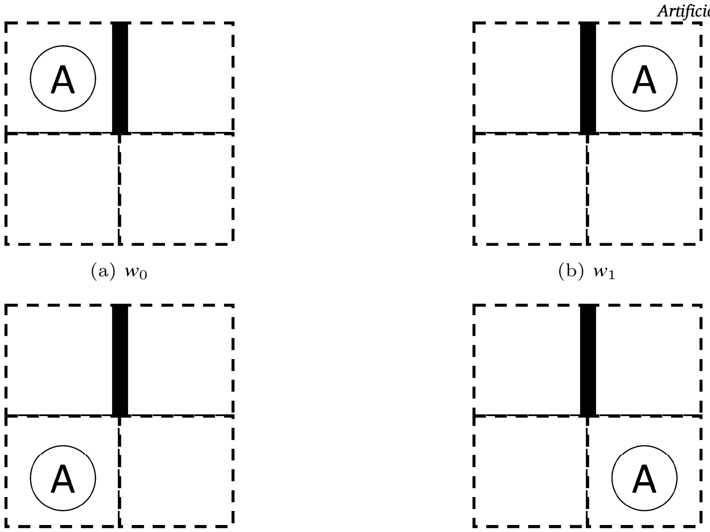
4. Beyond SBDRs

In the previous section, we showed under which conditions, those of forming a (symmetry) group, SBDRL representations can be expressed within our framework, as defined in Section 2. We will now consider worlds that are accounted for using our framework but that do not satisfy either world condition 1 or world condition 2, that is, transformations that do not form a group action and for which the assumption of action-homogeneous worlds does not necessarily hold. We have selected examples with features that are common in simple reinforcement learning scenarios to illustrate the potential of our approach in extending SBDRL to symmetries that go beyond group actions.

In order to do so, we use two methods of treating actions that are not allowed to be used in certain world states (e.g., the agent trying to move through a wall or eat a consumable in a state where the agent is not in the same location as the consumable). Method 1 (Sections 4.1.1 and 4.1.3) lets the agent select the actions but any actions that would have been undefined in a state w have the same effect in w as the agent performing the identity action 1. Method 2 (Sections 4.2.1 and 4.2.2) does not let the agent select these actions and so considers them as undefined; this violates world condition 1. These two treatments of actions are common in reinforcement learning. We employ the computational methods outlined in Section 3.3 to generate the action algebras of the agent in these worlds and provide evidence for our statements.

4.1. Worlds without inverse actions

In this section, we consider worlds that do not necessarily satisfy world condition 2 but do satisfy world condition 1.



(a) w_0 (b) w_1

(c) w_2 (d) w_3

Fig. 6. The world states of a cyclical 2×2 grid world $\mathscr{W}_{wall}$ with a wall, where changes to the world are due to an agent moving either up, down, left, or right. The position of the agent in the world is represented by the position of the circled A. The treatment of the wall is explained when needed.

Table 8
Each entry in this table shows the outcome state of the agent performing the action given in the column label when in the world state given by the row label.

	1	U	D	L	R
w_0	w_0	w_2	w_2	w_1	w_0
w_1	w_1	w_3	w_3	w_1	w_0
w_2	w_2	w_0	w_0	w_3	w_3
w_3	w_3	w_1	w_1	w_2	w_2

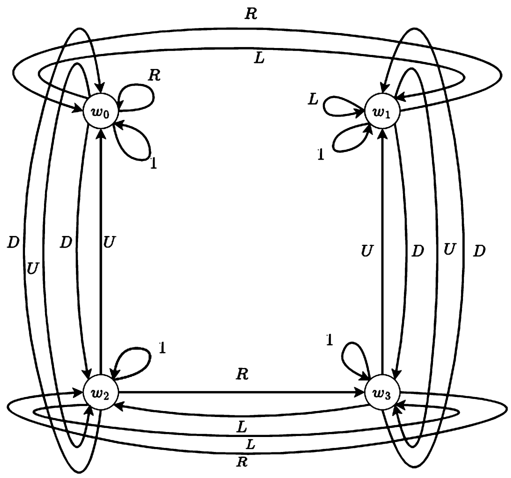


Fig. 7. A transition diagram of the labelled transitions in Table 8.

Proposition 4.1. *Consider a world $\mathscr{W}$ with a set W of world states and containing an agent with a set A of actions. If $\mathscr{W}$ satisfies world condition 1, then $*\colon (A/\sim) \times W \to W'$, where $W' \subseteq W$, is the action of a monoid $A/\sim$ on W.*

Proof. (1) Totality of $A/\sim$ is given by world condition 1. (2) Associativity of $A/\sim$ is given by Proposition 3.5. (3) Identity element of $A/\sim$ is given by Proposition 3.4. Since $A/\sim$ satisfies properties (1), (2), and (3), $A/\sim$ is a monoid.

$*$ is defined for any $a \in A$ and $w \in W$, therefore $*$ is a monoid action. $\square$

Table 9
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	Y
Identity	Y
Inverse	N
Associative	Y
Commutative	Y

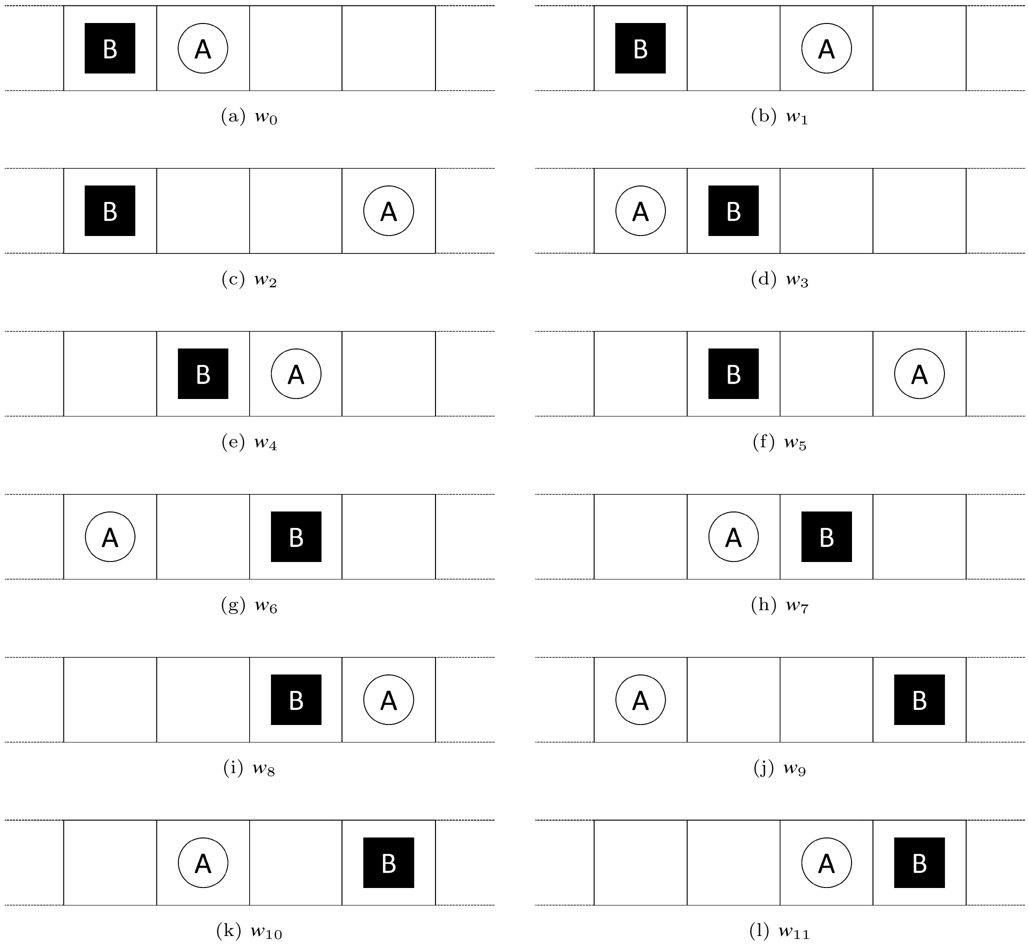


Fig. 8. World states of a world containing an agent and a movable block.

Remark 4.1.1. *If all actions are reversible in $\mathscr{W}$ then $W' = W$. If any action is irreversible in $\mathscr{W}$ then $W' \subset W$.*

4.1.1. Example 1: reversible action-inhomogeneous world

To turn the 2×2 cyclical grid world $\mathscr{W}_c$ used previously from a reversible action-homogeneous world to a reversible action-inhomogeneous world we add a wall to the world as shown in Fig. 6 to give world $\mathscr{W}_{wall}$.

This wall is said to *restrict* the actions of the agent. We say that actions restricted by the wall affect the world in the same way as the identity action; for example, if the agent is directly to the left of a wall and performs the 'move to the right' action, then this action is treated like the identity action and so the state of the world does not change (see Table 8 and Fig. 7).

