Equivariant RL

Max Welling

Joint Work with:
Elise van der Pol*
Daniel Worrall
Herke van Hoof
Frans Oliehoek

* Main contributor + created the slides

Co-Authors

Frans Oliehoek

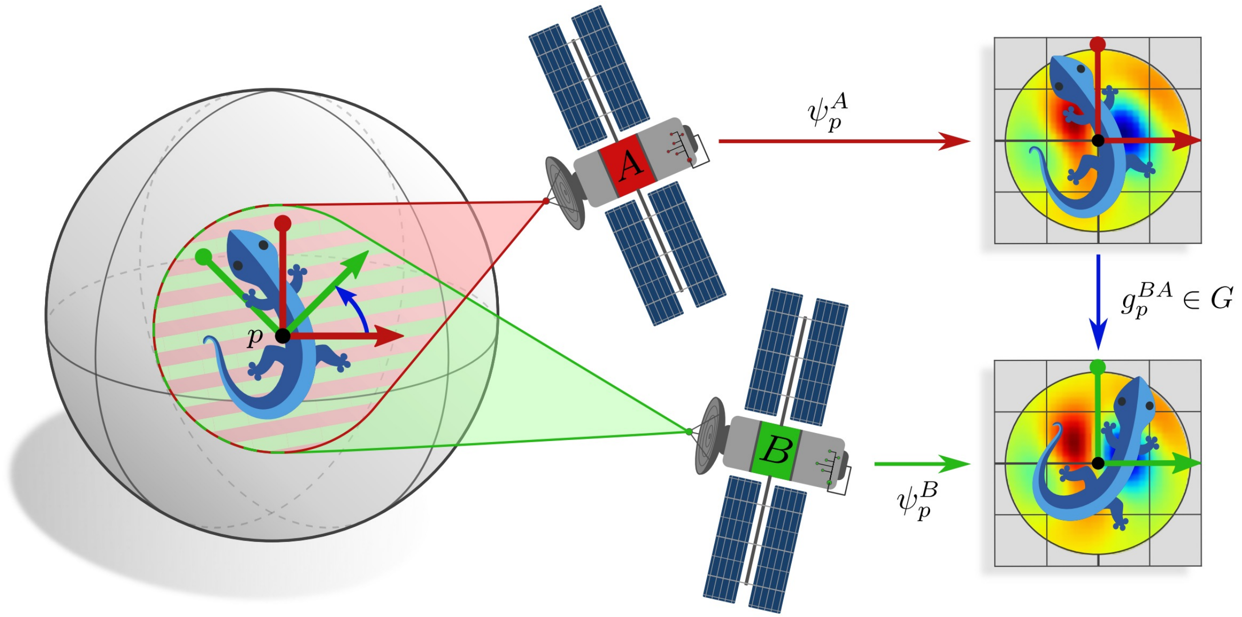
Elise van der Pol

Herke van Hoof

Daniel Worrall

Overview

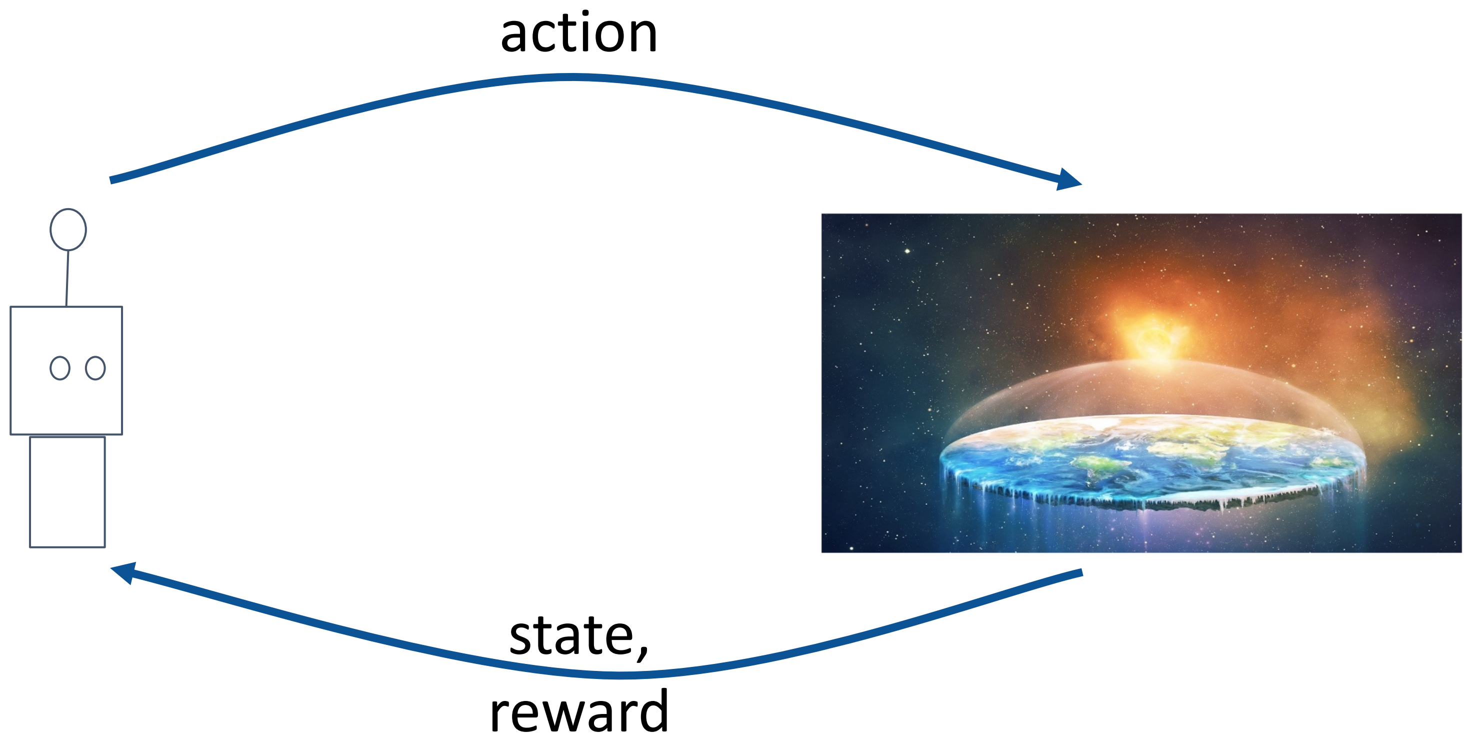
- RL & Optimization
- MDP homomorpisms & equivariance
- Equivariant multiagent systems



Picture created by Maurice Weiler

(Lizards adapted under the Creative Commons Attribution 4.0 International license by courtesy of Twitter.)

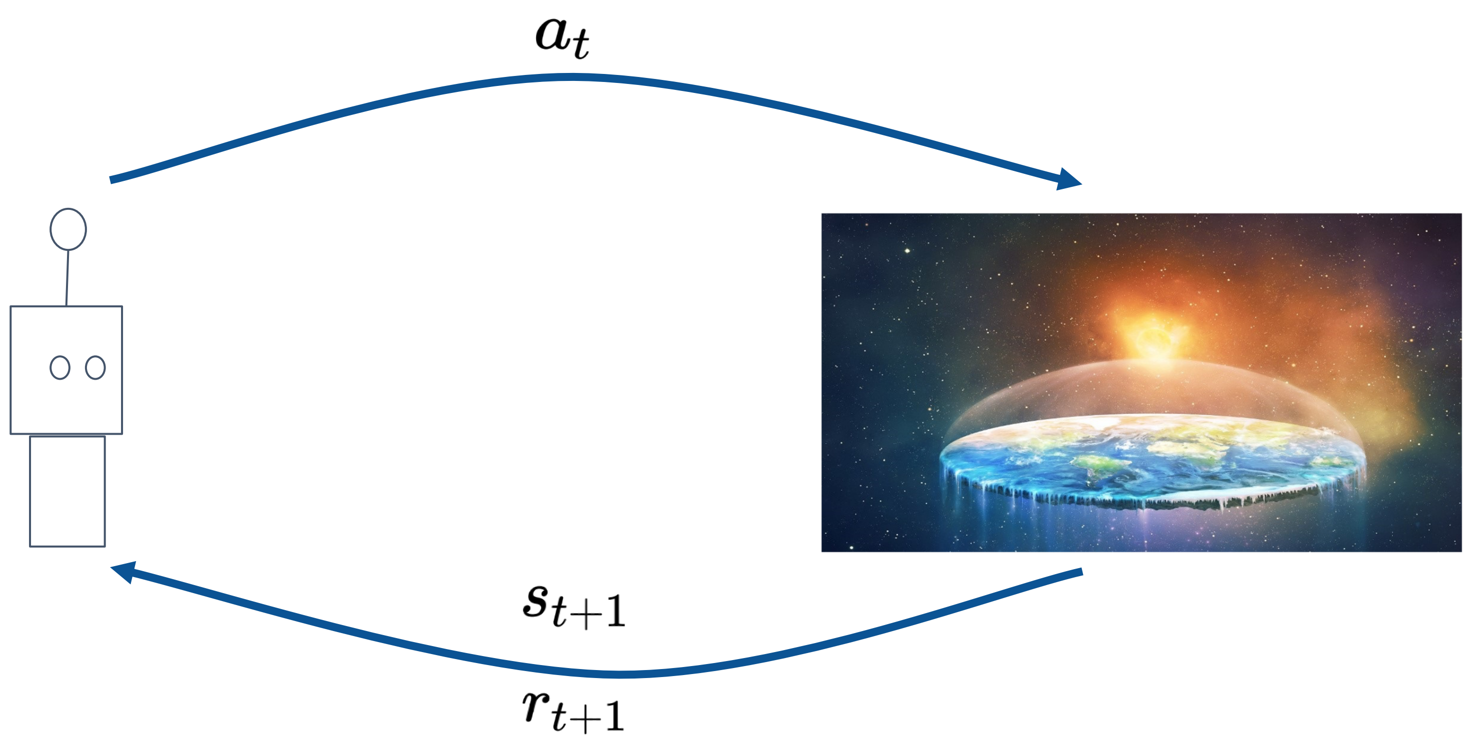
What is Reinforcement Learning?



Learning from trial and error

What is Reinforcement Learning?

$a_t = \pi(s_t)$



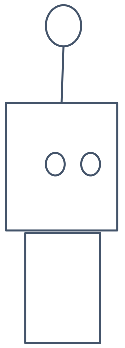
$s_{t+1} = T(s_t, a_t)$
$r_{t+1} = R(s_t, a_t)$

Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, T, R)$

What is Reinforcement Learning?

a_t

$a_t = \pi(s_t)$



$$s_{t+1} = T(s_t, a_t)$$
$$r_{t+1} = R(s_t, a_t)$$

s_{t+1}

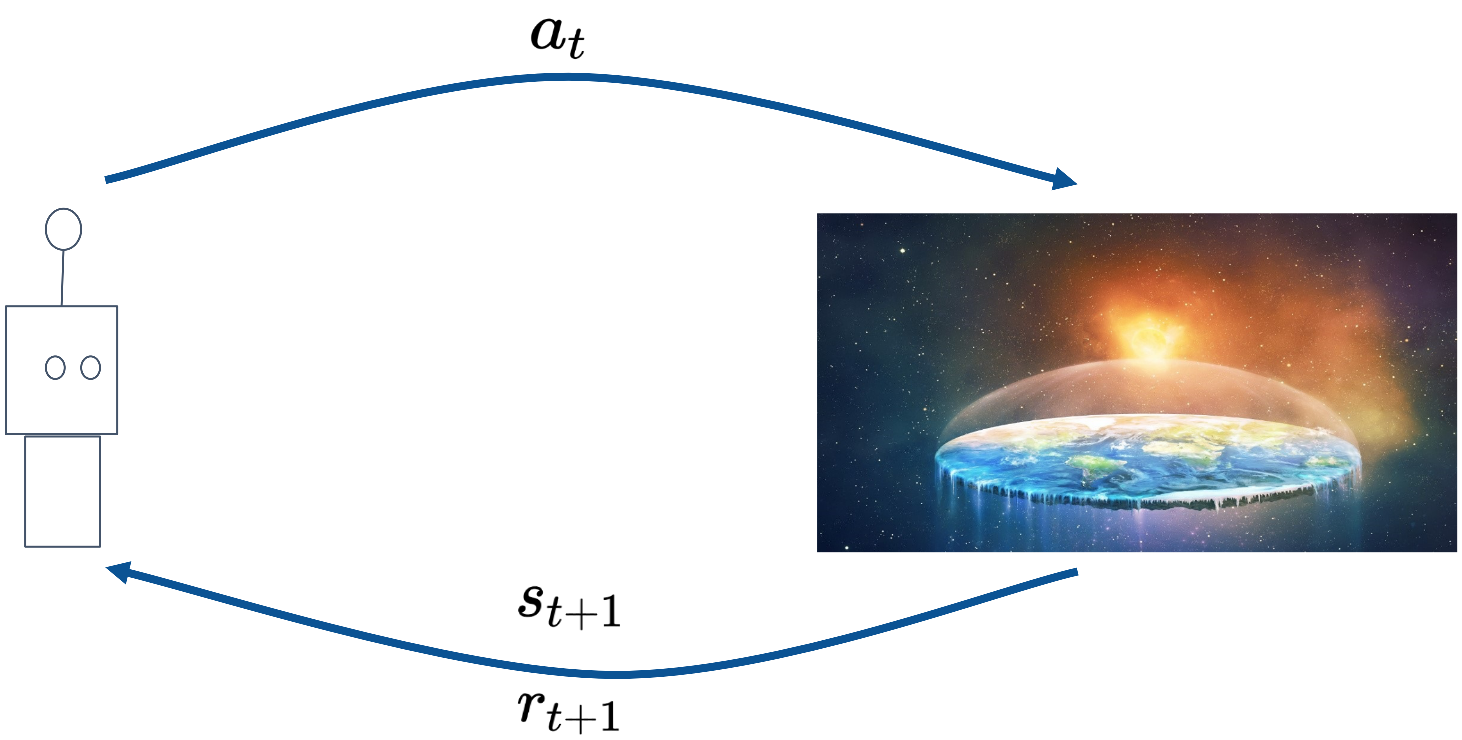
r_{t+1}

Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, T, R)$

states

What is Reinforcement Learning?

