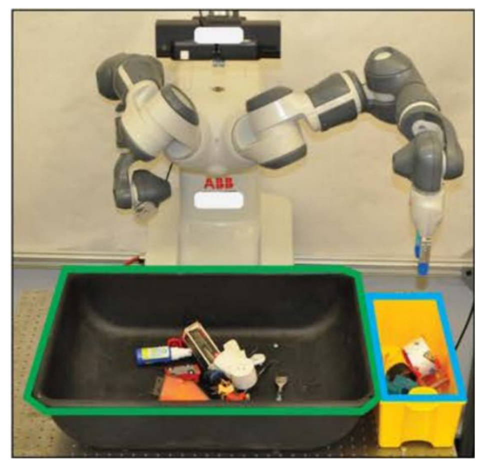
Deep Bin Picking with Reinforcement Learning

Jeff Chen, Tori Fujinami, Ethan Li

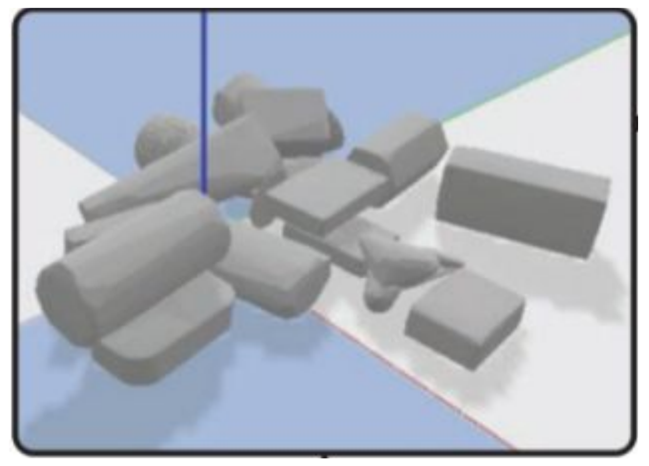
{jc1, fujinam2, ethanli}@stanford.edu - March 2018

Motivation

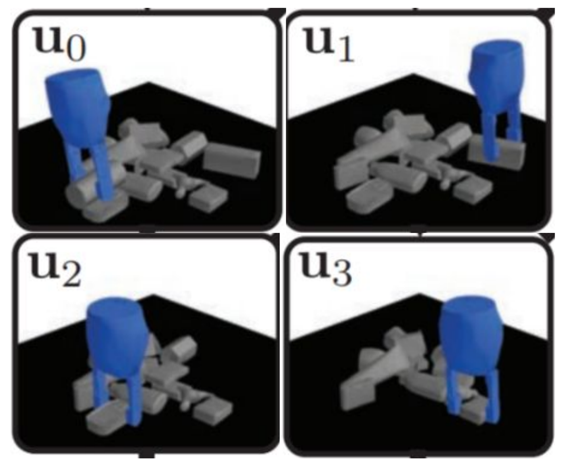
- Automated bin picking to enable robots to manipulate environment and assist or redirect human involvement in e-commerce logistics
- Bin clearing as sequential repeated bin picking: perception + grasp planning + item picking sequence planning
- State-of-the-art: deep CNNs for perception + grasp planning on **flat** cluttered pile with **heuristic** picking sequence policy [1]
- In deep pile, picking sequence may have delayed consequences



Example of shallow bin clearing task in [1]



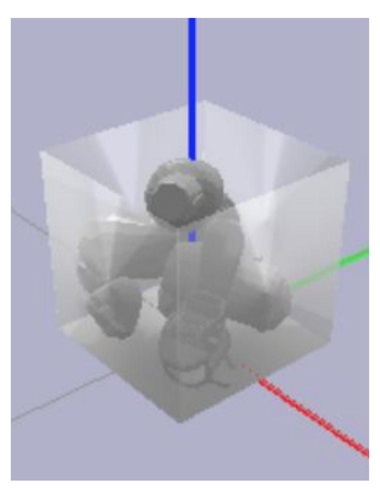
Example of simulated item pile from [1]



Example of item picking sequence from [1]

