Thesis Summary and Ongoing work:

Augmenting Behavior Trees with Reinforcement Learning Nodes for Robotic Assembly

Scott Scheraga

Thesis Advisors:

- Brad Hayes (Primary) –CSCI Dept.
- Nikolaus Correll –CSCI Dept.

Thesis Concept

To allow robots to reliably accomplish complex sequences of specific fine motor actions across a variety of assembly and repair tasks,

I propose a system that integrates high-level task management, defined by hardcoded Behavior Trees, along with trained RL nodes for both low-level control flow nodes and execution nodes.

Project inspiration is primarily taken from:

"**Integrating Reinforcement Learning into Behavior Trees by Hierarchical Composition**" Kartasev, Ögren , 2019 KTH Master's Thesis

- Implemented a number of different behavior trees with DQN and PPO nodes to fight monsters and assemble a house in Minecraft. https://www.diva-portal.org/smash/get/diva2:1368535/FULLTEXT01.pdf

"Deep Reinforcement Learning for High Precision Assembly Tasks" Inoue et al. , IROS 2017

- Developed a two -stage policy consisting of a DQN with LSTM layers to perform 10 µm clearance peg-in-hole insertion tasks on a real robot. https://arxiv.org/pdf/1708.04033.pdf

"A Learning Framework for High Precision Industrial Assembly" Fan et al. , ICRA 2019

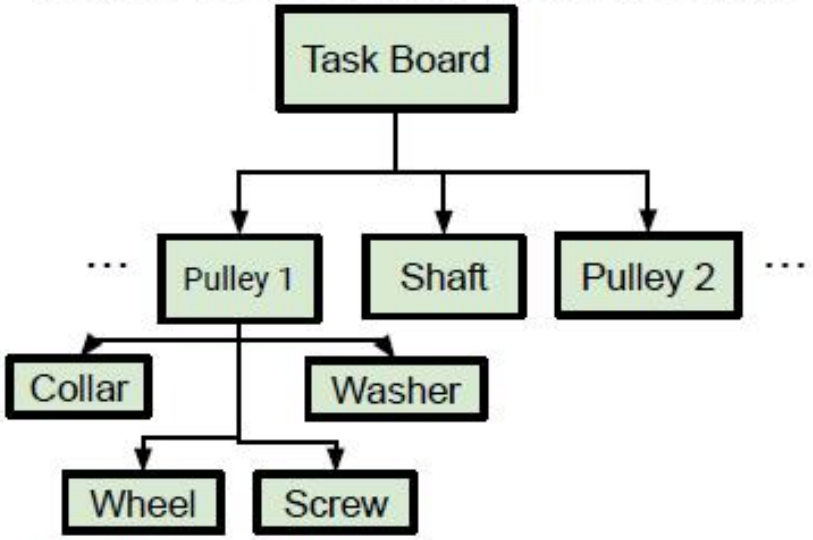
- Developed "Guided DDPG" algorithm that to assemble Lego bricks using a continuous action space. https://arxiv.org/abs/1809.08548

Contents

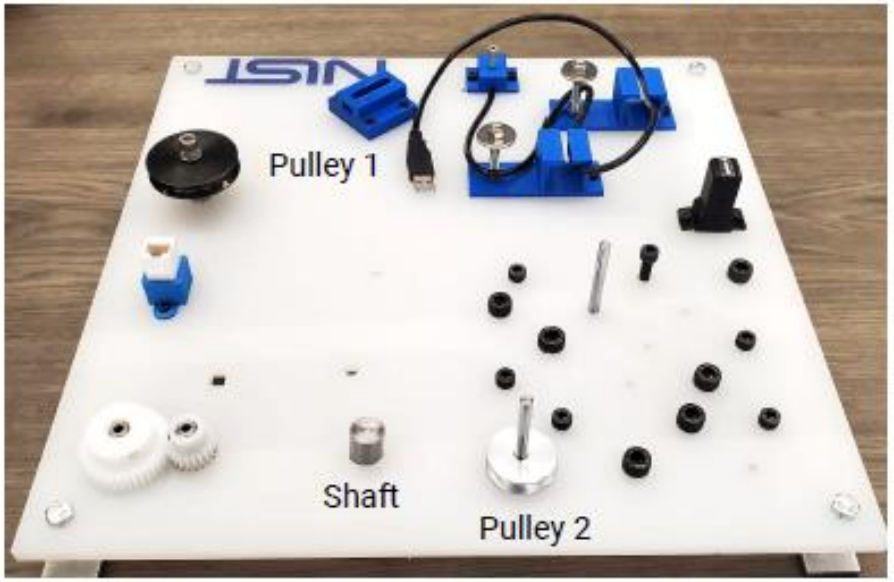
- Prior Correll Lab work
- Thesis Concept
- Summary of "**Integrating Reinforcement Learning into Behavior Trees by Hierarchical Composition"** (Kartasev, Ögren , 2019 KTH Master's Thesis)
- Summary of **Deep Reinforcement Learning for High Precision Assembly Tasks"** Inoue et al. , IROS 2017
- Short summary of **"A Learning Framework for High Precision Industrial Assembly"** Fan et al. , ICRA 2019
- Current progress
- Projected Timeline

Prior Related Correll Lab work

NIST Challenge (IROS 2019 and Beyond): Competitions providing a venue to test and demonstrate industrial automation techniques



Example Assembly Tree

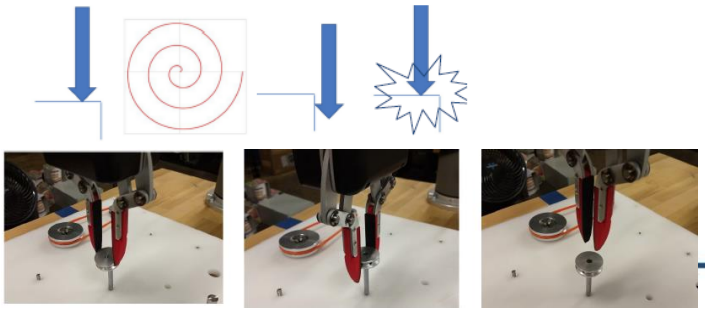


NIST taskboard requires different join operations, as well as parallel, sequential, and hierarchical actions

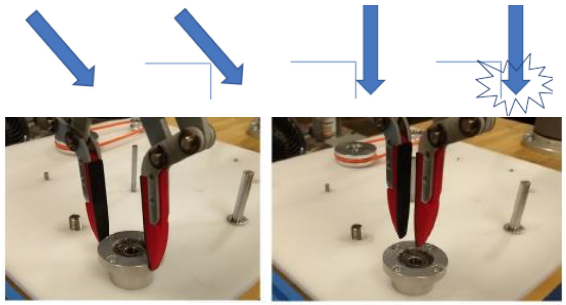
Prior Related Correll Lab work

Behavior tree actions with support vector
machine -classified failures/successes

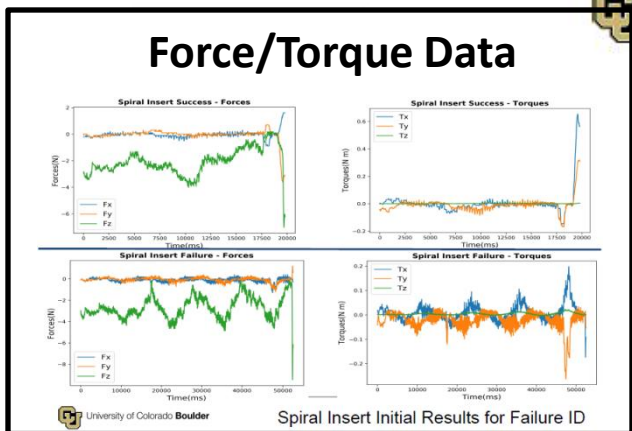
- **Spiral insert** (pulley onto shaft)
  Locate part and shaft, Move to contact,
  Spiral search until condition (Drop Force, Lateral Torque),
  Push, Tamp



- **Tilt Insert** (bearing into hole)
  Locate hole and bearing, Tilt and offset,
  Move to contact, Return to vertical and center part on hole, Tamp

