Split Deep Q-Learning for Robust Object Singulation

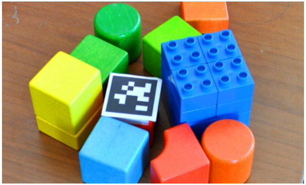
Iason Sarantopoulos, Marios Kiatos, Zoe Doulgeri and Sotiris Malassiotis

2020 International Conference on Robotics and Automation,
May 31 - June 4, 2020. Paris, France

Introduction

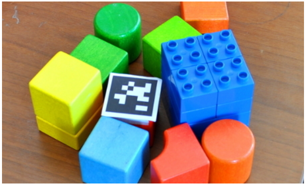
Task:

- Extracting a target object from a pile of other objects in a cluttered environment.
- Prehensile grasping is impossible due to clutter



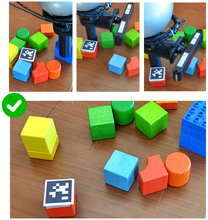
Task:

- Extracting a target object from a pile of other objects in a cluttered environment.
- Prehensile grasping is impossible due to clutter



Policy:

- Pushing policy for singulating the target object.
- A novel Split Q-learning algorithm is proposed

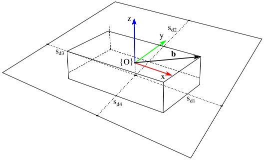


Environment and Assumptions

Environment and Assumptions

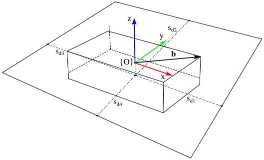
Environment:

- A target object with known pose $\{O\}$ and bounding box, placed on a support surface.
- Poses, dimensions and the number of obstacles are random and unknown.



Environment:

- A target object with known pose $\{O\}$ and bounding box, placed on a support surface.
- Poses, dimensions and the number of obstacles are random and unknown.



Assumptions:

- Availability of:
  - Robotic finger for pushes
  - RGB information for pose estimation of target
  - Depth information for state representation
- Collisions between the fingertip and an object can be detected.
- Pushing actions result in 2D motion of the target. No flipping is expected.

Singulation

Singulation means that the target object is separated from the closest obstacle by a minimum distance d_{sing}.

Singulation

Singulation means that the target object is separated from the closest obstacle by a minimum distance d_{sing}.

Objective

Singulate the target object from its surrounding obstacles by:

- using the minimum number of pushes and
- avoiding to throw the target off the support surface's limits.

MDP Formulation

- A pushing action:
  $\mathcal{P} = (p_0, \quad d, \quad \theta)$

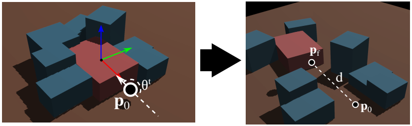


Figure 1: Push target

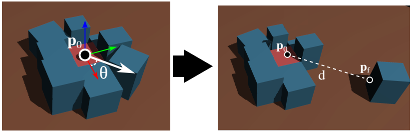


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (p_0, \quad d, \quad \theta)$
- Push target object

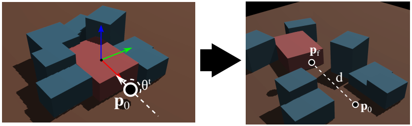


Figure 1: Push target

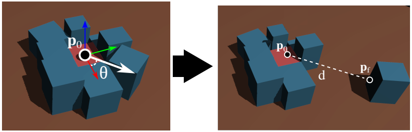


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target

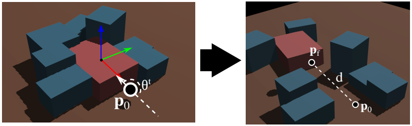


Figure 1: Push target

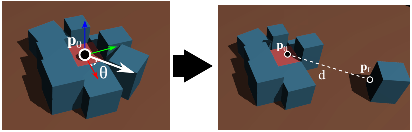


Figure 2: Push obstacle

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle

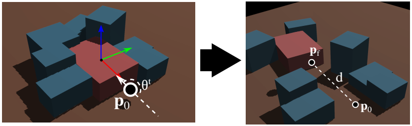


Figure 1: Push target

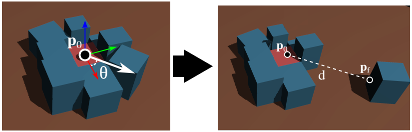


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle
- Push obstacle

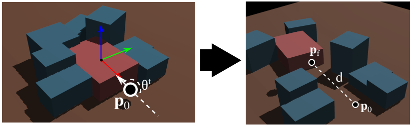


Figure 1: Push target

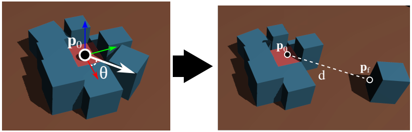


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (p_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle
- Push obstacle
  - Placing the finger above the target for pushing obstacles