Simulation Environment

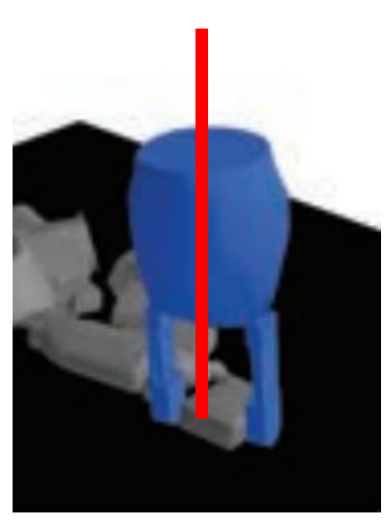
- Custom built environment using Pybullet [2] simulations of cubic crate filled with up to 10 randomly stacked items
- 96 item classes [1], no duplicates per crate
- Objects pulled upwards sequentially from crate
- Collision checking [1] prevents object removal attempts involving gripper-object collisions



Example of simulated crate

MDP Model

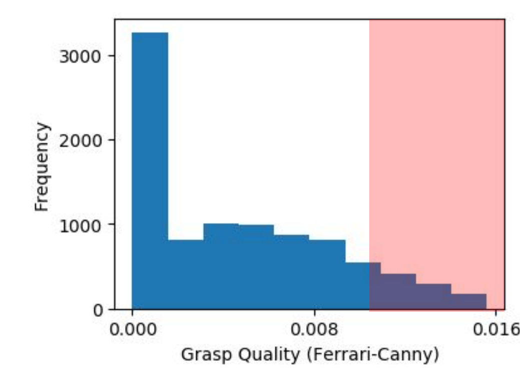
- States: Cartesian positions *(x,y,z)* and quaternion orientations (q_0,q_1,q_2,q_3) of all items in crate
- Actions: (Item class, Grasp (x,y,d,θ), Grasp Success Probability p_g) tuples for all objects in crate
- Rewards: 1 if selected item successfully removed
  - -10 for extraneously removed items
  - 0 otherwise
- State transitions: state at static equilibrium after attempted item removal, determined by p_g and simulation physics.
- Gamma: 0.9



Example grasp [1] of depth d and angle θ about axis through planar point (x,y)

Action Space Approximation

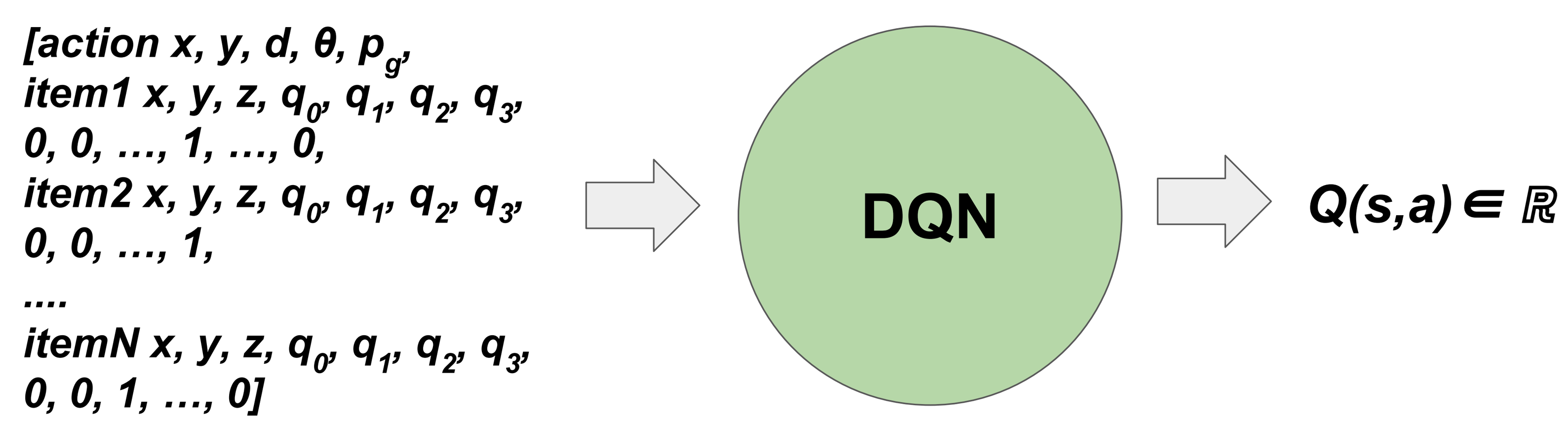
- [1] provides finite set of sampled grasps & precomputed grasp qualities per item
- Heuristic approximation of p_g from Ferrari-Canny grasp quality metrics: linearly rescale & clamp so that top 10% of grasps have $p_g = 0.9$
- Filtering heuristic: remove actions with $p_g < 0.1$
- Collision checking between gripper and items [1] to prune away infeasible actions with collisions
- Pruning heuristic: first pass checks collisions for 3 of the top 25 actions per item; when fewer than 3 feasible actions found, second pass checks collisions for 10 random actions per remaining item

