SYMMETRY-BASED DISENTANGLED REPRESENTATION LEARNING REQUIRES INTERACTION WITH ENVIRONMENTS

Hugo Caselles-Dupré[1, 3], Michael Garcia-Ortiz[3], and David Filliat[1]

[1]Flowers Laboratory (ENSTA ParisTech & INRIA)
[3]Softbank Robotics Europe

ABSTRACT

Finding a generally accepted formal definition of a disentangled representation in the context of an agent behaving in an environment is an important challenge towards the construction of data-efficient autonomous agents. Higgins et al. (2018) recently proposed Symmetry-Based Disentangled Representation Learning, a definition based on a characterization of symmetries in the environment using group theory. We build on their work and make observations, theoretical and empirical, that lead us to argue that Symmetry-Based Disentangled Representation Learning cannot only be based on fixed data samples. Agents should interact with the environment to discover its symmetries. All of our experiments can be reproduced on Colab: `http://bit.do/eKpqv`.

1 INTRODUCTION

Disentangled Representation Learning aims at finding a low-dimensional vectorial representation of the world for which the underlying structure of the world is separated into disjoint parts (i.e., disentangled) corresponding to the actual compositional nature of the world. Previous work (Raffin et al., 2019) has shown that agents capable of learning disentangled representations can perform data-efficient policy learning. However, there is no generally accepted formal definition of disentanglement in Representation Learning, which prevents significant progress in this emerging field.

Recent efforts have been made towards finding a proper definition (Locatello et al., 2018). In particular, Higgins et al. (2018) defines Symmetry-Based Disentangled Representation Learning (SBDRL), by taking inspiration from the successful study of symmetry transformations in Physics. Their definition focuses on the transformation properties of the world. They argue that transformations that change only some properties of the underlying world state, while leaving all other properties invariant, are what gives exploitable structure to any kind of data. They distinguish between linear disentangled representation which models these transformation effect on the representation linearly and non-linear ones. Supposedly, the former should be more useful for downstream tasks such as Reinforcement Learning or auxiliary prediction tasks. Their definition is intuitive and provides principled resolutions to several points of contention regarding what disentangling is. For clarity, we refer to a representation as SB-disentangled if it is disentangled in the sense of SBDRL, and as LSB-disentangled if linear disentangled.

We build on the work of Higgins et al. (2018) and make observations, theoretical and empirical, that lead us to argue that SBDRL requires interaction with environments. The necessity of having interaction has been suggested before (Thomas et al., 2017). We are able to give theoretical and empirical evidence of why it is needed for SBDRL. As in the original work, we base our analysis on a simple environment, where we can formally define and manipulate a SB-disentangled representation. This simple environment is 2D, composed of one circular agent on a plane that can move left-right and up-down. The world is cyclic: whenever the agent steps beyond the boundary of the world, it is placed at the opposite end (e.g. stepping up at the top of the grid places the object at the bottom of the grid).

We prove, for this environment, that the minimal number of dimensions of the representation required for it to be LSB-disentangled is counter-intuitive. Indeed, the natural number of dimensions required to describe the state of the world is not enough to describe its symmetries in a linear way, which is supposedly ideal for subsequent tasks. Additionally, learning a non-linear SB-disentangled representation is possible, but current approaches are not designed to model the effect of the world's symmetries on the representation, a key aspect of SBDRL which we present later. We thus ask: how is one supposed to, in practice, learn a (L)SB-disentangled representation?

We propose two options that arise naturally, one where representation and world symmetries effect on it are learned separately and one where they are learned jointly. For both scenarios, we formally define what could be the proper representation to learn, using the formalism of SBDRL. We propose empirical implementations that are able to successfully approximate these analytically defined representations. Both empirical approaches make use of transitions (o_t, a_t, o_{t+1}) rather than still images o_t, which validates the main point of this paper: Symmetry-Based Disentangled Representation Learning requires interaction with the environment.

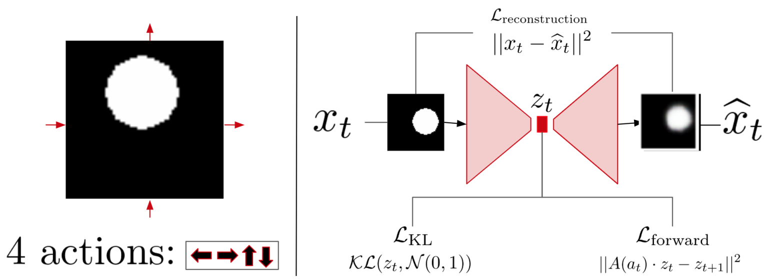

Figure 1: **Left:** Environment studied in this paper. **Right:** Proposed architecture for learning a LSB-disentangled representation in the environment at the left.

