Extended Abstract Track

# Representing Repeated Structure in Reinforcement Learning Using Symmetric Motifs

**Editors:** List of editors' names

## Abstract

Transition structures in reinforcement learning can contain repeated motifs and redundancies. In this preliminary work, we suggest using the geometric decomposition of the adjacency matrix to form a mapping into an abstract state space. Using the successor representation (SR) framework, we decouple symmetries in the transition structure from the reward structure, and form a natural structural hierarchy by using separate SRs for the global and local structures of a given task. We demonstrate our method can achieve high policy evaluation accuracy while using representations that are significantly compressed.

**Keywords:** symmetry, reinforcement learning, successor representation

## 1. Introduction

Knowledge of a task's structure can enable more efficient policy learning in reinforcement learning (RL) (1). For example, policies can be constructed that are temporally hierarchical (2) or an environment's state space can be reduced by recognising its symmetries (3). Unfortunately, discovering an environment's latent structure is difficult. Existing methods usually find reduced state-action spaces by learning a global mapping between the environment's Markov decision process (MDP) and a more abstract MDP (4; 5). In this paper we use a different strategy—we use results from graph theory and network analysis (6; 7) to isolate repeated structural motifs in an MDP, allowing for the compression of its transition function into a more compact form. Additionally, our approach is formulated using the Successor Representation (SR) (8), meaning structural regularity can be decoupled from the reward function. Empirically, we demonstrate effective compression of SRs formed over small world graphs (9; 10), opening up future avenues of work using hierarchical reinforcement learning approaches within the SR.

## 2. Related Work

To operate in large state spaces, knowledge of the structural regularities of an environment is essential. Traditionally structural knowledge is obtained implicitly by learning an intermediate representation useful for value judgements—or explicitly by learning a compressed latent model of the environment (11). Other approaches encode inductive biases into policies about structural regularities (1). For example, (3) build knowledge of global symmetries into policy networks, while (4) learn the equivariances inherent in the dynamics of a given MDP. The SR is a biologically plausible middle ground between model-based and model-free reinforcement learning that contains information about an environment's structure—decoupled into its reward function and expected transition dynamics (8; 12). However, as far as we are aware, we are first to build an abstraction on top of the SR to compress it into a more compact form, which we describe next.

© 2022 .

## 3. Geometric Decomposition

For a given MDP, the transition function $T(s, a, s')$ implies an adjacency graph $G = (\mathcal{S}, E)$ where $\mathcal{S}$ is the set of states in the MDP, and $E$ is the set of edges: pairs of states $(s, s')$ that are connected under some action. Only deterministic MDPs are considered. Structural regularities in this graph can be described using the notion of *graph automorphisms*. An automorphism is a permutation of nodes that preserves the adjacency structure of said nodes. The set of nodes being permuted is known as the orbit. Within the overall adjacency graph of the MDP, there may exist smaller subgraphs that are symmetric and therefore can be described by some automorphism that permutes their nodes within the bounds of the local symmetry. These automorphisms are often repeated across the structure of a graph—they are hence referred to as motifs (6). From a given automorphism group, a graph can be represented as the union of the vertices not belonging to any symmetric subgraph, $\mathcal{S}_0$, and the the vertices of each motif *i.e.* $\mathcal{S} = S_0 \cup S_{M_1} \cup S_{M_2} \ldots \cup S_{M_m}$ where $S_{M_i}$ is the subgraph of vertices that are symmetric under a specific automorphism. This decomposition is the geometric decomposition of a graph (6). The automorphism group for a given graph is not unique; we follow (7) and generate subgraphs that are non-overlapping, and therefore automorphisms that are support-disjoint. We use `saucy` (13) to generate our automorphism groups, and in-line with (7) we restrict the motifs found to two orbits in size.

## 4. Structural Hierarchies with Local Symmetry and Successor Representations

Once an automorphism group is generated and the geometric decomposition of the graph is found, we generate a reduced graph with adjacency matrix $\hat{A}$ by representing every occurrence of a symmetric subgraph with a single node, whilst also keeping an index of which node represents which subgraph and the local transition structure of each symmetric subgraph $A_i$. For the purposes of this work, we will only consider the task of policy evaluation under a random walk policy. If one were to sample trajectories from this graph following a policy, or compute a policy through sample-based policy improvement, the graph-subgraph structure would need to be represented hierarchically using the options framework (2), where nodes representing a symmetric subgraph would also lie in the initiating set of an option that would allow the agent to execute a policy specified for the subgraph.

A random walk SR is generated for the reduced graph $\hat{A}$, and for each symmetric subgraph $A_i$. Let $\mathcal{S}_{A_i}$ represent the state space of subgraph $A_i$. We generate expected random walk state transition dynamics using the analytic form of the SR: $\psi^\pi = (\mathbb{I} - \gamma \mathbf{A})^{-1}$, where $\pi$ is the policy, $\psi$ is the SR and $\gamma$ is the discount factor. A policy over the MDP is defined for $m$ subgraphs as $\pi(a|s) = \text{argmax}_a \left(\psi^\pi(R(s))\right.$— $\psi$ is either the SR of the reduced graph, or a subgraph depending on the state being evaluated. $a$ represents actions and $s$ represents graph states. When evaluating states in $\mathcal{S}_{\hat{A}}$, $R(s)$ for states representing compressed subgraphs is the mean of the rewards contained in the subgraph. There is an inherent bias in estimating the value function in this way, as the state occupancies obtained for $\mathcal{S}_{\hat{A}}$ will not account for occupancies accumulated inside the subgraphs. Future work will study deriving a bound for this value approximation error, but empirically this is not a significant malus.