$a_t = \pi(s_t)$



$$s_{t+1} = T(s_t, a_t)$$
$$r_{t+1} = R(s_t, a_t)$$

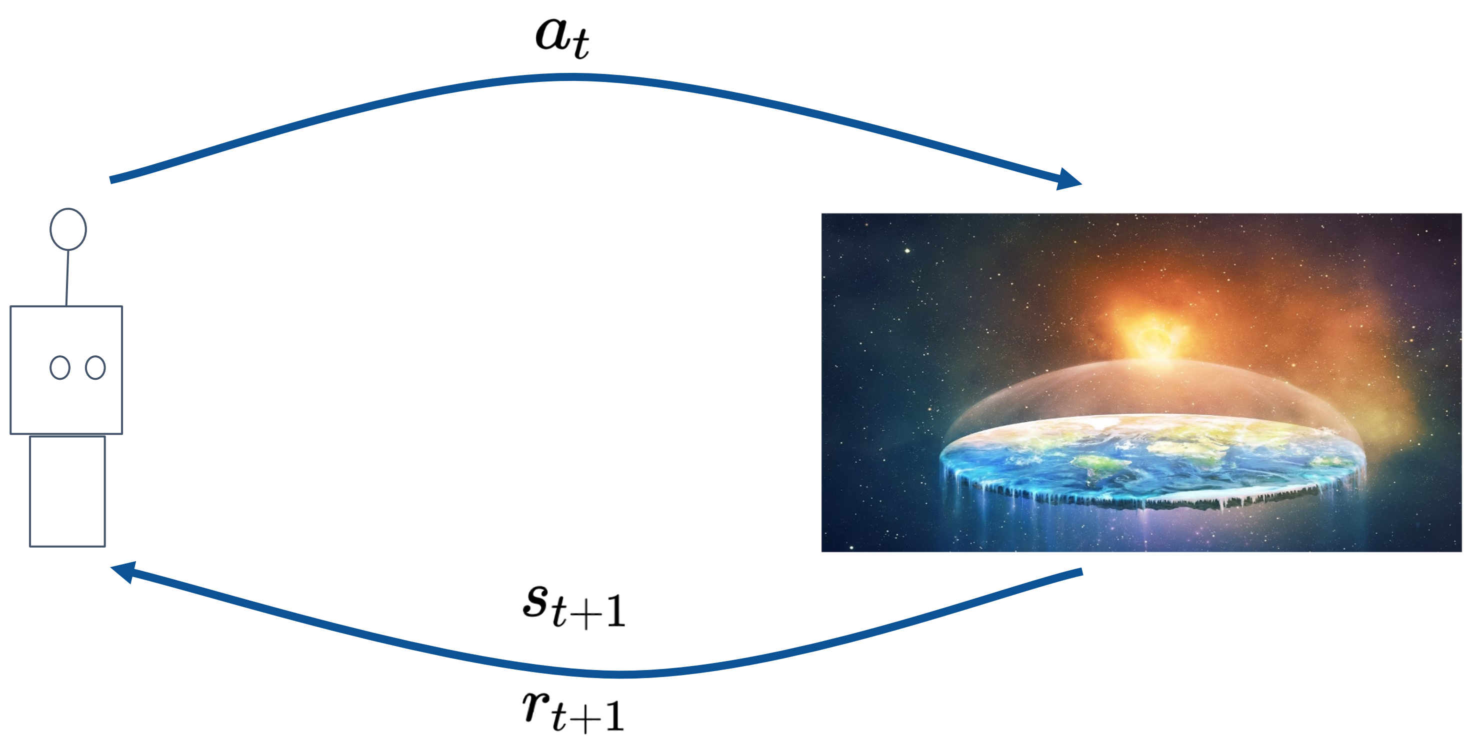
actions

Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, T, R)$

states

What is Reinforcement Learning?

$$a_t = \pi(s_t)$$



$$s_{t+1} = T(s_t, a_t)$$
$$r_{t+1} = R(s_t, a_t)$$

actions

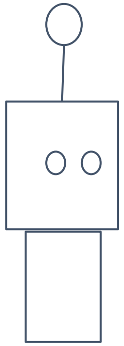
Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, T, R)$

states transition function

What is Reinforcement Learning?

a_t

$a_t = \pi(s_t)$



$$s_{t+1} = T(s_t, a_t)$$
$$r_{t+1} = R(s_t, a_t)$$

s_{t+1}

r_{t+1}

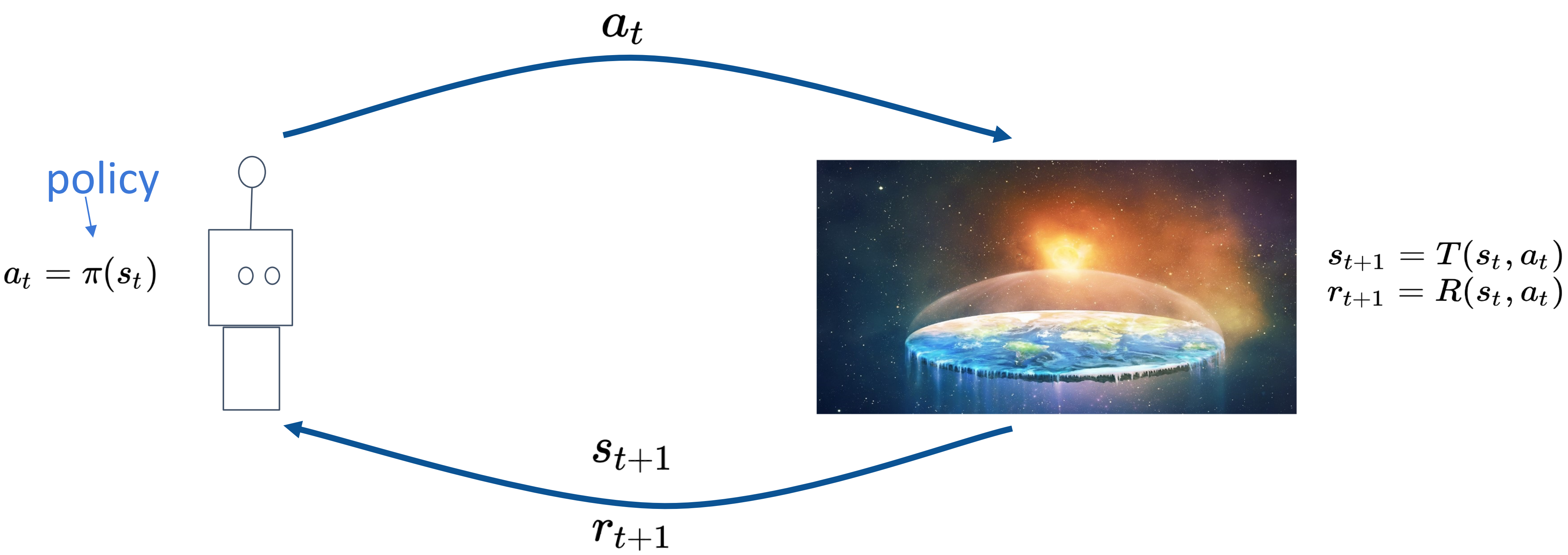
reward function

actions

Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, T, R)$

states   transition function

What is Reinforcement Learning?



Goal: Policy that maximizes cumulative reward

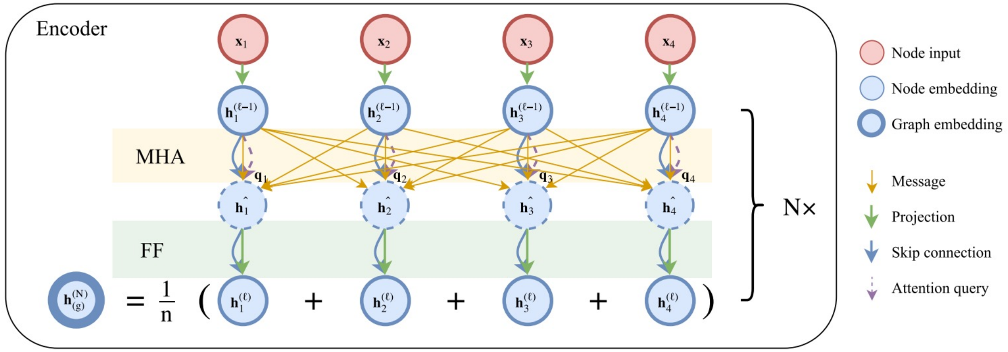
Relation between RL & Optimization

ATTENTION, LEARN TO SOLVE ROUTING PROBLEMS!

Wouter Kool
University of Amsterdam
ORTEC
w.w.m.kool@uva.nl

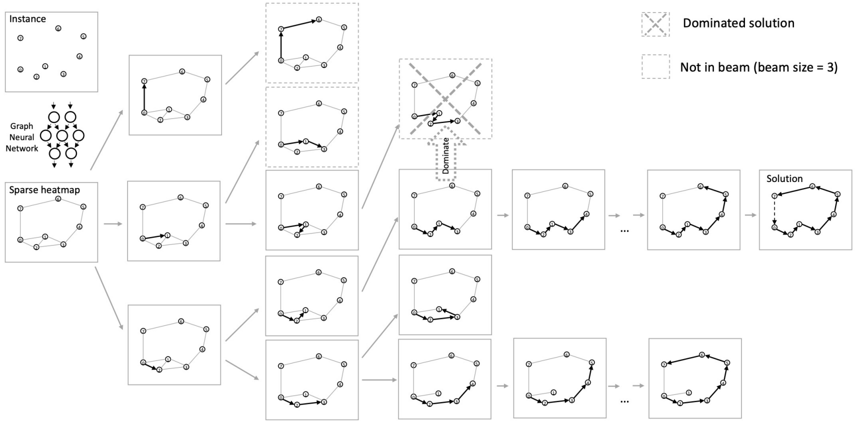
Herke van Hoof
University of Amsterdam
h.c.vanhoof@uva.nl

