HYBRID DEEP REINFORCEMENT LEARNING WITH ROBOTIC GRASP PLANNING

Yefan Zhou, Xiangyu Zhou & Jerry Ge Department of Electrical Engineering and Computer Science
University of California, Berkeley
{yefan0726, josiath, yuzhouge}@berkeley.edu

ABSTRACT

Robotic grasping is a critical control task that involves in all areas of real-life application scenarios all the way from manufacturing sites to regular households. Due to the high practicality of robotic grasping task, the number of research projects and products has been increasing significantly during the recent years. With the emerging of data-driven machine learning techniques, more novel ideas and methods have been proposed. However, most of the robotic problems have to be solved in very complex environments which consists of both continuous and discrete decision variables. While most of the current work rely on supervised learning methods, e.g., GGCNN Morrison et al. (2018) or discretize continuous action space Zeng et al. (2018). In this work, we hypothesize and experimentally show that it is feasible to learn the robotic grasping task with hybrid action space through deep reinforcement learning techniques. We define gripper position as a discrete action variable and gripper rotation as a continuous action variable. We propose Hybrid Policy Gradient (**H-PG**) method using both discrete and continuous action variables which can achieve a noticeable better performance compared with our baseline methods: **Heuristic**, **GGCNN** and **Vanilla Policy Gradient** methods. You are welcomed to checkout our video demo of the project through the link at https://www.youtube.com/watch?v=YFV3_CLpEdk

1 INTRODUCTION

Robotic grasping is a critical control task that involves in all areas of real-life application scenarios all the way from manufacturing sites to regular households. Due to the high practicality of robotic grasping task, the number of research projects and products has been increasing significantly during recent years. With the emerging of data-driven machine learning techniques, more novel ideas and methods have been proposed.

There has been some work Zeng et al. (2018); Quillen et al. (2018); Kalashnikov et al. (2018) investigated the grasping task using supervised learning methods, e.g., GGCNN with either discrete or continuous action spaces. However, most problems in robotics need to be solved in very complex environments where the action space consists of both continuous and discrete decision variables. For example, the grasping pixel position is a discrete variable while the rotation of the gripper is a continuous variable.

As the result, this paper investigates the continuous-discrete hybrid set-up for robotic grasp planning tasks. The specific task studied here is planning a planar grasp in a cluster environment given a depth image, and it is modeled as a Deep Reinforcement Learning (DRL) problem with hybrid action spaces. This paper proposes a Hybrid Policy Gradient (H-PG) method to solve the problem set-up.

We evaluate the grasping success rate for the proposed Hybrid Policy Gradient (H-PG) method compared with multiple baseline methods: Grasp Heuristic Zeng et al. (2018) and Supervised Learning methods, e.g., GGCNN Morrison et al. (2018) and the vanilla Policy Gradient method. The evaluated results (single object grasp) showed the H-PG method outperforms all of the baseline methods with YCB daily objects, e.g., Hammer, Banana, Pear, Tennis Ball, etc. Calli et al. (2015) The results clearly show the proposed H-PG approach can be used for solving the grasp planning task with satisfying results in hybrid action spaces.

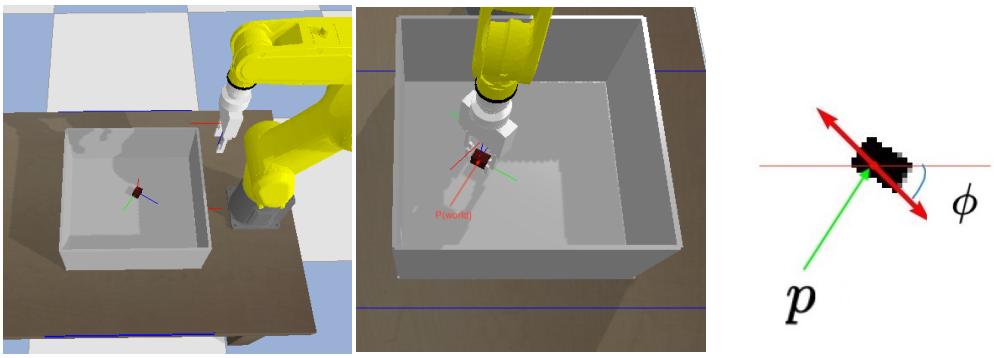
The rest of this paper is organized as follows. Related works are discussed in Section 2. The detailed problem formulation will be discussed in Section 3. The main methods will be proposed in Section 4. Section 5 will introduce the experiment details and results. In the end, we will have conclusions and further discussions in Section 6.

2 RELATED WORK

Grasping There have been many articles about Robotic grasping in the past 5 years. Basically, we can classify them into 2 kinds of methods: Analytic and Empirical. Using analytic methods, we tend not to get good results in real world due to the difficulty of modeling real world connections. Now, most of the researchers have changed their attention to empirical methods. Empirical methods are model-based or experience-based approaches.