Extended Abstract Track

### 4.1. Results

To test this approach, 100 Barabási-Albert (9) and 100 Connected Watts-Strogatz (10) graphs were generated, each with 100 nodes. The Barabási-Albert graphs were generated with $m = 1$ edges preferentially reattached, and the Watts-Strogatz graphs were generated with nodes initially attached to $k = 2$ neighbours and rerouted with $p = 0.33$. The Watts-Strogatz graphs were more densely connected than the Barabási-Albert graphs. We used `networkx` (14) to generate these graphs. To convert these to MDPs, an action space was chosen arbitrarily with size equal to the degree of the most connected node in a graph. For each graph, the value function under a random walk policy was found for 100 reward functions, where the reward function $r(s, a, s') := r(s') := \mathcal{N}(0, 1)$; that is, 100 reward functions for each of the 200 graphs.

The exact value functions can be recovered within symmetric motifs using their random walk SRs and the rewards associated with each state. In the reduced graph, there will be a discrepancy between the value functions as the state occupancy for states in the core will not reflect transitions made inside motifs. As such the difference between the optimal greedy policies obtained by the reduced random walk SR combined with a given reward vector, and the true optimal policies found using the SR defined over the original state space, was quantified as the total variation $TV := \frac{1}{|\mathcal{S}'|} \sum_{s \in \mathcal{S}'} \frac{1}{2} ||\pi(s) - \pi^*(s)||_1$. The TV gives an indication of, on average, the proportion of actions taken under the inferred policy that deviate from the optimal policy. The numbers quoted in Table 1 correspond to a deviation of around 4% and 0.1% for the Barabási-Albert and Watts-Strogatz graphs respectively. The mean squared error between the derived value and optimal value is also shown—in practice this is dominated by a small number of graphs per node that are massively overestimated. The compression is shown as a percentage size of the resulting SRs compared to an SR defined over the original graph. For a graph with $n$ states, the number of elements required for storage is equal to $(n - (\sum_i |S_{A_i}| - 1))^2 + (\sum_i |S_{A_i}| + 1)^2$. An additional 1 is added to the second term to allow for a virtual state to be represented in the SRs of the subgraphs that allows for an agent to leave the subgraph and return to the main graph. This is an upper bound and assumes that every subgraph is generated according to a unique motif.

Table 1: Summary statistics of evaluation of the random walk policy over 100 graphs with 100 rewards functions per graph

| **Graph Generator** | **TV**$(\pi, \pi^*)$ | $\|\|\boldsymbol{V}, \boldsymbol{V}^*\|\|_2^2$ | **Compression** |
|---|---|---|---|
| Barabási-Albert | 0.0231±0.0074 | 19.4±0.8 | 33.6%± 7.7% |
| Watts-Strogatz | 0.0015±0.0001 | 22.1±1.2 | 94.3%± 3.5% |

## 5. Future Work

An additional improvement that can be made is the use of the internal connectivities to adjust the SR of the $\hat{A}$ to reduced the value approximation error. (7) use this information

in the form of their *Basic Symmetric Motifs* to adjust pairwise network measures over compressed graphs. The largest constraint is the need to know the full adjacency matrix—this is not tractable in reinforcement learning with high dimensional observations. Instead, we are interested in using the knowledge of symmetric graphs to allow agents to form expectations of local structures given their prior experience.

## References


[1] Balaraman Ravindran and Andrew G Barto. Symmetries and model minimization in markov decision processes, 2001.

[2] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in rl. *Artificial intelligence*, 1999.

[3] Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in rl. *NeurIPS*, 2020.

[4] Elise van der Pol, Thomas Kipf, Frans A Oliehoek, and Max Welling. Plannable approximations to mdp homomorphisms. *arXiv:2002.11963*, 2020.

[5] Augustine N Mavor-Parker, Andrea Banino, Lewis D Griffin, and Caswell Barry. A simple approach for state-action abstraction using a learned mdp homomorphism. *arXiv:2209.06356*, 2022.

[6] Ben D. MacArthur, Rubén J. Sánchez-García, and James W. Anderson. Symmetry in complex networks. *Discrete Applied Mathematics*, 156(18):3525–3531, 2008.

[7] Rubén J Sánchez-García. Exploiting symmetry in network analysis. *Communications Physics*, 3(1):1–15, 2020.

[8] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 1999.

[10] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *Nature*, 1998.

[11] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv:1912.01603*, 2019.

[12] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

[13] Paul T Darga, Hadi Katebi, Mark Liffiton, Igor L Markov, and Karem Skallah. Saucy3: Fast symmetry discovery in graphs. 2012.

[14] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In *7th Python in Science Conference*, 2008.