Max Welling
University of Amsterdam
CIFAR
m.welling@uva.nl

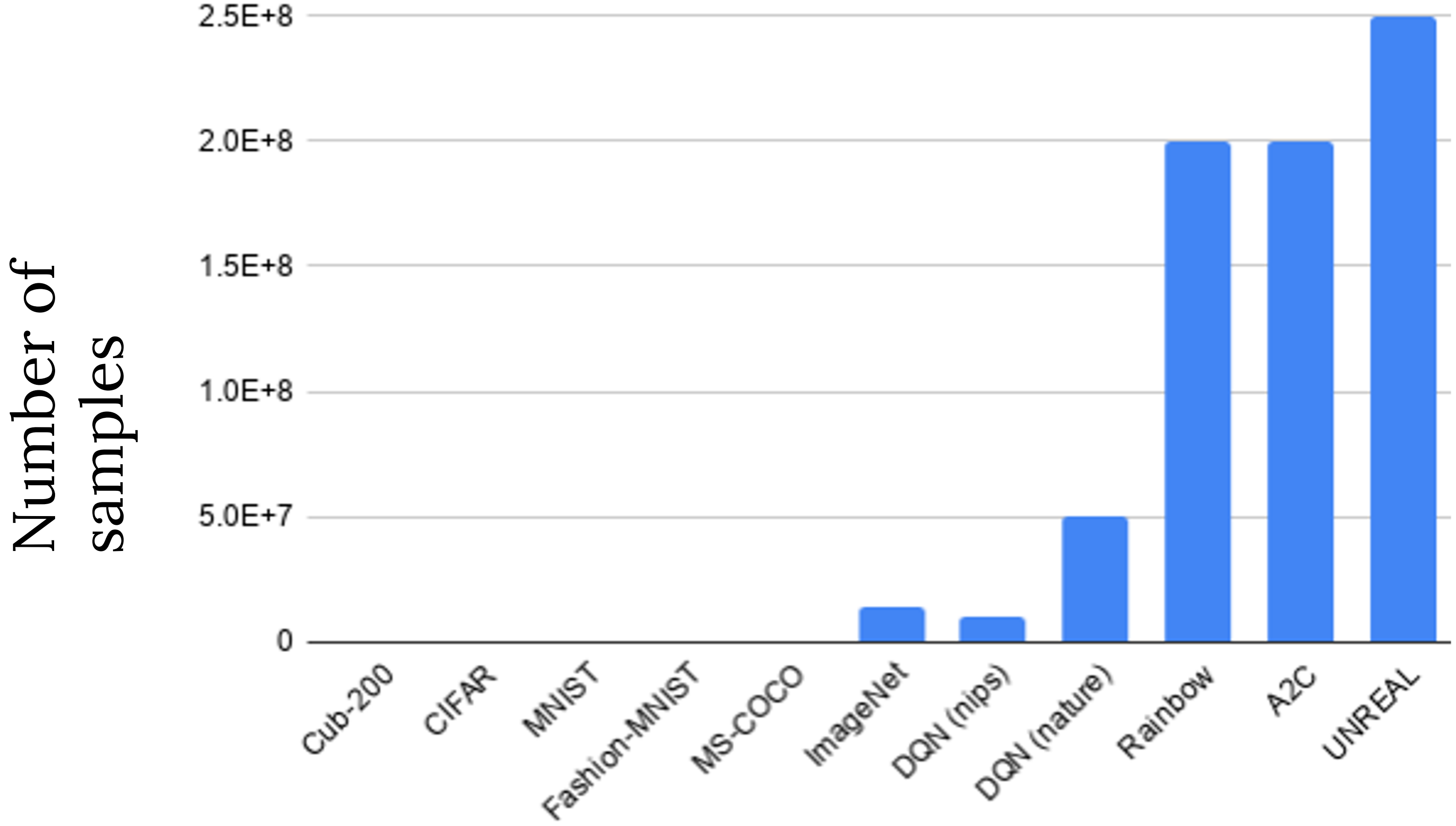


Deep Policy Dynamic Programming for Vehicle Routing Problems

Wouter Kool [1 2] **Herke van Hoof** [1] **Joaquim Gromicho** [1 2] **Max Welling** [1 3]

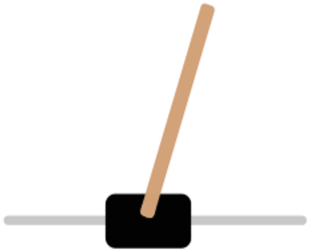


Reinforcement learning is very data hungry

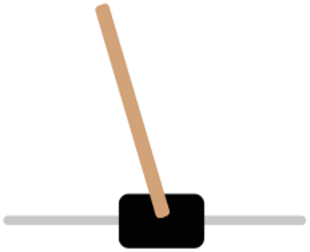


There are useful symmetries in RL!

?
≠

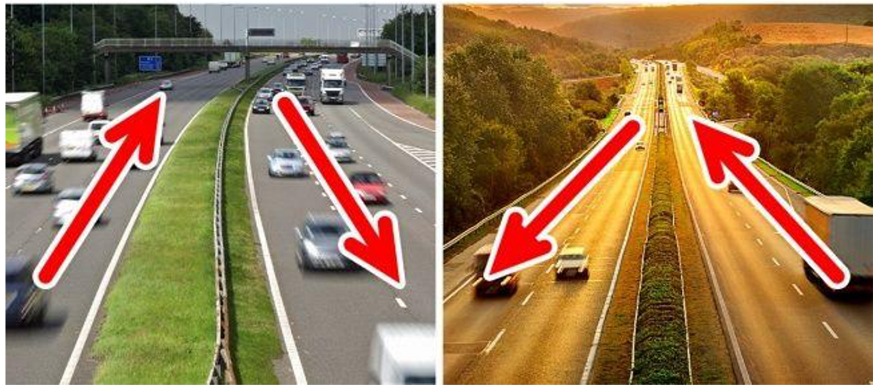


?
≠



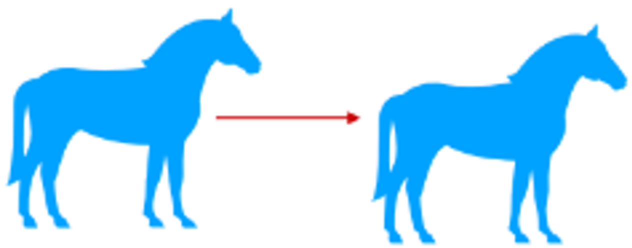
?
≠

Cambridge Amsterdam

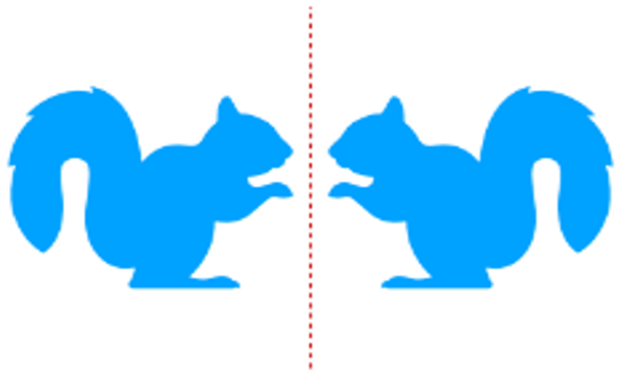


What is a group?

Examples:



Translations



Reflections

Rotations

A set with an operation obeying the group axioms

(identity, invertibility, closure, associativity)

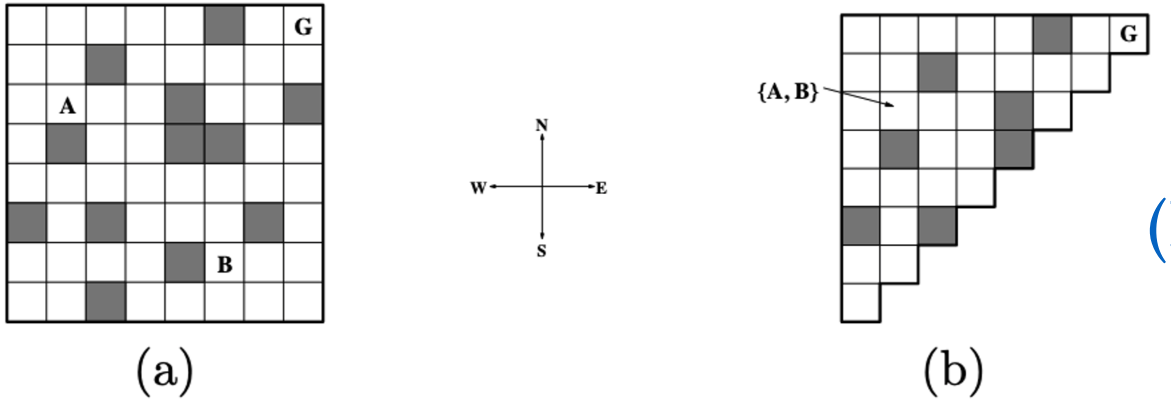
Figures from Daniel Worrall's MLSS slides

Symmetries in Reinforcement Learning

For all states and actions, and all group elements:

Rewards and dynamics are invariant under group transformations:

$$R(s,a) = R(gs,ga) \qquad T(s'|s,a) = T(gs'|gs,ga)$$

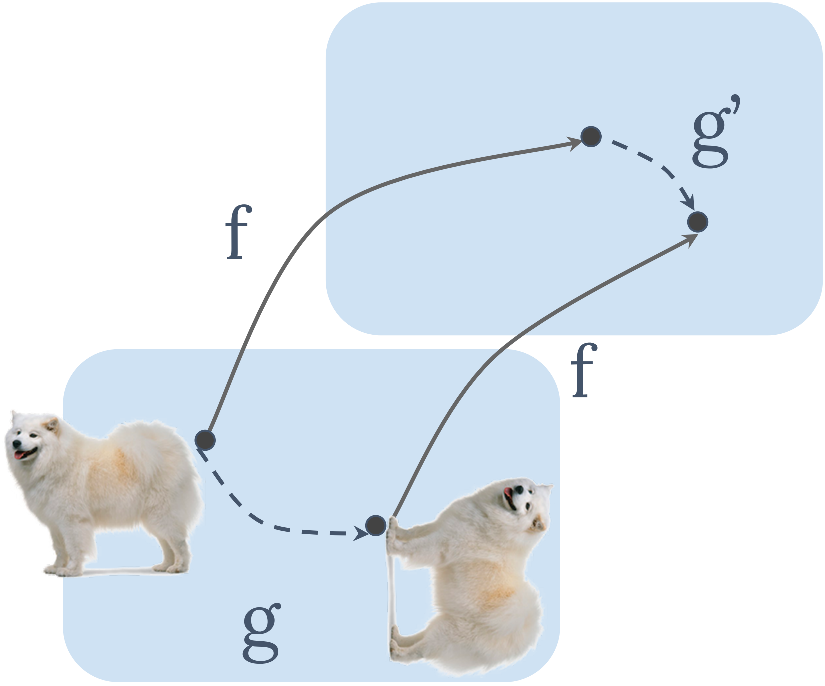


(Ravindran & Barto 2004)