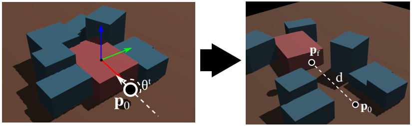


Figure 1: Push target

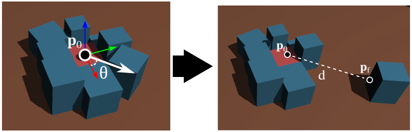


Figure 2: Push obstacle

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle
- Push obstacle
  - Placing the finger above the target for pushing obstacles
  - Risk of empty push

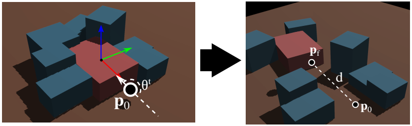


Figure 1: Push target

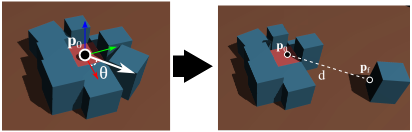


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle
- Push obstacle
  - Placing the finger above the target for pushing obstacles
  - Risk of empty push
- d predetermined and θ discretized in w pushing directions

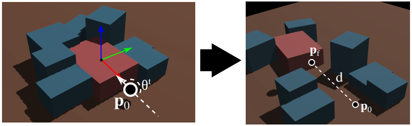


Figure 1: Push target

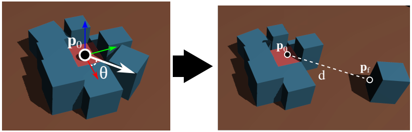


Figure 2: Push obstacle

MDP: Actions

- A pushing action:
  $\mathcal{P} = (\boldsymbol{p}_0, \quad d, \quad \theta)$
- Push target object
  - Placing the finger beside the target
  - Risk of undesired collision with an obstacle
- Push obstacle
  - Placing the finger above the target for pushing obstacles
  - Risk of empty push
- d predetermined and θ discretized in w pushing directions
- $2w$ total discrete actions.

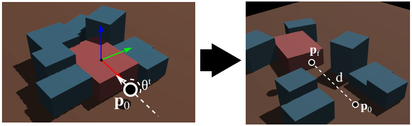


Figure 1: Push target

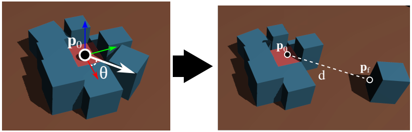
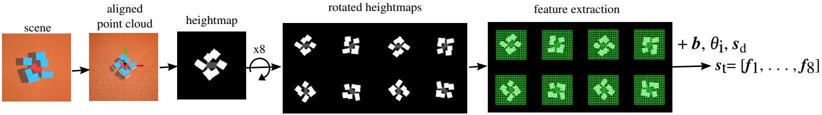


Figure 2: Push obstacle

MDP: State



- Transform the point cloud w.r.t. $\{O\}$
- Generate heightmap.
- Rotate heightmap w times.
- For each rotation:
    - Define a 16x16 region and average the values of the heightmaps

$$z_{ij} = \frac{1}{c_x \cdot c_y} \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} h_i(x, y)$$

    - Add the bounding box, the rotation angle and the distances of the target from the table limits $\boldsymbol{s}_d$.

MDP: Rewards and Terminal states

Sparse reward function for each timestep:

- Singulation $r = +10$ (successful terminal state)
- Falling off the table $r = -10$ (failed terminal state)
- Undesired collision $r = -10$ (failed terminal state)
- Empty pushes $r = -5$
- Otherwise: $r = -1$

Split DQN

Split DQN

- One fully connected network for each primitive.

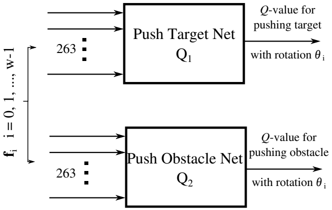


Figure 3: Split DQN

Split DQN

- One fully connected network for each primitive.
- Policy: $\operatorname{argmax}_{action} Q(state, action)$

$$\max Q = \max\left(\max Q_1(f_i), \max Q_2(f_i)\right)$$

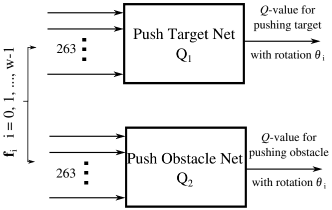


Figure 3: Split DQN

Split DQN

- One fully connected network for each primitive.
- Policy: $\text{argmax}_{action}\, Q(state, action)$

$$\max Q = \max\left(\max Q_1(f_i), \max Q_2(f_i)\right)$$

- Advantages:

