Motion Planner Augmented Reinforcement Learning for Robot Manipulation in Obstructed Environments

Yeeun Lim (임예은)

Motivation

- **Manipulation skills**

(a) 2D Push (b) Sawyer Push (c) Sawyer Lift (d) Sawyer Assembly

" Long- horizon planning capabilities "

" Obstructed Environment "

Motivation

- **Deep Reinforcement Learning**

  pros : continuous control problems

  cons : mostly operated controlled and uncluttered environment.
  In messy env, exploration becomes challenging.

- **Motion Planning**

  pros : easily find collision-free path in static environment.

  cons : struggles with rich interactions with objects

  (challenging to obtain accurate contact models)

  Struggles with complex manipulation task (e.g. object pushing)

Motivation

Deep Reinforcement Learning

+

Motion Planning

--

MoPA-RL : Motion Planner Augmented RL

how to balance the use of motion planner and primitive action.

Exploration by DRL is small perturbations in the action space,

struggling to find a path to the goal in obstructed environments.

-> harness MP techniques.

Main Idea

- **They has different size of action.**

DRL

Action: small joint angle

$$||a|| = \Delta q_{drl} = 0.05$$

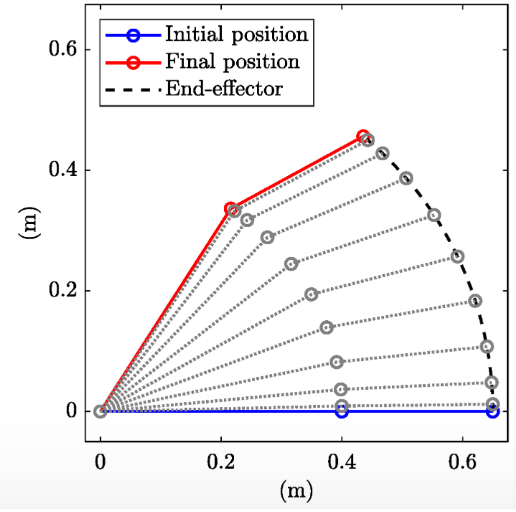
$$q : robot\ joint\ angle$$

Motion Planning

Action: large joint displacement

$$||a|| = \Delta q_{mp} = 0.5$$

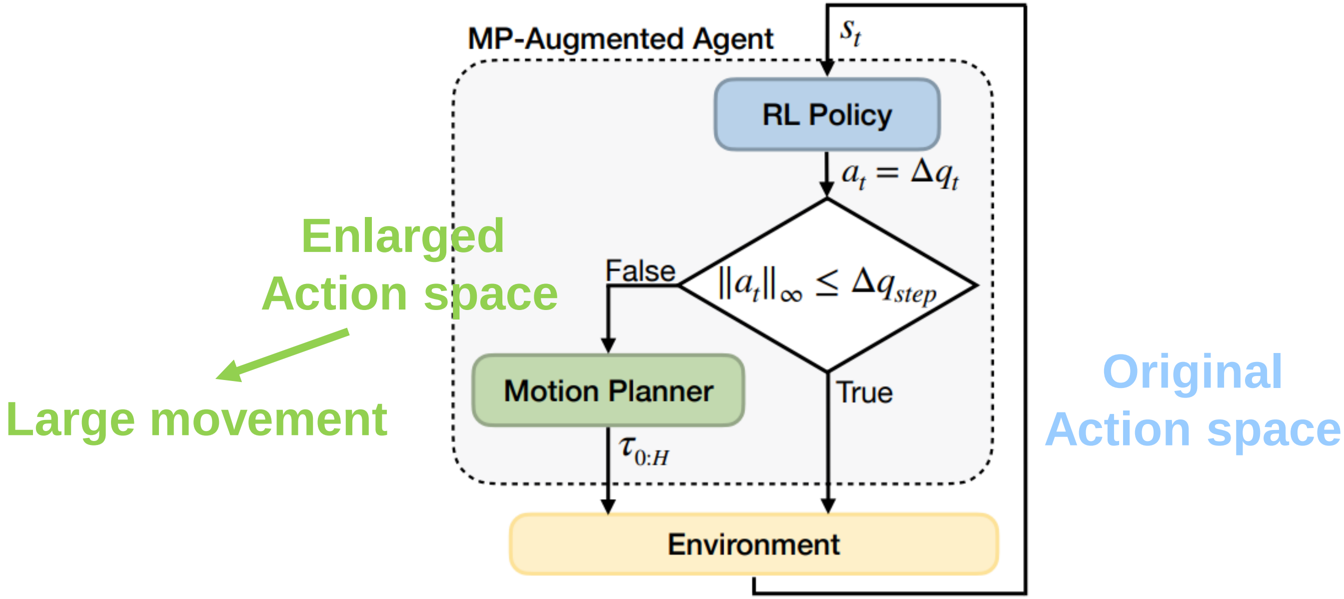
$$q : robot\ joint\ angle$$



- **Augmenting action space of agent !**

MP is integrated into the RL policy by enlarging the action space

Main Idea : MoPA-RL Framework



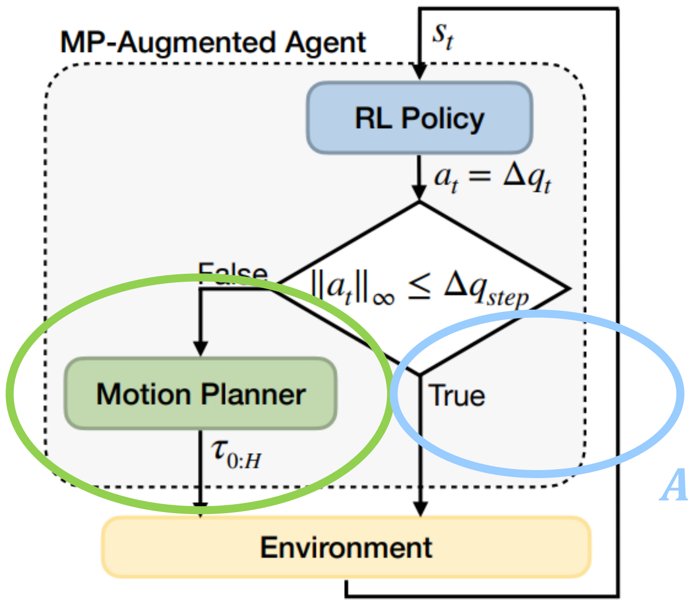
- **Allow an agent to freely switch between MP and direct action execution by controlling the scale of action.**
- **Naturally learn trajectories that avoid collisions by leveraging motion planning, while directly executing small actions for sophisticated manipulation**

Main Idea : MoPA-RL Framework

Enlarged Action space

$\widetilde{A} = [-\Delta q_{MP}, \Delta q_{MP}]^d$

$\widetilde{a} = \Delta\tau_{0:H}$



Original Action space

$A = [-\Delta q_{MP}, \Delta q_{MP}]^d$

- **Augmented MDP** $\widetilde{M}(S, \widetilde{A}, \widetilde{P}, \widetilde{R}, \rho_0, \gamma)$