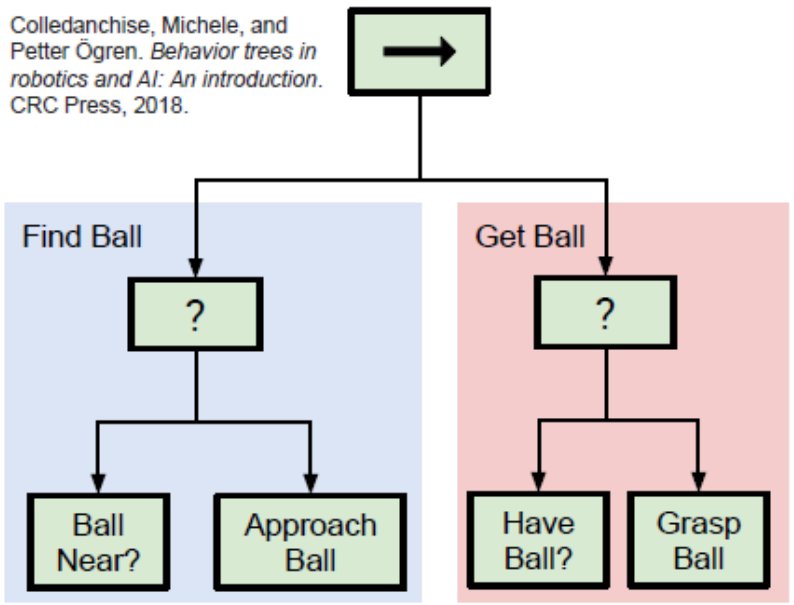


- **Nut threading**
  Locate nut and stud, Move to contact,
  While no TZ resistance:
    Grasp, Rotate CW 2 flats and descend,
    Release, Rotate CCW 2 flats



Advanced Robotics 34, no. 7-8 (2020): 546-559.

Behavior Trees in a Nutshell



- Flow control through distribution of ticks (visits)
- When running, a behavior returns “Running”, “Success” or “Failure”
- Modular.
- Any finite state machine has an equivalent Behavior Tree

Icon	Node type	Success condition	Failure condition	Running condition
→	Sequence	All children succeed	A child fails	A child returns running
?	Selector/Fallback	A child succeeds	All children fail	A child returns running

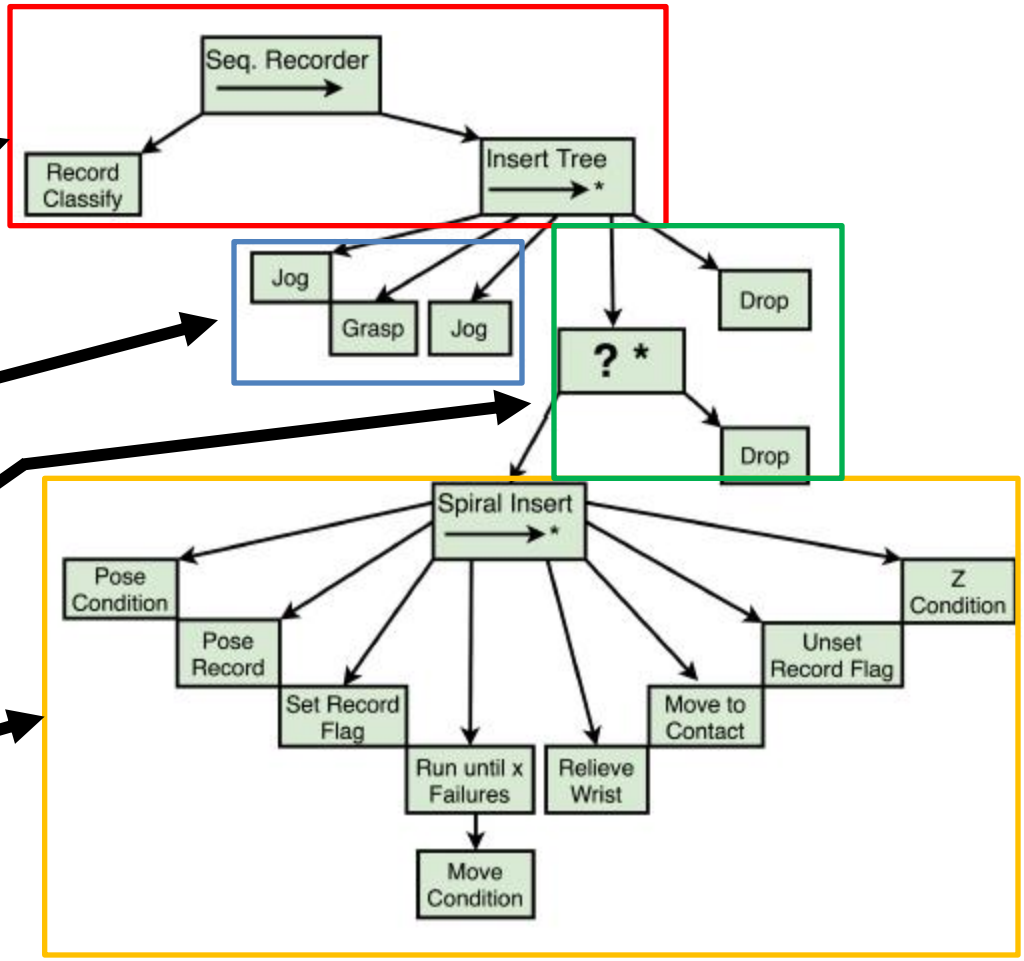
Spiral Insert Behavior Tree

Top Level:
Record and Run Task

Middle Level:
Retrieve Part

Run Spiral Insert Skill unless classifier determines fail state

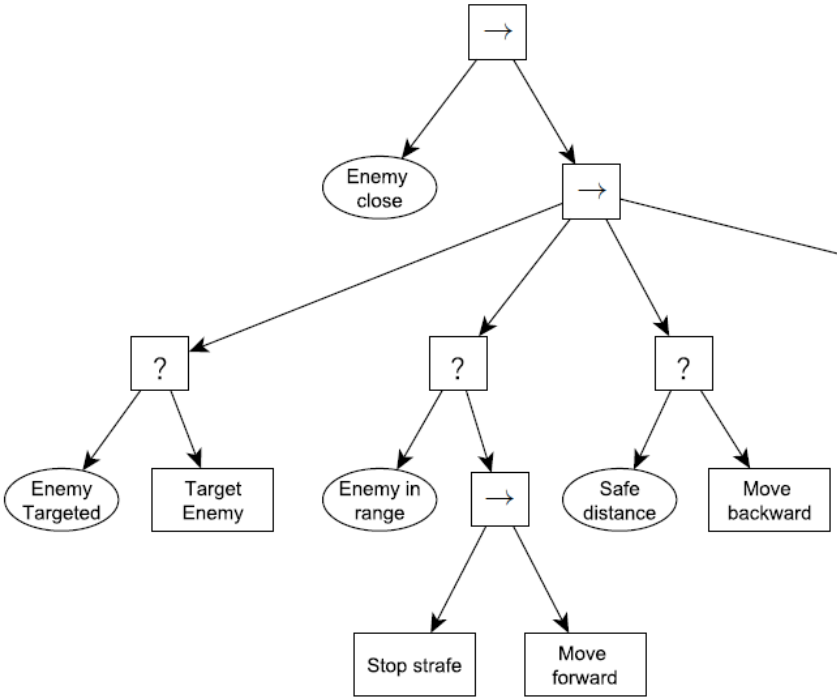
Bottom Level:
Spiral Insert Skill



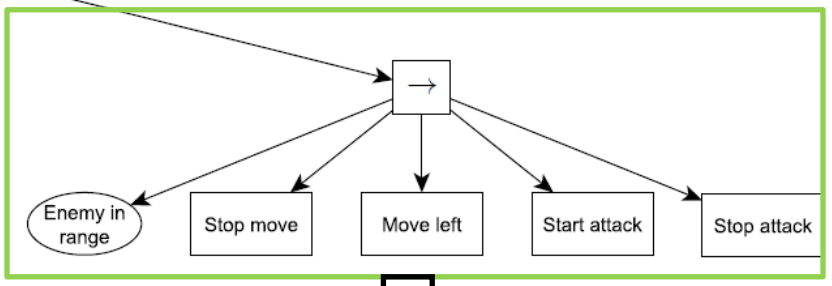
“Integrating Reinforcement Learning into Behavior Trees by Hierarchical Composition”

