Subassembly to Full Assembly: Effective Assembly Sequence Planning through Graph-based Reinforcement Learning

Chang Shu, Anton Kim, and Shinkyu Park

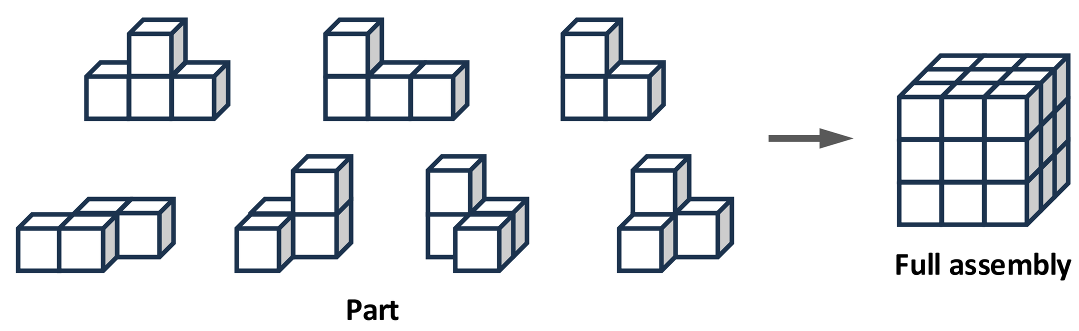
King Abdullah University of Science and Technology (KAUST), Saudi Arabia (chang.shu@kaust.edu.sa)

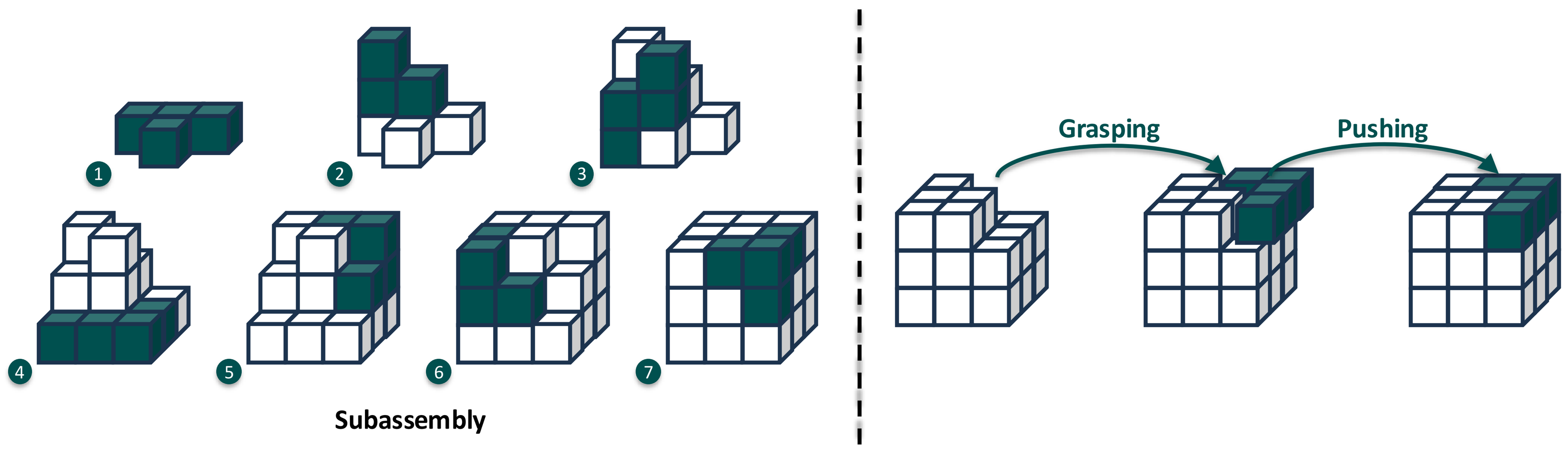
KAUST Robotics, Intelligent Systems, and Control Lab

