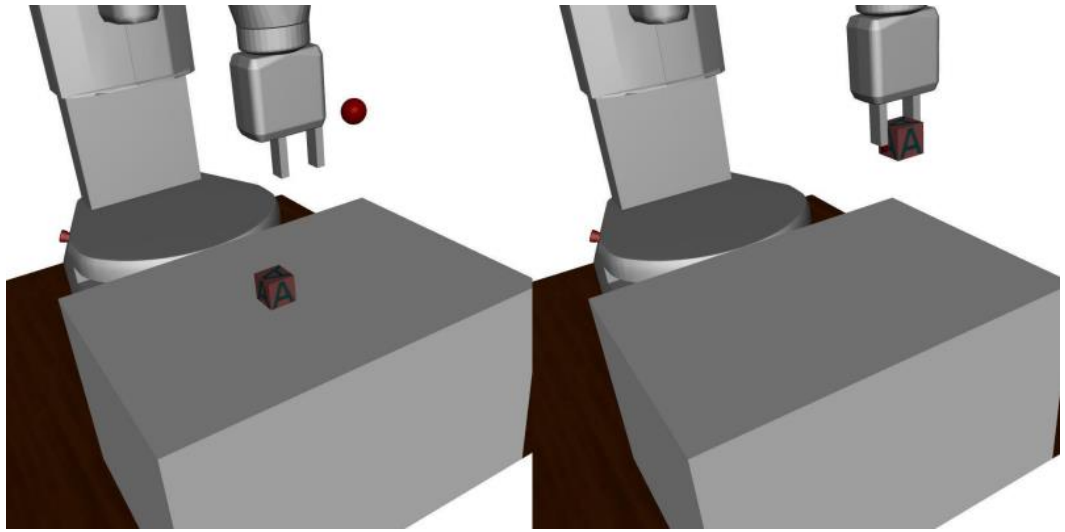
Overcoming Exploration in Reinforcement Learning with Demonstrations

Authors: Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, Pieter Abbeel

Presentation by: Scott Larter

Introduction

- Addresses problem of exploration in sparse reward tasks
- Focuses on tasks of moving objects with robotic arm
  - Pushing, sliding, pick-and-place, stacking
- Proposes new RL algorithm combining DDPG and demonstrations



Introduction

- Problem
  - In tasks with sparse rewards, agents may not receive any positive rewards for many consecutive timesteps
  - Very difficult to learn good policy with no indication whether actions taken are good or not
  - Random exploration does not work well
- Solution: eliminate random exploration phase using demonstrations
  - Perform a number of human demonstrations
  - Introduce auxiliary objective on demonstrations for training the actor
  - Introduce method to account for suboptimal demonstrations
  - Reset some training episodes using demonstration data

Related Work

- Imitation Learning
  - Behavior Cloning (BC) – uses supervised learning on demonstrations to learn policy
    - Autonomous driving, quadcopter navigation, and locomotion
  - Inverse Reinforcement Learning – infers reward function from demonstrations
    - Navigation, autonomous helicopter flight
  - This paper incorporates BC into new, improved method

Robotic Block Stacking

- Main success of paper is in robotic block stacking
- Recent work has tackled task, but require domain-specific engineering
- RL method, PILCO, has shown potential in block stacking but has trouble grasping blocks
- One-shot Imitation learns to adapt to new target configurations, but required over 100,000 demonstrations

Combining RL with Imitation Learning

- Imitation learning and RL have been combined before
  - Learn to hit a baseball and underactuated swing-up
- Deep Q-learning from Demonstrations (DQfD)
  - Shown potential in Atari games
  - Drawback: cannot learn past demonstrations
- Deep Deterministic Policy Gradients from Demonstrations (DDPGfD)
  - Applied to similar robotics tasks of peg insertion and other object manipulation
  - This paper extends and generalizes this approach

Background – Reinforcement Learning (RL)

- Standard MDP framework with fully observable environment E
- At timestep t, agent is in state x_t, takes action a_t, receives reward r_t, and transitions to state x_{t+1}
- Learns policy $a_t = \pi(x_t)$ to maximize return $R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r_i$ with horizon T and discount factor γ
- Bellman equation to estimate future rewards from given state after taking action
  - Action-value function: $Q^{\pi}(s_t, a_t)$

Background – Deep Deterministic Policy Gradients (DDPG)

- Off-policy, model-free, actor-critic RL algorithm
- Learns $Q(s, a)$ (critic) by minimizing Bellman error while learning $\pi(s)$ (actor) by maximizing Q w.r.t. policy parameters θ_π
- Maintains replay buffer R with tuples (s_t, a_t, r_t, s_{t+1})
- Alternates between collecting experience and updating parameters
- Training step:
    1) Sample N tuples from R
    2) Update critic function parameters by minimizing loss
    3) Update policy parameters with policy gradient