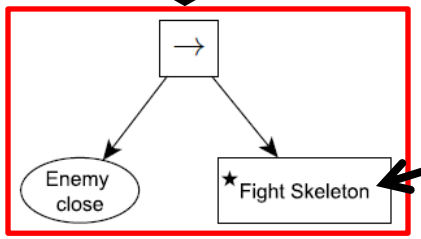
Kartasev, Ögren , 2019 KTH Master’s Thesis

Learning an execution node in a Behavior Tree in Minecraft (Project Malmo) Environment



Previous behavior tree





Learned BT node

Environment with skeleton and no obstacles

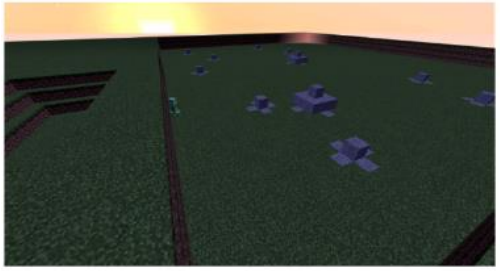
Observation Space and Rewards:

Source	Time step	Damaging skeleton	Skeleton health 0	Taking damage	Agent health 0
Reward	-0.10	5 to 20 *	1000	-5 to -20 *	-1000

Action Space:

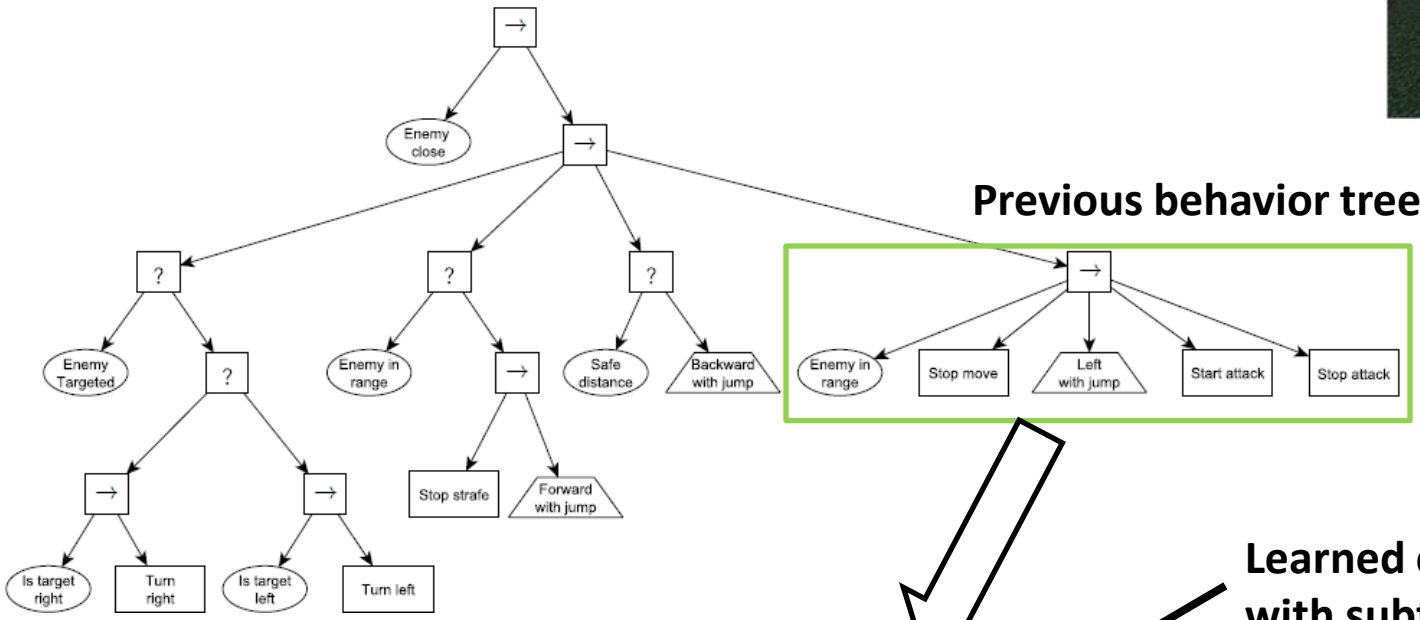
At each timestep, DQN agent chooses one
Or PPO agent simultaneously choose up to one action from each row below:
{Move forward, Stop motion, Move backward}
{Move left, Stop strafe, Move right}
{Start attack, No operation, Stop attack, Target enemy}