Properties and structure of $A/\sim$ The action Cayley table for this world with the identity treatment of the walls contains 26 elements. As shown in Table 9, $A/\sim$ is a monoid.

Adding a single wall to the world has massively increased the complexity of the transition algebra of the world. While the transition algebra of $\mathscr{W}_c$ has four elements, the transition algebra of $\mathscr{W}_{wall}$ with restricted actions treated as identity actions contains 26 elements.

The restrictiveness of the group inverse condition should be noted. For $\mathscr{W}_{wall}$ with restricted actions treated as identity actions, every action is reversible from a particular state $w \in W$. However, the action that takes a back to its starting state is not necessarily

Table 10
Each entry in this table shows the outcome state of the agent performing the action given in the column label when in the world state given by the row label.

	1	L	R
w_0	w_0	w_9	w_1
w_1	w_1	w_0	w_2
w_2	w_2	w_1	w_3
w_3	w_3	w_5	w_7
w_4	w_4	w_0	w_5
w_5	w_5	w_4	w_3
w_6	w_6	w_8	w_7
w_7	w_7	w_6	w_{11}
w_8	w_8	w_4	w_6
w_9	w_9	w_8	w_{10}
w_{10}	w_{10}	w_9	w_{11}
w_{11}	w_{11}	w_{10}	w_2

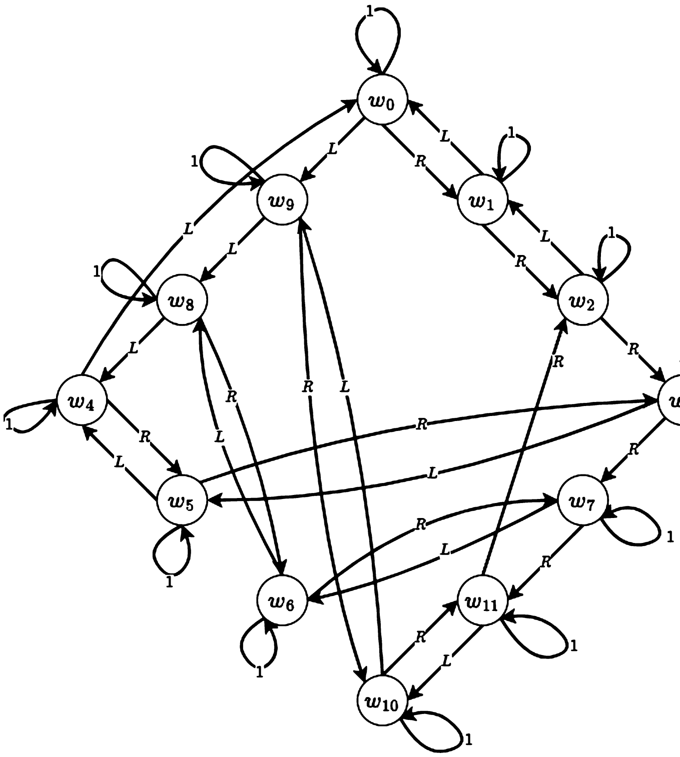


Fig. 9. A transition diagram of the labelled transitions in Table 10.

the same any starting state $w \in W$. For the inverse property to be present, the inverse for each element must be the same from any starting state (*i.e.*, the inverse must be independent of the starting state). $\mathscr{W}_{wall}$ with restricted actions treated as identity actions is proof that it is possible to have a world where all actions are reversible but for some of those actions to not have an inverse action.

4.1.2. Example 2: reversible action-inhomogenous world without walls

Consider a world $\mathscr{W}_{block}$ with the world states in Fig. 8 and with movement along a single 1D cyclical axis with a movable block. If the agent is in the location directly to the left of the block and moves into the block, the block moves one location in the direction of the agent's movement and the agent moves into the location previously occupied by the block (see Table 10 and Fig. 9).

The action Cayley table for $\mathscr{W}_{block}$ contains 17 elements. As shown by Table 11, $A/\sim$ is a monoid.

Remark 4.1.2. *Adding a wall or a moveable block breaks the symmetry of the original* 2×2 *cyclical world* $\mathscr{W}_c$*; this manifests as the action algebra of the world being much more complex.*

Table 11
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	Y
Identity	Y
Inverse	N
Associative	Y
Commutative	N

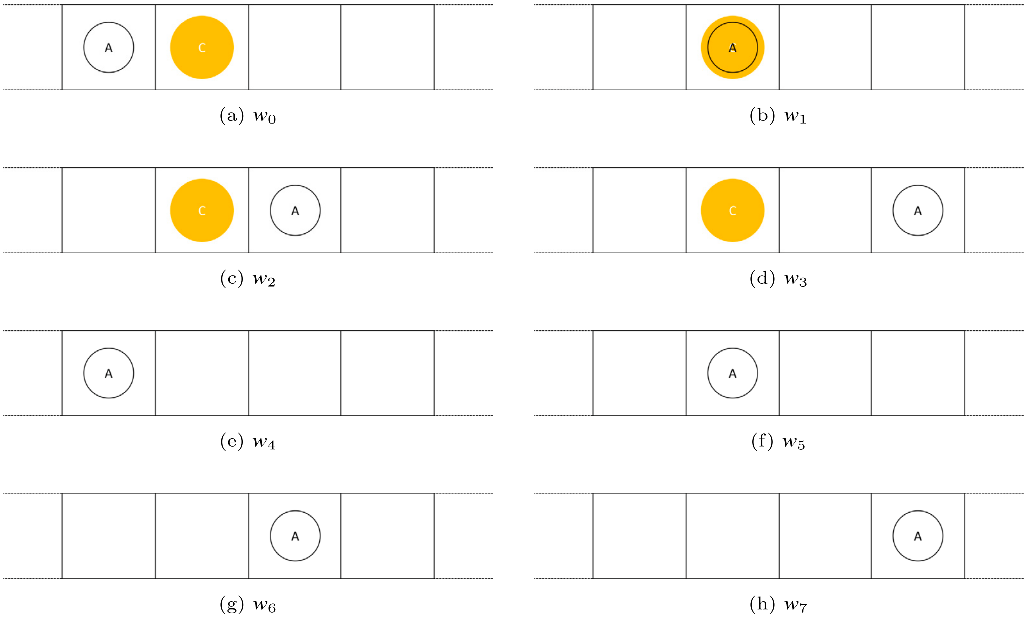


Fig. 10. World states of a world containing an agent and a consumable.

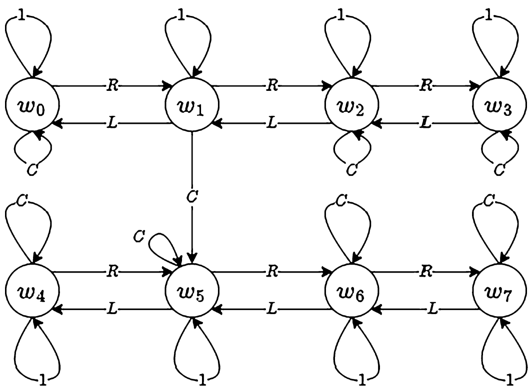


Fig. 11. Minimum action network for world $\mathscr{W}_{consumable}$.

4.1.3. *Example 3: irreversible inhomogeneous actions*

This example explores the transition algebras of worlds that are action-inhomogeneous and contain irreversible actions.

Consider a world $\mathscr{W}_{consumable}$ with movement along a single 1D cyclical axis with a single consumable. Let this world also contain an agent that can move left, right or consume the consumable if the agent is in the same place as the consumable (see Fig. 10).

There is not a consumable in every state, therefore if the agent performs the consume action in any state except w_1 then the effect will be the same as if the agent had performed the no-op action $1 \in A/\sim$ (see Fig. 11).

The action Cayley table for world $\mathscr{W}_{consumable}$ with restricted actions treated as identity actions contains 64 elements. As shown in Table 12, $A/\sim$ is a monoid.

Table 12
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	Y
Identity	Y
Inverse	N
Associative	Y
Commutative	N

Table 13
Each entry in this table shows the outcome state of the agent performing the action given in the column label when in the world state given by the row label.

	1	U	D	L	R
w_0	w_0	w_2	w_2	w_1	undefined
w_1	w_1	w_3	w_3	undefined	w_0
w_2	w_2	w_0	w_0	w_3	w_3
w_3	w_3	w_1	w_1	w_2	w_2

Table 14
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	N
Identity	Y
Inverse	N
Associative	Y
Commutative	N

Remark 4.1.3. *Considering this example, we note that the irreversible action moves from one 'reversible plane' to another. Perhaps we can treat an irreversible action as having a 'reversible action affecting part' and a 'world affecting part'; in this example, the reversible action affecting part would be the identity action since the reversible action network is unchanged by the irreversible action.*

4.2. Worlds without inverse actions or unrestricted actions

In this section, we consider worlds that do not necessarily satisfy world condition 2 or world condition 1.