Our contributions are the following:

- We prove that learning a LSB-disentangled representation of dimension 2 is impossible in the world considered in this paper, due to the cyclical nature of the environment dynamics.
- We propose alternatives for learning linear and non-linear SB-disentangled representation, both using transitions rather than still observations. We validate both approaches emprically.
- Based on these observations, we take a step back and argue that interaction with the environment, i.e. the use of transitions, is necessary for SBDRL.

2 Symmetry-Based Disentangled Representation Learning

Higgins et al. (2018) defines Symmetry-Based Disentangled Representation Learning (SBDRL) as an attempt to formalize disentanglement in Representation Learning. The core idea is that SB-disentanglement of a representation is defined with respect to a particular decomposition of the symmetries of the environment. Symmetries are transformations of the environment that leave some aspects of it unchanged. For instance, for an agent on a plane, translations of the agent on the y-axis leave its x coordinate unchanged. They formalize this using group theory. Groups are composed of these transformations, and group actions are the effect of the transformations on the state of the world and representation.

The proposed definition of SB-disentanglement supposes that these symmetries are formally defined as a group G that can be decomposed into a direct product $G = G_1 \times .. \times G_n$. We now recall the formal definition of a SB-disentangled representation w.r.t to this group decomposition. We advise the reader to refer to the detailed work of (Higgins et al., 2018) for any clarification. Let W be the set of world-states. We suppose that there is a generative process $b : W \to O$ leading from world-states to observations (these could be pixel, retinal, or any other potentially multi-sensory observations), and an inference process $h : O \to Z$ leading from observations to an agent's representations. We consider the composition $f : W \to Z, f = h \circ b$. Suppose also that there is a group G of symmetries acting on W via a group action $\cdot_W : G \times W \to W$. What we would like is to find a corresponding group action $\cdot_Z : G \times Z \to Z$ so that the symmetry structure of W is reflected in Z. We also want the group action $\cdot_Z$ to be disentangled, which means that applying G_i to Z leaves all sub-spaces of Z unchanged but the one corresponding to the transformation G_i. Formally, the representation Z is SB-disentangled with respect to the decomposition $G = G_1 \times .. \times G_n$ if:

1. There is a group action $\cdot_Z : G \times Z \to Z$
2. The map $f : W \to Z$ is equivariant between the group actions on W and Z:

$$\boxed{g \cdot_Z f(w) = f(g \cdot_W w)} \quad \leftrightarrow \quad \begin{array}{ccc} G \times W & \xrightarrow{\cdot_W} & W \\ {\scriptstyle id_G \times f}\downarrow & & \downarrow{\scriptstyle f} \\ G \times Z & \xrightarrow[\cdot_Z]{} & Z \end{array}$$

3. There is a decomposition $Z = Z_1 \times .. \times Z_n$ such that each Z_i is fixed by the action of all $G_j, j \neq i$ and affected only by G_i

This definition of SB-disentangled representations does not make any assumptions on what form the group action should take when acting on the relevant disentangled vector subspace. However, many subsequent tasks may benefit from a SB-disentangled representation where the group actions transform their corresponding disentangled subspace linearly. Such representations are termed linear SB-disentangled representations, which we refer to as LSB-disentangled representations.

3 Considered environment

In this paper, we consider a simplification of the environment studied in the original paper (Higgins et al., 2018). This environment is 2D, composed of one circular agent on a plane that can move left-right and up-down, see Fig.1. Whenever the agent steps beyond the boundary of the world, it is placed at the opposite end (e.g. stepping up at the top of the grid places the object at the bottom of the grid). The world-states can be described in two-dimensions: (x, y) position of the agent. All of our results are based on this environment. It is simple, yet presents the basis for a navigation environment in 2D. We chose this environment because we are able to define theoretically SB-disentangled representations, without making any approximation. We implement this simple environment using Flatland (Caselles-Dupré et al., 2018).

4 Theoretical analysis

We provide a theorem that proves it is impossible to learn a LSB-disentangled representation of dimension 2 in the environment presented in Sec.3 (the result also applies to the environment considered in Higgins et al. (2018)).