Learning a Control Flow Node

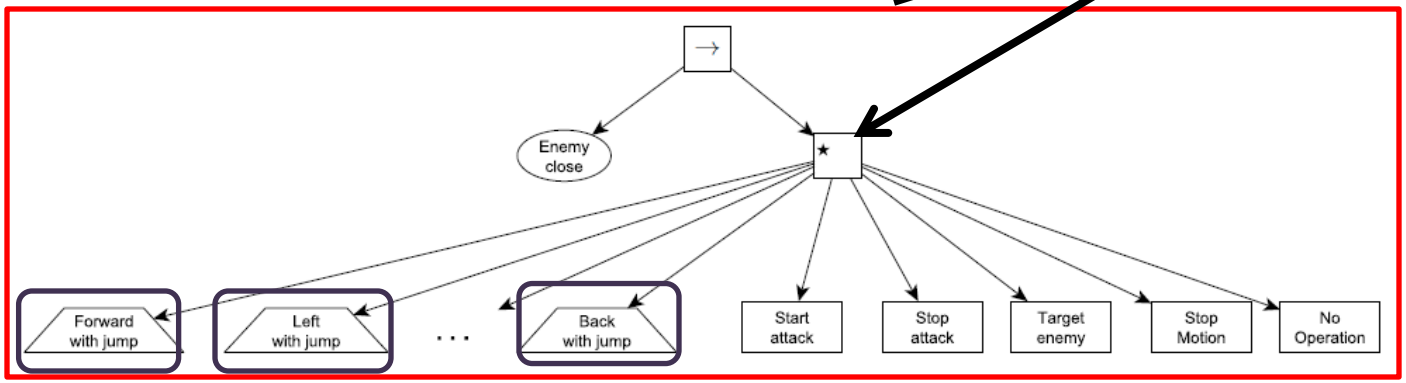


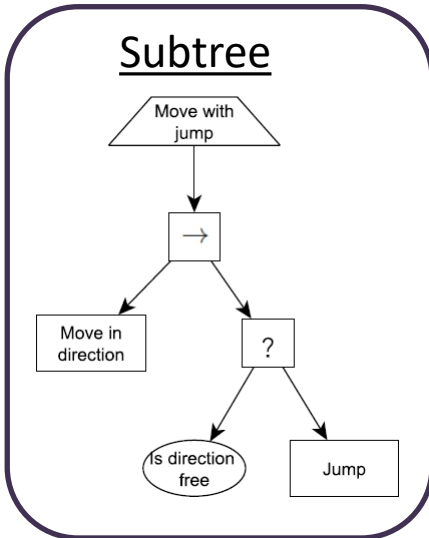
Env. with obstacles

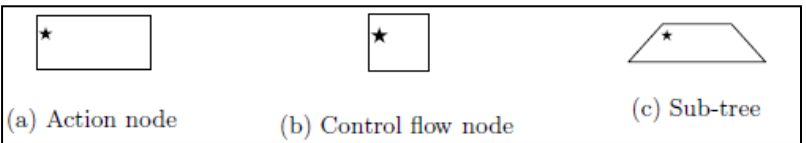
Previous behavior tree

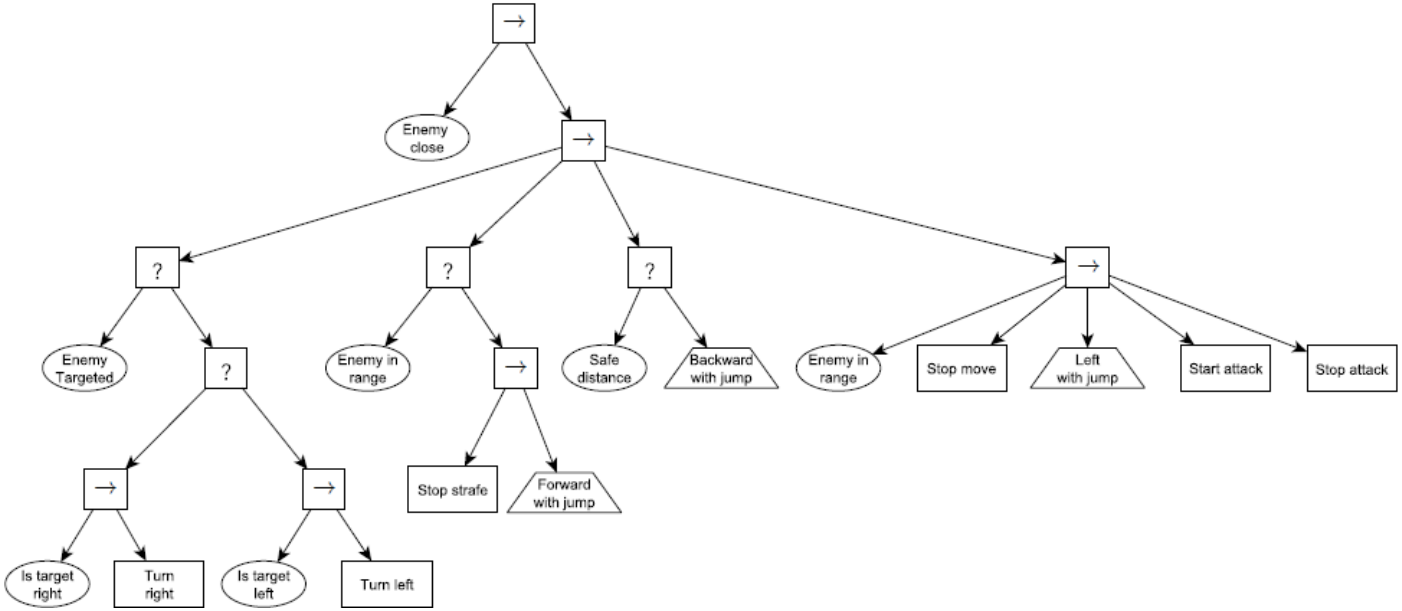


Learned control flow node, with subtree actions



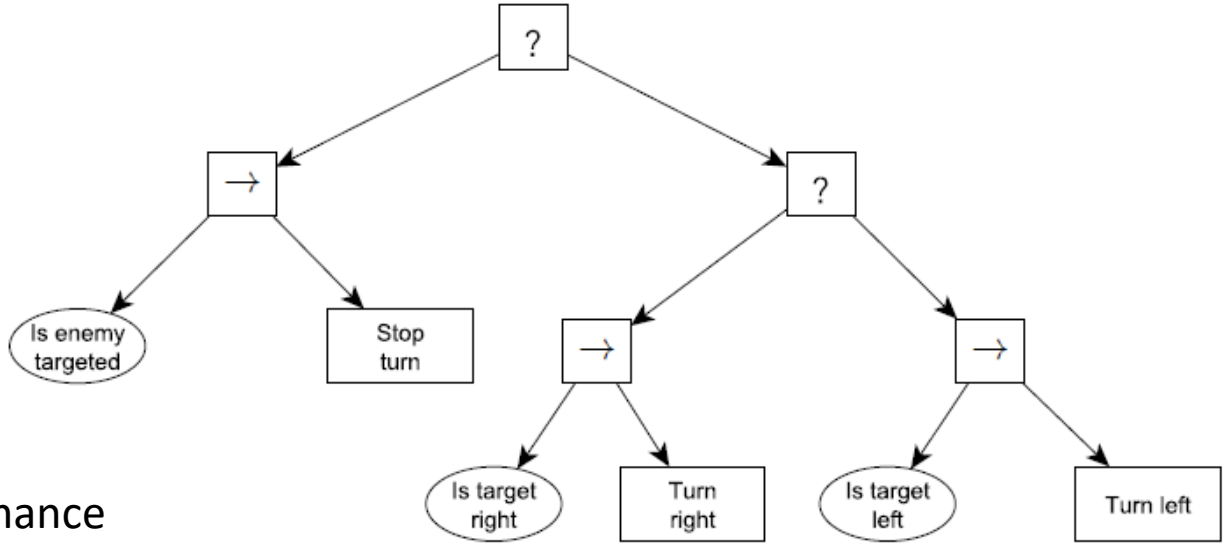





→
Enemy close
→
?
?
?
→
Enemy Targeted
?
Enemy in range
→
Safe distance
Backward with jump
Enemy in range
Stop move
Left with jump
Start attack
Stop attack
→
→
→
Stop strafe
Forward with jump
Is target right
Turn right
Is target left
Turn left

Learning a Control Flow Node

Replacement of “target enemy” node subtree in “+turn” training runs.

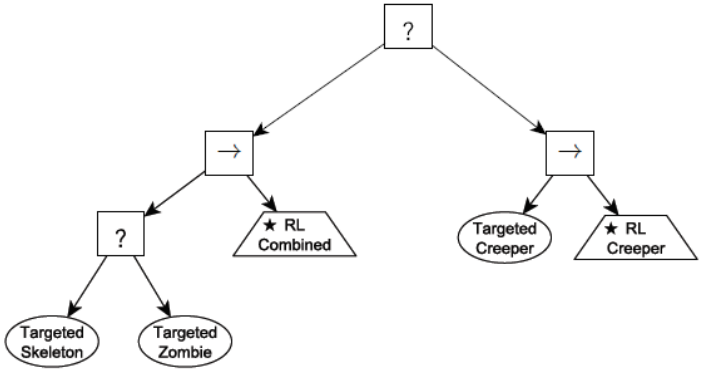


Best Performance

Configuration	Reward			Health remaining			Step count		
	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
DQN	2406.64	-263.92	1912.35	20.0	18.0	19.96	20317.0	454.0	1001.06
DQN+turn	2644.23	1604.04	2010.85	20.0	10.83	19.17	657.0	349.0	458.46
DQN+reuse	2122.25	1428.83	1834.32	20.0	13.0	19.0	1849.0	346.0	500.38
DQN+reuse+turn	2216.95	1458.86	1851.24	20.0	11.0	18.59	683.0	327.0	485.4
Manual	1984.58	1404.64	1630.4	20.0	12.5	19.72	893.0	353.0	620.18

“+reuse” uses a pre-trained model from the previous experiment.

Learning multiple RL policies in a Behavior Tree




(a) Skeleton (b) Creeper (c) Zombie

Figure 30: Tree with 2 types of policies for type specific learning experiment

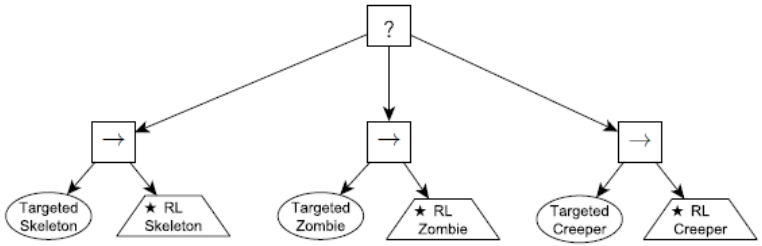
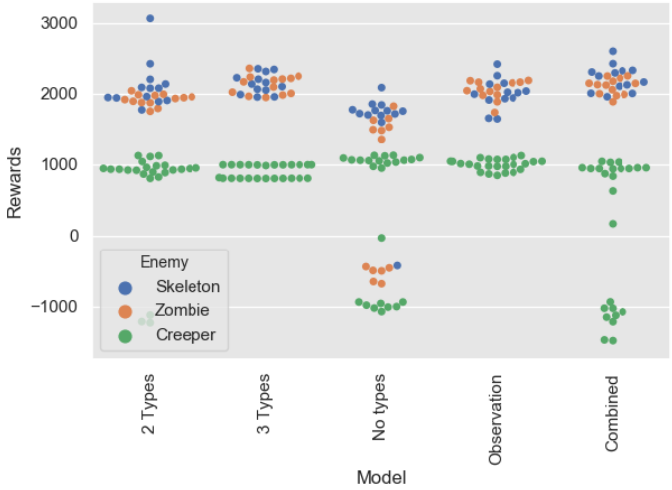


