Exploiting symmetries in Markov Decision Processes for Reinforcement Learning

Group 33

Context

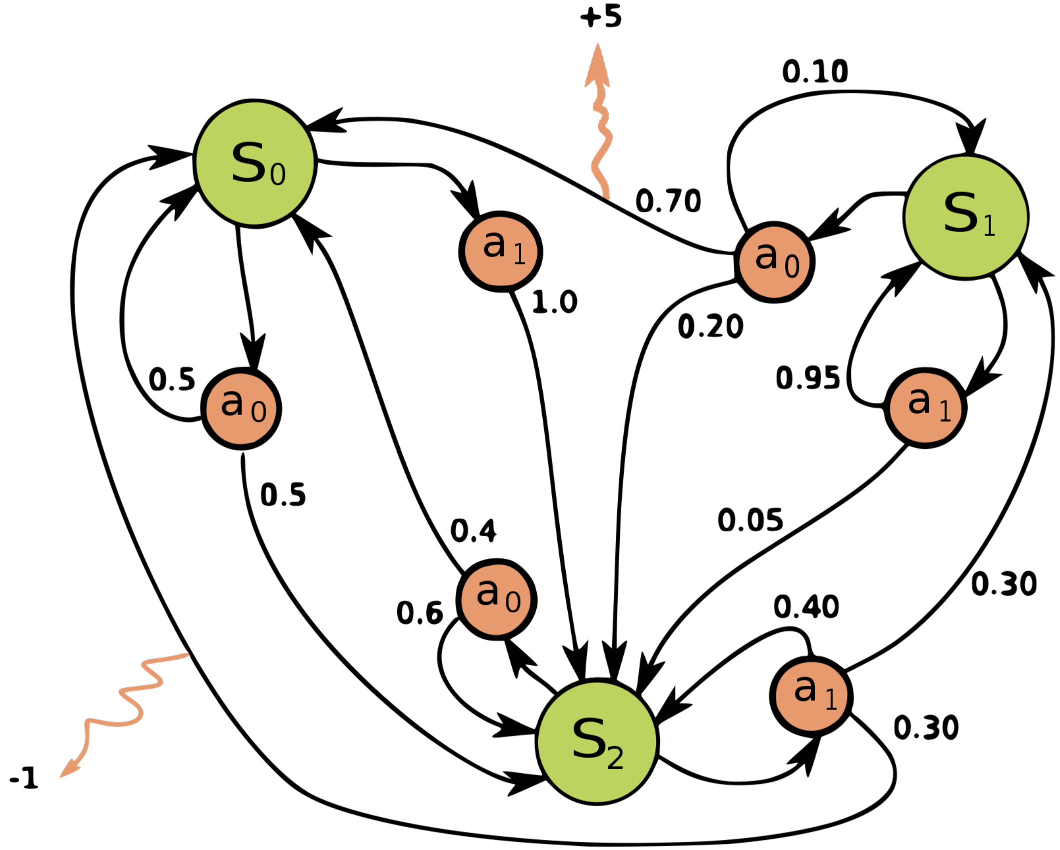
- Markov decision processes
- Reinforcement learning
- Unsupervised learning
- Geometric learning
- Homomorphism

MDP
Homomorphic
Networks

Van der Pol et al. “MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning” (2020)

Markov decision processes (MDPs)