Theorem 1. *For the considered world, there exists no LSB-disentangled representation Z w.r.t to the group decomposition $G = G_x \times G_y$, such that $\dim(Z) = 2$ and Z is not trivial.*

Proof. Proof by contradiction. The key element of the proof is that the two actual dimension of the environment are not linear but cyclic. Hence the impossibility of modelling two cyclic dimensions using two linear dimensions. See Appendix A for full proof.

5 Symmetry-Based Disentangled Representation Learning in practice requires transitions

We now consider the problem of learning, in practice, SB-disentangled and LSB-disentangled representations for the world considered in Sec.3.

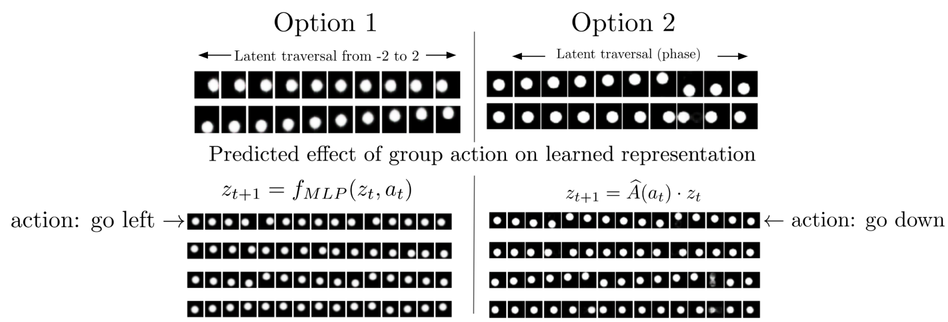

Figure 2: **Left:** First option: SB-disentanglement. Latent traversal spanning from -2 to 2 over each of the representation's dimensions, followed by the predicted effect of the group action associated each action (left, right, down, up). **Right:** Second option: LSB-disentanglement. The representation is complex: latent traversal over the phase of each of the representation's dimensions, followed by the predicted linear effect of the group action associated each action (down, left, up, right).

5.1 First option: SB-disentanglement with 2 dimension

Theorem 1 states that we cannot learn a 2-dimensional LSB-disentangled representation for the environment. We thus consider learning a 2-dimensional SB-disentangled representation. We started by reproducing the results in (Higgins et al., 2018): we used a variant of current state-of-the-art disentangled representation learning model CCI-VAE. The learned representation corresponds (up to a scaling factor) to the world-state W, i.e. the (x, y) position of the agent. This intuitively seems like a reasonable approximation to a disentangled representation.

However, once the representation is learned, we have no idea how the group action of symmetries affect the representation, even though it is at the core of the definition of SBDRL. This is where the necessity for transitions $(o_t, a_t, o_{t+1})_{t=1..n}$ rather than still observations $(o_t)_{t=1..n}$ comes into play. Indeed, learning about the effect of transformations on the world implies learning about the dynamics of the environment, which requires transitions.

Starting from the learned non-linear SB-disentangled representation, we propose to learn the group action of G on Z using a separate model. This way, we have a complete description of the SB-disentangled representation. This approach is effectively decoupling the learning of physics from vision as in (Ha & Schmidhuber, 2018). The second option would be to jointly learn vision and physics, which we demonstrate in the next experiment with LSB-disentangled representations.

In practice, we learn $h : O \to Z$ with a variant of CCI-VAE, and then use a multi-layer perceptron to learn the group action on Z $\cdot_Z : G \times Z \to Z$, such that $f = h \circ b$ is an equivariant map between the actions on W and Z. The results are presented in Fig.2, where we observe that the learned group action correctly approximate the cyclical movement of the agent. We thus have learned a properly SB-disentangled representation of the world, w.r.t to the group decomposition $G = G_x \times G_y$.