Figure 31: Tree with 3 types of policies for type specific learning experiment



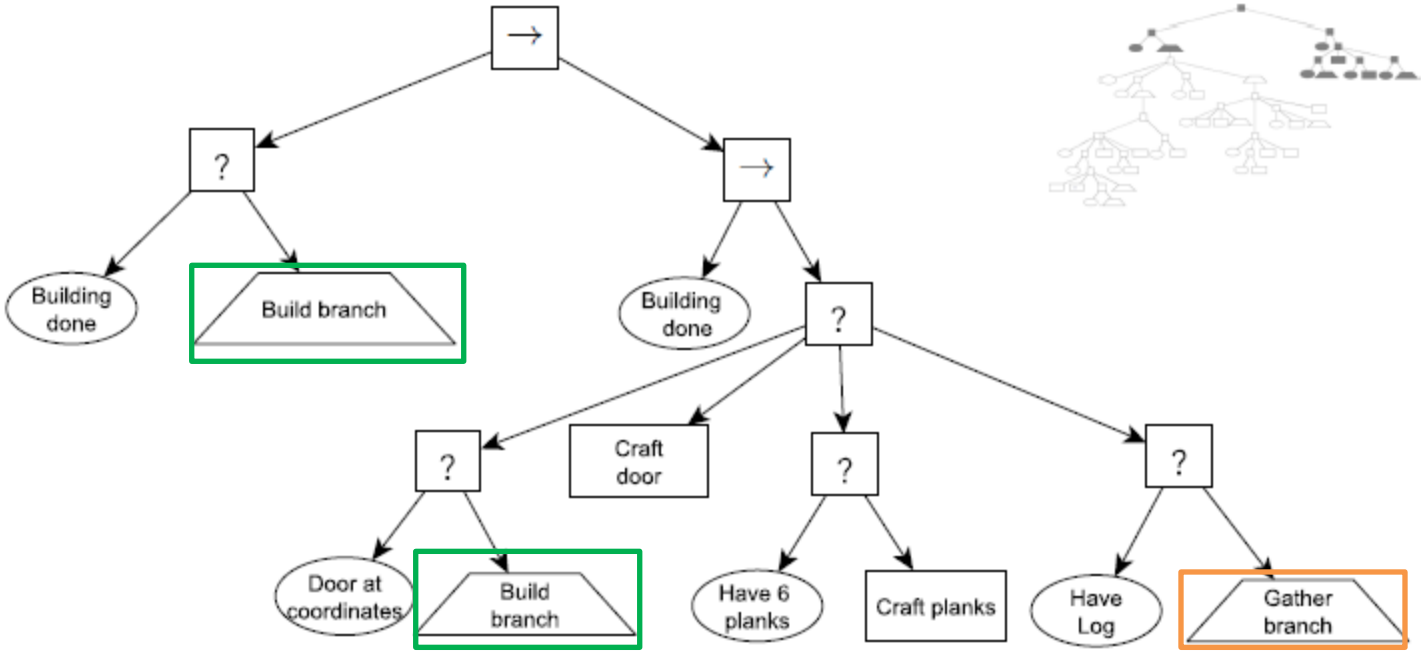
Configuration	Reward			Health remaining			Step count		
	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
2 types	3064.18	-1223.9	**1375.72**	20.0	0.0	**14.06**	623.0	58.0	327.2
3 types	2351.17	801.5	**1539.1**	20.0	1.0	**10.1**	653.0	54.0	276.16
No type info	2083.36	-1068.69	730.34	20.0	0.0	5.1	649.0	50.0	294.12
Observation	2410.44	845.24	**1531.68**	20.0	2.0	**12.37**	658.0	52.0	282.88
Combined	2590.14	-1472.9	**1176.59**	20.0	10.0	**17.38**	**7982.0**	48.0	**878.16**