Supervised learning Supervised learning is one kind of empirical methods. The procedure for supervised learning is almost the same. It will first learn grasp quality function and then rank the points with quality value. Finally, by choosing the optimal position, a robot in real world or simulator maps the position in a picture to the position in 3D world. This method relates closely to the precision of robot control and camera mapping. GG-CNN, inputs a depth map into the network and predicts quality and pose of grasps at every pixel, successfully reduces the computation time and avoids the discrete sampling of grasp candidates.

Hybrid RL Although there have been studies on hybrid DRL algorithms that mostly study computer games Xiong et al. (2018); Bester et al. (2019a); Fan et al. (2019); Neunert et al. (2020); Delalleau et al. (2019), few of them are verified in challenging robotic tasks, e.g., grasping in clusters. Besides, most works in grasping community Zeng et al. (2018); Quillen et al. (2018); Kalashnikov et al. (2018) commonly approximate hybrid action space as fully discrete or fully continuous space without considering a hybrid definition. This work Zeng & et al. (2018) transforms the continuous angle to 16 discontinuous fixed angles and gets a good result. Q-learning is normal in Reinforcement learning grasping for estimating the state-action qualities with iterative updates. We want to use Policy gradient on hybrid settings to see whether it may get a better result.



(a) (b) (c)

Figure 1: (a) Scenerio of grasping in a bin. (b) Grasp configuration in world coordinate. (c) Grasp configuration in image coordinate.

3 Problem Formulation

In this section, we introduce the notions of the grasping task, and our formulation of the task as a Markov decision process solved by deep RL method.

Grasp Representation Following the definition in the literature Johns et al. (2016), we consider the problem of detecting a grasp configuration on objects in a bin, given a depth image of scene in the bin. The scenario is shown in Figure 1a. Let the $I \in R^{H \times W}$ define a depth image with height H and W. As shown in Figure 1b and 1c, we define a three degree of freedom grasp configuration $g = (p, \phi)$, which is perpendicular to the x-y plane. $p = (u, v)$ is the gripper's centre position in image coordinates (pixels). ϕ is the gripper's rotation around the camera's z axis.

Bin Grasping as a MDP We formulate the grasping problem as a Markov decision process. At the beginning of a rollout, a number of N_{obj} objects are randomly distributed in a bin. At time t, given a depth image I as a state s_t capturing the poses of objects in the bin, the agent chooses an action a_t according to a policy π_{s_t}, $a_t \in \mathbf{A}$ and $\mathbf{A}$ is denoted as action space. The a_t is defined the same as the grasp configuration g. During the execution of a_t, if the gripper successfully grasps an object and stably lifts it up, the grasp is labeled as a success and an reward r_t is granted. The grasped object will be removed from the bin. After execution, the objects are manipulated by the gripper and another depth image is taken capturing the updated object poses as the next state s_{t+1}. The goal of RL problem is to find a optimal policy π^* that maximizes the expected sum of reward in given time length T given by $R_t = \sum_{i=t}^{T} r_i$. A rollout terminates when all the objects are grasped and removed, or the elapsed time step reaches the limit T. At each time step t, the robot only plans and executes one grasp a_t. In this work, we only study the problem of single object grasping, i.e., only one object is thrown to the bin in one rollout, $N_{obj} = 1$.

Action Space For action a_t at time t, We consider a discrete-continuous hybrid action space $\mathbf{A}$, $a_t \in \mathbf{A}$, $\mathbf{A} = (P, \Phi)$, gripper's centre position $p \in P$, gripper's rotation $\phi \in \Phi$. P is a discrete action space with dimension $H \times W$, which is the number of pixels in the depth input. Φ is a continuous action space, $\Phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Reward Calculation If a grasp is successful, a reward of 1 is granted, otherwise the reward is 0. A successful grasp is defined as the following. If the object's z coordinate is significantly above the table ground for a period of time, we consider the object is stably grasped and lifted up.

Exploration For our proposed method, we use an unusual exploration policy. The normal exploration strategy is random exploration. It just gets random action over the whole action space, which is suitable for small action space. The action space defined is extremely large, as the dimension of gripper position p is $H \times W$. If we use random policy, the sample efficiency is pretty low and takes a long training time. Thus, we use a Heuristic method as our exploration policy, which will be described in Section 5. The Heuristic policy will output about 30% successful grasp. The exploration rate starts from 1 and linearly decays to 0.05 after 30 % of total iterations, then it remains 0.05 until the end.