Background – Multi-Goal RL and HER

- Multi-goal RL
  - Agents have parametrized goals
  - In this case, goals are the target locations of all the objects
  - Goal of episode is sampled and given to π and Q as input
- Hindsight Experience Replay (HER)
  - Experiences are stored twice in R
    - With original goal
    - With goal corresponding to final state of episode
  - Failed rollouts still counted as successful by assuming end state was goal

Solution

- Four main components of solution method
  - Demonstration buffer
    - Maintain second replay buffer for demonstration data
    - Sample minibatches and use data in update step for both actor and critic
  - Behavior Cloning Loss
    - Introduce auxiliary loss function on demonstrations used to train the actor
    - Loss function used in gradient updates for actor parameters θ_{π}
  - Q-filter
    - Only apply BC loss when $Q(s, a)$ determines demonstration is better than the actor
  - Resets to demonstration states
    - Resets a training episode by sampling a demonstration and uniformly sampling a state x_i
    - Final state of the demonstration is used as goal of training episode

Advantages

- Ensures agent receives positive rewards early
- Behavior Cloning Loss prevents the learned policy from improving much past the demonstration policy
- Q-filter handles suboptimal demonstrations as actor is not tied back to demonstrations where the learned policy is better
- Resetting training episodes to demonstrations handles sparse rewards by exposing agent to higher rewards during training
- Advantage over previous work is highlighted in block stacking task
  - Complex task with sparser reward and longer horizon

Disadvantages

- Relies on demonstrations which cannot always be easily obtained
  - However, in the absence of demonstrations, successful rollouts could be used in their place
- Not very sample efficient
  - Requires a lot of experience which is not always feasible outside of simulation

Comparison to Prior and Related Work

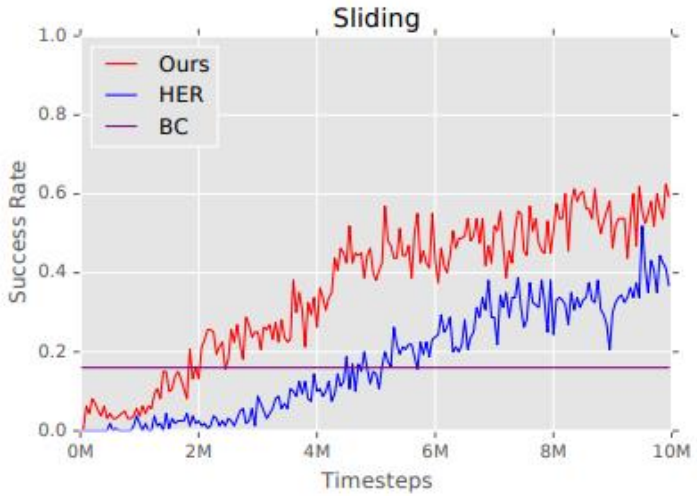
- HER has been used to deal with robotic arm tasks with sparse rewards [1]
  - Does not use demonstrations or behavior cloning
  - Tested on pushing, sliding, and pick-and-place tasks
- Leveraging demonstrations to guide exploration (DDPGfD) has been used [2]
  - Similar robotic arm tasks: inserting pegs and other objects into slots and holes
- Approach in this paper merges features from both papers into single method

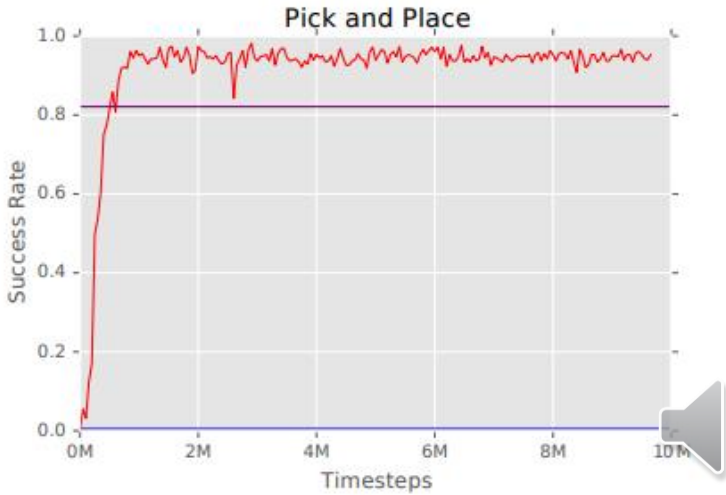
[1] M. Andrychowicz *et al.*, "Hindsight experience replay," in *Advances in neural information processing systems* (NIPS), 2017.

[2] M. Vecerik *et al.*, "Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards," *arXiv preprint arxiv:1707.08817*, 2017.