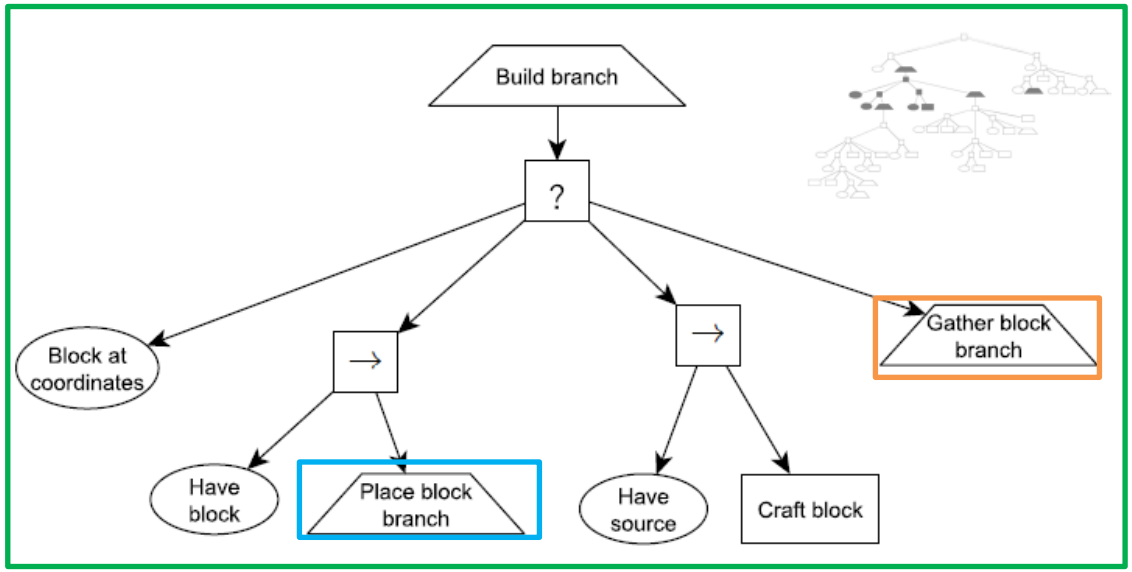
Manual-crafted BT for Building a House

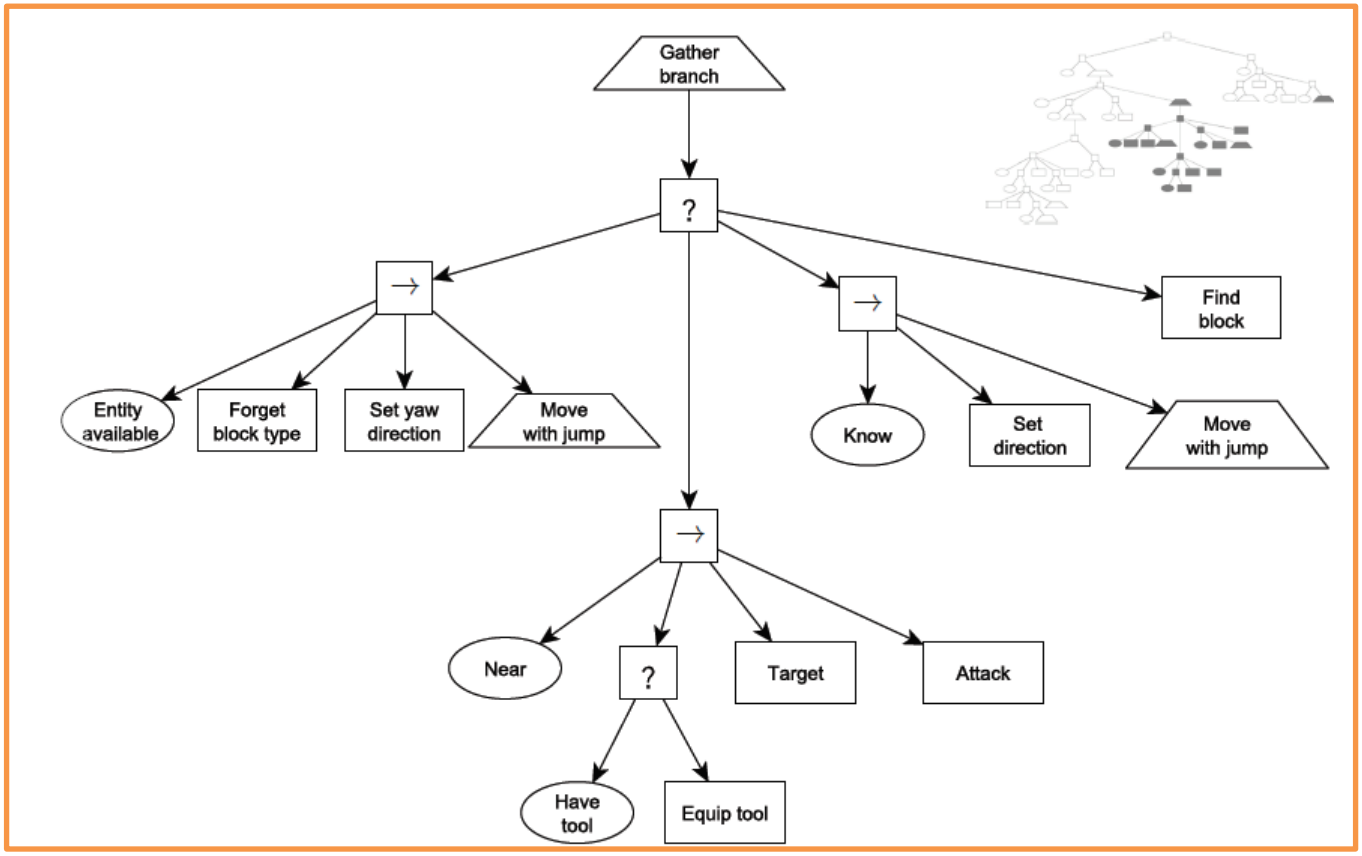
(a) Foundation

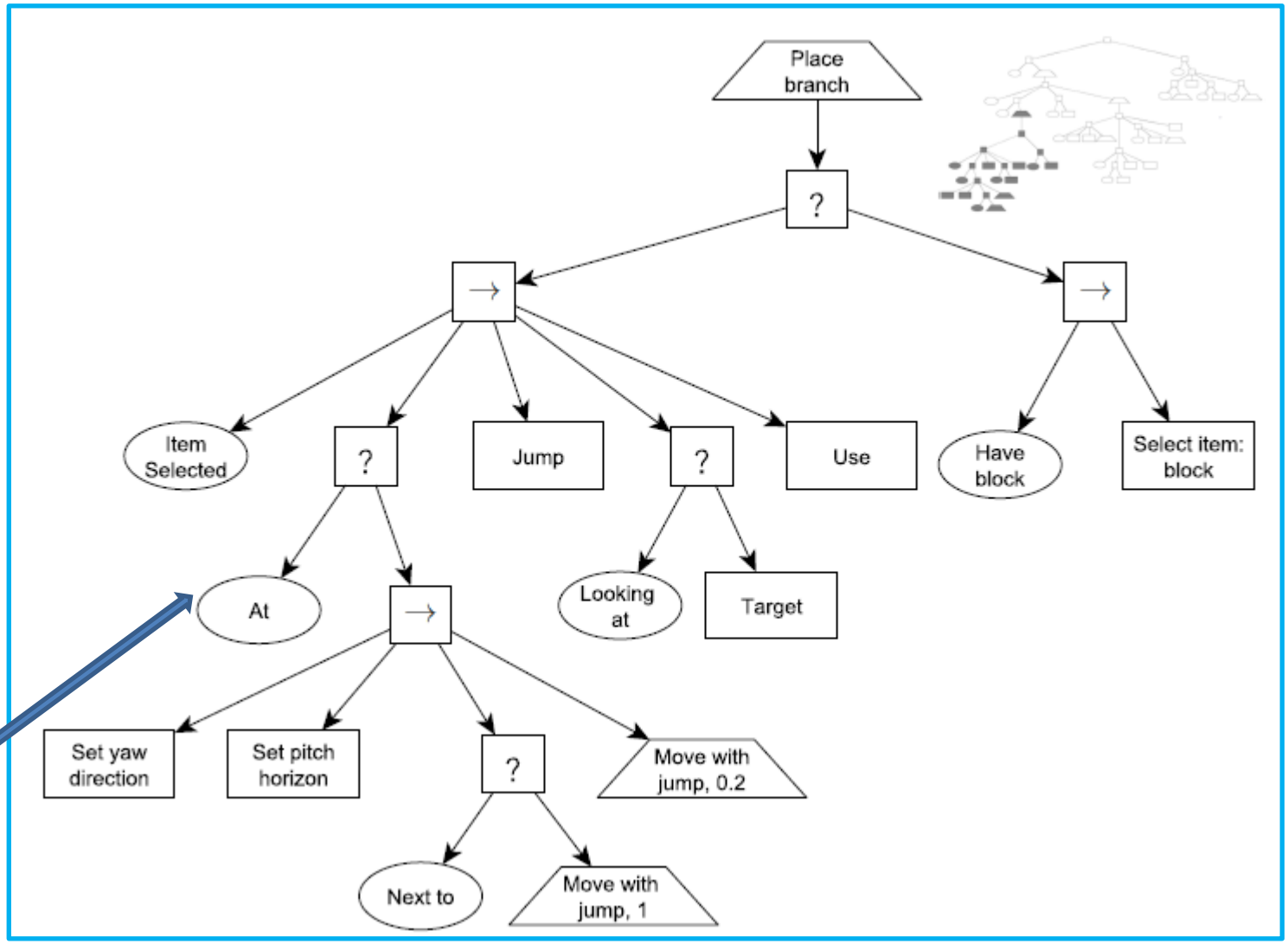
(b) In progress

(c) Completed



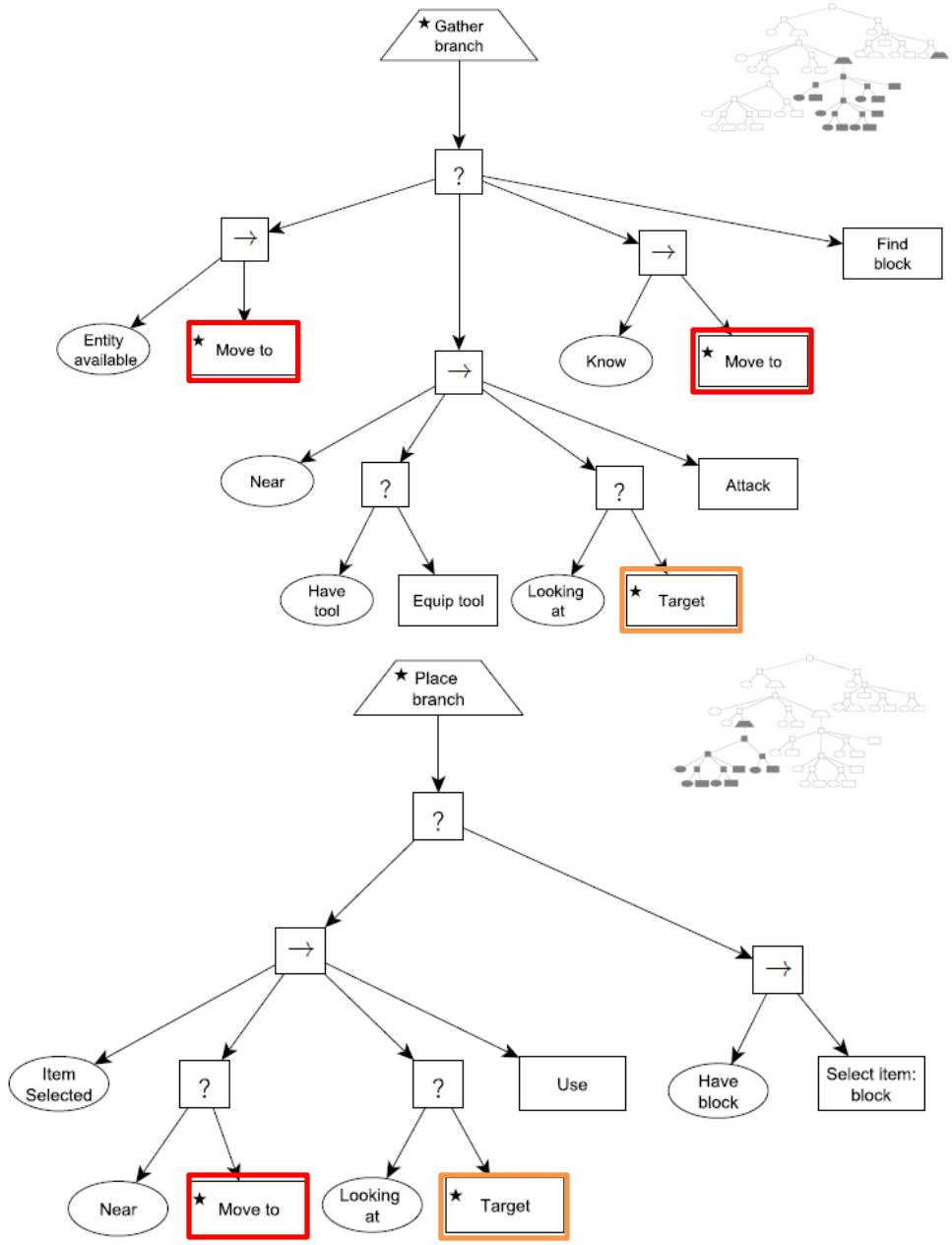

Build branch
?
Block at coordinates
→
→
Gather block branch
Have block
Place block branch
Have source
Craft block


