A Review of Generative Attention Learning

Jaime Canizales

City University of New York

jaime.canizales@hunter.cuny.edu

April 26, 2024

Overview

1. Introduction

2. Technical Information

3. Conclusions and possible extensions

Introduction

Problem Statement

How to solve the problem of **smartly** grasping(e.g., grabbing a pitcher by the handle) an object in a cluttered real world environment, using a high degree of freedom(DOF) robot hand(up to 6 degrees) not restricted to any particular camera view or grasping strategy to simplify the problem(e.g., top down grasps, where the robot hand acts as a claw machine and grabs the object from above).

Introduction(cont.)

Solution proposed in paper

Using a reinforcement learning framework, which takes a single depth image as input; we can find the end-effector position and orientation, as well as a set of finger joint angles in a timely manner(close to real time).

Why is this hard?

- As the DOF increases in a robot hand, the training data needed to find a reliable solutions increases exponentially.
- Real world problems must inherently deal with cartesian spaces as their solution spaces(Finding solutions is difficult in cartesian spaces, because the spaces are infinite, continuous and the number of possible solutions may be infinite).

Why is this hard?(cont.)

- Not restricted to only top-down grasping solutions(top-down grasping solutions decrease the complexity of the problem in two ways: End-effector position and orientation solution is simplified from $\mathbb{R}^6$ to $\mathbb{R}^4$, constraints are added to solution space decreasing complexity).
- slight perturbations of a good solution can lead to bad solutions.

Alleviating these complexities

- Uses a deep convolutional Reinforcement learning(RL) framework to solve the problem in reasonable time.
- Trained in simulation but used in the real world.
- A zooming mechanism is applied to objects to optimize good grasps.
- Action space, states, and RL policies are all learned and optimized in pixel space(smaller and discrete).

Technical Information

- Uses an infinite horizon Partially Observable Markov Decision Process(POMDP), because agent(robot) can't observe rgb info, and the complete 3D geometry of objects and scene.
- infinite horizon(no clear start or end state) implies the magnitude for the set of instructions to achieve goal is unknown $[\pi_1, \pi_2, \pi_3, \pi_4, ...]{=}\pi$ (policy).

POMDP

- Six tuple definition $<$S, A,ρ_0,R,T, $\gamma >$
- observation space S - states of the agent(robot) in the environment(observations).
- Action space A - set of actions that can be performed by the agent.
- ρ_0 initial state distribution $\rho_0 \in \Pi(S)$ - initial probability of being in any given state(randomly initialized).
- Dynamics model T: S $\times$ A $\rightarrow \Pi(S)$ - for all state-action pairs, return the probabilities of going to the next states.
- γ is discount factor $\in [0, 1]$ - used more as a hyper parameter to help convergence in infinite horizon POMDP.
- Reward function R: S $\times$ A $\rightarrow \mathbb{R}$ - Binary indicator function. 1 for successful grasp and 0 otherwise.

POMDP(cont.)

- The agent acts according to stationary stochastic policies(π does not change over time) π : S $\rightarrow$ Π(A), which specify action choice probabilities for each observation.
- Each policy π has a corresponding Q_π : S $\times$ A $\rightarrow$ $\mathbb{R}$, function that defines the expected discounted cumulative reward for taking action a from observation S and following the policy π from that point onward. $\sum_t \gamma^{t-1} r_t$

Policy Gradient

Policy Gradient

Policy gradient methods are a type of RL techniques that rely upon optimizing parameterized policies w.r.t. expected return(long term cumulative reward) by gradient descent.

Soft proximal policy optimization(SPPO)

The policy gradient method used in this paper, which follows the formula: $maximize_\theta\ L\ = \mathbb{E}_{\pi_\theta}[\pi_\theta(a_t|S_t)Q_{\pi_\theta}(S_t, a_t)]$ (θ is the network weights, t is the time step)

SPPO modifications

SPPO modified to use Clipped Surrogate Objective (Schulman et al. 2017) and apply a soft advantage target to balance between exploration and exploitation (Haarnoja et al. 2018).

Hyper Parameters

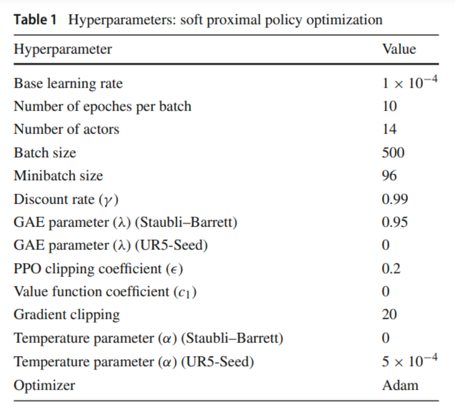


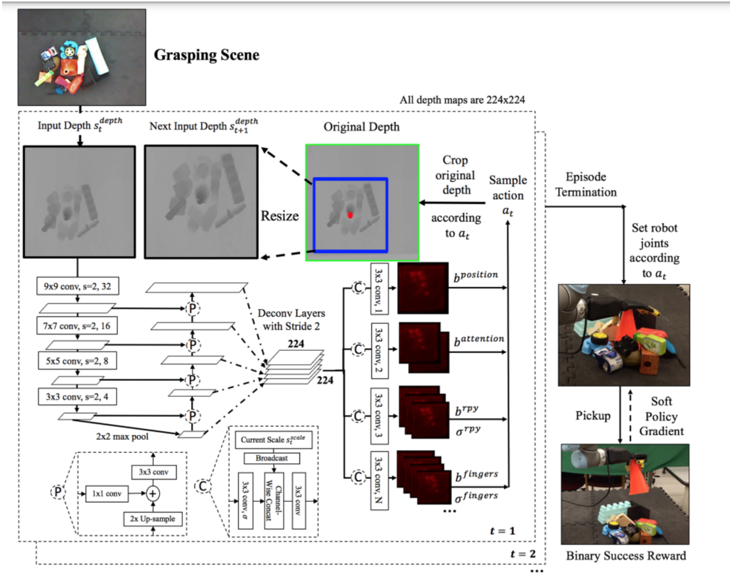
Table 1 Hyperparameters: soft proximal policy optimization