Proposition 4.2. *Consider a world $\mathscr{W}$ with a set W of world states and containing an agent with a set A of actions. $*: (A/\sim)\times W \to W'$, where $W' \subseteq W$, is the action of a small category $A/\sim$ on W.*

Proof. (1) Associativity of $A/\sim$ is given by Proposition 3.5. (2) Identity element of $A/\sim$ is given by Proposition 3.4. Since $A/\sim$ satisfies properties (1) and (2), $A/\sim$ is a small category. Therefore $*$ is the action of a small category. □

Remark 4.2.1. *We can consider two equivalent perspectives of $*$:*

1. *$A/\sim$ is a small category and $*$ is a full action of $A/\sim$ on W.*
2. *$A/\sim$ is a monoid and $*$ is a partial action of $A/\sim$ on W.*

4.2.1. Example 1: reversible action-inhomogeneous world

We once again consider the world $\mathscr{W}_{wall}$ (see Fig. 6 for world states); however, now instead of treating restricted actions like the identity action, we *mask* the restricted actions. Masking restricted actions involves not allowing the agent to perform the restricted actions - the restricted actions are hidden (masked) from the agent - and so, mathematically, we treat the restricted actions as *undefined*; for example, if $\mathscr{W}_{wall}$ is in state w_0, then the agent cannot perform the R action because $R * w_0$ is undefined (see Table 13 and Fig. 12.

The action Cayley table for $\mathscr{W}_{wall}$ with the masked treatment of the walls contains 59 elements. As shown in Table 14, $A/\sim$ is a small category.

4.2.2. Example 2: irreversible action-inhomogeneous world

We will now apply the masking treatment of restricted actions to the world $\mathscr{W}_{consumable}$ (see Fig. 13 for world states); for example, if $\mathscr{W}_{consumable}$ is not in state w_1 then the consume action is undefined.

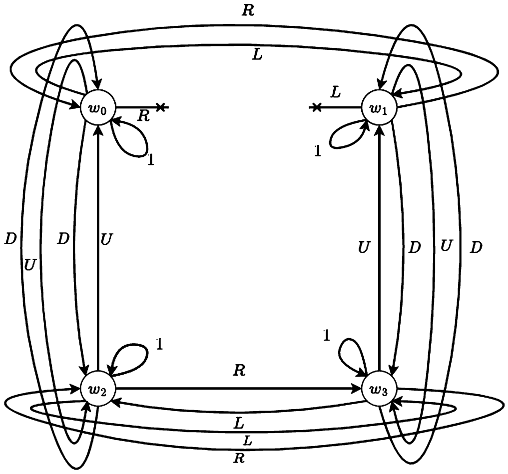


Fig. 12. A diagram of the labelled transitions in Table 13.

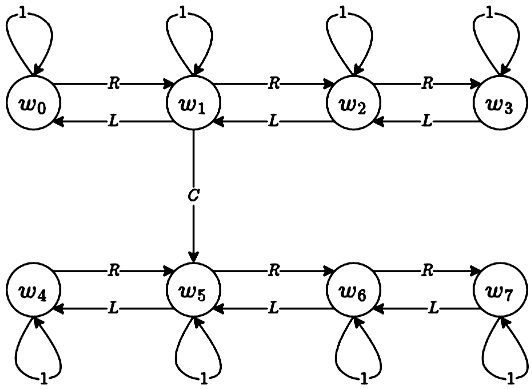


Fig. 13. Minimum action network for a world containing an agent and a consumable.

Table 15
Properties of the $A/\sim$ algebra.

Property	Present?
Totality	N
Identity	Y
Inverse	N
Associative	Y
Commutative	N

The action Cayley table for $\mathscr{W}_{consumable}$ with the masked treatment of the walls contains 20 elements. As shown in Table 15, $A/\sim$ is a small category.

Remark 4.2.2. *For any world, once the agent has the action Cayley table for any initial world state it can produce the action Cayley table for any other initial world state by applying an action that transitions from the old initial world state to the new initial world state to every element of the action Cayley table.*

Proposition 4.3. *The algebra $A/\sim$ of the actions of an agent depends on the treatment of the actions of the agent (e.g., masked treatment vs identity treatment) even if the world states remain the same.*

Proof. $A/\sim$ for the identity treatment of $\mathscr{W}_{wall}$ contains 26 elements, while $A/\sim$ for the masked treatment of $\mathscr{W}_{wall}$ contains 59 elements.

Additionally, $A/\sim$ for the identity treatment of $\mathscr{W}_{consumable}$ contains 64 elements, while $A/\sim$ for the masked treatment of $\mathscr{W}_{consumable}$ contains 20 elements. $\square$

We have also demonstrated that neither the identity treatment nor the masked treatment produces simpler algebra in every world. For $\mathscr{W}_{wall}$, the identity treatment contains fewer elements than the masked treatment (26 elements vs 59 elements), while for $\mathscr{W}_{consumable}$, the masked treatment contains fewer elements than the identity treatment (20 elements vs 64 elements).

In this section, we have selected several standard reinforcement learning scenarios that go beyond the expressive power of the symmetries represented in SBDRL as groups and that can nevertheless be accounted for using our general framework. It was not the purpose of this section to identify which algebraic structures would correspond with such representations, say semi-groups or groupoids for instance, but rather to establish a proof of concept, through examples generated using Algorithms 1 and 2, that SBDRL is not robust or generalisable enough and that our framework subsumes and extends it to bring about substantial representations.

5. Generalising SBDRL

So far, we have introduced a formal framework that allows agents to learn richer representations than those of SBDRL. In order to do so, we have proved in Section 3 that SBDRL is subsumed by our framework, that is, that the symmetric disentangled representations that SBDRL accounts for are also derivable using a generative algorithm of the corresponding Cayley tables within our approach. In Section 4, we have extended our study to simple yet challenging reinforcement learning cases where the conditions for the formation of groups, intrinsic to SBDRL, are not satisfied and that fall nevertheless under the remit of our framework. In other words, we have proved that our framework is more powerful than SBDRL and that it allows the representation of world transitions that can be structured in any algebra, in principle. Although formal, this was nevertheless an exploratory exercise, and, to express it with mathematical rigour we recur to category theory in this section. In order to do so, after some preliminary definitions, we first generalise SBDRL's equivariance condition categorically for both single and, importantly, multi-object categories, and then the notion of disentanglement for transformations that cannot be described by groups. That is, unlike in Section 4, the purpose of Section 5 is not to present further proof cases of the expressive power of our framework nor to revisit the examples in the previous section categorically (it would be impractical to go through the many algebraic structures that might result from our framework within and beyond SBDRL) but rather to provide a solid, formal characterisation that, in turn, broadens the rationale and capabilities of the original framework.

A fundamental result in category theory is the Yoneda Lemma, which posits that the properties of mathematical objects are completely determined by their relationships to other objects [53,54]. This result is similar to the shift in perspective in Artificial Intelligence representations from studying world states to gaining insight into their structures by considering how they are transformed by the dynamics of the environment they are immersed in, including the actions of agents. Due to the Yoneda Lemma, category theory already incorporates this approach of considering the transformation properties of objects built in, which arguably makes it the natural choice for their representation. More generally, category theory has been extensively used in computer science in the analysis of relational databases and functional programming [55,56]. Let us track back the main argument: generally speaking, symmetries define invariance, that is, impunity to possible alterations. Interestingly, the fact that the parts that are related by means of an equivalence relation correspond to the family of operations transforming the parts into each other while leaving the whole invariant satisfies the conditions for constituting a group. Consequently, it has traditionally been assumed that group theory is the language of symmetries and indeed groups have been customarily used to exploit symmetries in mathematics and physics, from E. Galois' studies of the structures that underlie the number and form of the solutions for equations of arbitrary degrees, to the formulation of the Standard Model that classifies all elementary particles and their interactions according to their flavour, charge and colour symmetries as the $SU(3) \bigotimes SU(2) \bigotimes U(1)$ group.

Now, a group is simply a category with a single object where every morphism is an isomorphism, that is, the object only relates to itself. Thus, although groups are certainly sufficient to formalise homogeneous structures, there are nonetheless plenty of transformations that exhibit what we clearly recognise as symmetries that are nontrivial (think of a bowling ball; it may not display the symmetries of a perfect sphere, such as a basketball, but it nonetheless poses symmetries that can be identified and exploited). To detect such symmetries, we need to "zoom out", and this is precisely why adopting a categorical approach in representation learning makes sense: category theory does consider objects, but its focus is on the relationships (morphisms) between those objects, rather than the internal structure of the objects themselves. The axioms of a category do not require objects to have elements or any specific internal structure. Category theory thus becomes the ideal tool for the study of structures that show partial symmetries and symmetries that apply to multiple objects, and those that result in higher-dimensional, increasingly more abstract categories. In other words, the ontology of algebraic groups is suitable to express symmetric structures that can be represented at the object-level in a straightforward way, but does not scale to other types that benefit from viewing the problem from a relation-level. Using the framework of Section 2, in Section 4 we have identified several symmetric transformations that cannot be expressed in groups, the structure favoured by SBDRL. In this section, we formalise the conditions that accommodate such structures as categories. Our claim is that the insights provided in this section might be taken as the starting point for a deeper exploration of category theory as the formalism in which to establish and derive symmetries beyond group representations in reinforcement learning and, more generally, Artificial Intelligence [57,58].