Task and Motion Planning for Robotic Assembly

Problem Definition

- **Assembly Sequence Planning**
  Which part first
- **Assembly Path Planning**
  How to assemble





Object Manipulation Actions

A Combinatorial Problem

For N-part assembly,

- # subassemblies = 2^N
- # assembly sequences = $N!$

Feasibility Constraints

- Collision-free
- Structural stable

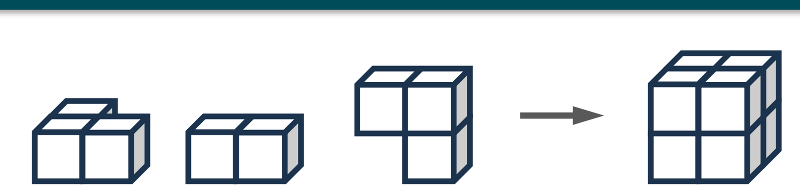
Similar to the Traveling Salesman Problem

ASP	TSP
assemble all the parts	visit all the cities
each part assembled once	each city visited once
start from the non-assembly	start from any city
stop at the full assembly	stop at the starting city

Representing Assembly Sequences on Graphs

Assembly Problem Instance $\mathcal{P}$

- $\mathcal{P} = \{p_1, p_2, \dots, p_M\}$

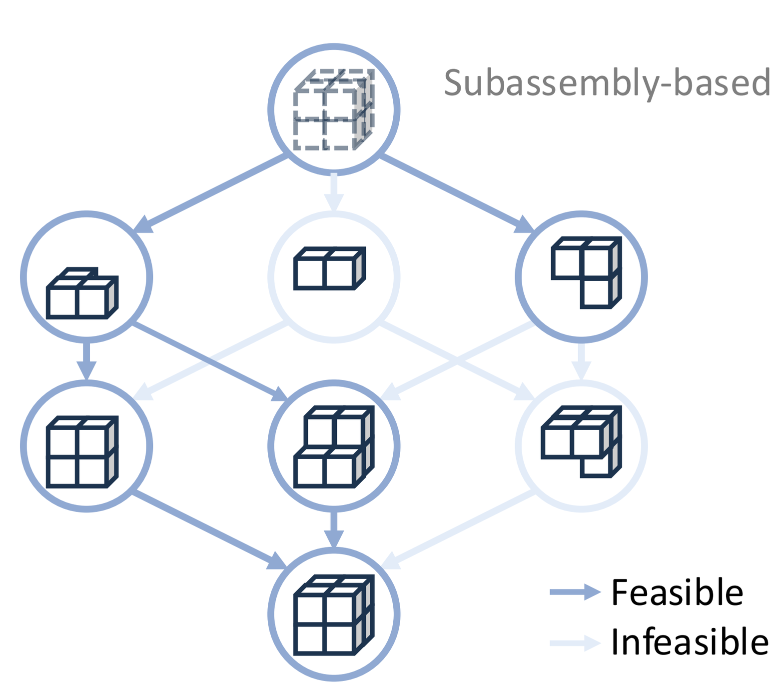


The directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- **Node** $v \subset 2^{\mathcal{P}}$: a subassembly
- **Edge** $(u, v) \in \mathcal{E}$: an assembly step
- $|\mathcal{V}| = N = 2^M$, $|\mathcal{E}| = M \cdot 2^{M-1}$