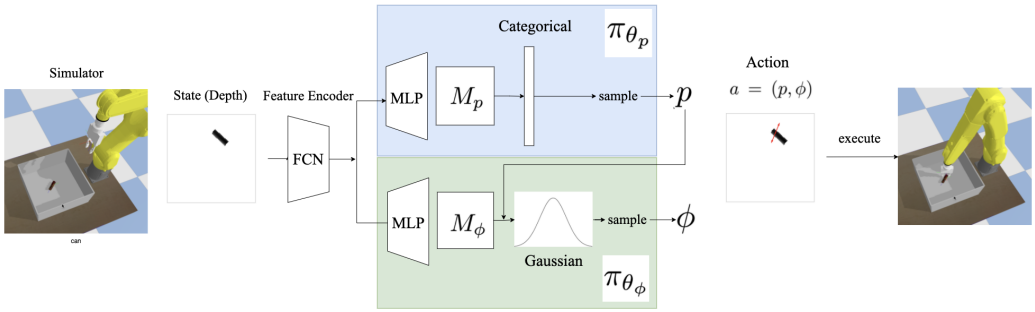


Figure 2: Overview of hybrid RL system. FCN: fully convolutional network. MLP: multi-layer perceptron.

4 METHODOLOGY

This section introduces the proposed hybrid RL system and the H-PG algorithm, and presents the network architecture of PG actors.

4.1 SYSTEM OVERVIEW

As stated in Section 3, we define a hybrid action space $\mathbf{A} = (P, \Phi)$, each action $a \in \mathbf{A}$ is comprised of two sub-actions. The proposed hybrid policy π_h predicts a hybrid action $a = (p, \phi)$ at each timestep, which is comprised of one discrete action gripper centre $p \in P$ and one continuous action gripper rotation $\phi \in \Phi$. The gripper position p is conditioned on the state s (input depth image). It means that we choose where to locate gripper's centre based on the local geometric information of depth image, e.g. a successful grasp is commonly located on the part of objects that could satisfy the force closure criterion Ferrari & Canny (1992). The gripper rotation ϕ is conditioned on the state s and gripper position p because when p changes, the local geometry could vary, ϕ should be adjusted. We formulate the policy as:

$$\pi(a \mid s) = \pi(p, \phi \mid s) \tag{1a}$$
$$= \pi(p \mid s) \cdot \pi(\phi \mid s, p) \tag{1b}$$
$$= \pi_{\theta_p}(p \mid s) \cdot \pi_{\theta_\phi}(\phi \mid s, p) \tag{1c}$$

Based on the formulation, we build our RL system as shown in Figure 2. Given a depth image I as state s_t at time t, the feature encoder encodes the s_t, and then condition on the feature embedding, the discrete policy π_{θ_p} uses actor network predicts a map $M_p \in R^{H \times W}$, which are totally $H \times W$ values for all the pixels, we form a discrete Categorical distribution based on the M_p, and the position p_t to take is random sampled by $\pi_{\theta_p}(p_t \mid s_t)$. The continuous policy π_{θ_ϕ} uses another actor network outputs a map $M_\phi \in R^{H \times W}$, we use the value of M_ϕ at pixel p_t as the mean to form a Gaussian distribution, the rotation ϕ_t to take is sampled by $\pi_{\theta_\phi}(\phi_t \mid s_t, p_t)$. Therefore, a grasp action $a_t = (p_t, \phi_t)$ is sampled from our hybrid policy $\pi_h = (\pi_{\theta_p}, \pi_{\theta_\phi})$.

4.2 HYBRID POLICY GRADIENT

The hybrid policy gradient optimization (H-PG) takes the hybrid actor architectures in Figure 2 and uses online policy gradient as the policy optimization method for both its discrete policy π_{θ_p} and continuous policy π_{θ_ϕ}. For each iteration of training, we collect new data samples in simulation. In the conventional policy gradient method, the policy update formula is given by

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(a_{i,t} \mid s_{i,t} \right) \right) A(s_{i,t}, a_{i,t}) \tag{2}$$

$$\theta \leftarrow \theta + \nabla_\theta J(\theta) \tag{3}$$

In H-PG, we update the hybrid policy as

$$\begin{aligned}\nabla_\theta J(\theta_p) &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_{\theta_p}\log\pi_{\theta_p}\left(p_{i,t}\mid s_{i,t}\right)\right)A(s_{i,t},a_{i,t})\\ \nabla_\theta J(\theta_\phi) &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_{\theta_\phi}\log\pi_{\theta_\phi}\left(\phi_{i,t}\mid s_{i,t},p_{i,t}\right)\right)A(s_{i,t},a_{i,t})\end{aligned} \tag{4}$$

$$\begin{aligned}\theta_p &\leftarrow \theta_p + \alpha\nabla_\theta J(\theta_p)\\ \theta_\phi &\leftarrow \theta_\phi + \alpha\nabla_\theta J(\theta_\phi)\end{aligned} \tag{5}$$