5.1. Preliminaries

Technically, a category consists of objects connected by arrows, which represent structure-preserving maps between the objects. Importantly, we can also relate categories via functors, which are ways to transform a category into another category while preserving the relationships between the objects and arrows of the domain category. In turn, natural transforms are ways to replace one functor with another while preserving the structure of the first. That is, we can iteratively construct higher-order categories, where (*n*)-categories become objects of (*n+1*)-categories. It is worth emphasizing that (a) the fundamental concept is the concept of morphism (transitions in our case) since an object is in fact defined as its collection of identity morphisms; (b) in relating categories these do not need to be of the same class, rather the power of category theory becomes visible when, through a process of categorisation, that is, by forgetting the details of its objects, we learn properties of a category by relating it to other category whose structure is well-known (similarly, we can recover the details of the domain category by a process of decategorisation); (c) as a result, in practice, categories provide a rich ontology to represent the concept of similarity beyond strict equivalence. That is, category theory allows us to compare (and build) a great variety of structures.

We will now introduce relevant category theory concepts. The reader is referred to [59] for a more detailed account of category theory and its uses.

Definition 5.1 *(Category). A category C consists of a class of objects, denoted $\boldsymbol{Ob}(C)$, and, for each pair x, y of objects, a class of morphisms $\alpha : x \to y$, denoted $C(x, y)$, satisfying the following:*

- ***Composition law.*** *Given two morphisms $\alpha \in C(x, y)$ and $\beta \in C(y, z)$ there exists a morphism $\beta \circ \alpha \in C(x, z)$ called the composition of α and β.*
- ***Existence of units.*** *Given an object x, there exists a morphism denoted by $1_x \in C(x, x)$ such that for any morphism $\alpha \in C(x, a)$, $\alpha \circ 1_x = \alpha$ and for any morphism $\beta \in C(b, x)$, $1_x \circ \beta = \beta$.*
- ***Associativity.*** *Given three morphisms $\alpha \in C(x, y)$, $\beta \in C(y, z)$, $\gamma \in C(z, u)$, then the following associative law is satisfied: $\gamma \circ (\beta \circ \alpha) = (\gamma \circ \beta) \circ \alpha$.*

Definition 5.2 *(Isomorphism). A morphism $\alpha : x \to y$ in a category $\mathbf{C}$ is an isomorphism if there exists another morphism $\beta : y \to x$ in $\mathbf{C}$ such that $\beta \circ \alpha = 1_x$ and $\alpha \circ \beta = 1_y$. This can be denoted by $x \overset{\alpha}{\cong} y$.*

Definition 5.3 *(Group). A group is a category that has a single object and in which every morphism is an isomorphism (i.e., every morphism has an inverse).*

Definition 5.4 *(Small category). A category C is a small category if its collection of objects $\boldsymbol{Ob}(C)$ is a set, and the collection of morphisms $Hom_C(X, Y)$, where $X, Y \in \boldsymbol{Ob}(C)$ is also a set.*

Definition 5.5 *(Hom-set). Given objects x and y in a small category C, the hom-set $Hom_C(x, y)$ is the collection of all morphisms from x to y. A category is said to be small if each of its hom-sets is a set instead of a proper class.*

Definition 5.6 *((covariant) Functor). A functor is a structure-preserving map between two categories. For two categories $\mathcal{A}$ and $\mathcal{B}$, a functor $F : \mathcal{A} \to \mathcal{B}$ from $\mathcal{A}$ to $\mathcal{B}$ assigns to each object in $\mathcal{A}$ an object in $\mathcal{B}$ and to each morphism in $\mathcal{A}$ a morphism in $\mathcal{B}$ such that the composition of morphisms and the identity morphisms are preserved. A functor transforms objects and morphism from one category to another in a way that preserves the structure of the original category.*

More precisely, a functor $F : \mathcal{A} \to \mathcal{B}$ consists of two maps:

1. *A map that assigns an object $F(A)$ in $\mathcal{B}$ to each object A in $\mathcal{A}$.*
2. *A map that assigns a morphism $F(f)$ in $\mathcal{B}$ to each morphism f in $\mathcal{A}$ such that:*
   - *(a) F respects composition for any two composable morphisms $f, g \in \mathcal{A}$, $F(fg) = F(f)F(g)$.*
   - *(b) F preserves identities: for any object A in $\mathcal{A}$, $F(id_A) = id_{F(A)}$.*

Definition 5.7 *(Functor category). For categories C and $\mathcal{D}$, the functor category denoted D^C or $[C, \mathcal{D}]$ is the category whose: (1) objects are functors $F : C \to \mathcal{D}$, and (2) morphisms are natural transforms between these functors.*

Definition 5.8 *(Natural transform). For categories C and $\mathcal{D}$ and functors $F, G : C \to \mathcal{D}$, a natural transform $\alpha : F \Rightarrow G$ between F and G is an assignment to every object x in C of a morphism $\alpha_x : F(x) \to G(x)$ in $\mathcal{D}$ such that for any morphism $f : x \to y$ in C, the following diagram commutes:*

$$\begin{tikzcd}
F(x) \arrow[r, "F(f)"] \arrow[d, "\alpha_x"'] & F(y) \arrow[d, "\alpha_y"] \\
G(x) \arrow[r, "G(f)"'] & G(y)
\end{tikzcd}$$

Definition 5.9 *(Delooped category). Given an algebraic structure A, we can construct the delooped category $\boldsymbol{B}A$ whose morphisms correspond to the elements of A with the relevant composition: $A \xrightarrow{deloop} \boldsymbol{B}A$. $\boldsymbol{B}$ id called the base of the category and contains the objects. The number of objects in $\boldsymbol{B}$ is the number of objects necessary for the morphisms of $\boldsymbol{B}A$ to correspond to the elements of A.*

Definition 5.10 *(Monoid). A monoid is a category with a single object.*

Definition 5.11 *(Categorical product). The categorical product of two categories C_1 and C_2 is the category $C_1 \times C_2$. The objects of $C_1 \times C_2$ are the pairs of objects (B_1, B_2), where B_1 is an object in C_1 and B_2 is an object in C_2. The morphisms in $C_1 \times C_2$ are the pairs of morphisms (f_1, f_2), where f_1 is a morphism in C_1 and f_2 is a morphism in C_2. The composition of morphisms is defined component-wise.*

Definition 5.12 *(Sub-functors of sub-categories). We can define a functor $F : C_1 \times C_2 \to C$ as follows:*

1. *For each object (B_1, B_2) in $C_1 \times C_2$, $F(B_1, B_2)$ is the object in C that corresponds to the pair (B_1, B_2).*
2. *For each morphism $(f_1, f_2) : (B_1, B_2) \to (B_1', B_2')$ in $G_1 \times G_2$, $F(f_1, f_2)$ is the morphism in G that corresponds to the pair (f_1, f_2).*

We can now define sub-functors F_1 of G_1 and F_2 of G_2 on an object B of G as follows:

1. *$F_1(g_1, B) = F(g_1, 1_{G_2})(B)$ for all $g_1 \in G_1$.*
2. *$F_2(g_2, B) = F(1_{G_1}, g_2)(B)$ for all $g_1 \in G_1$.*

We can decompose F into the sub-functors F_1 and F_2 as $F = F_1 \times F_2$, if there is a decomposition $B = B_1 \times B_2$ of B into two sub-objects B_1 and B_2 such that:

1. *For all $g_1 \in G_1$ and $B_2 \in G_2$, we have $F_1(g_1, B_1 \times B_2) = F_1(g_1, B_1 \times B_2$.*
2. *For all $g_2 \in G_2$ and $B_1 \in G_1$, we have $F_2(g_2, B_1 \times B_2) = B_1 \times F_2(g_2, B_2)$.*

5.2. The equivariance condition

In this section, we use category theory to generalise [1]'s equivariance condition. In doing so we show that the equivariance condition can apply to worlds where the actions of an agent cannot be fully described by groups.

5.2.1. Group equivariance in category theory

We will now convert [1]'s group equivariance condition into the language of category theory.

A left group action $G \times S \to S$ is a homomorphism from a group G to the group of bijections of a set S that G is acting upon. Since S is itself an object in the category **Set** of sets, and bijections from S to itself are the invertible morphisms in $hom_{\mathbf{Set}}(S, S)$, the left group action $G \times S \to S$ is an object of the category $\mathbf{Set}^G$ of covariant functors from G to **Set**.

