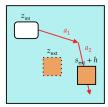
7 Appendix

7.1 Proofs of Theoretical Results

Theorem 5. Suppose that the representation $\varphi: \mathcal{O} \to \mathcal{Z}$ satisfies:

- 1. φ is equivariant (Definition 1),
- 2. φ is injective,
- 3. for all $o \in \mathcal{O}$ and $a \in \mathcal{A}$ it holds that $z'_{\text{ext}} \neq z_{\text{ext}}$ if and only if $z_{\text{ext}} \in [z_{\text{int}}, z_{\text{int}} + a]$ where $(z_{\text{int}}, z_{\text{ext}}) = \varphi(o)$ and $(z'_{\text{int}}, z'_{\text{ext}}) = \varphi(a \cdot o)$.

Then $\varphi \circ \omega$ is a translation i.e., there is a constant vector $h \in \mathbb{R}^n$ such that for all $s \in S$ it holds that $\varphi(\omega(s)) = s + h$. In particular, φ is an isometry w.r.t. the Euclidean metric on both S and Z.



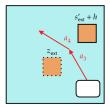


Fig. 7: Graphical depiction of the proof of Theorem 4.

Proof. Pick an arbitrary state $s^0 \in \mathbb{S}$ together with its represented internal state z_{int}^0 and set $h = z_{\mathrm{int}}^0 - s_{\mathrm{int}}^0$. For any state s, consider the action $a = s_{\mathrm{int}} - s_{\mathrm{int}}^0$. Equivariance then implies that $z_{\mathrm{int}} = z_{\mathrm{int}}^0 + a = s_{\mathrm{int}} + h$. This shows that the claim holds for internal states.

To prove that the same happens for external states, suppose by contradiction that there is a state s such that $z_{\text{ext}} \neq s_{\text{ext}} + h$. Consider any path during which the agent interacts with the object without passing through z_{ext} . Formally, this means considering a sequence of actions a_1, \dots, a_r such that (see Figure 7, left):

- $-z_{\text{ext}}$ and $s_{\text{ext}} + h$ do not belong to $[z_{\text{int}} + a_1 + \cdots + a_{i-1}, z_{\text{int}} + a_1 + \cdots + a_i]$ for every $i = 1, \dots, r-1$,
- $-z_{\text{ext}}$ does not belong to $\lfloor z_{\text{int}} + a_1 + \dots + a_{r-1}, z_{\text{int}} + a_1 + \dots + a_r \rfloor$ but $s_{\text{ext}} + h$ does.

The existence of such a path follows from Assumptions 1 and 3. After interaction the state becomes $s' = a_r \cdot (a_{r-1} \cdots (a_1 \cdot s))$ with $s'_{\text{ext}} \neq s_{\text{ext}}$ because of Assumption 2. One can then consider a path back to the initial agent's position z_{int} i.e., another sequence of actions a_{r+1}, \cdots, a_R such that (see Figure 7, right):

 $-s'_{\text{ext}} + h$ and z_{ext} do not belong to $\lfloor z_{\text{int}} + a_1 + \dots + a_{i-1}, z_{\text{int}} + a_1 + \dots + a_i \rfloor$ for every $i = r+1, \dots, R$,

```
-a_1+\cdots+a_R=0.
```

All the conditions imply together that the representation of the object remains equal to $z_{\rm ext}$ during the execution of the actions a_1, \cdots, a_R . Since the actions sum to 0, the representation of the agent does not change as well. But then $\varphi(\omega(s)) = \varphi(\omega(s_{\rm int}, s'_{\rm ext}))$ while $s_{\rm ext} \neq s'_{\rm ext}$, contraddicting injectivity. We conclude that $z_{\rm ext} = s_{\rm ext} + h$ and thus z = s + h as desired.

7.2 Pseudocode for Loss Computation

Algorithm 1 Loss Computation

```
Input: Batch \mathcal{B} \subseteq \mathcal{D}, models \varphi_{\text{int}}, \varphi_{\text{ext}}, \varphi_{\text{cont}}
Output: Loss \mathcal{L}
   \mathcal{L} = 0
   for all (o, a, o') \in \mathcal{B} do
         Compute z_{\rm int} = \varphi_{\rm int}(o), \ z_{\rm ext} = \varphi_{\rm ext}(o), \ z_{\rm int}' = \varphi_{\rm int}(o'), \ z_{\rm ext}' = \varphi_{\rm ext}(o'), \ w = 0
   \varphi_{\text{cont}}(o), w' = \varphi_{\text{cont}}(o')
   end for
   Compute the classes C_-, C_+ via Otsu's algorithm based on \{d_W(w, w')\}
   for all (o, a, o') \in \mathcal{B} do
         Compute \mathcal{L}_{\text{int}}(o, a, o') via Equation 3
         Compute A = \{d_{\mathcal{W}}(w', w''), d(z'_{\text{int}}, z''_{\text{int}})\} for o'' marginalized from \mathcal{B}
         Based on A compute \mathcal{L}_{\text{cont}}(o, o') via Equation 5
         if d_{\mathcal{W}}(w,w') \in C_{-} then
                Compute \mathcal{L}_{\text{ext}}(o, a, o') = \mathcal{L}_{-}(o, a, o') via Equation 4 (left)
                Compute \mathcal{L}_{\text{ext}}(o, a, o') = \mathcal{L}_{+}(o, a, o') via Equation 4 (right)
         end if
         \mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\mathrm{int}} + \mathcal{L}_{\mathrm{ext}} + \mathcal{L}_{\mathrm{cont}}
   end for
```