Hyperparameter	Value
Base learning rate	1×10^{-4}
Number of epoches per batch	10
Number of actors	14
Batch size	500
Minibatch size	96
Discount rate (γ)	0.99
GAE parameter (λ) (Staubli–Barrett)	0.95
GAE parameter (λ) (UR5-Seed)	0
PPO clipping coefficient (ϵ)	0.2
Value function coefficient (c_1)	0
Gradient clipping	20
Temperature parameter (α) (Staubli–Barrett)	0
Temperature parameter (α) (UR5-Seed)	5×10^{-4}
Optimizer	Adam

Learning attention

- Decide whether the robot should zoom into the depth map or start grasping.
- Determine the level of zooming(scale)(end-effector as center of zoomed image).

Architecture



Architecture(Cont.)

- A feature four branch pyramid branch CNN.
- The “P” blocks indicate feature pyramid blocks, giving scale invariance capability to the CNN.
- The “C” blocks indicate how GenerAL introduces the current zoom level into each branch.
- All convolutional layers have ReLU activations and strides of 1.
- Each red-black map proposes pixel-wise grasp configurations and their probabilities.

Training

- GenerAL is trained entirely in simulation.
- During training, a single seen object or a cluttered scene of multiple seen objects is loaded with equal probability.
- Random number of objects from 2 to 30 for a simulated cluttered scene.
- The number of training grasp attempts required to reach convergence range from 5000 to 15,000.
- 24 hour training time on one gpu, 13 virtual cpu machine.

Testing

- GenerAL is tested both in simulation and real-world. Using the ShapeNet Repository in simulation.
- 200 + seen objects from the YCB and KIT datasets.
- 100+ novel objects from the BigBIRD dataset.
- Evaluated 500 grasp attempts per experiment in simulation.

Performance

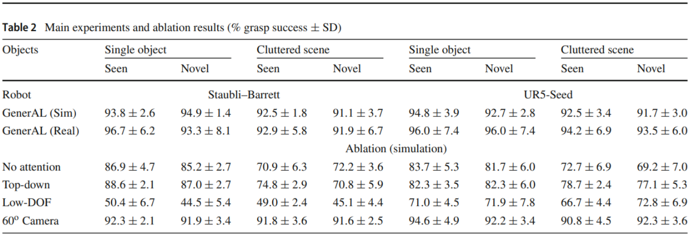


Table 2 Main experiments and ablation results (% grasp success ± SD)

Objects	Single object		Cluttered scene		Single object		Cluttered scene	
	Seen	Novel	Seen	Novel	Seen	Novel	Seen	Novel
Robot	Staubli–Barrett				UR5-Seed			
GenerAL (Sim)	93.8 ± 2.6	94.9 ± 1.4	92.5 ± 1.8	91.1 ± 3.7	94.8 ± 3.9	92.7 ± 2.8	92.5 ± 3.4	91.7 ± 3.0
GenerAL (Real)	96.7 ± 6.2	93.3 ± 8.1	92.9 ± 5.8	91.9 ± 6.7	96.0 ± 7.4	96.0 ± 7.4	94.2 ± 6.9	93.5 ± 6.0
	Ablation (simulation)							
No attention	86.9 ± 4.7	85.2 ± 2.7	70.9 ± 6.3	72.2 ± 3.6	83.7 ± 5.3	81.7 ± 6.0	72.7 ± 6.9	69.2 ± 7.0
Top-down	88.6 ± 2.1	87.0 ± 2.7	74.8 ± 2.9	70.8 ± 5.9	82.3 ± 3.5	82.3 ± 6.0	78.7 ± 2.4	77.1 ± 5.3
Low-DOF	50.4 ± 6.7	44.5 ± 5.4	49.0 ± 2.4	45.1 ± 4.4	71.0 ± 4.5	71.9 ± 7.8	66.7 ± 4.4	72.8 ± 6.9
60^{o} Camera	92.3 ± 2.1	91.9 ± 3.4	91.8 ± 3.6	91.6 ± 2.5	94.6 ± 4.9	92.2 ± 3.4	90.8 ± 4.5	92.3 ± 3.6

possible extensions

- By adding segmentation you can accomplish object specific grasping.
- Solve for collision avoidance.

References

Bohan Wu, Iretiayo Akinola, Abhi Gupta, Feng Xu, Jacob Varley, David Watkins-Valls, and Peter K. Allen (2020)

Generative Attention Learning: a “GenerAL” framework for high-performance multi-fingered grasping in clutter

© Springer Science+Business Media, LLC, part of Springer Nature 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018)

Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.

International conference on machine learning(ICML).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017).

Proximal policy optimization algorithms.

arXiv preprint