Since the objects of the category $\mathbf{Set}^G$ are the functors from G to **Set**,[1] then the morphisms of $\mathbf{Set}^G$ are natural transforms between these functors. For the left action $A_1 : G \to \mathbf{Set}$ that maps the single object b of G to a set S_1 and the left action $A_2 : G \to \mathbf{Set}$ that maps the single object b of G to a set S_2, the natural transform $\eta : A_1 \to A_2$ has a single component $\eta_b : S_1 \to S_2$ because there is a single object b in G. Every morphism, which is a group element $g \in G$, must satisfy the naturality condition $\eta_b(g \cdot_{S_1} s) = g \cdot_{S_2} \eta_b(s)$ for all $s \in S_1$, where $\cdot_{S_1}$ denotes the action of G on set S_1 and $\cdot_{S_2}$ denotes the action of G on set S_2.

Now consider an agent with a set A of actions and a set Z of representation states in a world $\mathscr{W}_0$ that has a set W of world states and that satisfies world conditions 1 and 2. For $\mathscr{W}_0$, $A/\sim$ is a group. Since the set of world states W and the set of representation states Z are both acted on by the same group $A/\sim$, there are two group actions $*_W : (A/\sim) \times W \to W$ and $*_Z : (A/\sim) \times Z \to Z$, and both group actions give functors from $A/\sim$ to objects (W and Z) in the category **Set**. A structure-preserving map between these functors is a natural transform with the single component $\eta_b : W \to Z$ that satisfies $\eta_b(a *_W w) = a *_Z \eta_b(w)$ for all $w \in W$ and for all $a \in A/\sim$, where $*_W$ denotes the action of $A/\sim$ on set W and $*_Z$ denotes the action of $A/\sim$ on set Z. This component η_b is [1]'s equivariant condition ($g \cdot_Z f(w) = f(g \cdot_W w)$) in the language of category theory.

5.2.2. Equivariance for single-object categories

We will now take the category theory argument we used to derive the group equivariance condition for worlds where $*_W : (A/\sim) \times W \to W$ is a group action (reproducing [1]'s equivariant condition) and generalise our argument to worlds where $*_W : (A/\sim) \times W \to W$ is the (full) action of any algebraic structure $A/\sim$ that can be delooped to form a single-object category $\mathbf{B}(A/\sim)$.[2] Any single-object category is a monoid (Definition 5.10), therefore an equivariance condition for single-object categories will hold for any world where $A/\sim$ is a monoid. The derivation of this condition is trivially the same argument as given for the group action case since we did not require that the morphisms in the single-object category with morphisms giving $A/\sim$ be isomorphisms.

[1] The objects of the category $\mathbf{Set}^G$ are the maps from the morphisms in G to the morphisms in **Set**.

[2] Formally, in section 5.2.1 we follow the argument in this section and deloop the group $A/\sim$ to form the associated category $\mathbf{B}(A/\sim)$.

1. Setup Now consider an agent with a set A of actions and a set Z of representation states in a world $\mathscr{W}_1$ that has a set W of world states and where $A/\sim$ is a monoid with elements $a_1, a_2, \dots, a_n$.

2. Category of actions Let $\mathbf{B}(A/\sim)$ be the delooped category of $A/\sim$ and let $\mathbf{B} = \{b\}$ be the set of objects of $\mathbf{B}(A/\sim)$. The set of morphisms of $\mathbf{B}(A/\sim)$ is $A/\sim$.

3. Functors for actions on sets The action $*_W : (A/\sim) \times W \to W$ of $A/\sim$ on the set W gives a functor $\rho : \mathbf{B}(A/\sim) \to W$, where ρ encodes the properties of the algebraic structure of $(A/\sim)$. Similarly, the action $*_Z : (A/\sim) \times Z \to Z$ of $A/\sim$ on the set Z gives a functor $\tau : \mathbf{B}(A/\sim) \to Z$, where τ encodes the properties of the algebraic structure of $(A/\sim)$.

4. Structure-preserving morphism (natural transform) We want the property that $A/\sim$ acts on W and Z in the same way, so we want to preserve the structure between the functors ρ and τ. The objects of the functor category $\mathbf{Set}^{\mathbf{B}(A/\sim)}$ are the functors, including ρ and τ, from $\mathbf{B}(A/\sim)$ to **Set**; the morphisms (structure-preserving maps) between the objects of $\mathbf{Set}^{\mathbf{B}(A/\sim)}$ are natural transforms. The structure-preserving map between ρ and τ is the natural transform $\eta : W \to Z$ with the single component $\eta_b : W \to Z$ that satisfies $\eta_b(a *_W w) = a *_Z \eta_b(w)$ for all $w \in W$ and for all $a \in A/\sim$, where $*_W$ denotes the action of $A/\sim$ on set W and $*_Z$ denotes the action of $A/\sim$ on set Z.

5. Generalised equivariance condition The generalised equivariance condition for the single-object case is:

$$\eta_b(a *_W w) = a *_Z \eta_b(w) \text{ for all } w \in W \text{ and for all } a \in A/\sim.$$

In other words, the diagram

$$\begin{tikzcd}
w \arrow[r, "a*_W"] \arrow[d, dashed, "\eta_b"'] & a *_W w \arrow[d, dashed, "\eta_b"'] \\
\eta_b(w) \arrow[r, "a*_Z"] & \eta_b(a *_W w) = a *_Z \eta_b(w)
\end{tikzcd}$$

commutes.

We have now generalised [1]'s group equivariance condition to worlds where $*_W : (A/\sim) \times W \to W$ is a (full) monoid action. We have shown that the equivariance condition depends only on the number of objects in the delooped category of $A/\sim$, and so the equivariance condition for monoid action $*$ is structurally the same as for group action $*$. Our derived equivariance condition is valid in some worlds where agents can perform irreversible actions since monoids can have elements with no inverse; the inability of [1]'s original formalism to deal with irreversible actions has been explicitly stated [51, page 4]. In fact, from Proposition 4.1, this equivariance condition for the single-object category case is valid for any world that satisfies world condition 1.

5.2.3. Equivariance for multi-object categories

We will now generalise our argument to derive equivariance conditions for worlds where $*_W : (A/\sim) \times W \to W$ is the action of any algebraic structure $A/\sim$ that can be delooped to form any small category $\mathbf{B}(A/\sim)$. Let us adapt our argument for single-object categories from the previous section step-by-step to the multi-object category case:

1. Setup Consider an agent with a set A of actions and a set Z of representation states in a world $\mathscr{W}_2$ that has a set W of world states and where $A/\sim$ is a small category with elements $a_1, a_2, \dots, a_n$.

2. Category of actions Let $\mathbf{B}(A/\sim)$ be the delooped category of $A/\sim$ and let $\mathbf{B} = \{b_1, b_2, \dots, b_m\}$ be the set of objects of $\mathbf{B}(A/\sim)$. For the multi-object category case, the set of morphisms $f : b_i \to b_j$ of $\mathbf{B}(A/\sim)$ is not the elements $a_1, a_2, \dots, a_n$ of $A/\sim$ - the elements from the algebraic structure $A/\sim$ do not appear directly as components of morphisms; instead they inform how the objects and morphisms within the category interact. The morphisms $f : b_i \to b_j$ of $\mathbf{B}(A/\sim)$ can be thought of as arrows connecting object b_i to object b_j. Each morphism is then labelled by an element of $A/\sim$; this labelling indicates how the element from $A/\sim$ maps b_i to b_j within the categorical framework.[3] This is analogous to the treatment of transitions (analogous to morphisms) and actions (analogous to elements of $A/\sim$) given in Section 2.2.1.

3. Functors for actions on sets As before, the action $*_W : (A/\sim) \times W \to W$ of $A/\sim$ on the set W gives a functor $\rho : \mathbf{B}(A/\sim) \to W$, where ρ encodes the properties of the algebraic structure of $(A/\sim)$. Similarly, the action $*_Z : (A/\sim) \times Z \to Z$ of $A/\sim$ on the set Z gives a functor $\tau : \mathbf{B}(A/\sim) \to Z$, where τ encodes the properties of the algebraic structure of $(A/\sim)$. The functors ρ and τ now have components, $\rho(b_i)$ and $\tau(b_i)$, for each object $b_i \in \mathbf{B}$ as well as components, $\rho(f)$ and $\tau(f)$, for each morphism $f \in \mathbf{B}(A/\sim)$.

[3] In the single-object category case ($\mathbf{B} = b$), all morphisms are between the same object (they are *endomorphisms*) and so there is a one-to-one correspondence between these endomorphisms $f : b \to b$ and the elements of $A/\sim$; this means we can treat the elements of $A/\sim$ as the morphisms of $\mathbf{B}(A/\sim)$.

4. Structure preserving morphisms (natural transform) We again want the property that $A/\sim$ acts on W and Z in the same way, so we want to preserve the structure between the functors ρ and τ. The objects of the functor category $\mathbf{Set}^{\mathbf{B}(A/\sim)}$ are the functors, including ρ and τ, from $\mathbf{B}(A/\sim)$ to **Set**; the morphisms between the objects of $\mathbf{Set}^{\mathbf{B}(A/\sim)}$ are natural transforms. So we again want to preserve the structure between the functors ρ and τ using a natural transform $\eta : W \to Z$. However, since there are multiple objects in $\mathbf{B}(A/\sim)$, our natural transform η has a component for each object in **B**. For each object $b_i \in \mathbf{B}$, there is a component $\eta_{b_i} : W \to Z$ such that it satisfies $\eta_{b_i}(a *_W w) = a *_Z \eta_{b_i}(w)$ for all $w \in W$ and for all $a \in A/\sim$, where $*_W$ denotes the action of $A/\sim$ on set W and $*_Z$ denotes the action of $A/\sim$ on set Z.