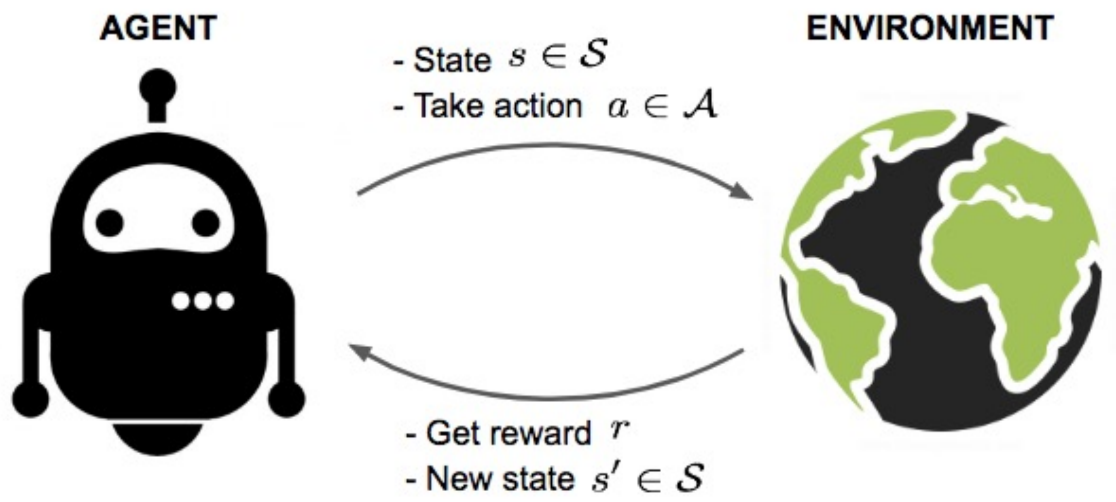
- States S
- Actions A
- State transition P
- Reward function R
- Policy π(S)
- MDPs scale poorly!



Waldoalvarez via Wikipedia

Reinforcement learning (RL)

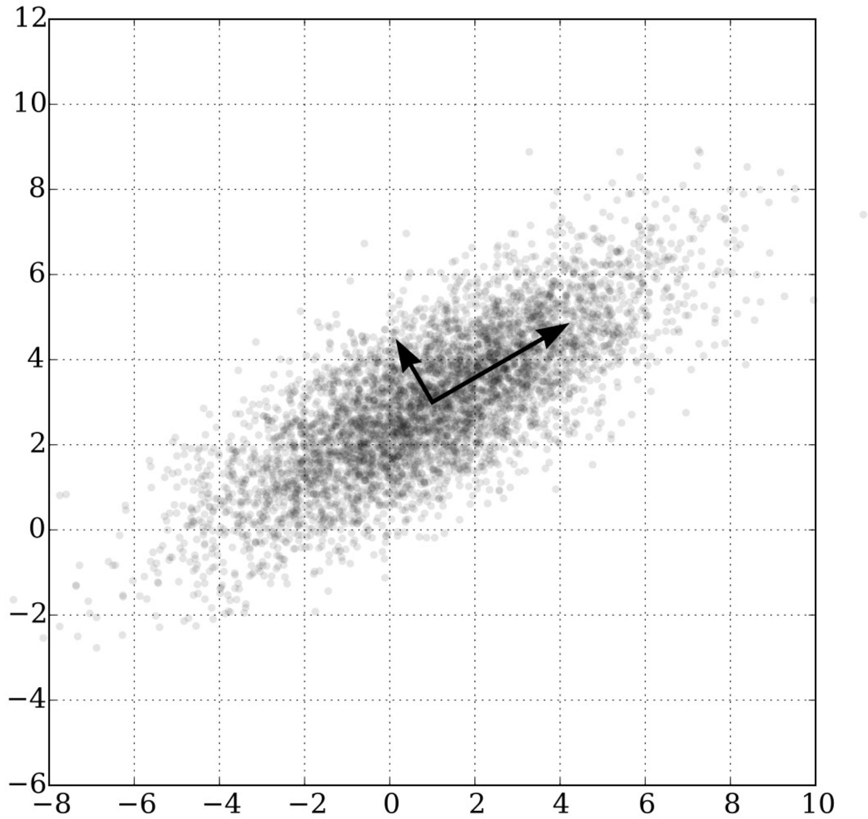
- Decision maker (agent)
- Environment
- States, actions and rewards
- MDPs for mathematical formulation