5.2 Second option: LSB-disentanglement with 4 dimensions

We now propose a method to learn a LSB-disentangled representation and the group action effect on it. To accomplish this, we start with a theoretically constructed LSB-disentangled representation. It is based on a example given in Higgins et al. (2018). The representation is defined as following, using 4 dimensions:

- $f : \mathbb{R}^2 \to \mathbb{C}^2$ is defined as $f(x, y) = (e^{2i\pi x/N}, e^{2i\pi y/N})$
- $\rho(g) : \mathbb{C}^2 \to \mathbb{C}^2$ is defined as $\begin{cases} \rho(g_x)(z_x, z_y) = & (e^{2i\pi n_x/N} z_x, z_y) \\ \rho(g_y)(z_x, z_y) = & (z_x, e^{2i\pi n_y/N} z_y) \end{cases}$

In this representation, the (x, y) position is mapped to two complex numbers (z_x, z_y). For each translation (on the x-axis or y-axis), the associated group representation is a rotation on a complex plane associated to the specific axis. This representation linearly accounts for the cyclic symmetry present in the environment. Using CCI-VAE with 4 dimensions fails to learn this representation: we verified experimentally that only 2 dimensions were actually used when learning (for encoding the (x, y) position), and the two remaining were ignored. We need to take into account transitions and enforce linearity in order to learn this specific representation.

We propose a method that allows to learn this LSB-disentangled representation. Once again, rather than using still observations, we generate a dataset of transitions, and use it to learn the 4-dimensional LSB-disentangled representation with a specific VAE architecture we term Forward-VAE. The core idea is to enforce linearity in transitions in the learned representation space.

We begin by re-writing the complex-valued function $\rho(g) : \mathbb{C}^2 \to \mathbb{C}^2$ as a real-valued function:

$$\rho(g) : \begin{array}{c} \mathbb{R}^4 \to \mathbb{R}^4 \\ v \to \rho(g)(v) = A^*(g) \cdot v \end{array} \tag{1}$$

$A^*(g)$ is a 2x2 block-diagonal matrix, composed of 2x2 rotation matrices. Let's consider the environment in Sec.3. The agent has 4 actions: go left, right, up or down. We associate each action with a corresponding matrix with trainable weights.

For instance, if $g = g_x \in G_x$ is a translation on the x-axis, the corresponding matrix is $A^*(g_x)$ and we associate actions go right/left with corresponding matrices $\hat{A}(a_t)$, where:

$A^*(g_x) = \begin{bmatrix} \cos(n_x) & -\sin(n_x) & 0 & 0 \\ \sin(n_x) & \cos(n_x) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $\hat{A}(g_x) = \begin{bmatrix} \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, where $\cdot$ represents trainable parameters.

We would like the representation model that we learn to satisfy $\rho(g)(v_t) = \hat{A}(g) \cdot v_t = v_{t+1}$. We thus enforce the representation to satisfy it, as illustrated in Fig.1. The training procedure is presented in Algorithm 1 in Appendix B. For each image in a batch, we compute $f(o_t) = z_t$ and $f(o_{t+1}) = z_{t+1}$ using the encoder part of the VAE. Then we decode z_t with the decoder and compute the reconstruction loss $\mathcal{L}_{reconstruction}$ and annealed KL divergence $\mathcal{L}_{KL}$ as in (Caselles-Dupré et al., 2019). Then we compute $\hat{A}(a_t) \cdot z_t$ and compute the forward loss, which is the MSE with z_{t+1}: $\mathcal{L}_{forward} = (\hat{A}(a_t) \cdot z_t - z_{t+1})^2$. We then backpropagate w.r.t to the full loss function of Forward-VAE:

$$\mathcal{L}_{Forward-VAE} = \mathcal{L}_{reconstruction} + \gamma_t \cdot \mathcal{L}_{KL} + \mathcal{L}_{forward} \tag{2}$$

The results are presented in Fig.2 and Appendix C. Forward-VAE correctly learns a representation where the two complex dimensions correspond to the position (x, y) of the agent. Moreover, we observe that the learned matrices $(\hat{A}_i)_{i=1..4}$ are very good approximation of the ideal matrices $(A_i^*)_{i=1..4}$ defined above, with $n_x \approx \frac{\pi}{3}$. The mean squared difference is very small (order of 10^{-4}).

6 Discussion & Conclusion

Discussion. We used inductive bias given by the theoretical construction of a LSB-disentangled representation theory to design the action matrices and its trainable weights. This construction is specific to the chosen example. However, the idea of having an action matrix for each action is extendable. If each action is high-level and associated to a symmetry, then you can perform SBDRL. Still, it requires high level actions that represent these symmetries. One potential way to find these actions is through active search (Soatto, 2011), as suggested in (Higgins et al., 2018).

Learning a LSB-disentangled representation is supposedly beneficial for subsequent tasks. However, this remains to be demonstrated, as other work have even challenged the benefit of learning disentangled representations over non-disentangled ones (Locatello et al., 2018). Therefore, our future work will study whether (L)SB-disentangled representations are beneficial for subsequent tasks. Overall, the field of Disentangled Representation Learning needs more investigation on this matter in order to move forward. We further discuss our results and the approach of SBDRL in Appendix D.