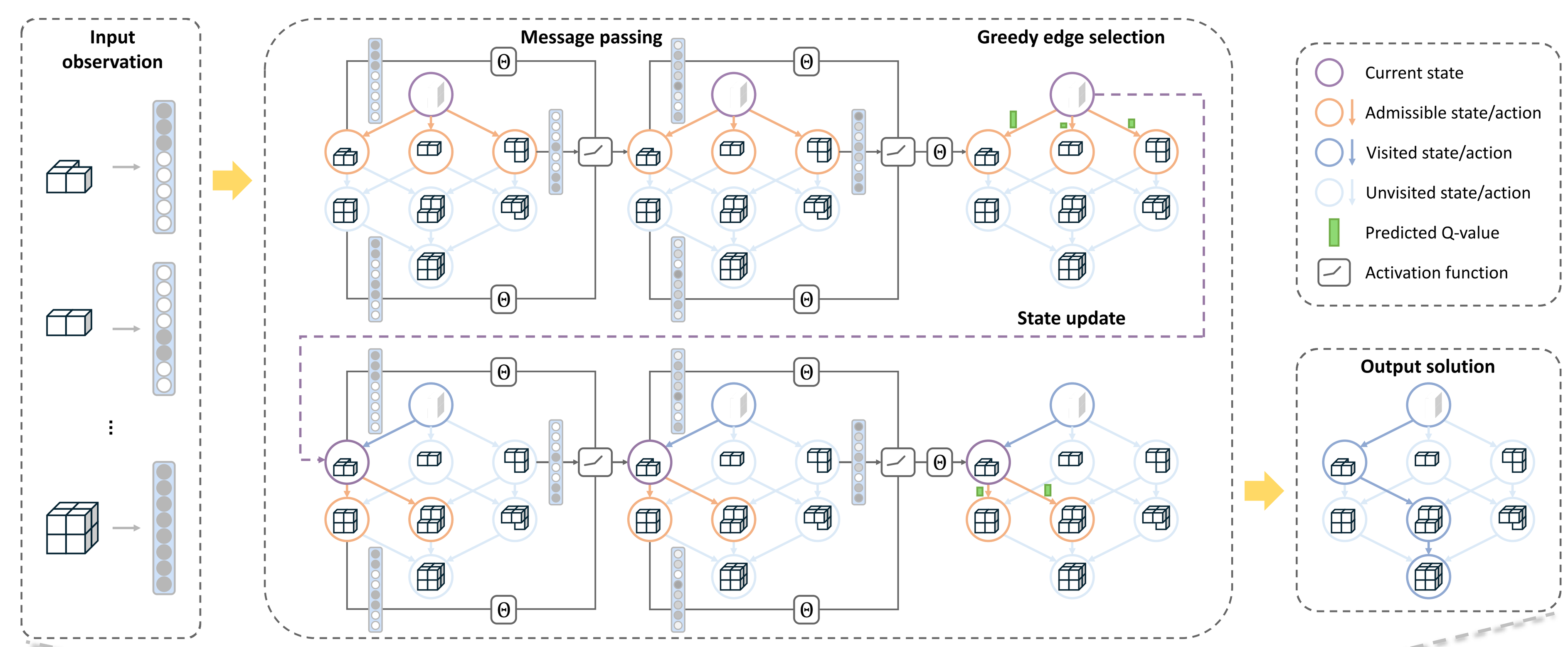
Markov Decision Process

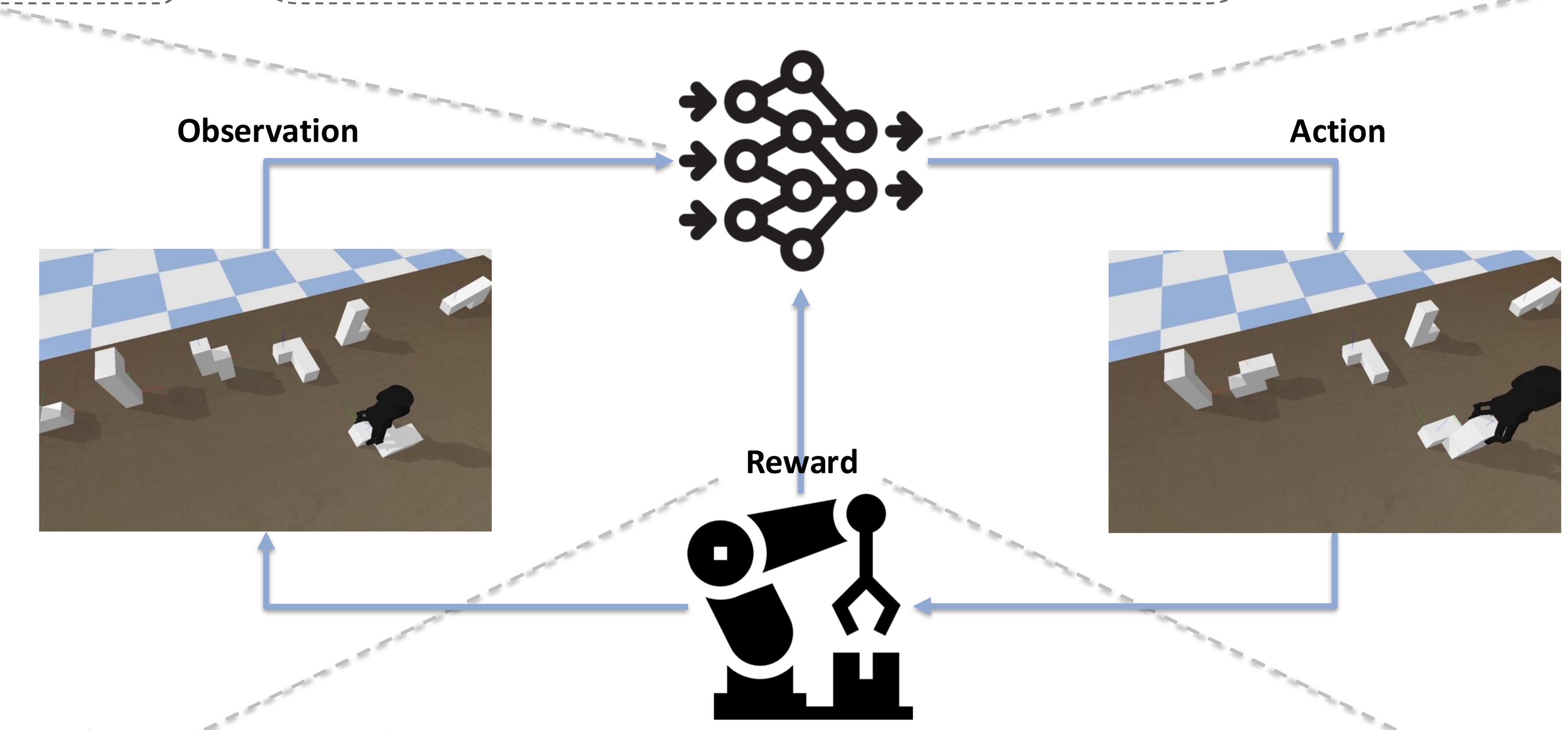
- **State** $s_t = v$: a subassembly
- **Action** $a_t = v \cup \{p\},\ p \in \mathcal{P} \backslash v$: adding a new part to the current state
- **Transition** $P(s_{t+1} | s_t, a_t)$