lilianweng.github.io/posts/2018-02-19-rl-overview/

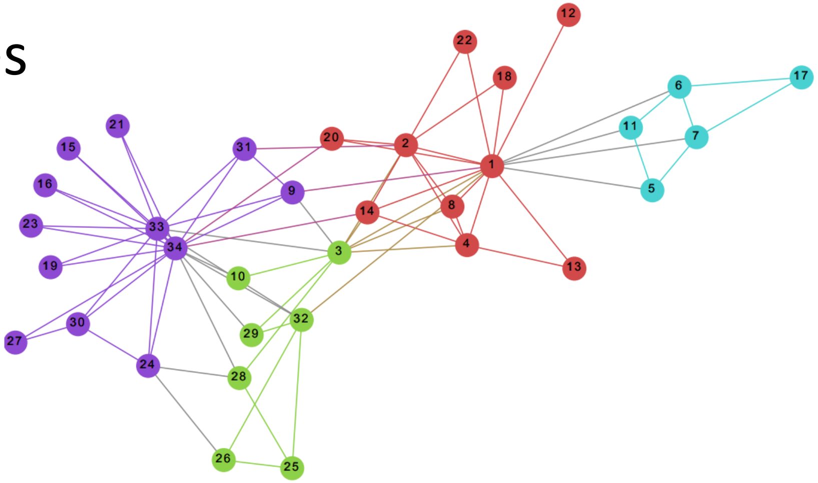
Unsupervised learning

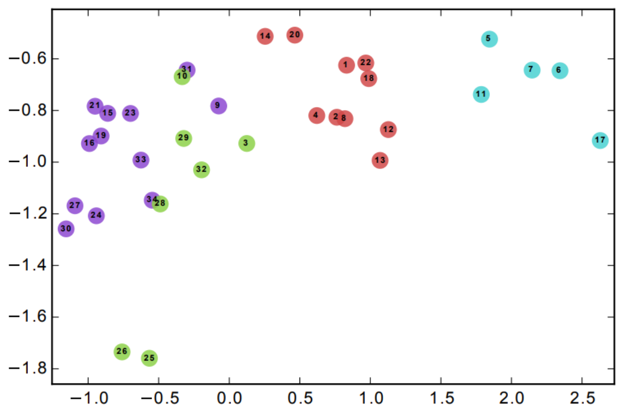
- Find structure in data without extra information
- Reduce dimensionality
- Principal component analysis



Geometric learning

- Non-euclidean data
- Graphs, manifolds
- Inherent relationships, connections and shared properties

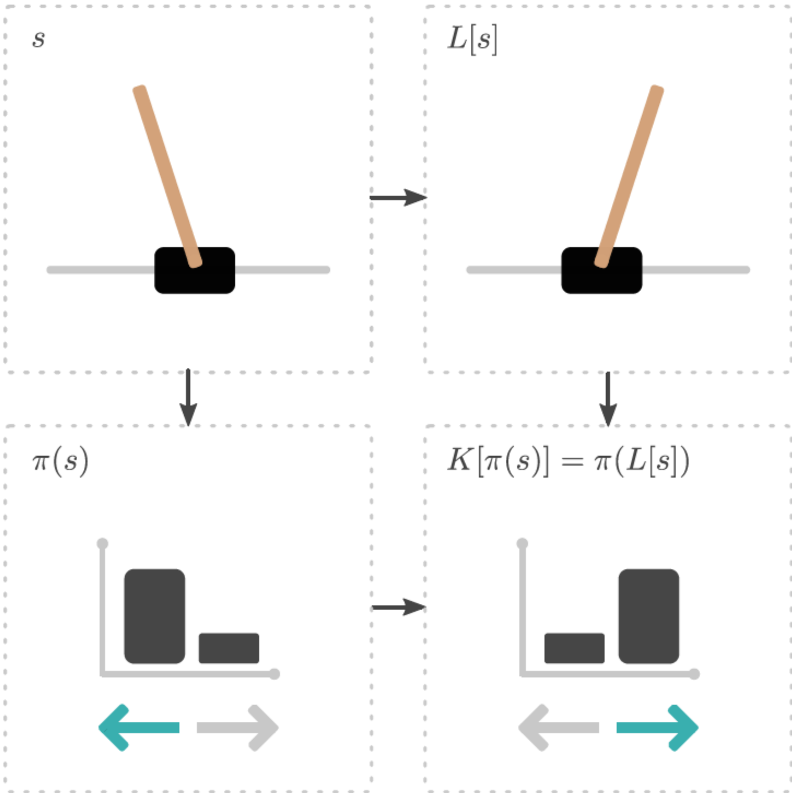




Perozzi et al. “DeepWalk: Online Learning of Social Representations” (2014)