Gather branch
?
→
→
Find block
Entity available
Forget block type
Set yaw direction
Move with jump
Know
Set direction
Move with jump
→
Near
?
Target
Attack
Have tool
Equip tool



The “At” node refers to a lookup table to place blocks at a list of locations

Adding DQN Agents for “Move to” and “Target” nodes



Observation space

- Distance to target [0, 50] ([0, 5] for targeting)
- Yaw to target, [-180, 180]
- Pitch to target, [-180, 180] (Only for targeting)
- 3x4x3 grid of blocks surrounding the agent

Rewards

Observations	Time step	Target	Distance
Targeting	-0.50	1000	-3000
Movement	-0.50	0	$\Delta \cdot 25$

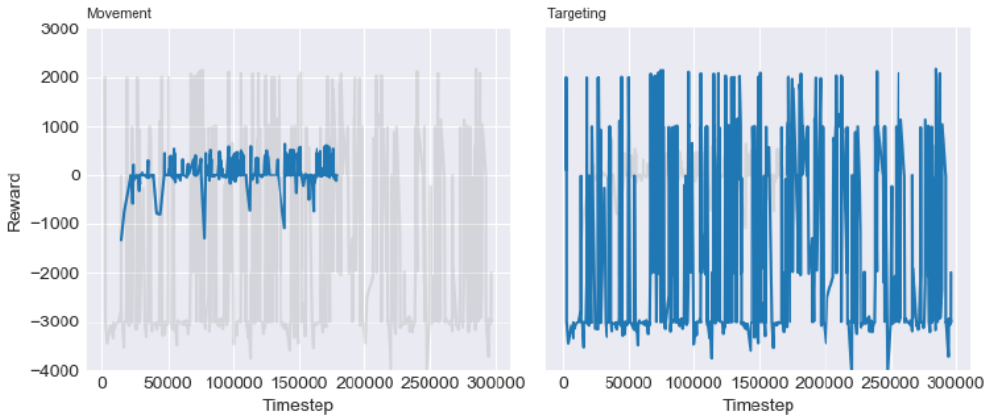
Action space for Move-to and Target nodes (Target node uses slower speeds)

- Move forward
- Move backward
- Move left
- Move left & forward
- Move left & back
- Move right & forward
- Move right & back
- Move right
- Turn right
- Turn left
- Jump
- No operation

“Target” node exclusive actions:

- Pitch up
- Pitch down

Their Conclusions



Configuration	Step count		
	Max	Min	Mean
Learned	19385.0	5503.0	9758.48
Manual	3835.0	1094.0	1645.8

Table 8: Multiple node decomposition evaluation over 25 missions

Their targeting node was never able to truly stabilize and, while it achieves its function during evaluation, it does not perform very well by qualitative observation

From their other experiments- learning nodes should not be nested , unless underlying functions are pre-trained. The Options Framework is a possible solution

“Deep Reinforcement Learning for High Precision Assembly Tasks” Inoue et al. IROS 2017

- Deep Q Learning with 2 LSTM layers, and a discrete action space
- **Search Phase:**

Chooses action of translational and downward force. End phase if peg’s z- position is 0.5mm below start position.

State: $\mathbf{s} = \left[F_x, F_y, F_z, M_x, M_y, \tilde{P}_x, \tilde{P}_y\right]$

Actions:

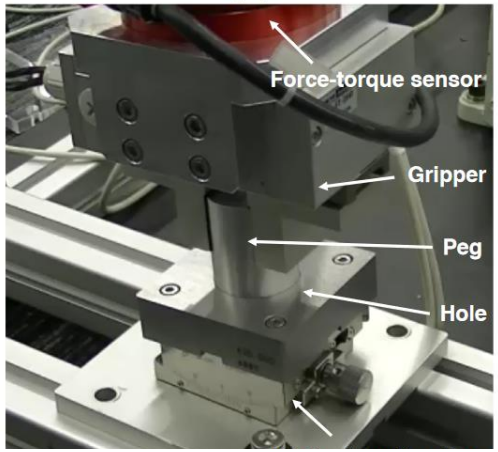
$$\begin{array}{ll} 1) & \left[+F_x^d, 0, -F_z^d, 0, 0\right] \\ 2) & \left[-F_x^d, 0, -F_z^d, 0, 0\right] \\ 3) & \left[0, +F_y^d, -F_z^d, 0, 0\right] \\ 4) & \left[0, -F_y^d, -F_z^d, 0, 0\right] \end{array}$$

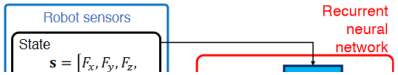
- **Insertion Phase:**

State: $\mathbf{s} = [0, 0, F_z, M_x, M_y, 0, 0]$

Actions:

$$\begin{array}{ll} 1) & \left[0, 0, -F_z^d, 0, 0\right] \\ 2) & \left[0, 0, -F_z^d, +R_x^d, 0\right] \\ 3) & \left[0, 0, -F_z^d, -R_x^d, 0\right] \\ 4) & \left[0, 0, -F_z^d, 0, +R_y^d\right] \\ 5) & \left[0, 0, -F_z^d, 0, -R_y^d\right] \end{array}$$





Approach	(1)	(2)	(2)	(2)
Clearance [μm]	≥ 10	10	10	20
Angle error [°]	≤ 1.0	0	0	1.6
Initial position error [mm]	≤ 1.0	1.0	3.0	1.0
Search time (s)	–	0.97	2.26	0.95
Insertion time (s)	–	1.40	1.33	2.31
Total time (s)	~5.0	3.47	4.68	4.36

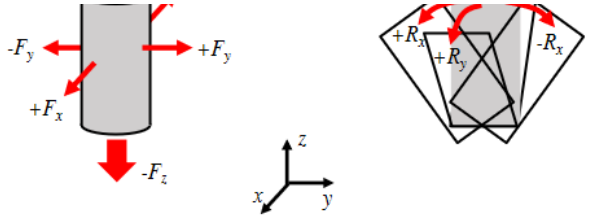


Fig. 4. Elementary movement: (a) Force movements (b) Rotation movements

“A Learning Framework for High Precision Industrial Assembly”

- Framework combines model-based RL with trajectory optimization (Guided Policy Search) and model-free actor-critic (DDPG), to assemble Lego bricks in a variety of configurations.
- Continuous action space.
- Trajectory optimization was used to res

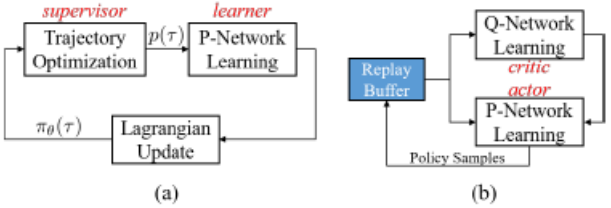


Fig. 1: (a) Guided Policy Search (GPS) and (b) Deep deterministic policy gradient (DDPG).