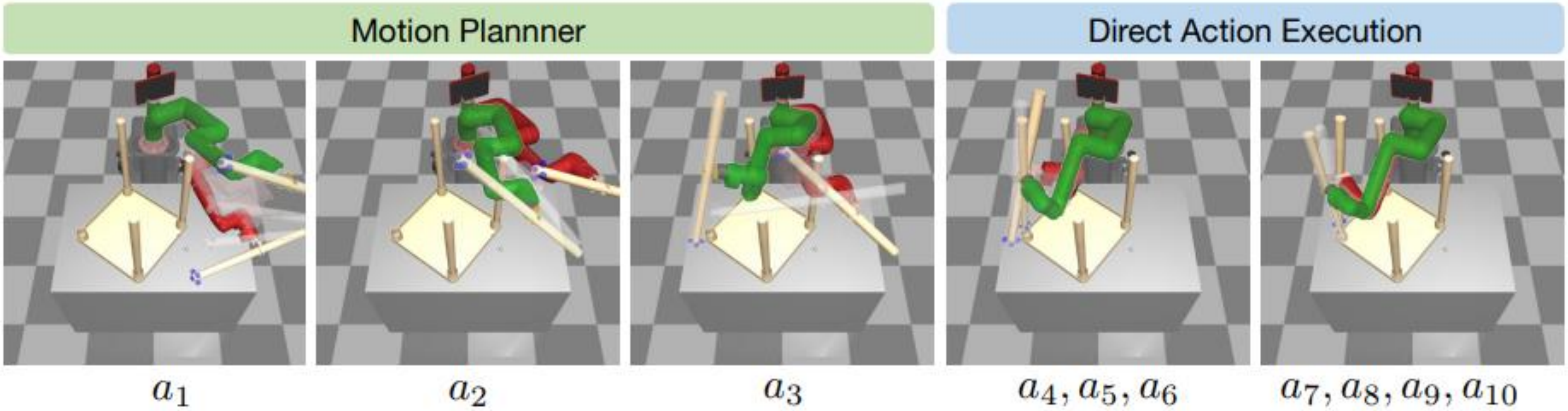
  $\widetilde{P}(s'|s, \widetilde{a})$

  $\widetilde{R}(s, \widetilde{a})$

- $\pi_\varphi(\widetilde{a}|s)$ **-> maximize** $E_{\pi_\varphi}[\sum_{t=0}^{\widetilde{T}} \gamma^t \widetilde{R}(s_{\tilde{t}}, \widetilde{a}_{\tilde{t}})]$