Homomorphism

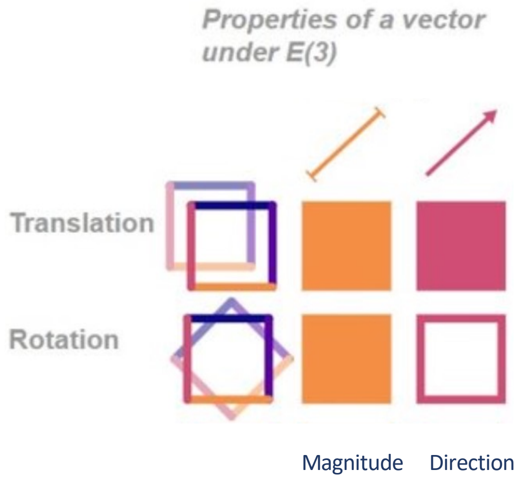
- Structure preserving map
- Symmetry in state-action space of MDP
- Reduce the solution space
- MDP homomorphic networks!

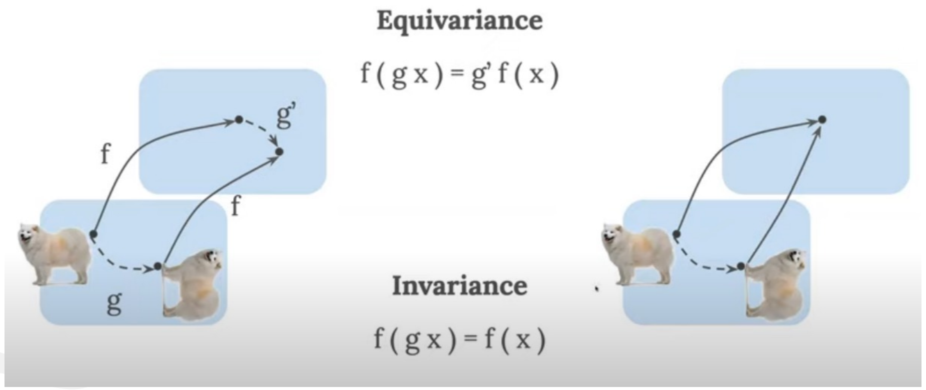


Van der Pol et al. “MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning” (2020)

Symmetry and Equivariance

- Symmetric to a transformation means that the output does not change
- Equivariant to a transformation means that the output changes deterministically
- Invariance is a special case of equivariance
- The **orbit $\boldsymbol{O_x}$** of point x is the set of points reachable from x via transformation operator g



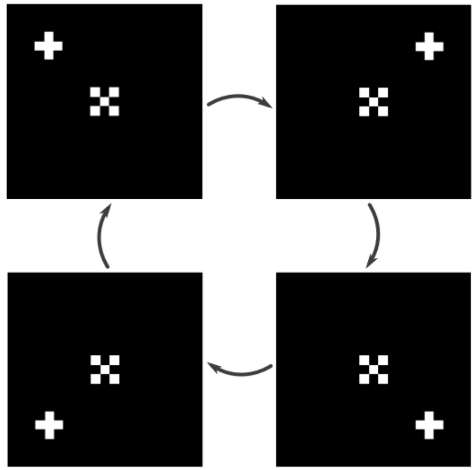


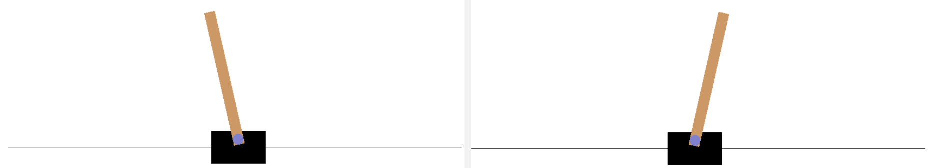
MDPs with Symmetries

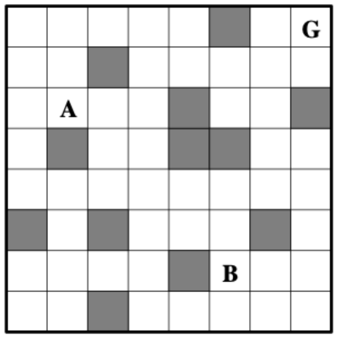
- In an MDP with symmetries there is a set of transformations on the state-action space, which leaves the reward function and transition operator invariant

- The reward and transition function are invariant along the orbits defined by L_g and K_g^s.