Remark 5.2.1. *There is a different η_{b_i} for each object $b_i \in \mathbf{B}$.*

5. Generalised equivariance condition The generalised equivariance condition for the multi-object category case is a collection of equivariance conditions, one for each object $b_i \in \mathbf{B}$, with the corresponding natural transformation components η_{b_i} satisfying the condition:

$$\eta_{b_i}(a *_W w) = a *_Z \eta_{b_i}(w) \text{ for all } w \in W \text{ and for all } a \in A/\sim.$$

In other words, the diagram

$$\begin{tikzcd}
w \arrow[rr, "a*_W"] \arrow[d, dashed, "\eta_{b_i}"'] & & a *_W w \arrow[d, dashed, "\eta_{b_i}"'] \\
\eta_{b_i}(w) \arrow[r, "a*_Z"] & \eta_{b_i}(a *_W w) = a *_Z \eta_{b_i}(w) &
\end{tikzcd}$$

commutes.

This generalised equivariance condition ensures that the action of $A/\sim$ on W and Z is preserved consistently across all objects of $\mathbf{B}(A/\sim)$.

Remark 5.2.2. *For each object b_i, we have an associated natural transformation component $\eta_{b_i} : W \to Z$. This component represents how the action of $A/\sim$ on the set W relates to the set Z for the specific object b_i. In other words, η_{b_i} encodes how the algebraic structure $A/\sim$ interacts with that particular object b_i.*

We have generalised [1]'s group equivariance condition to worlds where $*_W : (A/\sim) \times W \to W$ is a (full) small category action or, equivalently, a partial monoid action. Our derived equivariance condition is valid in some worlds where agents have actions that are not defined by some world states.

5.3. Disentangling

We will now generalise the definition of disentanglement given by [1] to give a definition that works for worlds with transformations that cannot be described by groups. We prove that the equivalence condition can be disentangled.

5.3.1. Disentangling of group actions in Category Theory terms

In Category Theory, the group action $\cdot : G \times X \to X$, where G is a group and X is a set, gives a functor $\alpha : \mathbf{B}G \times \mathbf{Set} \to \mathbf{Set}$, where $\mathbf{B}G$ is the (single-object) delooped category of G and **Set** the category of sets. α satisfies:

1. $\alpha_1 = \mathrm{id}_X$, where 1 is the identity element of G.
2. $\alpha_{g_1} \circ \alpha_{g_2} = \alpha_{g_1 g_2}$ for all $g_1, g_2 \in G$.

The set X is an object in the category **Set**.

If G decomposes into the direct product of subgroups $G = G_1 \times \ldots \times G_n$, then $\mathbf{B}G$ decomposes into a categorical product of the relevant delooped categories $\mathbf{B}_i G_i$ as $\mathbf{B}G = \mathbf{B}_1 G_1 \times \ldots \times \mathbf{B}_n G_n$. The object of $\mathbf{B}_1 G_1 \times \ldots \times \mathbf{B}_n G_n$ is the n-tuple of objects $(b_1, \ldots, b_n)$ where each b_i is the (single) object in $\mathbf{B}_i$. The morphisms of $\mathbf{B}_1 G_1 \times \ldots \times \mathbf{B}_n G_n$ are the n-tuples of morphisms $(f_1, \ldots, f_n)$ where f_i is a morphism $f_i : \mathbf{B}_i G_i \to \mathbf{B}_i G_i$. The composition of morphisms is defined component-wise.

We say the functor $\alpha : G \times X \to X$ is *disentangled* with respect to a decomposition $G = G_1 \times \ldots G_n$ of G if:

1. There exists a decomposition $X_1 \times \ldots X_n$ and an isomorphism $X \cong X_1 \times \ldots X_n$.
2. α can be decomposed into sub-functors $\alpha = \alpha_1 \times \alpha_2 \times \ldots \times \alpha_n$, where each sub-functor $\alpha_i : (\mathbf{B}_i G_i) \times X_i \to X_i$ corresponds to the action of the subgroup G_i on the subspace X_i that is invariant under all other subgroups: $\alpha(g, x) = \alpha(g_1 \times \ldots \times g_n, x_1 \times \ldots \times x_n) = \alpha_1(g_1, x_1) \times \ldots \times \alpha_n(g_n, x_n)$.

If G has additional structure such as a vector space or a topological space, then we can consider a subcategory consisting of objects with that structure, and require the functors α and α_i to preserve that structure.

We will now extend the definition of disentangling given by [1] to worlds with transition algebras that are described by multi-object categories. First, we convert [1]'s definition of disentangling into category theory terms. We then extend this to any single-object category, and finally to any multi-object category.

The definition of a disentangled functor given previously extends to the case of any algebra, including those expressed as multi-object categories, by letting G be the algebra $A/\sim$ and letting $\mathbf{B}$ contain one or more objects. These changes do not change the definition of a disentangled functor given previously.

5.3.2. Disentangling and the equivariance condition

We now want to see how disentangling affects the equivariance condition. Specifically, we want to see if the natural transform η disentangles.

Let $\rho : \mathbf{B}(A/\sim) \to W$ and $\tau : \mathbf{B}(A/\sim) \to Z$ be disentangled functors of the actions $*_W : (A/\sim)\times W \to W$ and $*_Z : (A/\sim)\times Z \to Z$ respectively.

By definition, there are decompositions $W = \prod_{i=1}^{n} W_i$ and $Z = \prod_{i=1}^{n} Z_i$ where each W_i (respectively Z_i) is fixed by the sub-functor $\rho_{j\neq i}$ (respectively $\tau_{j\neq i}$) and is only affected by the functor ρ_i (respectively τ_i). Now let $\eta : W \to Z$ be a natural transform between ρ and τ (*i.e.*, for each $a, w \in (A/\sim)\times W$, we have $\eta(\rho(a, w)) = \tau(a, \eta(w))$). We want to show that η is itself disentangled with respect to the decomposition of $\mathbf{B}(A/\sim)$.

Let $w = (w_1, ..., w_n) \in W$ be a point in the decomposition of W, let $a = (a_1, ..., a_n) \in A/\sim$ be a point of decomposition of $\mathbf{B}(A/\sim)$, and let $a_i \in (A/\sim)_i$ be an element in the ith factor of $A/\sim$. Say we have a decomposition of η such that $\eta = (\eta_1, ..., \eta_n)$ where η_i is the ith component of η. Therefore, we have $\eta(w) = (\eta_1(w_1), ..., \eta_n(w_n))$ where $\eta_i(w) \in Z_i$. Consider

$$\begin{aligned}\eta(\rho(a_i, w)) &= \eta((\rho_1(a_i, w_1), ..., \rho_i(a_i, w_i), ..., \rho_n(a_i, w_n))) \\ &= (\eta_1(\rho_1(a_i, w_1)), ..., \eta_i(\rho_i(a_i, w_i)), ...\eta_n(\rho_n(a_i, w_n)))\end{aligned}$$

Since W_i is fixed by the action of $(A/\sim)_{j\neq i}$, we have $\rho_j(a_i, w_j) = w_j$ for $i \neq j$, and so:

$$\eta(\rho(a_i, w)) = (\eta_1(w_1), ..., \eta(\rho_i(a_i, w_i)), ..., \eta_n(w_n)) \tag{1}$$

Now consider

$$\tau(a_i, \eta(w)) = (\tau_1(a_i, \eta_1(w_1)), ..., \tau_i(a_i, \eta_i(w_i)), ..., \tau_n(a_i, \eta_n(w_n)))$$

Since Z_i is fixed by the action of $(A/\sim)_{j\neq i}$, we have $\tau_j(a_i, \eta(w_j)) = \eta(w_j)$ for $i \neq j$, and so:

$$\tau(a_i, \eta(w)) = (\eta_1(w_1), ..., \tau_i(a_i, \eta_i(w_i)), ..., \eta_n(w_n)) \tag{2}$$

We can see from combining equations (1) and (2) that if $\eta_i(\rho_i(a_i, w_i)) = \tau_i(a_i, \eta_i(w_i))$ (*i.e.*, the relevant diagrams commute for each component η_i of η), then η_i is disentangled with respect to the action of $(A/\sim)_i$ on W_i and Z_i, and therefore η is itself disentangled with respect to the decomposition of $A/\sim$.

We have shown that η can be decomposed into sub-natural transforms $\eta_1, ..., \eta_n$, where each η_i is a natural transform between sub-functions ρ_i and τ_i. Each $\eta_i : W_i \to Z_i$ preserves the structure of the corresponding subcategory of W and Z; therefore, if W_i and Z_i have additional structure, then the sub-natural transform η_i must preserve that structure. This opens up the possibility of a decomposition of $A/\sim$ and therefore W, Z, ρ, τ, and η such that some additional structure is confined to a single set of components $(A/\sim)_i$, W_i, Z_i, ρ_i, τ_i, and η_i.