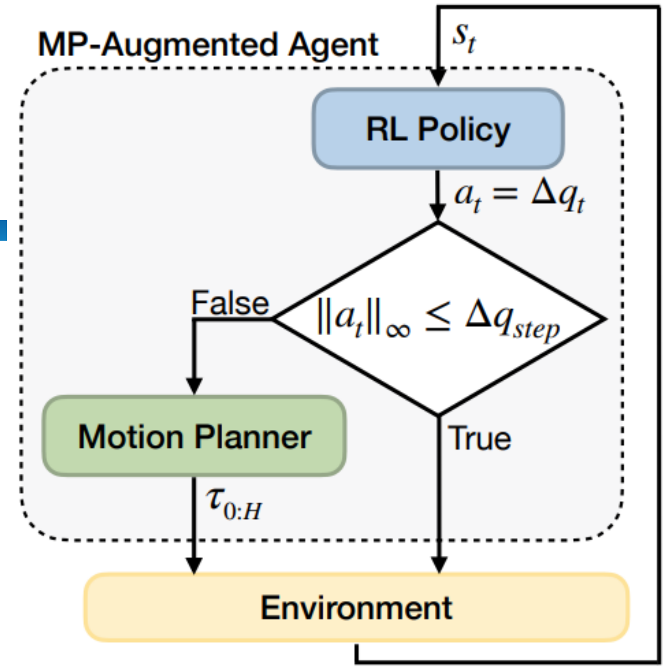
KAIST

Main Idea : execution

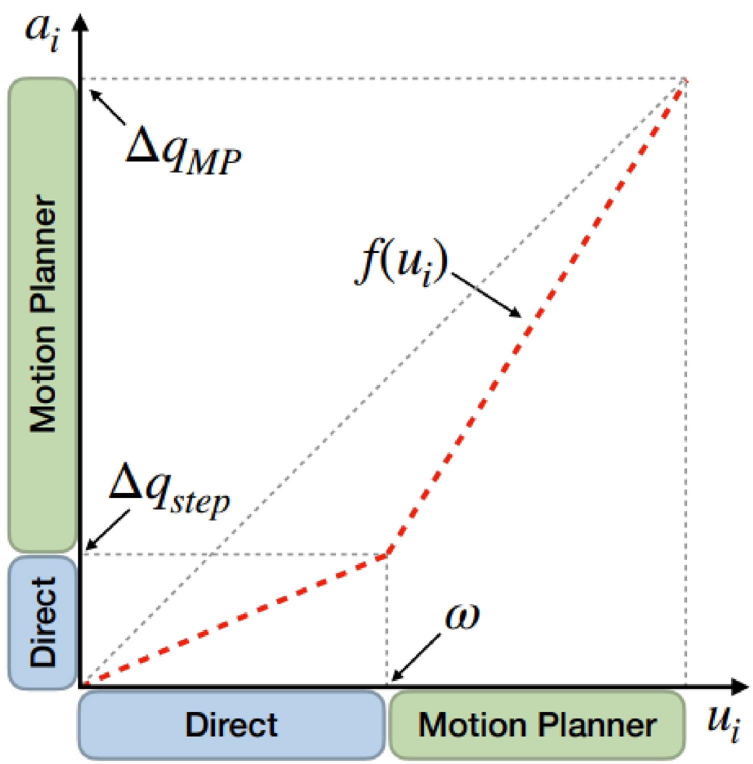


Technique

- **Action space rescaling**



Output from RL Policy



Input to Robot

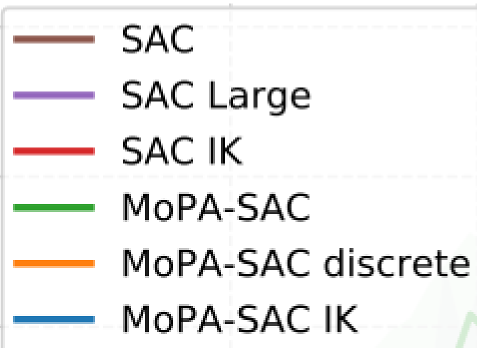
Probability of selecting direct action

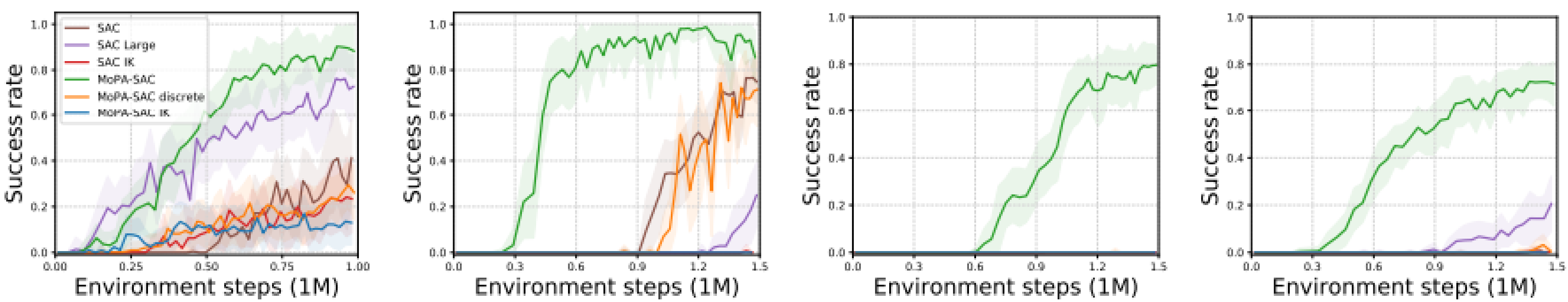
$$\left(\Delta q_{\text{step}} / \Delta q_{\text{MP}}\right)^d$$