Conclusion. Using theoretical and empirical arguments, we have demonstrated that SBDRL (Higgins et al., 2018), a proposed definition for disentanglement in Representation Learning, requires interaction with the environment. We thus proposed two methods to perform SBDRL in practice, both of which are successful empirically. We believe SBDRL provides a new perspective on disentanglement which can be promising for Representation Learning in the context of an agent acting in a environment.

REFERENCES

Hugo Caselles-Dupré, Louis Annabi, Oksana Hagen, Michael Garcia-Ortiz, and David Filliat. Flatland: a lightweight first-person 2-d environment for reinforcement learning. arXiv preprint arXiv:1809.00510, 2018.

Hugo Caselles-Dupré, Michael Garcia-Ortiz, and David Filliat. S-trigger: Continual state representation learning via self-triggered generative replay. arXiv preprint arXiv:1902.09434, 2019.

Roger C Conant and W Ross Ashby. Every good regulator of a system must be a model of that system. International journal of systems science, 1(2):89–97, 1970.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. arXiv preprint arXiv:1809.01999, 2018.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. arXiv preprint arXiv:1812.02230, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359, 2018.

Antonin Raffin, Ashley Hill, Kalifou René Traoré, Timothée Lesort, Natalia Díaz-Rodríguez, and David Filliat. Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. arXiv preprint arXiv:1901.08651, 2019.

Stefano Soatto. Steps towards a theory of visual information: Active perception, signal-to-symbol conversion and the interplay between sensing and control. arXiv preprint arXiv:1110.2053, 2011.

Valentin Thomas, Jules Pondard, Emmanuel Bengio, Marc Sarfati, Philippe Beaudoin, Marie-Jean Meurs, Joelle Pineau, Doina Precup, and Yoshua Bengio. Independently controllable features. arXiv preprint arXiv:1708.01289, 2017.

A PROOFS

A.1 TRIVIAL REPRESENTATIONS

We first define trivial representations and then prove that they are LSB-disentangled. We will then use this definition to prove Theorem 1.

Definition 1. *Z is a trivial representation if and only if f is constant.*

If Z is a trivial representation, we thus have that each state of the world $w \in W$ has the same representation.

Proposition 1. *If Z is a trivial representation then Z is LSB-disentangled w.r.t to every group decomposition.*

We prove Proposition 1 which states that trivial representations are LSB-disentangled.

Proof. The definition of LSB-disentangled representation of dimension 2 is:

1. There is a linear action $\cdot_Z : G \times Z \to Z$. It thus can be viewed as a group representation $\rho : G \to GL(Z)$.

2. The map $f : W \to Z$ is equivariant between the actions on W and Z.
3. There is a decomposition $Z = Z_1 \times Z_2$ or $Z = Z_1 \bigoplus Z_2$ such that each Z_i is fixed by the action of all $G_j, j \neq i$ and affected only by G_i.

Let $\rho(g)$ be the identity function $\forall g \in G$, which is linear.

We have that $f : W \to Z$ is constant. We can verify that f is equivariant between the actions on W and Z:

$$\rho(g)(f(w)) = f(w) = f(g \cdot_W w) \tag{3}$$

Finally, Z has the same representation $\forall w \in W$, so Z is fixed by the action of any subgroup of G. Hence for all decomposition of G, point 3. of the definition is satisfied.

□

A.2 It is impossible to learn a LSB-disentangled representation of dimension 2 in the considered environment

We prove Theorem 1 which states that it is impossible to learn a LSB-disentangled representation of dimension 2 in the environment presented in Sec.3 (the result also applies to the environment considered in Higgins et al. (2018)).

Theorem. *For the considered world, there exists no LSB-disentangled representation Z w.r.t to the group decomposition $G = G_x \times G_y$, such that $\dim(Z) = 2$ and Z is not trivial.*

Proof. Proof by contradiction.
Suppose that there exists a LSB-disentangled representation Z w.r.t to the group decomposition $G = G_x \times G_y$, such that $\dim(Z) = 2$. Then, by definition:

1. There is a linear action $\cdot_Z : G \times Z \to Z$. It thus can be viewed as a group representation $\rho : G \to GL(Z)$.
2. The map $f : W \to Z$ is equivariant between the actions on W and Z.
3. There is a decomposition $Z = Z_1 \times Z_2$ or $Z = Z_1 \bigoplus Z_2$ such that each Z_i is fixed by the action of all $G_j, j \neq i$ and affected only by G_i.