Components of η when $\mathbf{B}(A/\sim)$ is a multi-object category If $\mathbf{B}(A/\sim)$ is a multi-object category, then the components of the sub-natural transform η_i between functors ρ_i and τ_i are $\eta_{i\,b_j} : \rho_i(b_j) \to \tau_i(b_j)$ for each object $b_j \in \mathbf{B}$.

In this section, we first generalised [1]'s equivariance condition using category theory; to do so, we converted our symmetry-based representation argument into category theory - this gave us insight into the relationship between the algebraic form and the categorical form. Category theory naturally generalises the group-equivariance condition given by [1] to any algebraic structure that has a categorical interpretation. We then showed that a form of the equivariance condition exists for a world with transformations that form *any* algebra. We also demonstrated that the equivariance condition is, in fact, a fundamental feature of category theory: the natural transform.

Next, we converted [1]'s definition of disentangling into category theory terms; category theory then provided a natural generalisation of the definition of disentangling to worlds with transformations that form any algebra. Finally, we explored the interplay between the generalised equivariance condition and the generalised definition of disentangling. We concluded that disentangled sub-algebras can each have their own individual equivariance sub-conditions, and thus, the learning of these sub-algebras, as well as their applications, can be treated independently. This result has important implications for learning algorithms; for example, since each disentangled subspace has its own individual equivariance condition, the learning of each subspace is also independent, and so different learning algorithms could be used on each disentangled subspace.

6. Discussion & conclusion

The question of what features of the world should be present in a ‘good’ representation that improves the performance of an artificial agent for a variety of tasks is key in representation learning. [1] proposes that the symmetries of the world are important structures that should be present in an agent’s representation of that world. They formalise their proposal using group theory as SBDRL, which is made up of a disentangling condition that defines transformations as commutative subgroups and a group equivariant condition.

In this paper, we have taken that programme one step further and posit that the relationships of transformations of the world due to the actions of the agent should be included in the agent’s representation of the world and not just the actions that form group symmetries. We then used this framework to derive and identify limitations in SBDRL. This approach reports two benefits: (1) it shows that the framework we put forward encompasses SBDRL, and (2) it identifies worlds where SBDRL cannot describe the relationships between the transformations of a world due to the actions of an agent. We use algorithmic methods, newly designed by us for this work, to extract the (not necessarily group) algebras of the actions of an agent in example worlds with transformations that cannot be fully described by SBDRL. We decided to use worlds that exhibit features commonly found in reinforcement learning scenarios because representation learning methods have been shown to improve the learning efficiency, robustness, and generalisation in such contexts. Finally, we use category theory to generalise core ideas of SBDRs, specifically the equivariance condition and the disentangling definition, to a much larger class of worlds than previously done, with more complex action algebras, including those with irreversible actions. We also propose that category theory appears to be a natural choice for the study of transformations of a world because it focuses on the transformation properties of objects per se, and the perspective that the properties of objects are completely determined by their relationship to other objects is a key result of category theory (the Yoneda Lemma).

The framework we have set out and its results have much room for expansion in future work, including the following: (1) How would we deal with transformations of the world that are not due to the actions of an agent? (2) How would partial observability affect the agent’s representations? (3) What effect would the use of continuous actions have? (4) What algebraic structures would be given by different equivalence relations? (5) Under what conditions can we disentangle reversible and irreversible actions? (6) Could our category theory generalisation of the SBDRL equivariance condition also be used to describe other uses of equivariance conditions in Artificial Intelligence, such as unifying the different equivariance conditions given by [25] through natural transforms? (7) How can our framework be used to develop better representation learning algorithms?

Despite such valid follow-up questions, the main contributions of our proposal stand clear: for agents to interact with their environment efficiently, it is paramount that they learn “good” representations, representations that take advantage of the symmetries of the transformations brought in by their own actions to reduce the task of data processing. For instance, if we learn that a square shows rotation and reflection symmetries, we can, in principle, operate directly in the world under such assumptions, rather than exploring the outcomes of every possible action. Whereas SBDRL and similar approaches have paved the way in formalising such transformations in the case of homogeneous groups, they fall short in representing more flexible symmetric structures. That is, while the study of symmetries is becoming more prominent in representation learning, in this work we have sought to take some key results of mathematical frameworks based on symmetries and generalise them to encompass all transformations of the world due to the actions of an agent: we have presented a formal framework that provides AI developers with the right tools to generate rich representations that show symmetries as groups (as SBDRL does) and beyond. The range of applications of AI models built with this underlying framework and mechanism may greatly extend the state of the art –for instance, reinforcement learning algorithms can directly incorporate knowledge of unknown until now symmetries in so-called world models, minimizing convergence speed; in natural language processing and computer vision tasks, for both LLMs and multimodal foundation models that rely on training vast amounts of data, information and its structured representation in various types of symmetries may be incredibly useful; even in relatively simple classification tasks executed on convolutional layers one could expect an improvement in performance if such convolutions take into account the symmetries formalised in our framework; and the same can be hypothesised about generative AI models such as GANs and transformers that depend on encoding and decoding embedded representations to and from latent spaces. We hope the work presented here will stimulate further work in this direction.

In addition, we also believe that a general framework for exploring the algebra of the transformations of worlds containing an agent, as proposed in this paper, has the potential to be used as a tool in the field of explainable AI since it may enable us to predict which algebraic structures should appear in the agent’s representation of the world at the end of the learning process.

CRediT authorship contribution statement

Alexander Dean: Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Eduardo Alonso:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Esther Mondragón:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (GPT-3.5) in order to improve the language and readability of their work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Algorithm 3 AddElementToStateCayleyTable: Fill state Cayley table row and column for element a.

```
Require: state_cayley_table, w: initial world state, a.
Add new row and new column labelled by a to state_cayley_table.
for column_label in state_cayley_table do
    state_cayley_table[a][column_label] ← column_label ∗ (a ∗ w).
end for
for row_label in state_cayley_table do
    state_cayley_table[row_label][a] ← a ∗ (row_label ∗ w).
end for
```

Algorithm 4 SearchForNewCandidates: Search for new candidate elements in state Cayley table.

```
Require: state_cayley_table, w: initial world state, candidate_cayley_table_elements.
for row_label in state_cayley_table do
    for column_label in state_cayley_table do
        a_C ← column_label ∘ row_label.
        equivalents_found ← SearchForEquivalents(state_cayley_table, w, a_C).    ⊳ See Algorithm 5.
        if len(equivalents_found) ≠ 0 then
            Add a_C to relevant equivalence class.
        else
            Add a_C to candidate_cayley_table_elements.
        end if
    end for
end for
return candidate_cayley_table_elements.
```

Algorithm 5 SearchForEquivalents: Search for elements in Cayley table that are equivalent to a.

```
Require: state_cayley_table, w: initial world state, a.
equivalents_found ← empty list.
a_row ← empty list.    ⊳ Generate state Cayley row for a.
for column_label in state_cayley_table do
    if column_label == a then
        Continue to the next iteration of the for loop.
    end if
    Append column_label ∗ (a ∗ w) to a_row.
end for
a_column ← empty list.    ⊳ Generate state Cayley column for a.
for row_label in state_cayley_table do
    if row_label == a then
        Continue to the next iteration of the for loop.
    end if
    Append a ∗ (row_label ∗ w) to a_column.
end for
for row_label in state_cayley_table do
    if (a_row, a_column) = = (row_label row, row_label column) then
        Append row_label to equivalents_found.
    end if
end for
return equivalents_found.
```

Algorithm 6 SearchForBrokenEquivalenceClasses: Find equivalence classes that are broken by a_C.

Require: $state\_cayley\_table$, w: initial world state, a_C.
for row_label in $state\_cayley\_table$ **do**
  $ec\_label\_outcome \leftarrow row\_label * (a_C * w)$.
  for $ec\_element$ in equivalence class labelled by row_label **do**
    if $ec\_label\_outcome \neq ec\_element * (a_C * w)$ **then**
      Create a new equivalence class labelled by $ec\_element$.
      Remove $ec\_element$ from the equivalence class labelled by row_label.
    end if
  end for
end for
return: New equivalence classes.

Data availability

GitHub repository has been linked to in the paper.