- **Reward** $r_t = R(s_t, a_t)$: 1 if complete the assembly, -1 if lead to dead end, and 0 otherwise



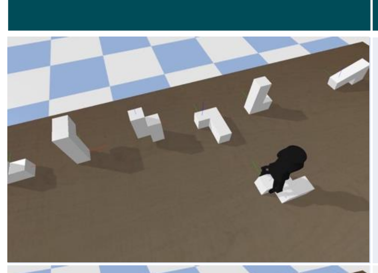
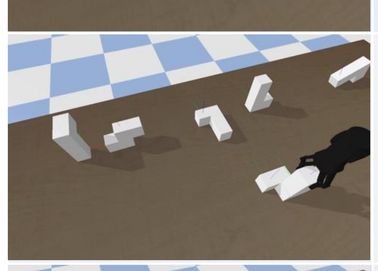
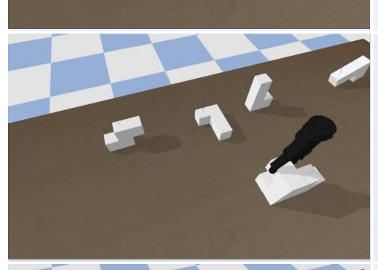
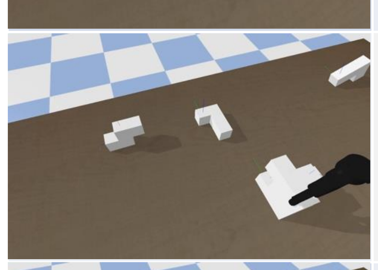
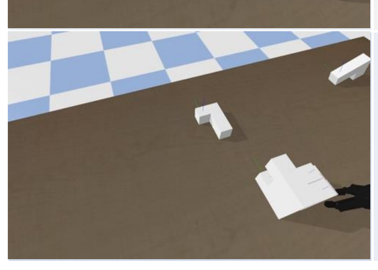
Subassembly to Assembly (S2A) Framework