d : dimension of robot joint

Result

- **Success rates**



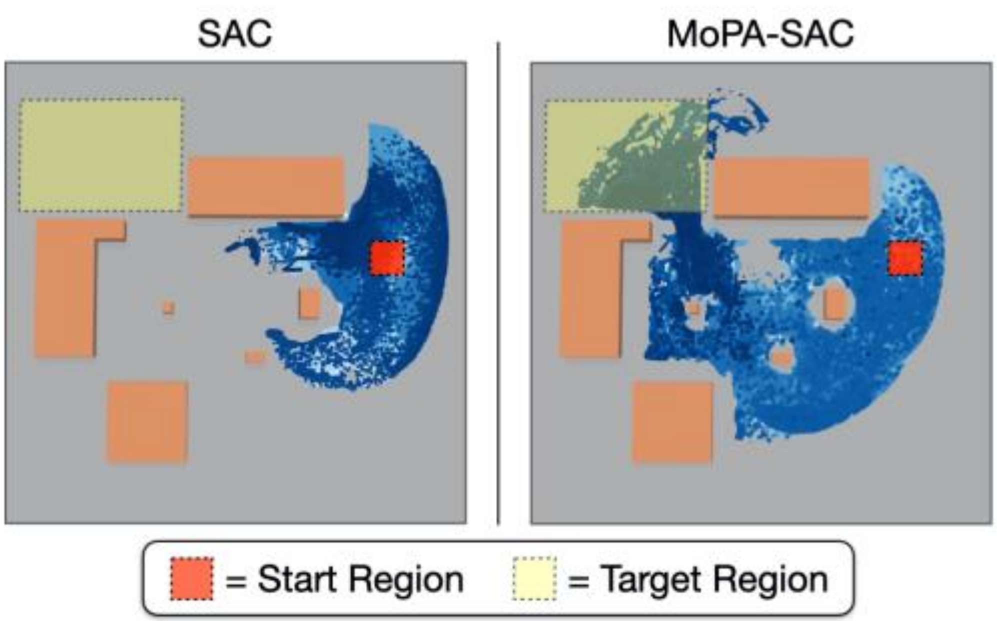


(a) 2D Push (b) Sawyer Push (c) Sawyer Lift (d) Sawyer Assembly

Result

- **Exploration efficiency (for training speed)**
- “Why augmenting RL agents with MP improves learning performance?”

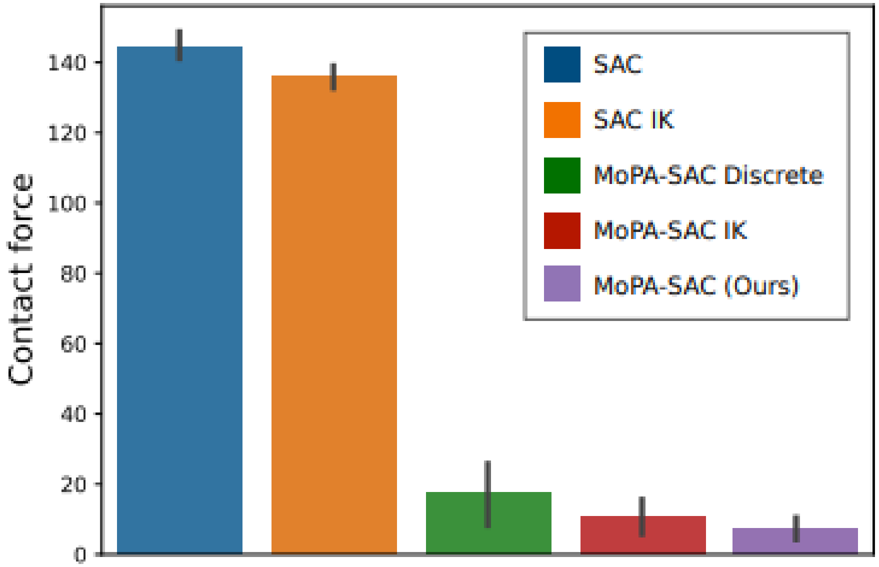
First 100k training steps



Result

- **Contact forces (collision-safe trajectory)**

Average contact force in episode over 7 executions in 2D Push



Conclusion

- **Contribution**

- flexible framework that combines the benefits of both motion planning and reinforcement learning, which has freedom to choose.

- Sample-efficient learning of continuous robot control in obstructed environments.