We now prove that if these conditions are verified, f is necessarily constant. Consequently, Z has the same representation for each state of the world, which is a trivial representation. So, if Z a LSB-disentangled representation of dimension 2 w.r.t to G, then Z is the trivial representation.

We thus suppose that there exists a LSB-disentangled representation Z of dimension 2 w.r.t to the group decomposition $G = G_x \times G_y$. Hence, we have, by point 2. of the definition:

$$g \cdot_Z f(w) = f(g \cdot_W w) \tag{4}$$

Since $\cdot_Z$ is linear, we can view it as a group representation ρ, as mentioned in point 1. of the definition:

$$g \cdot_Z f(w) = \rho(g)(f(w)) \tag{5}$$

Because $f(W) \in Z \subset \mathbb{R}^2$ and $W = W_x \bigoplus W_y = (x, y)$, we can re-write f as:

$$\begin{aligned} f(w) &= f((x, y)) \\ &= (f_1(x, y), f_2(x, y)) \end{aligned} \tag{6}$$

Hence, combining (4) and (5):

$$f(g \cdot_W (x,y)) = \rho(g)((f_1(x,y), f_2(x,y))) \tag{7}$$

We can decompose any $g \in G$ into the composition of functions of each subgroup of G, i.e. $\forall g \in G = G_x \times G_y, \exists (g_x, g_y) \in G_x \times G_y$ such that $g = g_x \circ g_y$. Plus, by definition of Z and because $W = W_x \bigoplus W_y = (x,y)$, the action of all G_i on W and Z is fixed by the action of all $G_j, j \neq i$ and affected only by G_i. We can thus re-write both terms of Equation (7).

$$f(g \cdot_W (x,y)) = (f_1((g_x(x), g_y(y))), f_2((g_x(x), g_y(y))) \quad \text{since } g \cdot_W (x,y) = (g_x(x), g_y(y)) \tag{8}$$

$$\rho(g)((f_1(x,y), f_2(x,y))) = (\rho_x(g_x)(f_1(x,y)), \rho_y(g_y)(f_2(x,y))) \quad \text{by definition of } \rho \tag{9}$$

Hence, Equation 7 becomes:

$$(f_1((g_x(x), g_y(y))), f_2((g_x(x), g_y(y))) = (\rho_x(g_x)(f_1(x,y)), \rho_y(g_y)(f_2(x,y))) \tag{10}$$

We will now prove that f_1 is necessarily constant. The same argument applies for f_2.

From Equation (10), we have:

$$f_1((g_x(x), g_y(y))) = \rho_x(g_x)(f_1(x,y)) \tag{11}$$

g_x and g_y are respectively translations on the x-axis and y-axis. Let N be the size of the grid, then $\exists (n_x, n_y) \in [|0, N|]$ s.t. $(g_x(x), g_y(y)) = ((x + n_x) \mod N, (y + n_y) \mod N)$. When at edge of the world, if the object translates to the right, it returns to the left, hence the modulo operation that represents this cycle. Hence:

$$\begin{aligned} f_1((g_x(x), g_y(y))) &= f_1(((x + n_x) \mod N, (y + n_y) \mod N)) \\ &= \rho_x(g_x)(f_1(x,y)) \end{aligned} \tag{12}$$

The key argument of the proof lies in the fact that $\rho_x(g_x)$ is necessary cyclic of order $2N$ (the minimal order can be inferior to N, but it is not useful to caracterize the minimal order in this proof). Let's compose $\rho_x(g_x)$ $2N$ times:

$$\begin{aligned} \rho_x(g_x)^{(2N)}(f_1(x,y)) &= f_1(((x + 2N \cdot n_x) \mod N, (y + 2N \cdot n_y) \mod N)) \\ &= f_1((x,y)) \end{aligned} \tag{13}$$

We now use the fact that $\rho_x(g_x)$ is a linear application of $\mathbb{R}$, thus:

$$\rho_x(g_x) \in GL(\mathbb{R}) \implies \exists (a(g_x), b(g_x)) \in \mathbb{R}^2 \quad s.t. \quad \forall x \in \mathbb{R} \quad \rho_x(g_x)(x) = a(g_x) \cdot x + b(g_x) \tag{14}$$

