sewts

Weekly Review

13/04/21

- **Tasks**
- Updated rewards to include positive values ✅
- Understand intuition behind SAC ⚠ ***In progress***
- Conduct experiments and evaluate results using GUI and plots ⚠ ***In progress***
- **Problems**
- Results of experiment still not improved
- **To-Do Items for Next Week**
- Compare organization of current repo with RLPYT original repository
- Update states, values, rewards till you get
- Define more reward functions, states, actions for our use case
- **To-Do Later**
- Explore usage of intermediate testing on simulation before sim-to-real transfer
- Define use-case (for different type of towels (colour, texture, etc.) / one type)

RL Problem for obtaining one flat seam

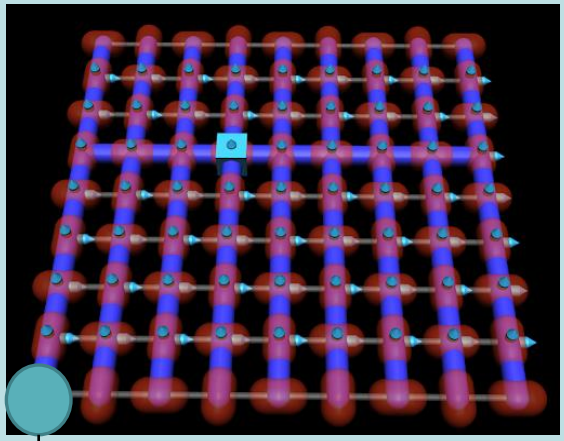
13/04/21

Goal

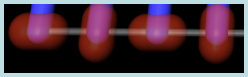
Obtain one flat seam

Given

Cloth in mujoco represented by 64 particles in 8*8 grid



Observations



[x,y,z] positions of 4 points adjacent to corner 1

Actions

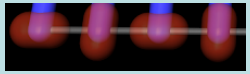
Random [x,y] movement of corner 1

RL Problem for obtaining one flat seam

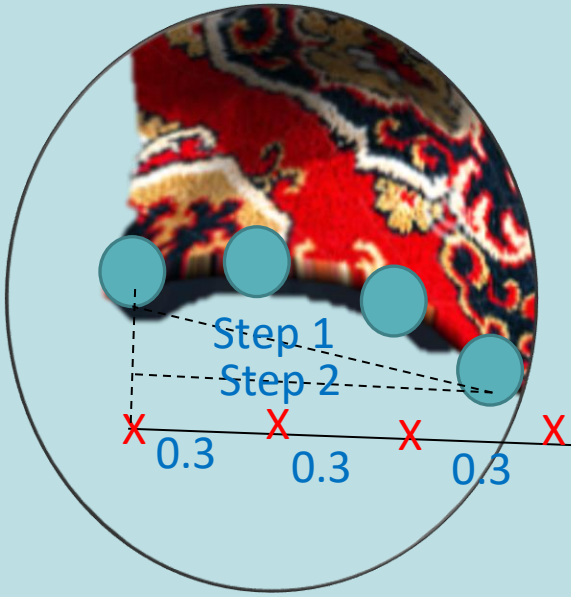
13/04/21

Goal

Obtain one flat seam -> Corner particle + 3 adjacent particles in a straight line



Reward



1. Join 1st point and last point
2. Project on x,y plane
3. Reward is proportional to :

(1 -1 * (x,y,z) distance from the ideal line)*10

Updated reward

STEP 1 : Sample actions from value network and store state, action, reward, next state, done in Experience Replay Buffer

Value Network (ψ)

Target Network (ψ)

Action sampled From mean, variance

Policy Network (φ)

Critic 1 Network (θ)

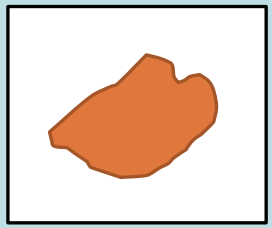
Critic 1 Network (θ)