Q & A

- **Thank you for listening ☺**

- **If you want more information, you can check appendix.**

Quiz

Q1. Match the alphabet and number corresponding to the advantages of each method

a) Deep Reinforcement Learning

b) Motion Planning

1) easily find collision-free path

2) more suitable contact-rich task

Q2. How does MoPA-RL decide whether to use a motion planner or direct control?

a) magnitude of the state

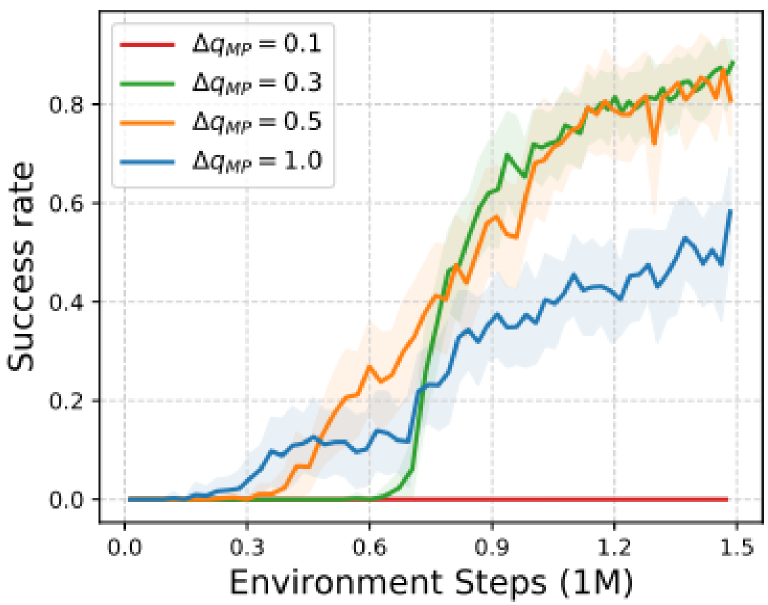
b) magnitude of the action

c) collision of environment

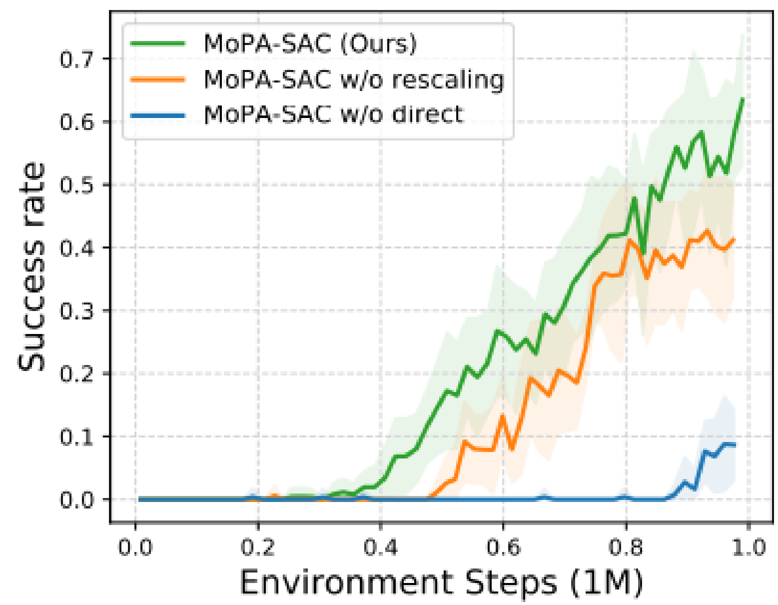
d) consider task

Appendix

- **Result of action scale**



(b) Action range Δq_{MP}



(c) Action space rescaling

Appendix

- **What kind of Motion Planning is used?**

- A simpler motion planner that attempts to linearly interpolate between the initial and goal states instead of the sampling-based motion planner.

Appendix

- **Then, is there no collision at all in the learning process?**

  -> Answer is "NO"

- They do collision checking before move the manipulator.

- If the path created by MP has collision, then RRT-Connect is used to find a collision-free path amongst obstacles.

- RL policy can predict a goal joint state that is in collision or not reachable. Iteratively reduce the action magnitude and check collision to prevents the policy from being stuck or wasting samples, which results in improved training efficiency.