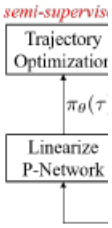
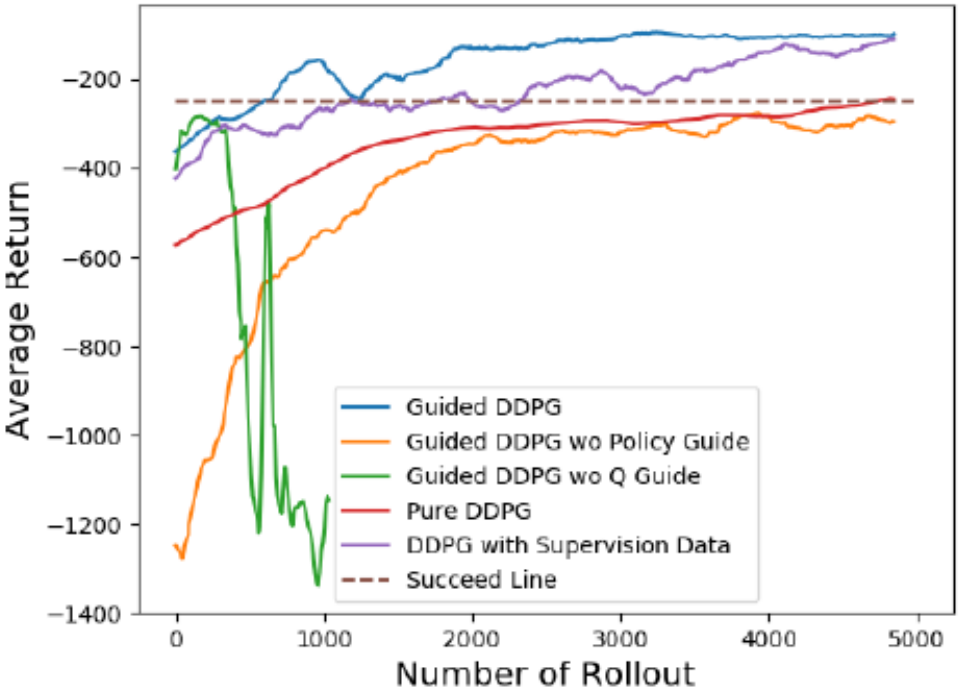


Fig. 2: Illustration
DDPG). Trajectory
actor and critic nets
nets gradually estab
pure supervised lea



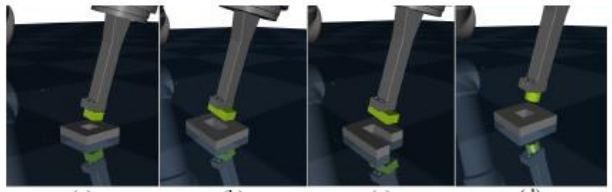
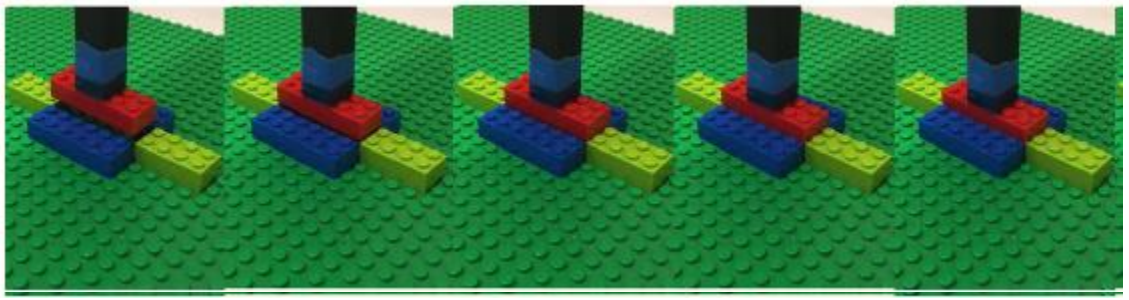
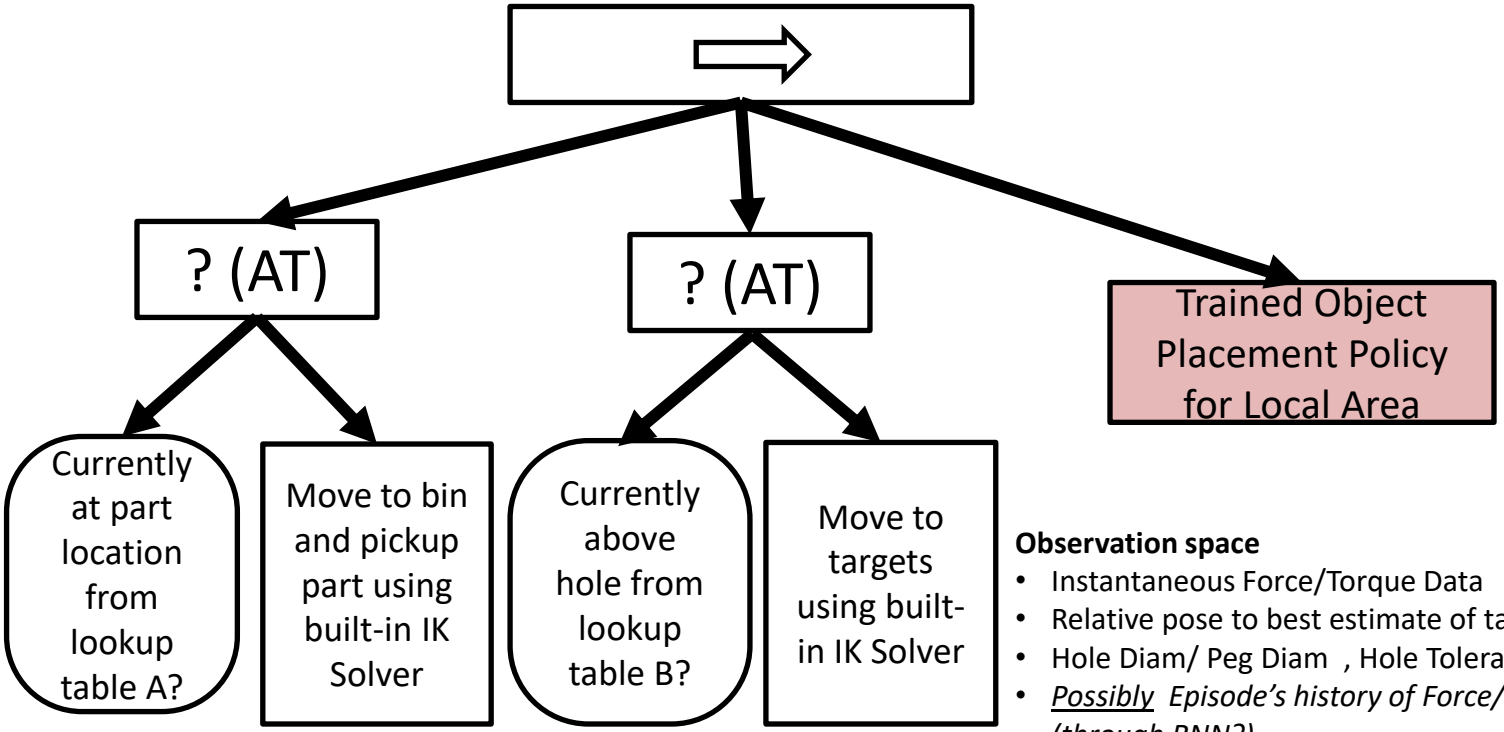


Fig. 8: Different shapes of the bricks and holes for adaptability test. (a) 2×2 brick used in training, (b) 4×2 brick, (c) 4×2 brick with incomplete hole, and (d) cylinder brick.



Architecture Proposal- Low Level



Lookup table A
(Bin locations)

Lookup table B
(Hole Locations)

Observation space

- Instantaneous Force/Torque Data
- Relative pose to best estimate of target position
- Hole Diam/ Peg Diam , Hole Tolerance?
- *Possibly Episode's history of Force/Torque Data (through RNN?)*
- *Possibly low-res point cloud data*

Rewards:

Z-condition records success

Action space (Gripper always points down)

- Jog X
- Jog Y
- Jog Z
- Spiral segment move CCW (with positioning based on memory)
- Erase spiraling position memory, set spiral parameters wider
- Erase spiraling position memory, set spiral parameters smaller
- No operation

Thank you