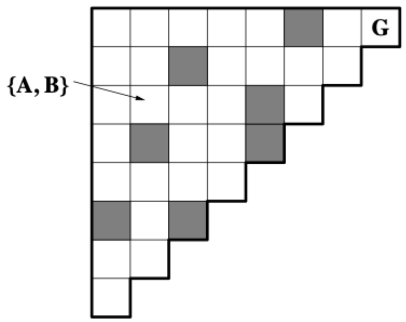
$$R(s,a) = R(L_g[s], K_g^s[a])$$
$$T(s'|s,a) = T(L_g[s']|L_g[s], K_g^s[a])$$
$$\forall g \in G, s \in S, a \in A$$







(a)



(b)

MDP homomorphism

- We call MDP homomorphism a map between an MDP $\mathrm{M} = (S, A, R, T, \gamma)$ and another MDP $\bar{M} = (\bar{S}, \bar{A}, \bar{R}, \bar{T}, \bar{\gamma})$ such that the new system, also called *abstract MDP* maintains the essential structure removing redundancies in the problem description.

- As result we obtain a smaller state-action space, upon which we may more easily build a policy.

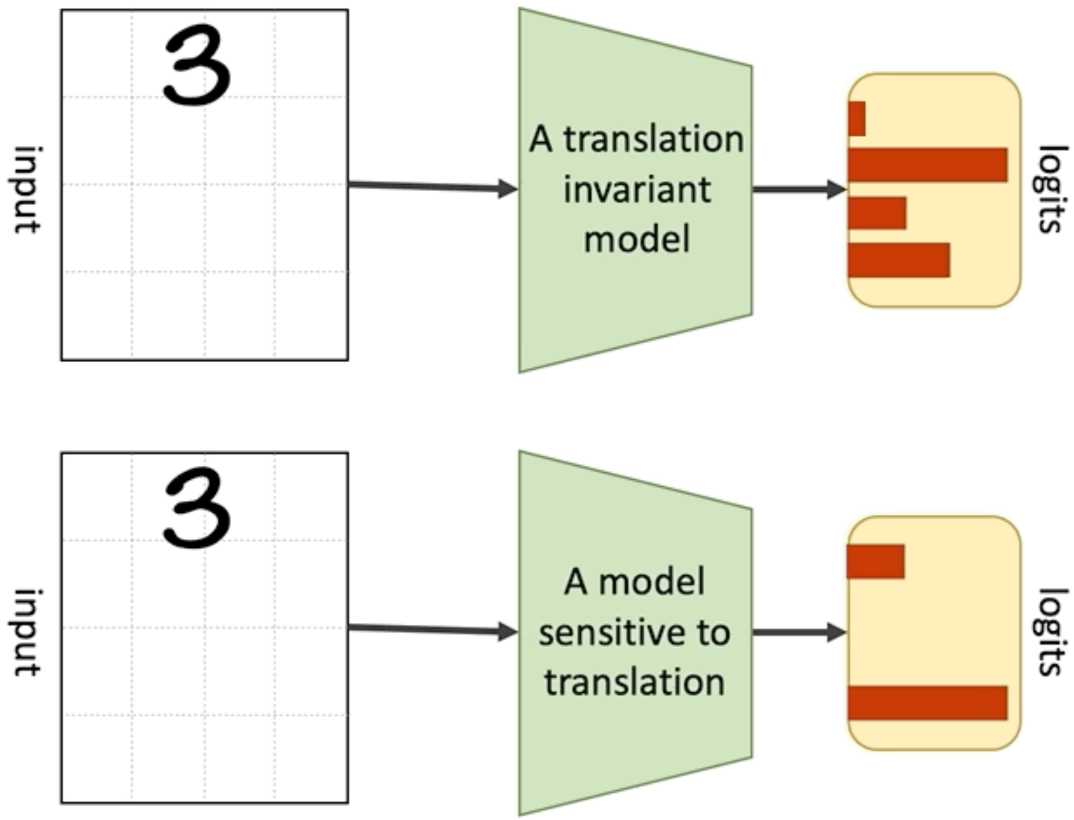
Ravindran and Barto «Approximate Homomorphisms: A framework for non-exact minimization in Markov Decision Processes» , 2004

MDP homomorphic network

- We define as *MDP homomorphic network* the connection between the **equivariant neural networks** and the symmetries in RL.