For notation purposes, we drop the dependence on g_x of the coefficients of the real linear application $\rho_x(g_x)$, and we can rewrite Equation (10):

$$\rho_x(g_x)(f_1(x,y)) = a \cdot f_1(x,y) + b \tag{15}$$

Hence, using Equation 13 we can develop the term $\rho_x(g_x)^{(2N)}(f_1(x,y))$:

$$\begin{aligned} \rho_x^{2N}(g_x)(f_1(x,y)) &= a^{2N} \cdot f_1(x,y) + b \cdot \sum_{i=0}^{2N-1} a^i \\ &= f_1(x,y) \end{aligned} \tag{16}$$

Define $c = b \cdot \sum_{i=0}^{2N-1} a^i$, we have:

$$\begin{aligned} & a^{2N} \cdot f_1(x,y) + c = f_1(x,y) \\ \iff \quad & (a^{2N} - 1) \cdot f_1(x,y) + c = 0 \end{aligned} \tag{17}$$

Equation (17) is verified $\forall (x,y) \in \mathbb{R}^2$. Let $((x_1,y_1),(x_2,y_2)) \in \mathbb{R}^2 \times \mathbb{R}^2$:

$$\begin{cases} (a^{2N} - 1) \cdot f_1(x_1,y_1) + c = & 0 \\ (a^{2N} - 1) \cdot f_1(x_2,y_2) + c = & 0 \end{cases} \implies (a^{2N} - 1) \cdot (f_1(x_1,y_1) - f_1(x_2,y_2)) = 0 \tag{18}$$

We can now derive conditions on f_1 or (a,b). From Equation (18) we know that either f_1 is constant or $(a^{2N} - 1) = 0 \implies a = 1$. If $a = 1$, then Equation (17) simplifies to $c = 0 \implies b = 0$. So either $\rho_x(g_x)$ is the identity function or f_1 is constant. The same argument applies to f_2 and $\rho_y(g_y)$, hence we have that either f is constant or $\rho(g) = \text{Id}(\mathbb{R}^2)$. By plugging the second option in Equation (7), we have that $\rho(g) = \text{Id} \implies f$ is constant.

Hence f is necessarily constant, which implies that Z is a trivial representation.

□

B Details about Forward-VAE

B.1 Definition of $\hat{A}$

$A^*(g)$ is a 2x2 block-diagonal rotation matrix of dimension 4. For instance, if $g = g_x \in G_x$ is a translation on the x-axis, the corresponding matrix is: $A^*(g_x) = \begin{bmatrix} \cos(n_x) & -\sin(n_x) & 0 & 0 \\ \sin(n_x) & \cos(n_x) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Similarly, for $g = g_y \in G_y$ which is a translation on the y-axis, the corresponding matrix is $A^*(g_y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(n_x) & -\sin(n_x) \\ 0 & 0 & \sin(n_x) & \cos(n_x) \end{bmatrix}$.

Let's consider the environment in Sec.3. The agent has 4 actions: go left, right, up or down. We associate each action with a corresponding matrix with trainable weights. Thus, we associate actions go up and go down with a matrix of the form $\hat{A} = \begin{bmatrix} \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and we associate actions go left and go right with a matrix of the form $\hat{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \end{bmatrix}$ where $\cdot$ represents trainable parameters.

B.2 Pseudo-code of Forward-VAE

Algorithm 1 Pseudo-code for training procedure of Forward-VAE

batch $= ((o_t, .., o_{t+k}), (a_t, .., a_{t+k-1})) = (\mathbf{o}, \mathbf{a})$
for batch in dataset **do**

— Forward model Loss—

$\mathbf{z} \leftarrow encoder\_mean(batch)$
$\mathbf{z_{before}} \leftarrow z[: -1]$
$\mathbf{z_{after}} \leftarrow z[1 :]$ *# targets*
$\hat{\mathbf{A}} \leftarrow [\hat{A}(a_t), .., \hat{A}(a_{t+k-1})]$ *# actions matrices corresponding to given action sequence*
$\mathbf{z_{prediction}} \leftarrow \hat{\mathbf{A}} \cdot \mathbf{z_{before}}$ *# predictions*
$\mathcal{L}_{forward}(batch) \leftarrow MeanSquaredError(\mathbf{z_{prediction}}, \mathbf{z_{after}})$