(s,a) and (gs,ga) are symmetric state-action pairs and have the same π^*

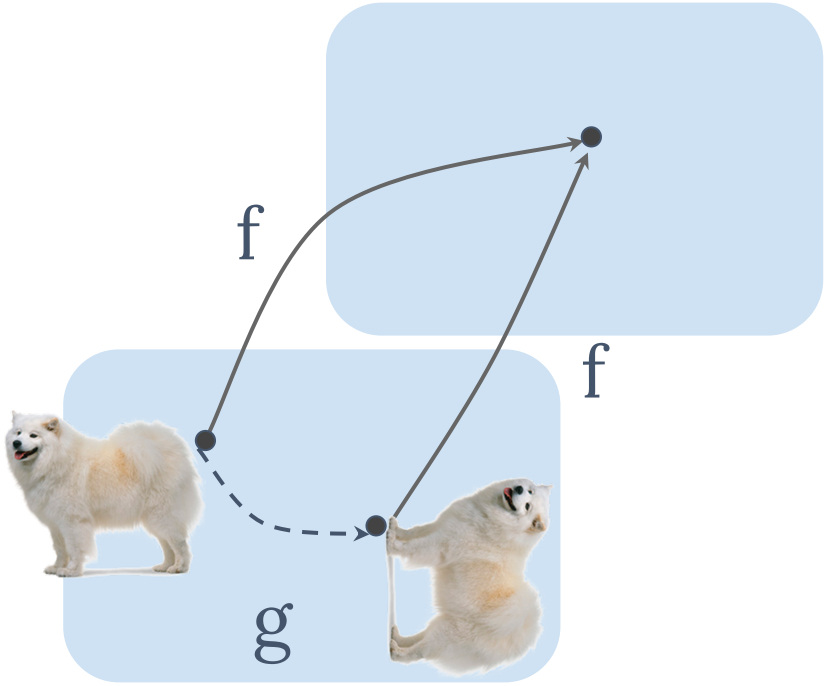
Equivariance

$$f(gx) = g' f(x)$$



Invariance

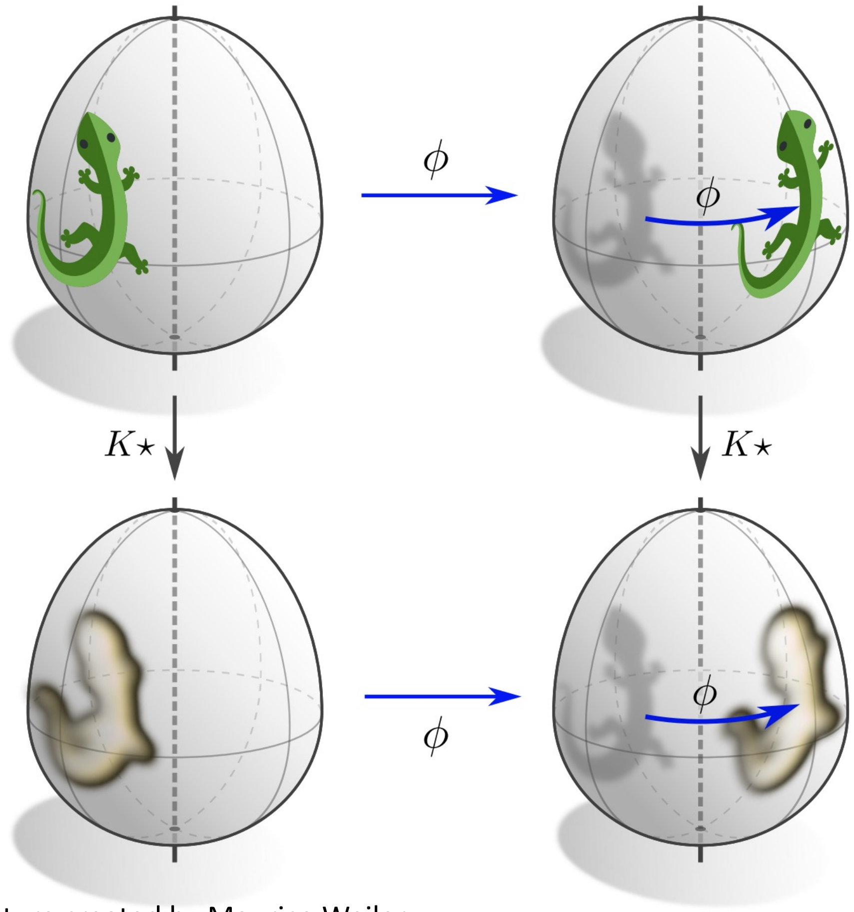
$$f(gx) = f(x)$$



Figures adapted from Daniel E. Worrall

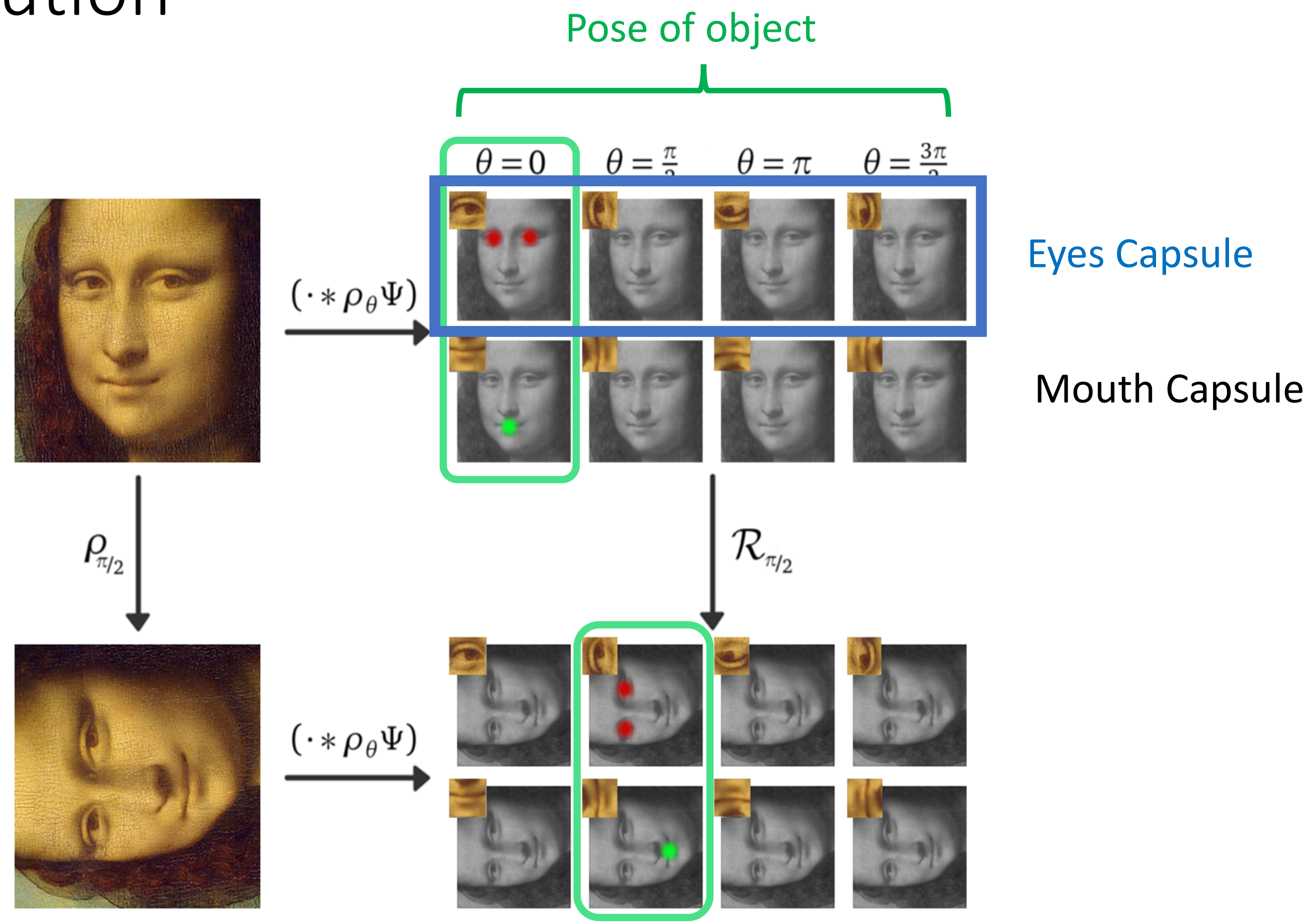
Advantages Equivariance

- First transform then convolve =
  First convolve then transform
- 'Encapsulates' symmetries of input
- Works on manifolds
- Advantages:
  - Data efficiency
  - Disentangling pose and presence
  - Creates easy patterns for next layer
- First appearance in ML: Group CNNs
  Cohen & W. '16, Dieleman et al, '16



Picture created by Maurice Weiler

Equivariant Convolution



Homomorphism

Structure-preserving map such that $f(x \cdot y) = f(x) \cdot f(y)$

Examples:

Linear map between vector spaces

$$T(v + w) = T(v) + T(w)$$

Exponential function between the reals and the positive reals

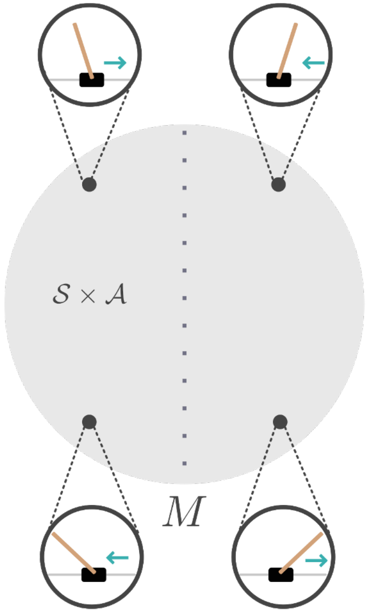
$$e^{x+y} = e^x e^y$$

Group representation between a group and the general linear group

$$\rho(g_1 g_2) = \rho(g_1)\rho(g_2)$$

MDP Homomorphisms

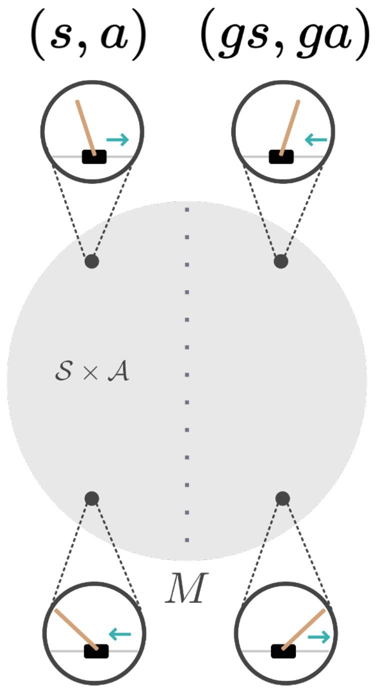
Map ground MDP → abstract MDP, preserve dynamics (Ravindran & Barto 2001)



Original problem

MDP Homomorphisms

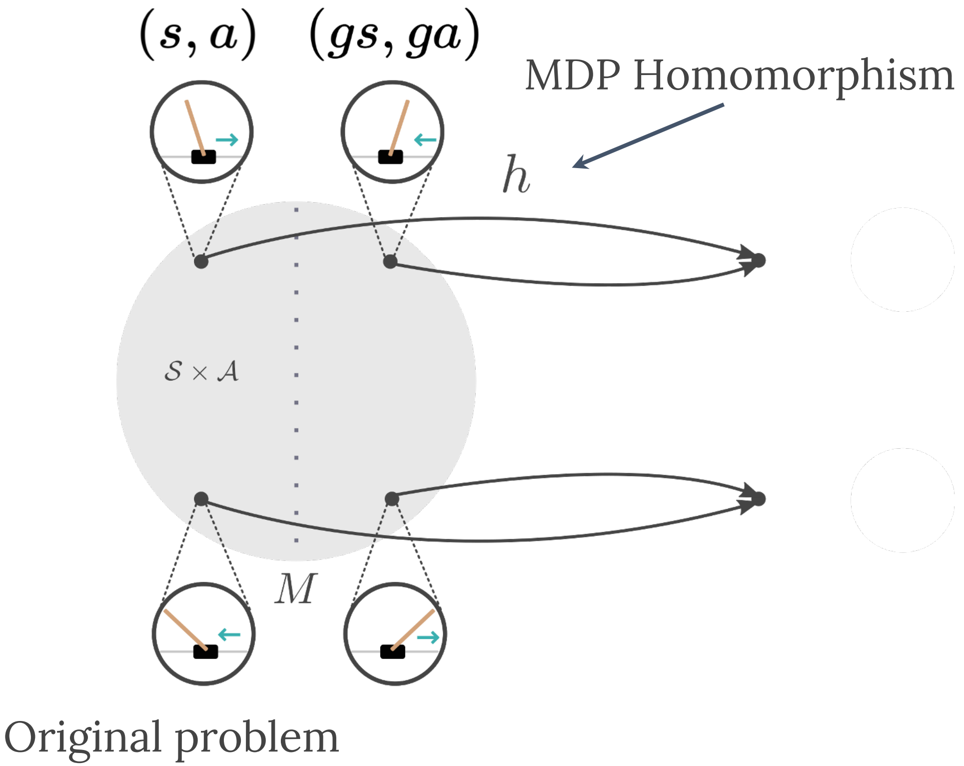
Map ground MDP → abstract MDP, preserve dynamics (Ravindran & Barto 2001)