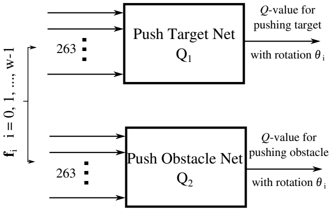


Figure 3: Split DQN

- One fully connected network for each primitive.
- Policy: $\text{argmax}_{action}\, Q(state, action)$

$$\max Q = \max\left(\max Q_1(f_i), \max Q_2(f_i)\right)$$

- Advantages:
  - The rotation invariant features simplifies learning

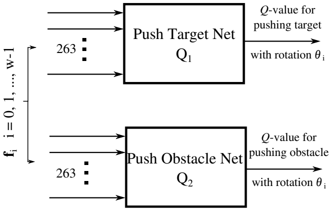


Figure 3: Split DQN

Split DQN

- One fully connected network for each primitive.
- Policy: $\text{argmax}_{action}\, Q(state, action)$

$$\max Q = \max\left(\max Q_1(f_i), \max Q_2(f_i)\right)$$

- Advantages:
  - The rotation invariant features simplifies learning
  - Training on data that comes from the same distribution (same primitive) results to faster learning

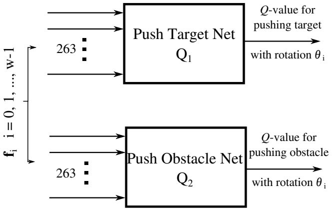


Figure 3: Split DQN

Split DQN

- One fully connected network for each primitive.
- Policy: $\text{argmax}_{action}\, Q(state, action)$

$$\max Q = \max\left(\max Q_1(f_i), \max Q_2(f_i)\right)$$

- Advantages:
  - The rotation invariant features simplifies learning
  - Training on data that comes from the same distribution (same primitive) results to faster learning
  - Inherent modularity for adding new primitives.

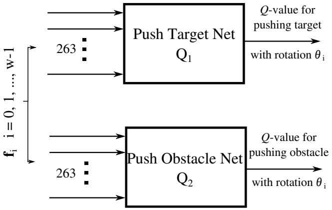


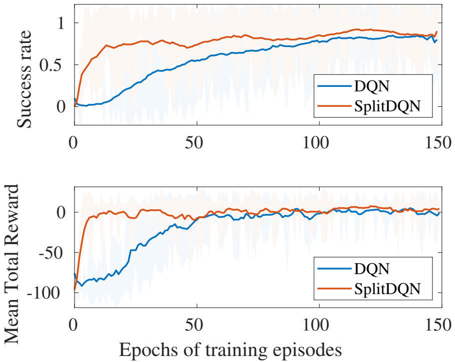
Figure 3: Split DQN

Experiments

Video

Results: Performance Evaluation

Policy	Success rate	Mean actions	Std actions	Mean reward	Std reward
Human	95.0%	2.46	0.88	7.51	4.36
SplitDQN	**88.6%**	**2.95**	1.43	**3.42**	18.56
DQN	77.1%	4.02	2.12	-1.924	23.01
Random	22.1%	5.79	3.24	-10.17	8.79
SplitDQN (Real)	75.0%	2.71	1.18	-1.37	5.60



Success rate
1
0.5
0
0
50
100
150
DQN
SplitDQN
Mean Total Reward
0
-50
-100
0
50
100
150
Epochs of training episodes
DQN
SplitDQN

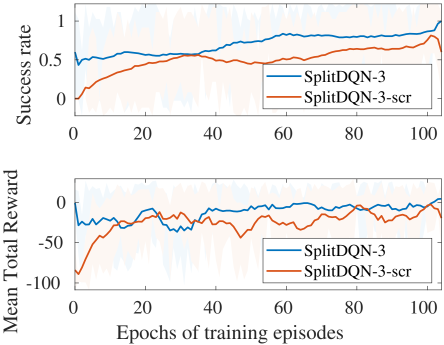
Extra primitive

Video

Extra primitive: Performance Evaluation

Policy	Success rate	Mean actions	Std actions	Mean reward	Std reward
SplitDQN-3	83.4%	3.19	1.43	-2.64	20.92
SplitDQN-2	59.6%	4.42	1.77	-20.35	40.95

Extra primitive: Training



Conclusions

Conclusion

- Splitting Q network to use one network per primitive results to faster convergence and increased success rate.
- The inherent modularity of the algorithm allows the addition of extra primitives.
- Effective training in a complex environment.
- Demonstrating that the policy can effectively transferred to a real world setup.

Thank you for watching.