References

[1] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, A. Lerchner, Towards a definition of disentangled representations, arXiv preprint, arXiv:1812.02230, 2018.
[2] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, I. Sutskever, AI Comput. (2018), openai.com/blog/ai-and-compute/.
[3] N.C. Thompson, K. Greenewald, K. Lee, G.F. Manso, The computational limits of deep learning, arXiv preprint, arXiv:2007.05558, 2020.
[4] T. Flesch, K. Juechems, T. Dumbalska, A. Saxe, C. Summerfield, Orthogonal representations for robust context-dependent task performance in brains and neural networks, Neuron 110 (7) (2022) 1258–1270.
[5] S. Bernardi, M.K. Benna, M. Rigotti, J. Munuera, S. Fusi, C.D. Salzman, The geometry of abstraction in the hippocampus and prefrontal cortex, Cell 183 (4) (2020) 954–967.
[6] T. Ito, T. Klinger, D. Schultz, J. Murray, M. Cole, M. Rigotti, Compositional generalization through abstract representations in human and artificial neural networks, Adv. Neural Inf. Process. Syst. 35 (2022) 32225–32239.
[7] I. Momennejad, Learning structures: predictive representations, replay, and generalization, Curr. Opin. Behav. Sci. 32 (2020) 155–166.
[8] L. Lehnert, M.L. Littman, M.J. Frank, Reward-predictive representations generalize across tasks in reinforcement learning, PLoS Comput. Biol. 16 (10) (2020) e1008317.
[9] E. Alonso, E. Mondragón, Associative Reinforcement Learning-a Proposal to Build Truly Adaptive Agents and Multi-Agent Systems, International Conference on Agents and Artificial Intelligence, vol. 2, SCITEPRESS, 2013, pp. 141–146.
[10] N.H. Kokkola, E. Mondragón, E. Alonso, A double error dynamic asymptote model of associative learning, Psychol. Rev. 126 (4) (2019) 506.
[11] Y. Niv, Learning task-state representations, Nat. Neurosci. 22 (2019) 1544–1553, https://doi.org/10.1038/s41593-019-0470-8.
[12] M.L. Mack, A.R. Preston, B.C. Love, Ventromedial prefrontal cortex compression during concept learning, Nat. Commun. 11 (1) (2020) 46.
[13] A. Jha, J.C. Peterson, T.L. Griffiths, Extracting low-dimensional psychological representations from convolutional neural networks, Cogn. Sci. 47 (1) (2023) e13226.
[14] H. Op de Beeck, J. Wagemans, R. Vogels, Inferotemporal neurons represent low-dimensional configurations of parameterized shapes, Nat. Neurosci. 4 (12) (2001) 1244–1252.
[15] R.N. Shepard, Toward a universal law of generalization for psychological science, Science 237 (4820) (1987) 1317–1323.
[16] S. Edelman, N. Intrator, Learning as Extraction of Low-Dimensional Representations, Psychology of Learning and Motivation, vol. 36, Elsevier, 1997, pp. 353–380.
[17] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[18] Y. Li, Deep reinforcement learning: an overview, arXiv preprint, arXiv:1701.07274, 2017.
[19] K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath, Deep reinforcement learning: a brief survey, IEEE Signal Process. Mag. 34 (6) (2017) 26–38.
[20] R. Nian, J. Liu, B. Huang, A review on reinforcement learning: introduction and applications in industrial process control, Comput. Chem. Eng. 139 (2020) 106886.
[21] I. Higgins, S. Racanière, D. Rezende, Symmetry-based representations for artificial and biological general intelligence, arXiv:2203.09250, 2022.
[22] Y. LeCun, Y. Bengio, et al., Convolutional networks for images, speech, and time series, in: The Handbook of Brain Theory and Neural Networks, vol. 3361 (10), 1995, pp. 255–258.
[23] Z. Dai, H. Liu, Q.V. Le, M. Tan, Coatnet: marrying convolution and attention for all data sizes, Adv. Neural Inf. Process. Syst. 34 (2021) 3965–3977.
[24] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv preprint, arXiv:1806.01261, 2018.
[25] M.M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: grids, groups, graphs, geodesics, and gauges, arXiv preprint, arXiv:2104.13478, 2021.
[26] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G.R. Lee, J. Wang, Q. Cong, L.N. Kinch, R.D. Schaeffer, et al., Accurate prediction of protein structures and interactions using a three-track neural network, Science 373 (6557) (2021) 871–876.
[27] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J.P. Mailoa, M. Kornbluth, N. Molinari, T.E. Smidt, B. Kozinsky, E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nat. Commun. 13 (1) (2022) 2453.
[28] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[29] J. Köhler, L. Klein, F. Noé, Equivariant flows: exact likelihood generative learning for symmetric densities, in: International Conference on Machine Learning, PMLR, 2020, pp. 5361–5370.
[30] C.P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint, arXiv:1804.03599, 2018.
[31] M. Jaderberg, V. Mnih, W.M. Czarnecki, T. Schaul, J.Z. Leibo, D. Silver, K. Kavukcuoglu, Reinforcement learning with unsupervised auxiliary tasks, arXiv preprint, arXiv:1611.05397, 2016.
[32] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[33] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[34] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[35] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al., Impala: scalable distributed deep-rl with importance weighted actor-learner architectures, in: International Conference on Machine Learning, PMLR, 2018, pp. 1407–1416.
[36] G. Marcus, Deep learning: a critical appraisal, arXiv preprint, arXiv:1801.00631, 2018.
[37] K. Cobbe, O. Klimov, C. Hesse, T. Kim, J. Schulman, Quantifying generalization in reinforcement learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 1282–1289.
[38] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2014) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50, arXiv:1206.5538.
[39] A. Raffin, A. Hill, R. Traoré, T. Lesort, N. Díaz-Rodríguez, D. Filliat, Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics, arXiv preprint, arXiv:1901.08651, 2019.
[40] X. Wang, H. Chen, S. Tang, Z. Wu, W. Zhu, Disentangled representation learning, arXiv preprint, arXiv:2211.11695, 2022.
[41] J.Y. Park, O. Biza, L. Zhao, J.W. van de Meent, R. Walters, Learning symmetric embeddings for equivariant world models, arXiv preprint, arXiv:2204.11371, 2022.
[42] R. Quessard, T. Barrett, W. Clements, Learning disentangled representations and group structure of dynamical environments, Adv. Neural Inf. Process. Syst. 33 (2020) 19727–19737.
[43] T. Miyato, M. Koyama, K. Fukumizu, Unsupervised learning of equivariant structure from sequences, Adv. Neural Inf. Process. Syst. 35 (2022) 768–781.
[44] D. Wang, J.Y. Park, N. Sortur, L.L. Wong, R. Walters, R. Platt, The surprising effectiveness of equivariant models in domains with latent symmetry, arXiv preprint, arXiv:2211.09231, 2022.
[45] H. Keurti, H.-R. Pan, M. Besserve, B.F. Grewe, B. Schölkopf, Homomorphism autoencoder–learning group structured representations from observed transitions, in: International Conference on Machine Learning, PMLR, 2023, pp. 16190–16215.
[46] X. Zhu, C. Xu, D. Tao, Commutative Lie group vae for disentanglement learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 12924–12934.
[47] T. Wang, Z. Yue, J. Huang, Q. Sun, H. Zhang, Self-supervised learning disentangled group representation as feature, Adv. Neural Inf. Process. Syst. 34 (2021) 18225–18240.
[48] D. Pfau, I. Higgins, A. Botev, S. Racanière, Disentangling by subspace diffusion, Adv. Neural Inf. Process. Syst. 33 (2020) 17403–17415.
[49] G. Mercatali, A. Freitas, V. Garg, Symmetry-induced disentanglement on graphs, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., 2022, pp. 31497–31511.
[50] G.L. Marchetti, G. Tegnér, A. Varava, D. Kragic, Equivariant representation learning via class-pose decomposition, in: F. Ruiz, J. Dy, J.-W. van de Meent (Eds.), Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, PMLR, vol. 206, 2023, pp. 4745–4756.
[51] H. Caselles-Dupré, M. Garcia Ortiz, D. Filliat, Symmetry-based disentangled representation learning requires interaction with environments, Adv. Neural Inf. Process. Syst. 32 (2019).
[52] H. Caselles-Dupré, M. Garcia-Ortiz, D. Filliat, On the sensory commutativity of action sequences for embodied agents, arXiv preprint, arXiv:2002.05630, 2020.
[53] E. Riehl, Category Theory in Context, Courier Dover Publications, 2017.
[54] M. Barr, C. Wells, Category Theory for Computing Science, vol. 1, Prentice Hall, New York, 1990.
[55] B. Fong, D.I. Spivak, An Invitation to Applied Category Theory: Seven Sketches in Compositionality, Cambridge University Press, 2019.
[56] B. Milewski, Category theory for programmers, Blurb, https://unglueit-files.s3.amazonaws.com/ebf/e90890f0a6ea420c9825657d6f3a851d.pdf, 2018.
[57] B. Gavranović, P. Lessard, A. Dudzik, T. von Glehn, J.G.M. Araújo, P. Veličković, Position: categorical deep learning is an algebraic theory of all architectures, arXiv:2402.15332, https://arxiv.org/abs/2402.15332, 2024.
[58] B. Gavranović, Fundamental components of deep learning: a category-theoretic approach, arXiv:2403.13001, https://arxiv.org/abs/2403.13001, 2024.
[59] D.I. Spivak, Category Theory for the Sciences, MIT Press, Cambridge, Massachusetts, 2014.