Original problem

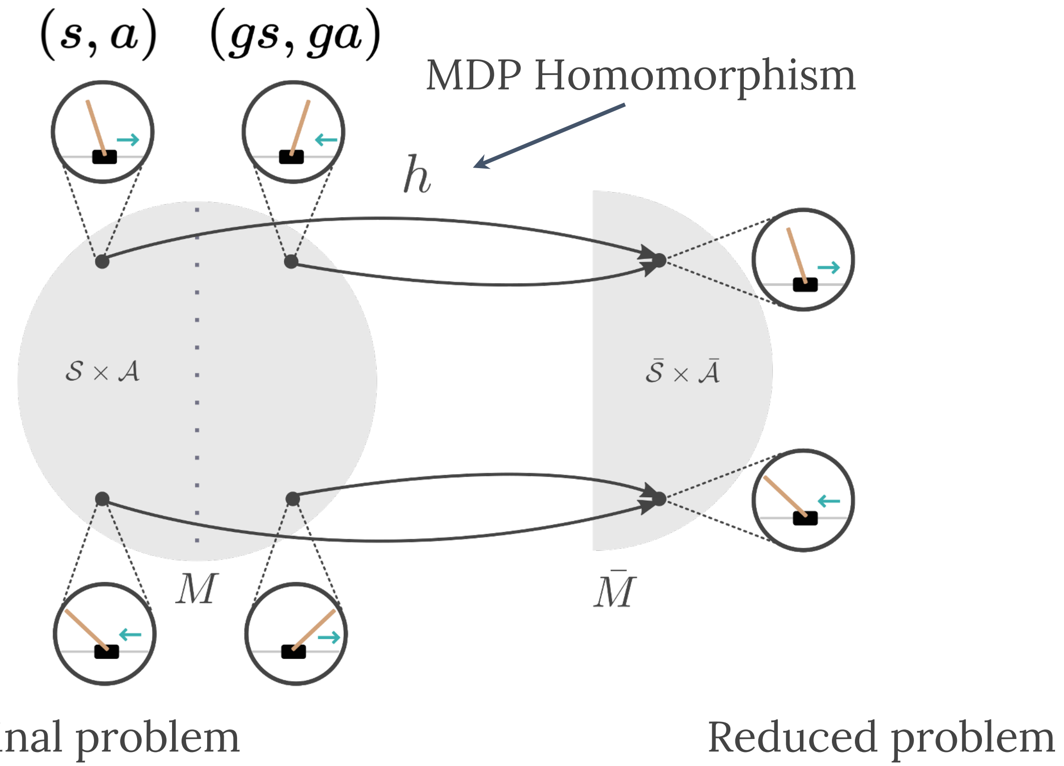
MDP Homomorphisms

Map ground MDP → abstract MDP, preserve dynamics (Ravindran & Barto 2001)



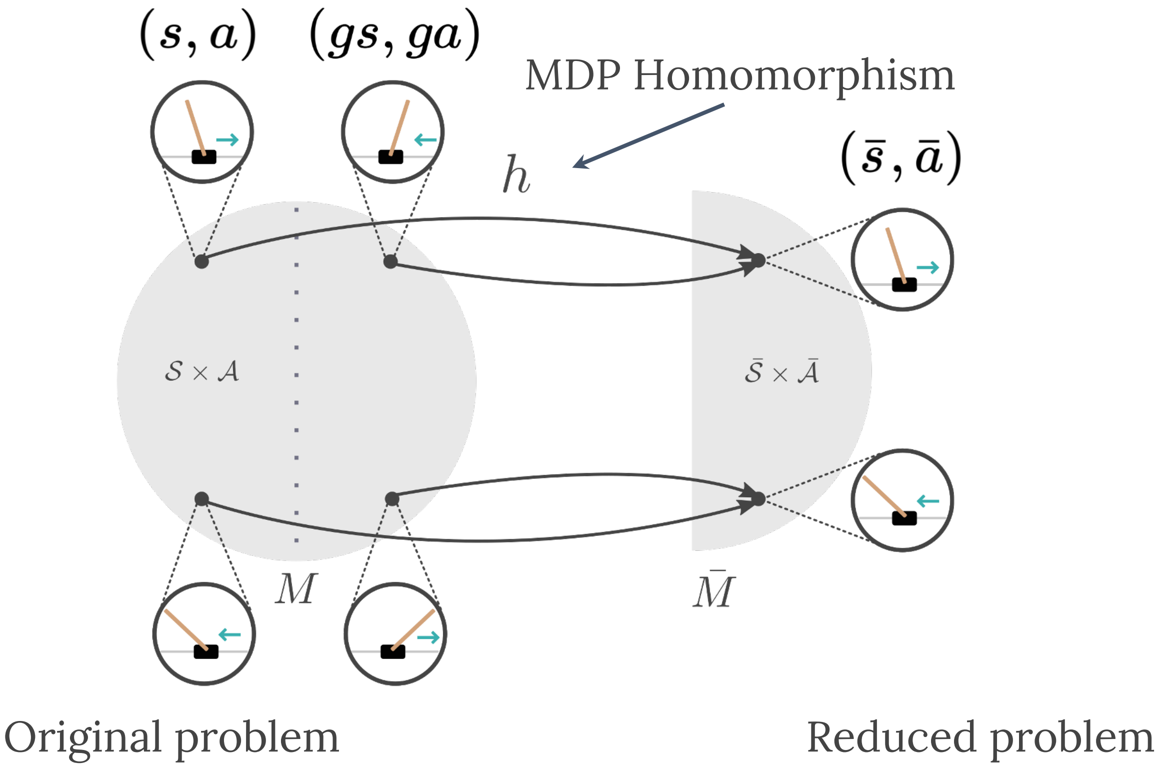
MDP Homomorphisms

Map ground MDP → abstract MDP, preserve dynamics (Ravindran & Barto 2001)



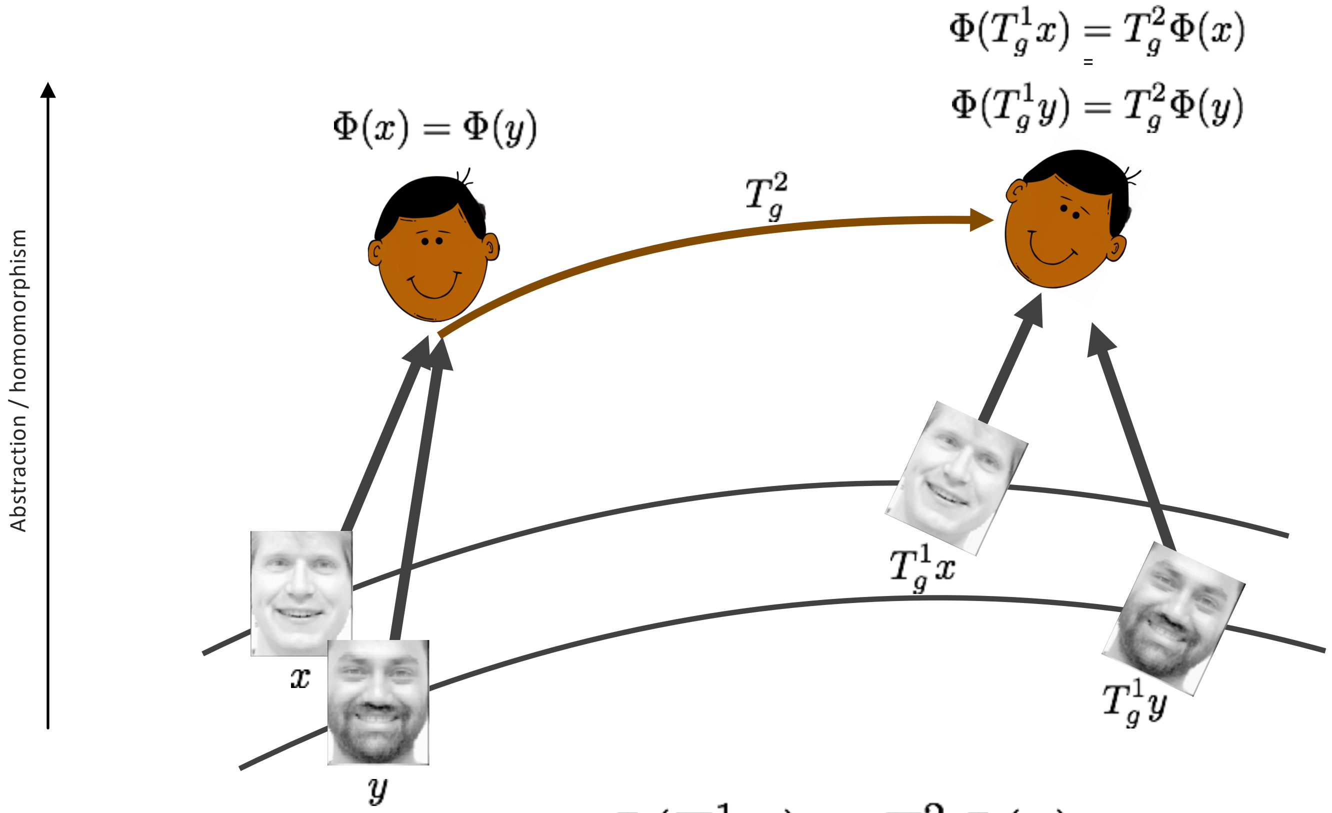
MDP Homomorphisms

Map ground MDP → abstract MDP, preserve dynamics (Ravindran & Barto 2001)



(s, a) and (gs, ga) are symmetric state-action pairs and have the same π^*

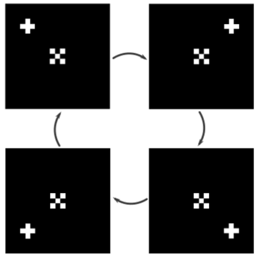
Abstractions Preserve Symmetries under Equivariance

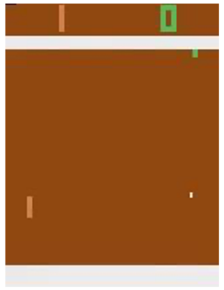


$$\Phi(T_g^1\, x) = T_g^2\, \Phi(x)$$

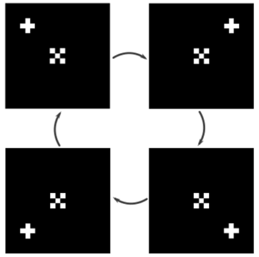
MDP Homomorphic Networks: Group Symmetries in RL

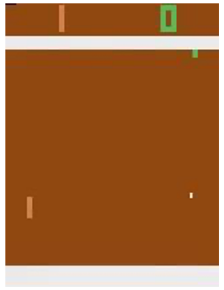
(van der Pol, Worrall, van Hoof, Oliehoek & Welling, NeurIPS 2020)

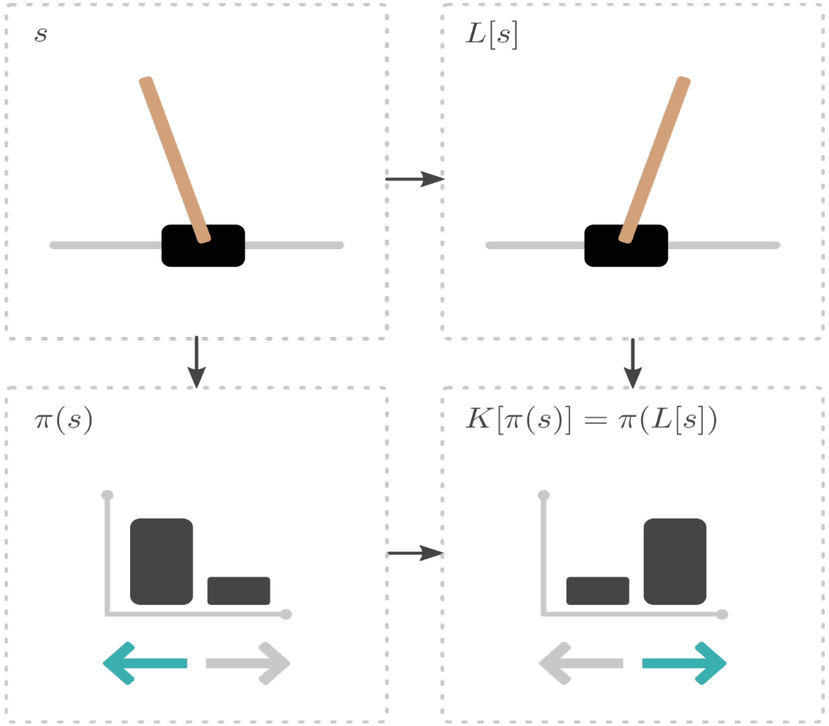




MDP Homomorphic Networks: Group Symmetries in RL







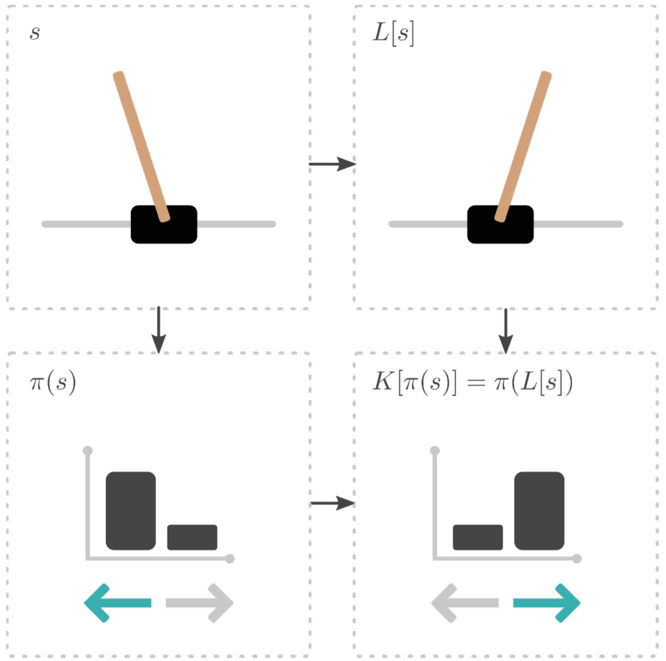
MDP Homomorphic Networks: Group Symmetries in RL