- Equivariant neural networks are a class of NNs, which have built-in symmetries.

- E.g., **Convolutional Neural Networks (CNNs)** are equivariant to translations but not to other transformations.

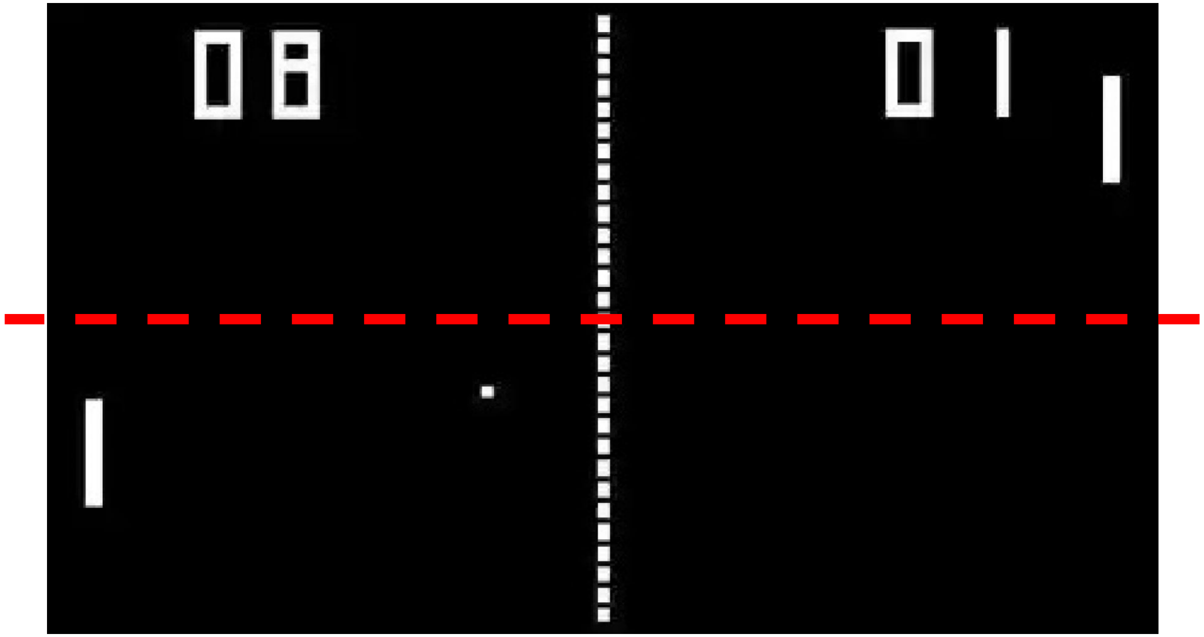


https://samiraabnar.github.io/articles/2020-05/indist

Applications: Pong

Atari video game developed in 1972

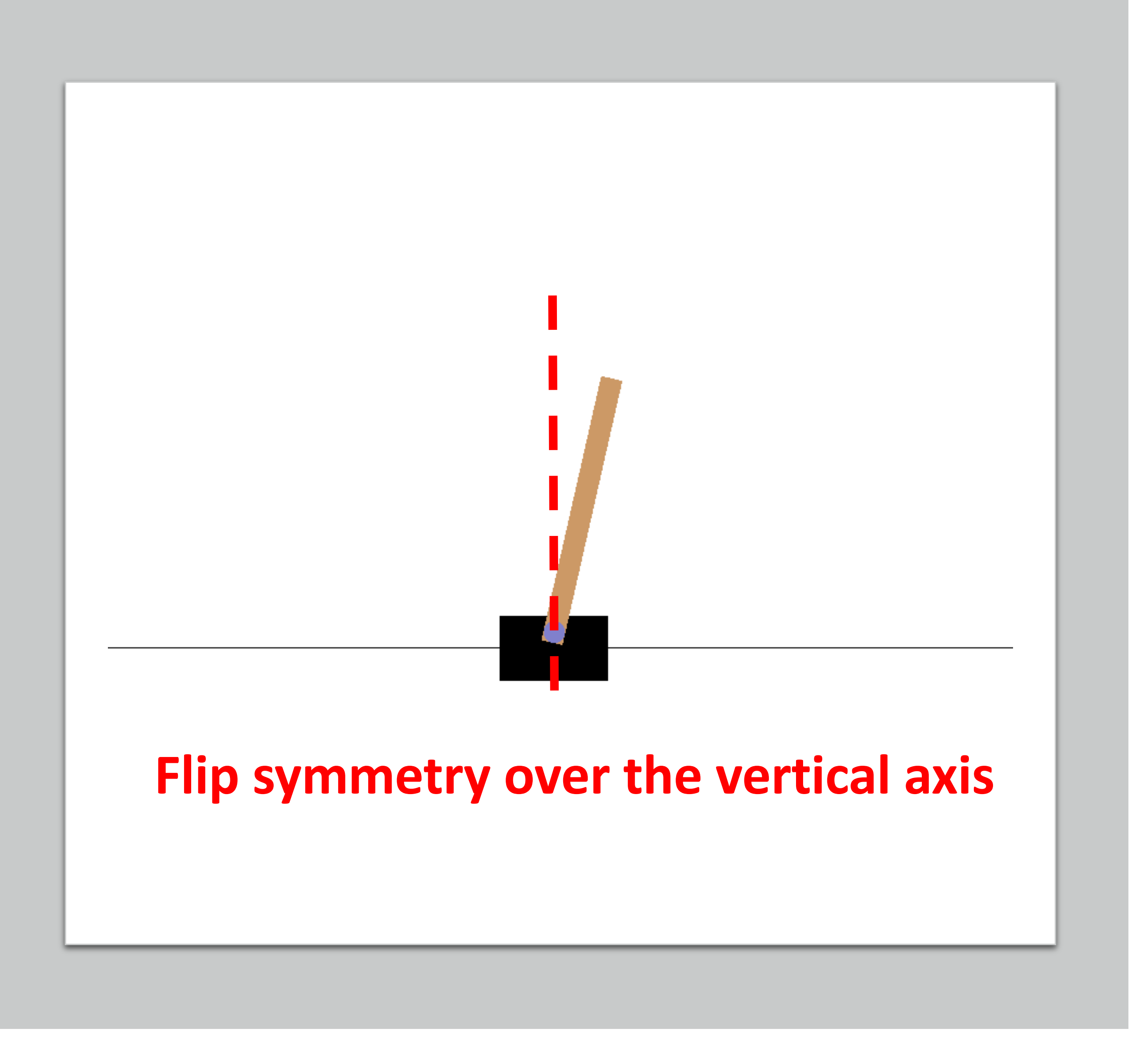
The symmetric design reflects the way it was created



Flip symmetry over the horizontal axis

