Hessisches Kompetenzzentrum
für Hochleistungsrechnen

# *Learning for Decentralized, Long-Horizon Manipulation*



Project Manager
Niklas Funk

Researchers
Sebastian Szelag

Principal Investigator
Prof. Jan Peters (PhD)

Project Term
2021 - 2022

Clusters
Lichtenberg II Cluster Darmstadt

Software
PyTorch

Additional Software
PyBullet

Institute
Intelligent Autonomous Systems

University
Technische Universität Darmstadt

## Introduction

Achieving long-horizon dexterous manipulation remains a challenging problem in robotics. There exists a long history of approaching long-horizon decision-making through task and motion planning methods which, however, typically are computationally very expensive. Thus, recently, there has been an increased interest in end-end learning approaches that benefit from flexible graph-based representations and have shown to be very effective. All of these works formulate long-horizon decision-making from the perspective of having one centralized agent. In this project, we, however, want to investigate a decentralized formulation for long-horizon manipulation, i.e., assembly tasks, to overcome the prohibitive combinatorial complexity of the hierarchical approaches. Due to the sample complexity of current reinforcement learning algorithms, it is inevitable to address this research topic from the direction of creating simulation environments and training the policies on the HPC. Also, as we have shown in previous work, transferring the trained policies to real robotic systems should be feasible.

## Methods

This work leverages three main components. In terms of representation, the project investigates graph-based neural networks that allow generalization across different numbers of building blocks for building the desired structures. In particular, every building block is an individual agent that is represented by a node in the graph. By making use of the graph attention mechanism, the graph is first encoded in multiple iterations, i.e.,

information is passed in between the different agent nodes, before in a final step, we decode the obtained graph to get the actions for every individual agent. For training the agents, and essentially refining the representation to output reasonable agent actions, we make use of reinforcement learning. In particular, this work leverages Soft Actor Critic (SAC). The last component necessary to realize this project was the creation and implementation of an assembly environment in Pybullet.

## Results

This project achieved a proof of concept for simple shapes consisting of less than 5 blocks. In these scenarios, the proposed method yielded the expected performance. Moreover, it showed generalization w.r.t. number of blocks and block initializations. Unfortunately, for larger and more complicated structures, the algorithm did not converge to an optimal solution. Even when increasing the training time from 24 to 48 hours, the results still remained sub-optimal. Additionally, because of the difficulties of training policies for assemblies of more than 5 blocks, all experiments have remained in the simplified setting of only optimizing the parts' positions, without considering orientations.

## Discussion

The main idea of the approach was that through decentralization, the combinatorial burden of long-horizon assembly could be overcome. While the results demonstrated a proof of concept for the entire approach, unfortunately, during this project, we could not scale the presented approach to problem sizes where the combinatorial complexity would really be a limiting factor for centralized approaches. The main factor counteracting the seamless scaling of the proposed approach is the action space. With an increasing number of building blocks, the number of agents and the action space grow linearly. The larger action spaces make exploration and credit assignment increasingly challenging. To mitigate this issue, in the future, we want to investigate curriculum learning. By carefully increasing the difficulty of the assembly environments, we hope to achieve more directed exploration and overcome the aforementioned problems.

## *Last Update:* 2024-01-18 15:17