Experience Replay Buffer

Maximize the reward + entropy of the policy

State, action, reward, next state, done

SAC Agent



Observation (64*64 img)

Environment (Mujoco)

Action

Pick point

Place point

STEP 2 : We train the Quality Network by minimizing the following error

$$\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p} \left[V_{\bar{\psi}}(\mathbf{s}_{t+1}) \right]$$

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t) \right)^2 \right]$$

Value Network (ψ)

Target Network (ψ)

Policy Network (φ)

Critic 1 Network (θ)

Critic 1 Network (θ)

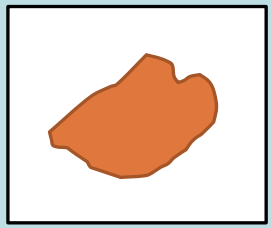
Experience Replay Buffer

Two networks are taken to avoid overestimation of Q values

Maximize the reward + entropy of the policy

State, action, reward, next state, done

SAC Agent



Observation (64*64 img)

Environment (Mujoco)

Action

Pick point

Place point

STEP 3 : We train the Value Network by minimizing the following error

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\frac{1}{2} \left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) \right] \right)^2 \right]$$

Value Network (ψ)

Target Network (ψ)

Policy Network (φ)

Critic 1 Network (θ)

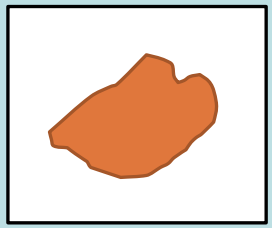
Critic 1 Network (θ)

Experience Replay Buffer

State, action, reward, next state, done

Maximize the reward + entropy of the policy

SAC Agent



Observation (64*64 img)

Environment (Mujoco)

Action

Pick point

Place point

STEP 4 : We update the Target Network

Value Network (ψ)

Lags the Value Network

Target Network (ψ)

Policy Network (φ)

Critic 1 Network (θ)

Critic 1 Network (θ)

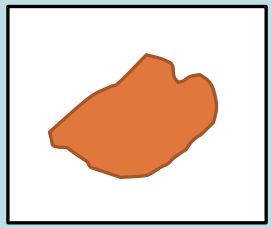
Target for Q Network depends on V Network and vice versa

Experience Replay Buffer

Maximize the reward + entropy of the policy

State, action, reward, next state, done

SAC Agent



Observation (64*64 img)

Environment (Mujoco)

Action

Pick point

Place point

Plan updation

13/04/21

V3	13/04/2021	<u>Planned</u> **Phase 3 : Implementation : 52 days (mid Feb- early Apr)** a) Setting up the Reinforcement Learning Platform and Simulation environment : 13 days b) Prepare a custom implementation taking existing states, actions, rewards : 9 days c) Redefine actions and rewards for our use case : 15 days d) Test the pipeline and iterate : 15 days **Phase 4 (Additional): Sim-to-real transfer : 28 days (early Apr- early May)** *Master's Thesis Registration* d) Perform domain randomization : 7 days e) Transfer to real robot : 21 days **Phase 5 (Additional): Enhancement : 31 days (May)** a) Exploration of alternate sim-to-real transfer approaches and Exploration of acceleration strategies : 10 days b) Incorporation of a proven acceleration approach : 15 days	Require more time to get valid results with SAC Algorithm

Plan updation

13/04/21

Update

Phase 3 : Implementation : 52 +22 days (mid Feb-end Apr)

e) Setting up the Reinforcement Learning Platform and Simulation environment : 13 days

f) Prepare a custom implementation taking existing states, actions, rewards : 9 days

g) Redefine actions and rewards for our use case : 15 days

h) Test the pipeline and iterate : 15 days

(+22 days for g) and h))

Phase 4 (Additional): Sim-to-real transfer : 31 days (May)

Master's Thesis Registration

f) Perform domain randomization : 7 days

g) Transfer to real robot : 21 days

sewts

THANK YOU