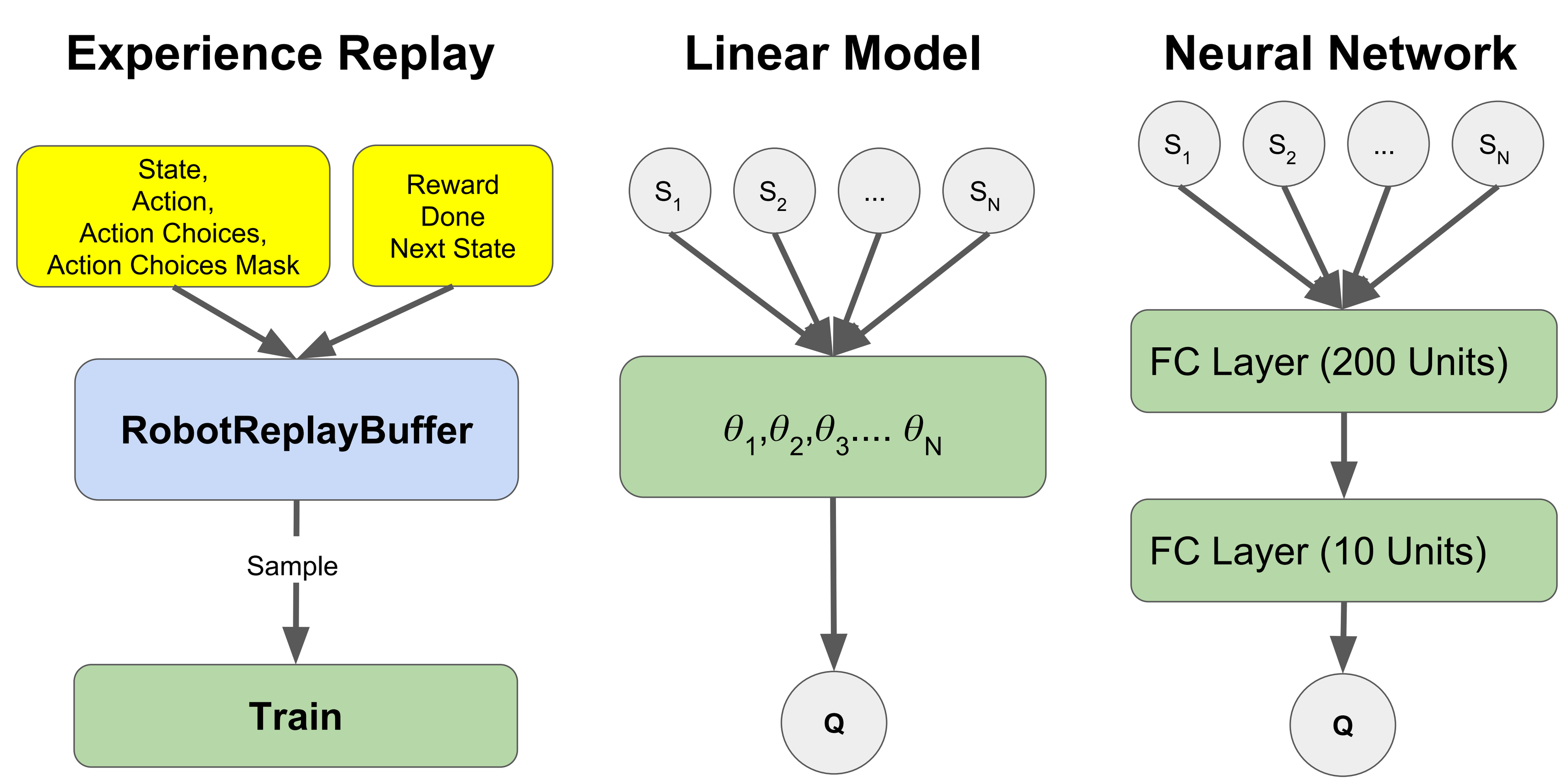


Precomputed grasp qualities from [1], with top 10% in red

DQN with Action Input

- Space of sampled actions is very large, requiring a DQN which takes an action in its input layer
- Input vector encoded as the concatenation of action, item poses and one-hot vectors of item classes
- Heuristically only evaluate $Q(s_t,a)$ for every item's best feasible action (highest p_g)
- Experience replay buffer stores s_t, a_t, and A_t

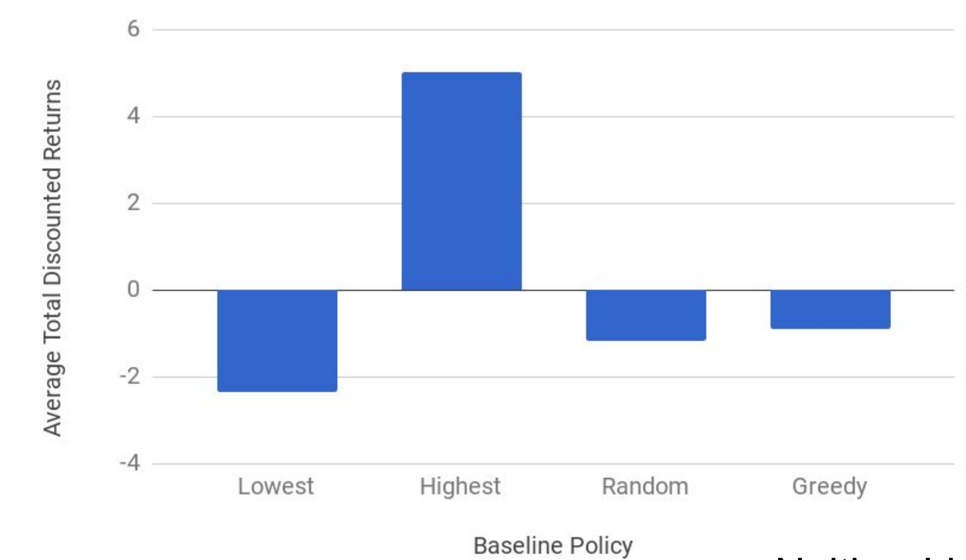




Results and Evaluation

Baseline Policies:

- Lowest Object First: Remove the object at the bottom of the pile (no feasibility check, assume guaranteed removal success)
- Highest Object First: Remove the object at the top of the pile (no feasibility check, assume guaranteed removal success)
- Greedy: Take the feasible action with highest p_g
- Random: Take a uniformly randomly selected feasible action



Lowest, Random, and Greedy policies all result in additional undesired objects being lifted and dropped outside the bin and thus get lower rewards. Highest overly optimistic

Models: DQN with Action Input

Neither Linear nor the NN train very well. This is expected for linear since it requires many more parameters and is less data efficient. We suspect NN also does poorly due to data deficiency leading to overfitting

Challenges and Next Steps

- Simulation speed: Large action space and slow collision checking result in slow simulations.
- Physics realism: Grasp success outcomes sampled over grasp success probabilities which are heuristically estimated from grasp quality metric, ignoring contacts with other items.
- Collision checking speedup heuristics: Further adjustment needed to balance physical realism with speed of identifying feasible actions.
- DQN training: Further architecture and hyperparameter tuning required for DQN to be able to train on task.

References

1. J Mahler, K Goldberg. Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences. *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 515-524. PMLR, 13-15 Nov 2017
2. E Coumans. Bullet physics library, 2012. https://pybullet.org/wordpress/