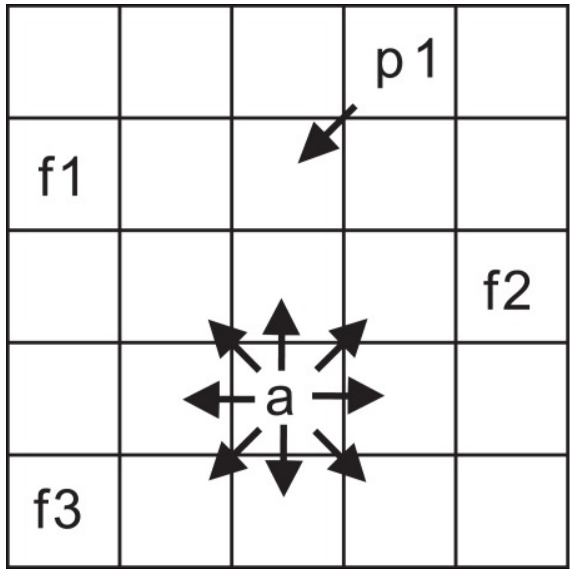
Applications: CartPole

- Game in the Open-AI Gym reinforcement learning environment

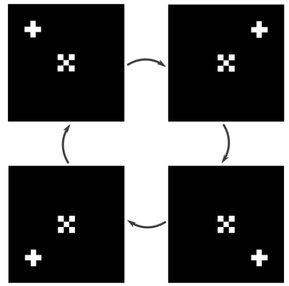


Applications: Grid world

Popular game in the reinforcement learning environment



Four-fold rotational symmetry



Conclusion

Faster convergence means better real-time applications.

Less computations needed mean cheaper hardware is required and the spread of technology will increase.