— VAE Loss (reconstruction and KL) —

$\mathbf{z} \leftarrow encoder\_sample(batch)$
$\hat{\mathbf{o}} \leftarrow decoder(\mathbf{z})$
$\mathcal{L}_{recon}(batch) \leftarrow MeanSquaredError(\hat{\mathbf{o}}, \mathbf{o})$
$\mathcal{L}_{KL}(batch) \leftarrow KL\_divergence(\mathbf{z}, \mathcal{N}(0, 1))$

— Backpropagation —

$\mathcal{L}_{Forward-VAE}(batch) \leftarrow \mathcal{L}_{recon}(batch) + \mathcal{L}_{KL}(batch) + \mathcal{L}_{forward}(batch)$
$encoder, decoder, (\hat{A}_1, .., \hat{A}_j) \leftarrow Backpropagation(\mathcal{L}_{Forward-VAE}(batch)$

C Additional results

We observe that the mean squared difference between the ideal matrices $(A_i)_{i=1..4}$ and the learned matrices $(\hat{A}_i)_{i=1..4}$ is very small (order of 10^{-4}). Hence, we have :

$$\hat{A}(\text{go left / go right}) \approx A^*(\text{go left / go right}) = \begin{bmatrix} \cos(\pm\alpha) & -\sin(\pm\alpha) & 0 & 0 \\ \sin(\pm\alpha) & \cos(\pm\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\hat{A}(\text{go up / go down}) \approx A^*(\text{go up / go down}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\pm\alpha) & -\sin(\pm\alpha) \\ 0 & 0 & \sin(\pm\alpha) & \cos(\pm\alpha) \end{bmatrix}$$

The result is quite surprising as we do not have completely explicitly optimized for this matrix (at least for the cos/sin part). Plus there is no instability in training.

One issue with the fact that the approximation is not exact, is unstability with composition. Rotation matrices' determinants are stable with composition, as we have:

$$\det(AB) = \det(A)\det(B)$$

As rotation matrices have a determinant equal to 1, the composition operation is cyclic for rotations.

However, as $(\hat{A})_{i=1..4}$ are only approximation of rotation matrices, their determinant is approximately 1 but not exactly. This is why, as many compositions are performed, the determinant of the resulting matrix either collapses to zero or explodes to $+\infty$. We provide evidence for this phenomenon in Fig.3.

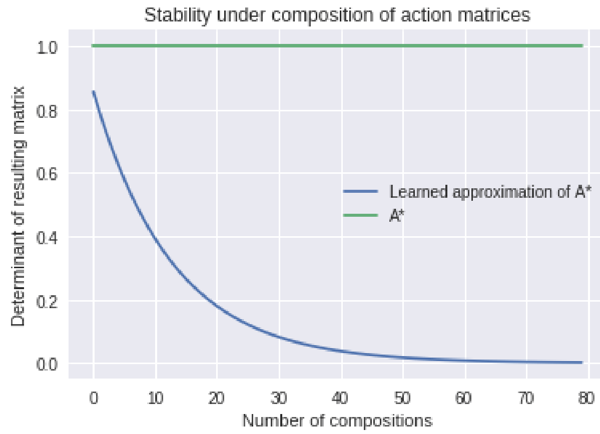

Figure 3: Determinant of real (A^*) and learned rotation matrix ($\hat{A}$) as a function of number of compositions. As many compositions are performed, the determinant of the approximation of the rotation matrix A^* collapses to zero.

D Discussion

It is important to note that Forward-VAE successfully learns even if there is a possible source of instability in training: the target for the physics loss $\mathcal{L}_{forward}$ are constantly changing throughout training, as the encoder is being trained.

The benefit of using transitions rather than still observations for representation learning in the context of an agent acting in a environment has been proposed, discussed and implemented in previous work (Thomas et al., 2017; Raffin et al., 2019). In this work however, we emphasize that using transitions is not an beneficial option, but is compulsory in the context of the current definition of SBDRL for an agent acting in an environment.

We make a connection with SBDRL and the Good Regulator Theorem (Conant & Ross Ashby, 1970). This principle states that, with regard to the brain, insofar as it is successful and efficient as a regulator for survival, it must proceed, in learning, by the formation of a model (or models) of its environment. In SBDRL, the aim is to find a representation that incorporates information about the dynamics of the environment.

Applying SBDRL to more complex environments is not straightforward. For instance consider that we add an object in the environment studied in this paper. Then the group structure of the symmetries of the world are broken when the agent is close to the object. However, the symmetries are conserved locally. One approach would be to start from this local property to learn an approximate SB-disentangled representation.