(van der Pol, Worrall, van Hoof, Oliehoek & Welling, NeurIPS 2020)

Symmetric (s, a) pairs have the same policy π :

$$K[\pi(s)] = \pi(L[s])$$

L is a transformation on states, K a transformation on policies



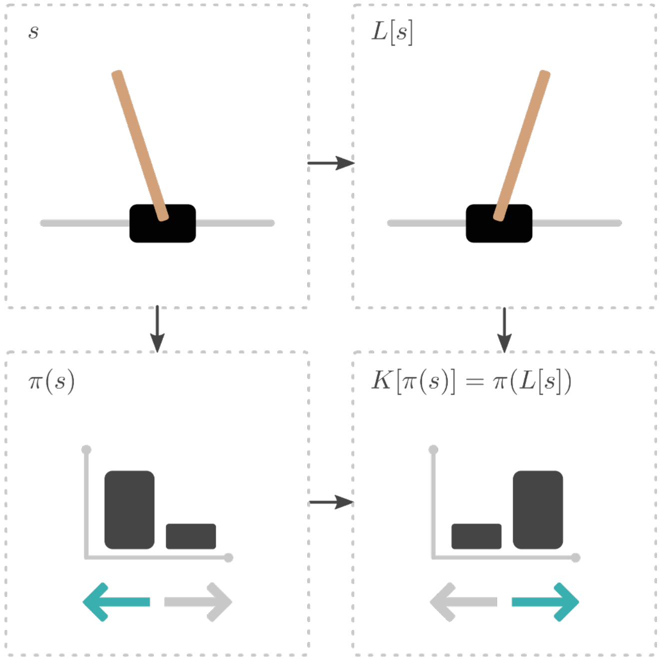
MDP Homomorphic Networks:
Group Symmetries in RL

(van der Pol, Worrall, van Hoof, Oliehoek & Welling, NeurIPS 2020)

Symmetric (s, a) pairs have the same policy π :

$$K[\pi(s)] = \pi(L[s])$$

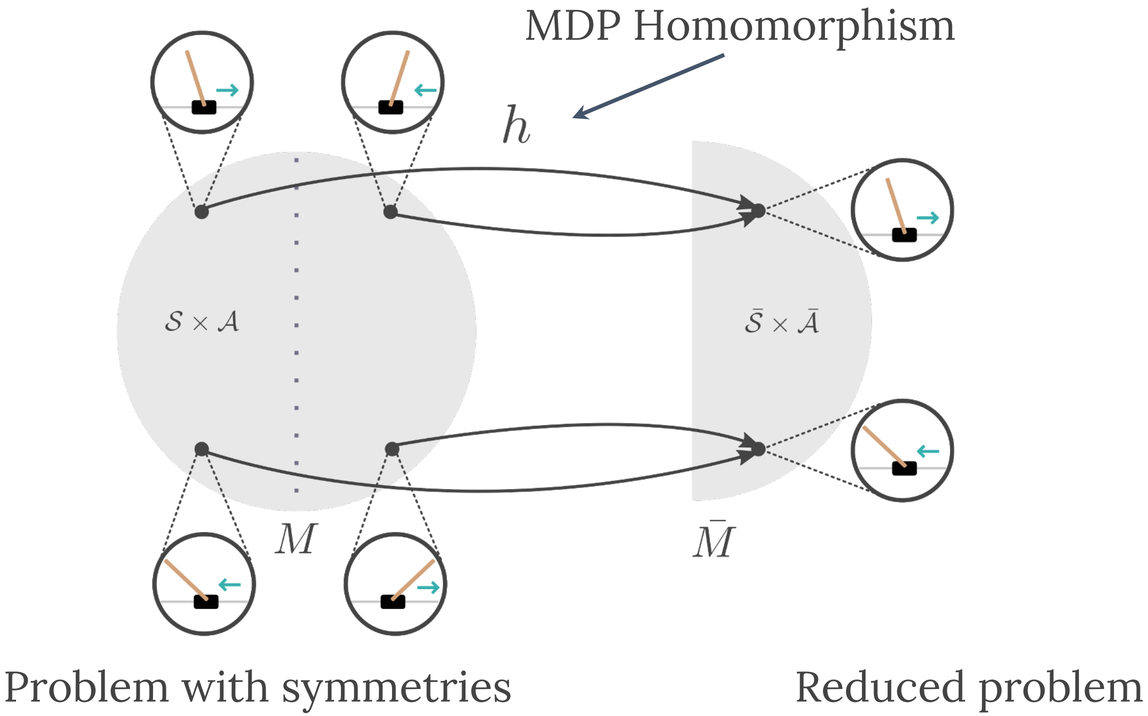
L is a transformation on states, K a transformation on policies



MDP homomorphic networks exploit symmetries in reinforcement learning

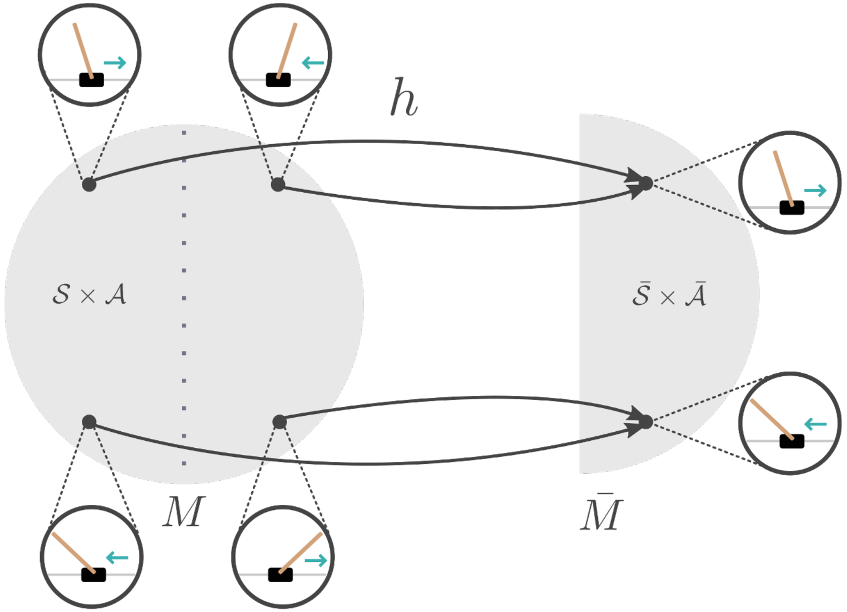
MDP Homomorphic Networks

We bridge MDP homomorphisms and equivariant networks



MDP Homomorphic Networks

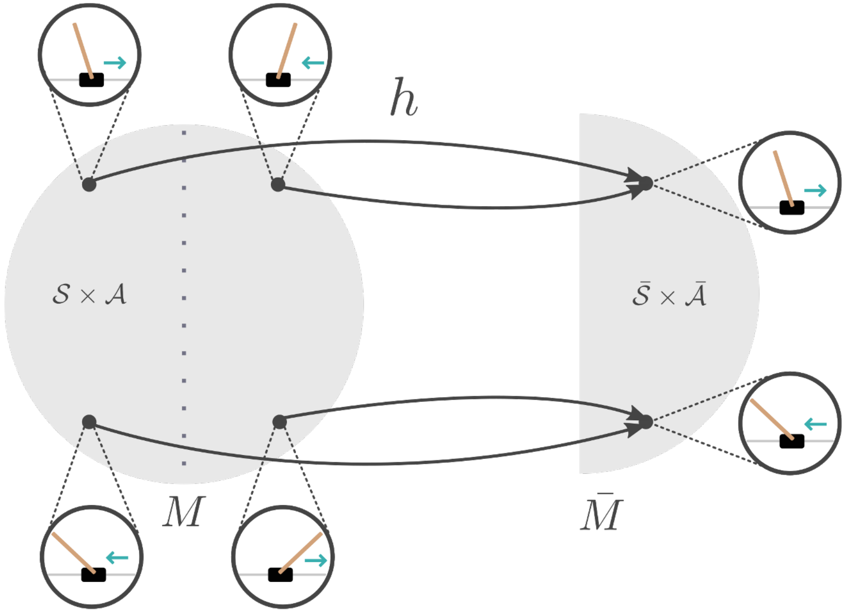
We bridge MDP homomorphisms and equivariant networks



We create deep networks constrained by MDP homomorphisms that enforce equivariance

MDP Homomorphic Networks

We bridge MDP homomorphisms and equivariant networks



We create deep networks constrained by MDP homomorphisms that enforce equivariance

We introduce a new method, the Symmetrizer, to construct equivariant weights

Group Equivariant CNN

Model policy with a group equivariant CNN

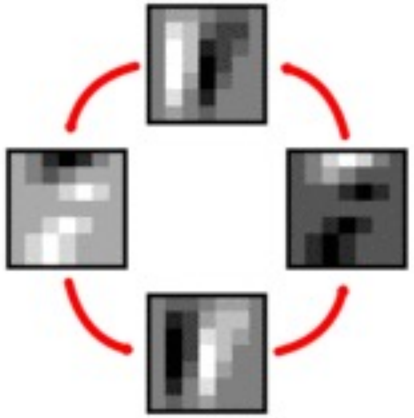
$$\pi(x) = \mathrm{GCNN}(x; w)$$

$$K\pi(x) = \mathrm{GCNN}(Lx; w)$$

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam

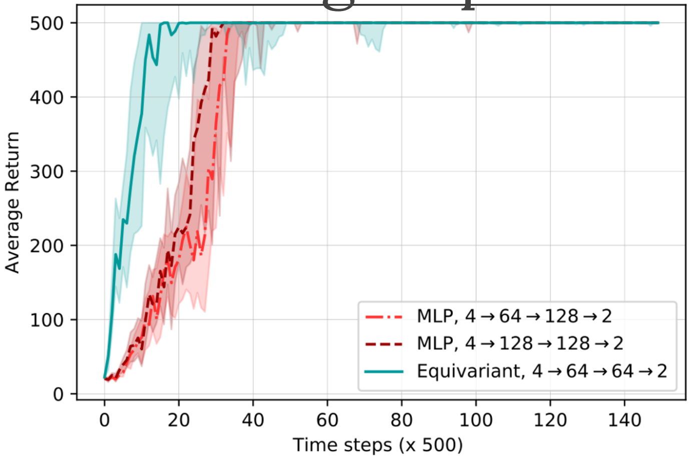
Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