Framework Overview





Delayed Reward Assignment

Simulation	Current Step	Full Assembly	Reward W/O Delay	Reward W/ Delay
	Feasible	Feasible	0	0
	Feasible	Feasible	0	0
	Feasible	Infeasible Stop Early	0	-1
	Feasible	Infeasible	0	/
	Infeasible Dead End	Infeasible	-1	/

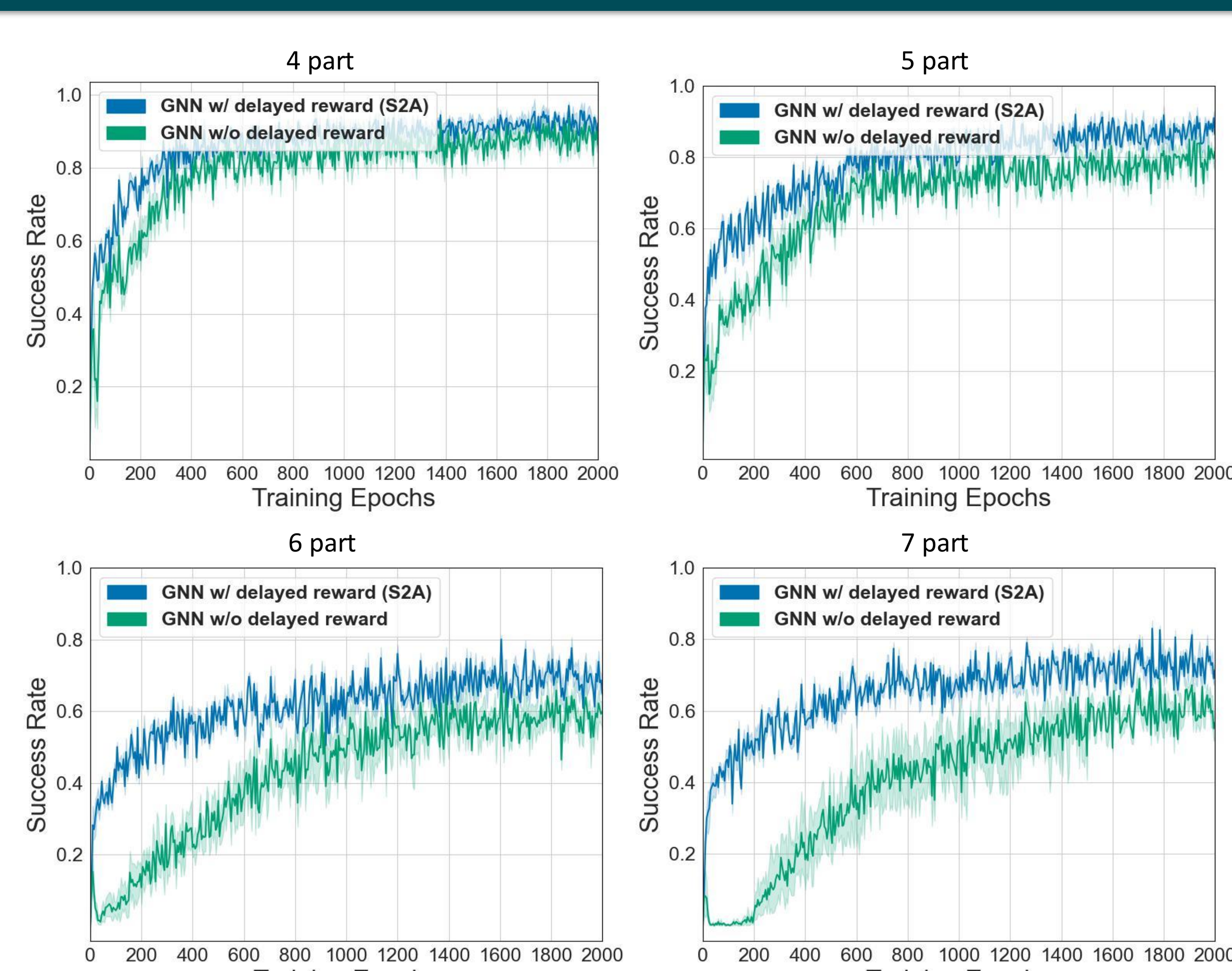
Experimental Results

- Comparison with baseline methods

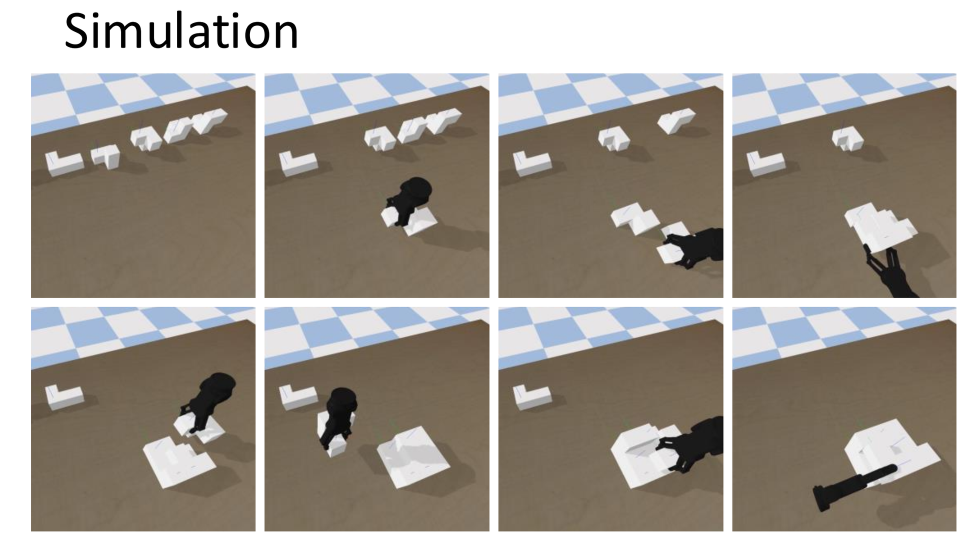
Method		Success Rate (%) M=4	M=5	M=6	M=7
Baseline	RandomWalk	8.15	3.81	1.33	0.62
	Heuristics	45.60	34.87	21.02	16.67
	GNN (w/o delay), BS1	87.72	78.91	59.50	59.72
	GNN (w/o delay), BS3	96.17	89.04	74.04	74.39
	GNN (w/o delay), BS5	96.90	91.38	77.78	76.97
Ours	S2A, BS1	91.50	87.85	70.00	72.43
	S2A, BS3	98.29	93.90	81.29	82.15
	S2A, BS5	**99.09**	**95.13**	**84.01**	**84.71**

- Generalization across problem sizes

Test → Train ↓	Success Rate (%) M=4	M=5	M=6	M=7
M=4	-	80.14	61.98	54.95
M=5	91.38	-	67.61	63.01
M=6	92.41	84.36	-	72.88
M=7	92.93	86.21	73.47	-



- Simulation



- Real world