Empirical Evaluation

- Setup:
  - Agent receives starting positions and goals of all objects
  - Initialize demonstration buffer with 100 human demonstrations with VR
  - Actor and critic function approximators π and Q are deep neural networks
- First tested on pushing, sliding, and pick-and-place tasks and compare with previous work



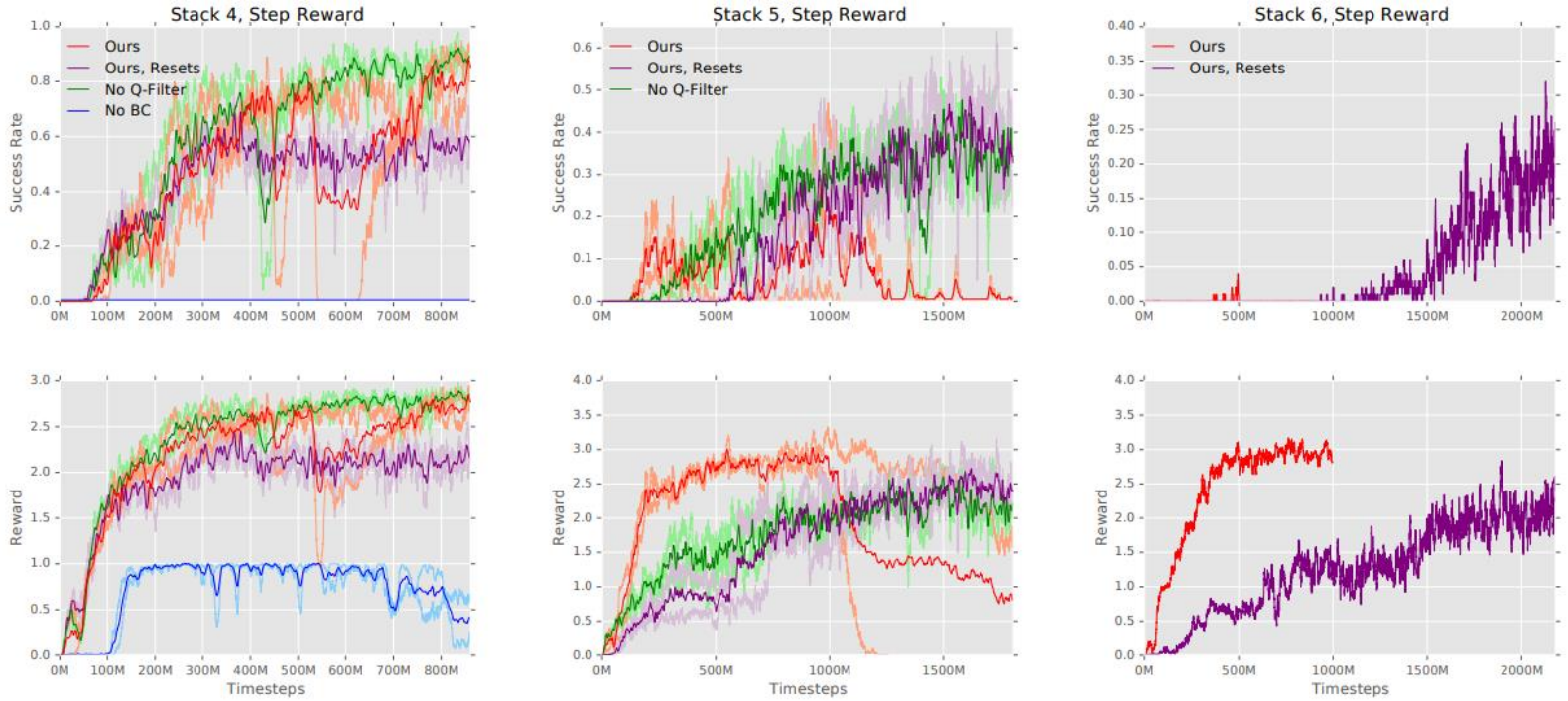


Block Stacking Task

- More interesting task with sparser rewards and longer horizon
- Blocks initialized at 6 random locations with one of those locations as the position of the tower
- Two reward functions
  - Fully sparse: only receives reward if all blocks are at goals
  - Step reward: receives reward whenever a block is moved to its goal position
- Compared method against baselines and ablations of own method
  - BC, HER, BC+HER
- Method shown to learn much longer horizon tasks better than baselines

Ablations of Own Method

- Strips away BC, Q-filter, and resets from demonstrations individually
- Studies effects and confirms necessity of each specific feature



Conclusion

- Main contribution: combining demonstrations with existing methods to guide exploration in complex multi-step tasks with sparse rewards
- Method is very general and not specific to robotic tasks
  - Can be applied to any continuous control task where demonstrations are possible
- Takeaway: demonstrations are invaluable in eliminating random exploration phase and speeding up learning significantly
- Future work: training policies directly on physical robots
  - Eliminate the simulation phase all together
  - Showed it feasible to train a physical robot with their method in a few hours