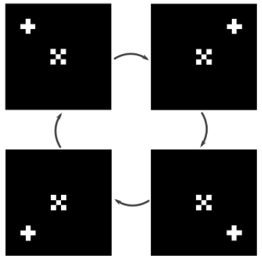


MDP Homomorphic Networks

Cartpole

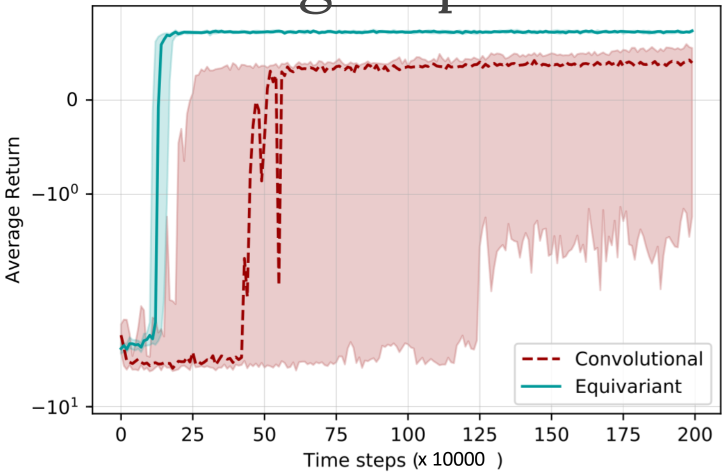
2 element symmetry group

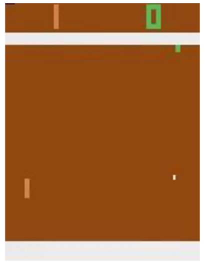




Grid World

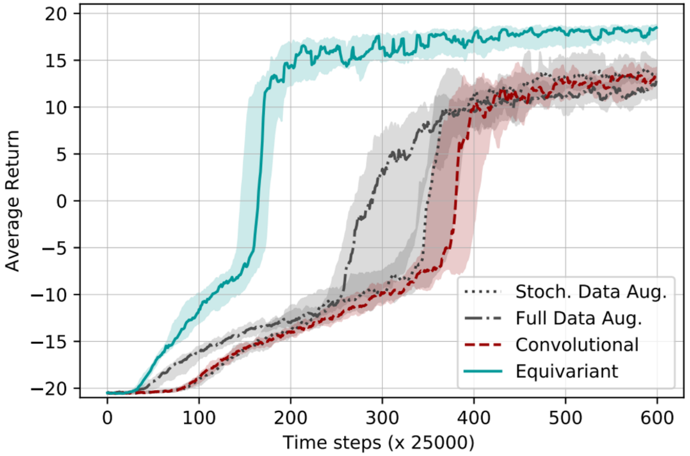
4 element symmetry group





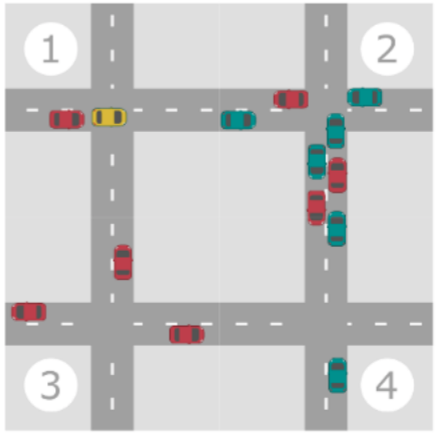
Pong

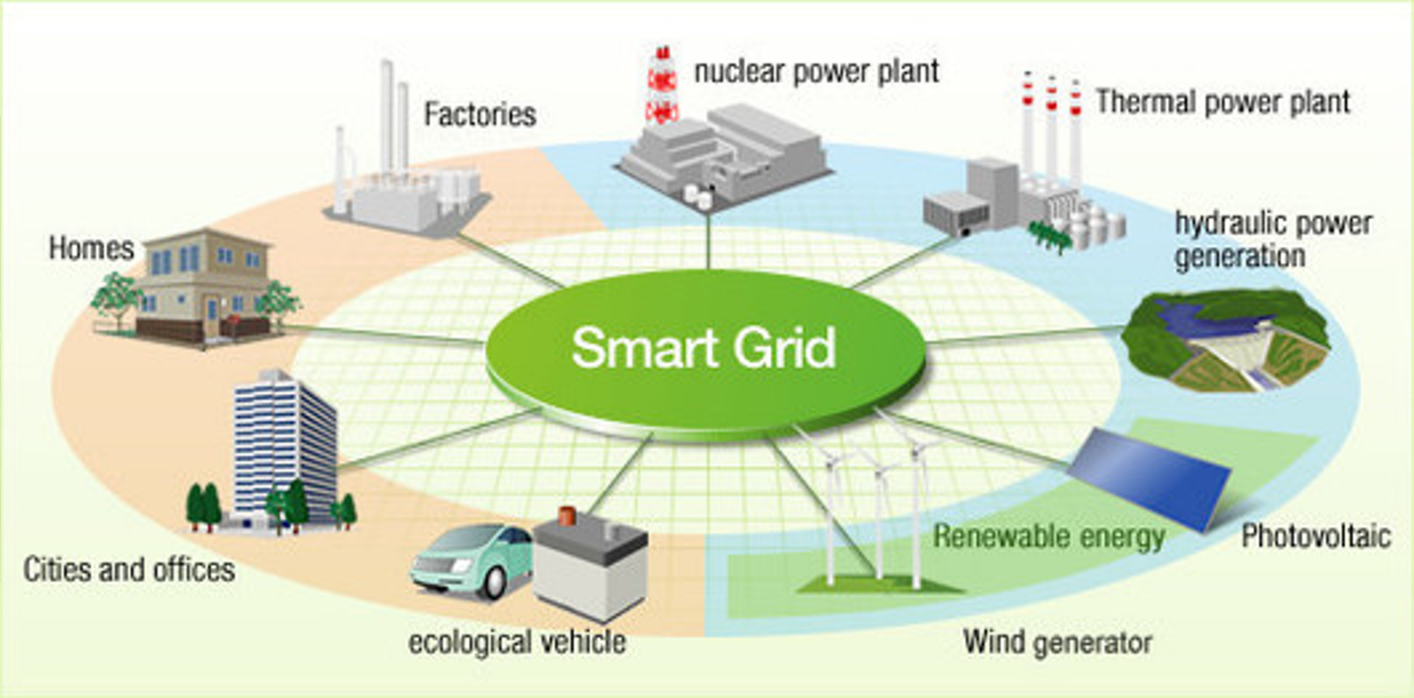
2 element symmetry group



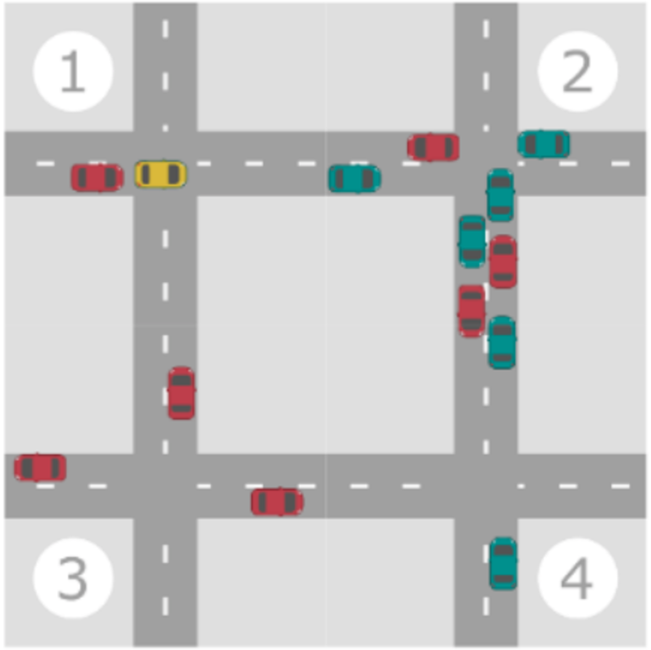
Fewer interactions with the world needed

Multi-Agent Systems

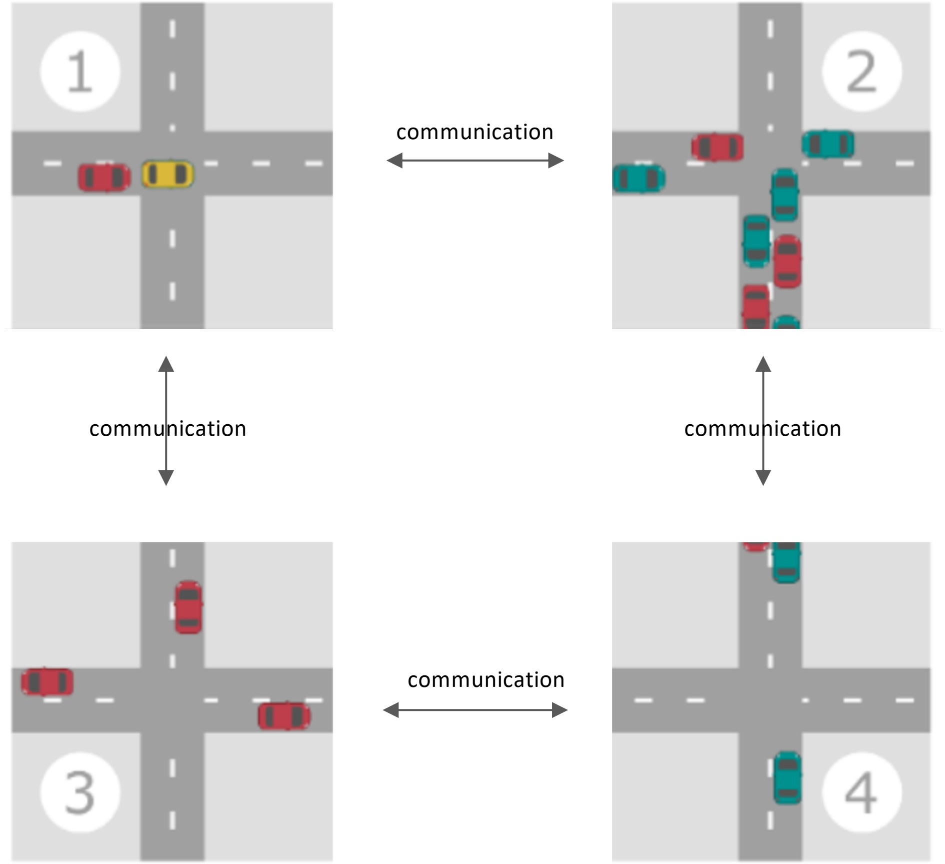




Centralized vs Distributed Multi-Agent Systems



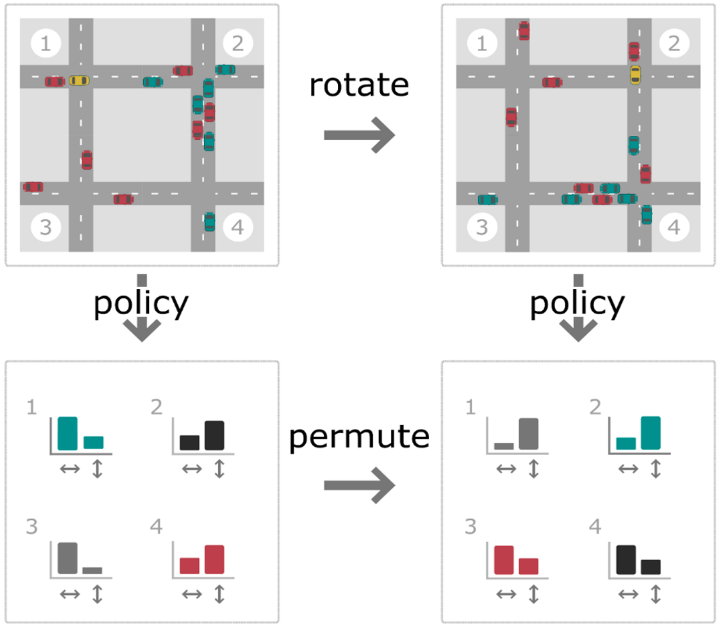
Centralized: CNN



Distributed: GNN

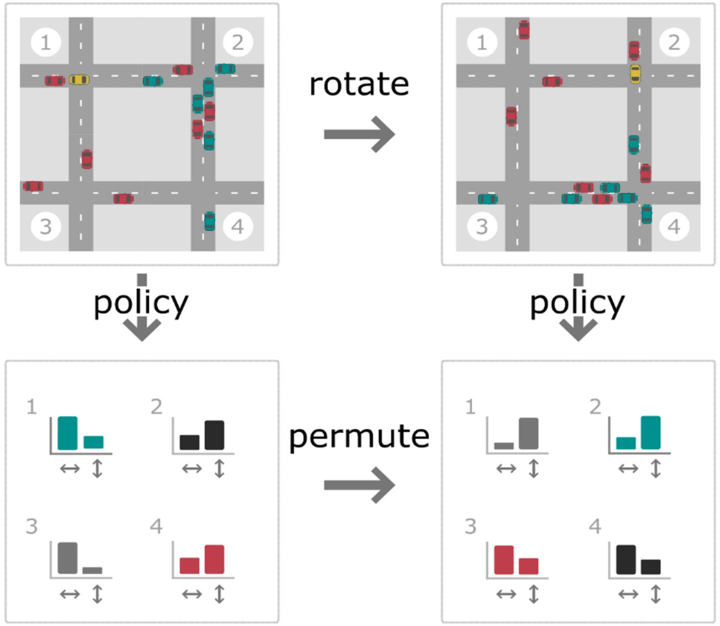
Multi-Agent MDP Homomorphic Networks

(van der Pol, van Hoof, Oliehoek & Welling, under review 2021)



Multi-Agent MDP Homomorphic Networks

Distributed: global equivariance through local equivariant computation & equivariant communication



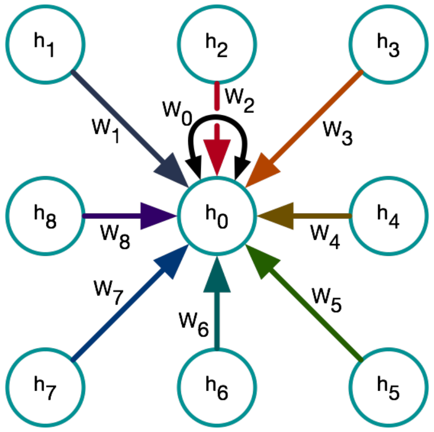
Model policy as equivariant Graph NN:

$$\pi(x) = \mathrm{EGNN}(x; w)$$

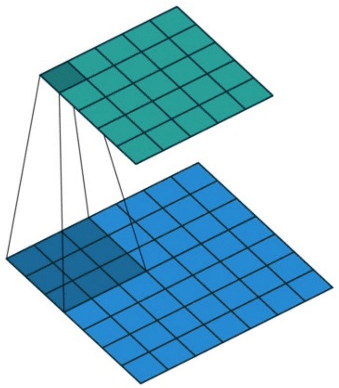
$$K\pi(x) = \mathrm{EGNN}(Lx; w)$$

Graph Neural Networks

Normal convolution



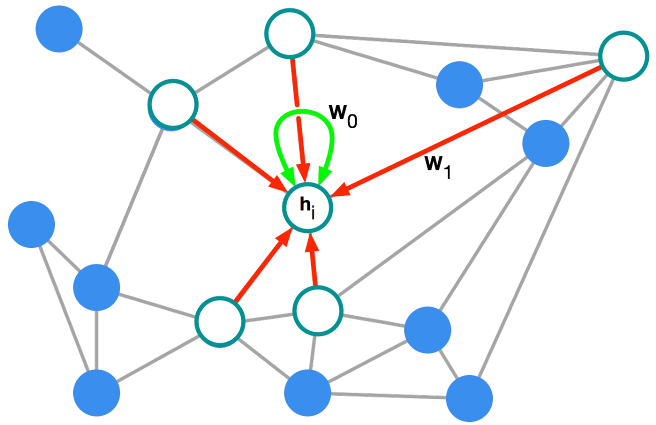
=



(Animation by
Vincent Dumoulin)

➔

Graph convolution



Convolution on a set

	GNN
$v \to e$	$m^t_{i \to j} = \phi_e(h^t_i, h^t_j, a_{ij})$
$e \to v$	$m^t_j = \sum_{i \in \mathcal{N}(j)} m^t_{i \to j}$
	$h^{t+1}_j = \phi_v([m^t_j, a_j], h^t_j)$

E(n) Equivariant Graph Neural Network

E(n) Equivariant Graph Neural Networks

Victor Garcia Satorras [1] **Emiel Hoogeboom** [1] **Max Welling** [1]

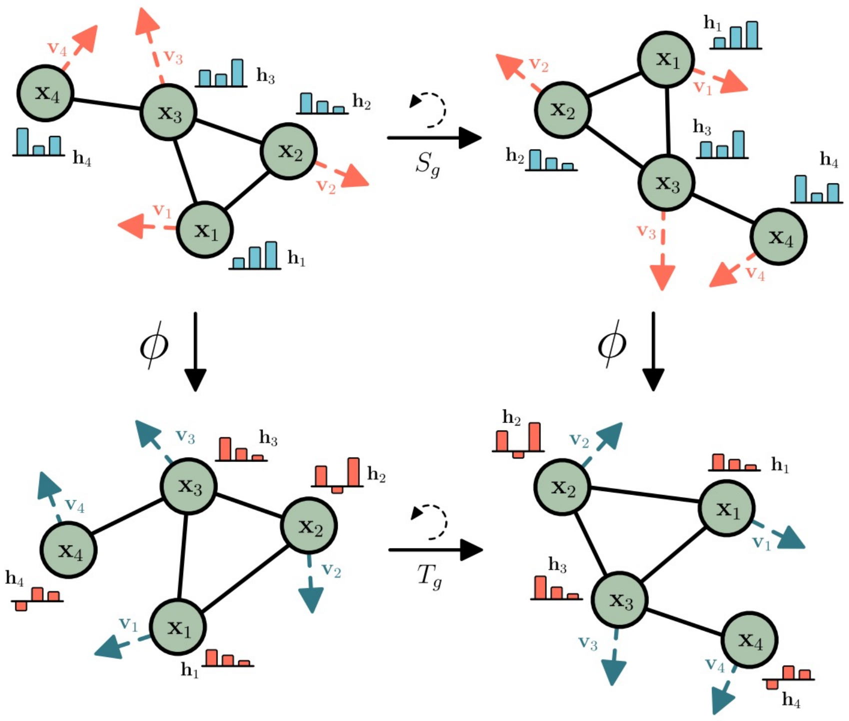


Figure 1. Example of rotation equivariance on a graph with a graph neural network ϕ

State: $\{h_i^t, x_i^t, v_i^t\}$

Features: $\{a_i, a_{ij}\}$

Invariants: $\{h_i, ||x_i - x_j||\}$

Equivariant Updates:

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right)$$

$$\mathbf{v}_i^{l+1} = \phi_v\left(\mathbf{h}_i^l\right)\mathbf{v}_i^l + \sum_{j \neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right)$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \mathbf{v}_i^{l+1}$$

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right)$$

SE(3) Equivariant GNNs

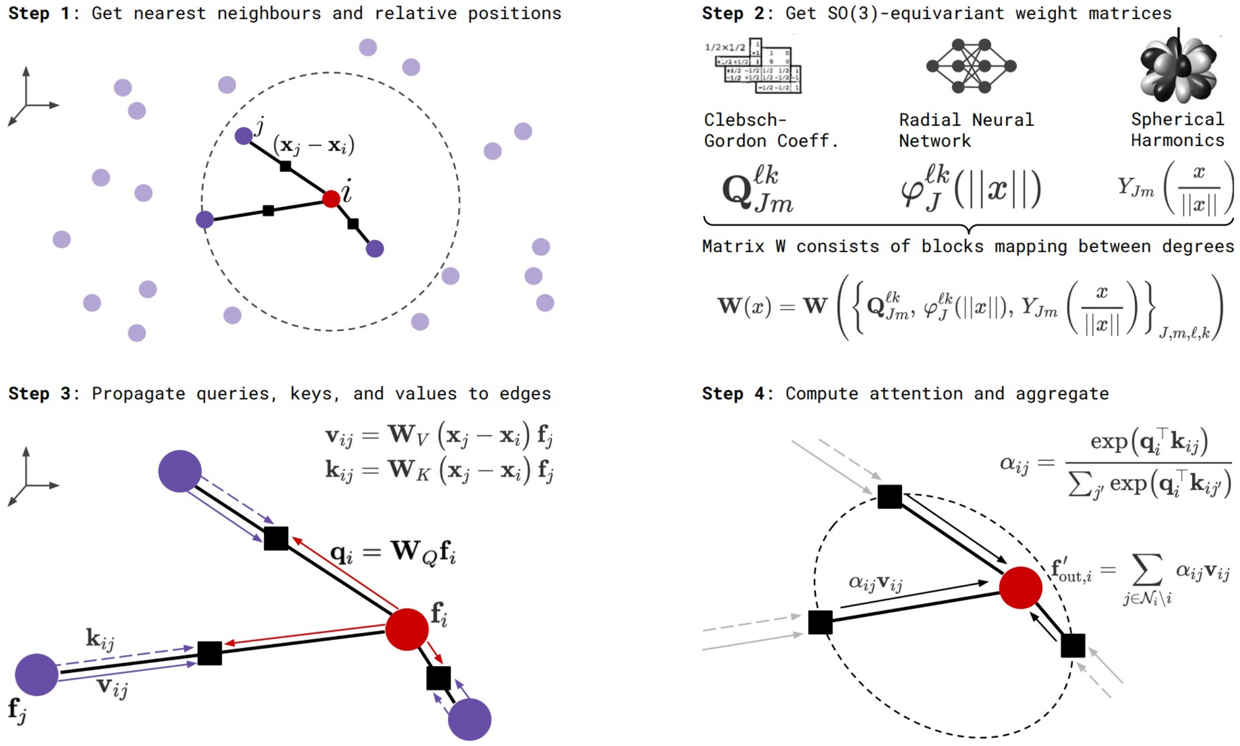
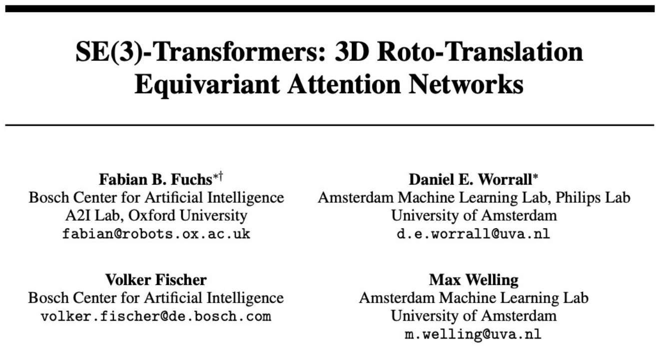
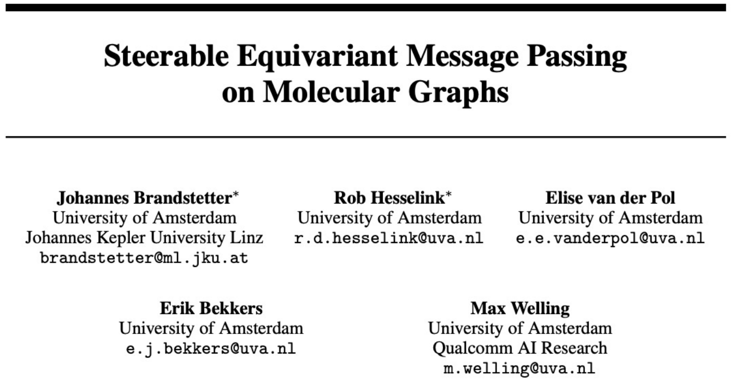


Figure 2: Updating the node features using our equivariant attention mechanism in four steps. A more detailed description, especially of step 2, is provided in the Appendix. Steps 3 and 4 visualise a graph network perspective: features are passed from nodes to edges to compute keys, queries and values, which depend both on features and relative positions in a rotation-equivariant manner.



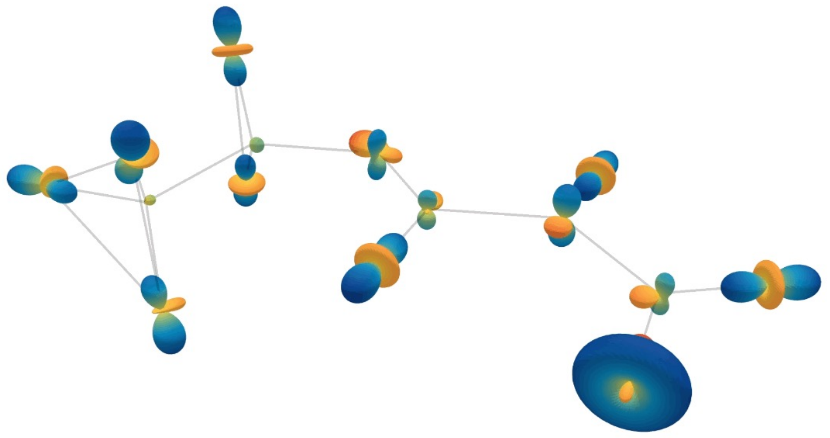
SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks

Fabian B. Fuchs*† — Bosch Center for Artificial Intelligence, A2I Lab, Oxford University, fabian@robots.ox.ac.uk
Daniel E. Worrall* — Amsterdam Machine Learning Lab, Philips Lab, University of Amsterdam, d.e.worrall@uva.nl
Volker Fischer — Bosch Center for Artificial Intelligence, volker.fischer@de.bosch.com
Max Welling — Amsterdam Machine Learning Lab, University of Amsterdam, m.welling@uva.nl



Steerable Equivariant Message Passing on Molecular Graphs

Johannes Brandstetter* — University of Amsterdam, Johannes Kepler University Linz, brandstetter@ml.jku.at
Rob Hesselink* — University of Amsterdam, r.d.hesselink@uva.nl
Elise van der Pol — University of Amsterdam, e.e.vanderpol@uva.nl
Erik Bekkers — University of Amsterdam, e.j.bekkers@uva.nl
Max Welling — University of Amsterdam, Qualcomm AI Research, m.welling@uva.nl



Conclusions

- By exploiting symmetries, the agent needs fewer experiences / interactions with the world / collect less data to perform well.
- There is nice relation between MDP homomorphisms and equivariance: the orbit of a group transformation is mapped to a single point in the abstract space and the policy transforms properly on these orbits.
- First papers to apply equivariance to action spaces.
- We have studied a global symmetry in for local distributed agents: Compute Locally, Coordinate Globally.
- Relevance to workshop: RL is often used to “learn to optimize”. With these tools, you can exploit symmetries. E.g. TSP