For the estimation of advantages $A(s_t, a_t)$, we do not use a critic network and only use sum of reward in a rollout, we use the reward-to-go techniques to minimize the variance.

$$A(s_t,a_t) = \sum_{t=t'}^{T} r\left(s_{t'},a_{t'}\right) \tag{6}$$

H-PG has a unique training process designed to further reduce variance and stabilize policy updates. Suppose we train the policy for 1500 iterations, for the first 150 iterations, we use the advantage defined in Eqn. 6. Then for the following iterations, we freeze the feature encoder network, and only update two actor networks. We reduce variance by standardizing the advantage as

$$A(\tau) = r\left(\tau\right) - \frac{1}{N}\sum_{n=1}^{N} r(\tau) \tag{7}$$

To show the superiority of the training process in H-PG , we propose a baseline method called Vanila-PG, in which we use advantages of Eqn. 6 for the whole training process. Another scratch is to use standardized advantages at the beginning of training. However, we find that training from scratch with advantage standardization makes the training fail in the early stage, shown as the evaluation average return remains to be unchanged. We analyze the underlying causes of this issue here. The policy estimation in H-PG $\pi(p|s)$ derived in 1b may not be accurate, a more reasonable formulation is to predict p by $\pi(p|s,\phi)$, which means predicting position conditioning on state and rotation. That is because a failed action could be caused by an incorrect ϕ when the p is correct, but using standardized advantage of Eqn. 7 will also minimize $\pi(p|s)$ instead of minimizing $\pi(p|s,\phi)$. In Section 5.3, we show that using the proposed training process can well resolve this issue. Vanila-PG is also able to tackle this issue in a "supervised-like" way, because the unstandardized advantage of Eqn. 6 is 0 when an action (p,ϕ) fails, then the gradient $\nabla_\theta J(\theta_p)$ calculated based on this sample is 0, the actor won't be updated. However, the evaluation performance of Vanila-PG is prone to fluctuate in the middle of training, when the exploration rate drops a lot. Note that the heuristic-based exploration used in our work gives the most positive demonstrations to agent.

4.3 Network Architecture

The feature encoder network and actor networks use a fully convolutional topology, similar to Satish et al. (2019). The encoder includes four downsampling layers, two dilated layers and two upsampling layers. Downsampling layers use kernel size of [11, 5, 5, 5], input feature dimensions of [1, 64, 64, 128] respectively, activated by ReLU and max-pooling. Two dilated layers apply [5, 5] kernels with dilation [2, 4] and feature dimensions of [128, 256]. Upsampling layers employ transpose convolutional kernels with size 3 and striding 2, feature dimensions of [256, 128]. The discrete and continuous actor network has identical architecture, including three fully connected linear layers activated by ReLU, the output feature dimensions of both networks are 1.

5 Experiments & Analyses

5.1 Experimental setup

Simulator In order to perform and simulate our grasping task, we use the PyBullet simulator with Fanuc LR Mate series 6DOF robotic arms. For the object to be grasped, we use the YCB Object Set

Calli et al. (2015) which consists of objects from daily life with different physical properties. Since some of the objects of the YCB Object Set are too large for LR Mate to grasp, we only include a subset of objects including Banana, Hammer, Clamp, Pear, Power Drill, Scissors, Strawberry and Tennis Ball. The grasping area is defined to be within the bin with a 100mm offset on the 4 sides to avoid the gripper hitting on the bin edges. The camera was put 625mm above the desktop and could get a depth image of size 128×128. See Figure 1. In this project, we only study the single object grasp case.

Platform All network computations were performed on the same PC running Ubuntu 18.04.6 with a 3.60GHz Intel Core i7-6850K CPU and NVIDIA GeForce 1080 graphics card. On this platform, it takes about 3s to compute for 16 depth images.

Environment Setting Our experiment is single-object grasping. There will only be 1 object in the box and no matter success or failure,it will reset the world for the next grasp. We evaluate with 4 different methods: Vanilla-PG, H-PG, GGCNN, Heuristic. We can take GGCNN and Heuristic methods as our baseline. For each method, we evaluate with 2 environments with different numbers of total objects: $n = 50$ and $n = 100$. For each environment we have five different seeds, the final evaluation result is the average of the results on 5 different seeds.

Evaluation metric
In the training progress, we evaluate the performance of our methods by defining the Average evaluation reward as:

$$\frac{\textit{total reward}}{\textit{total number of all grasps}}$$

When we compare our proposed methods with our baseline methods, we evaluate the performance of our methods by defining the single object grasp success rate as:

$$\frac{\textit{total number of success grasps}}{\textit{total number of all grasps}}$$

These two methods are essentially the same. Because in the training progress, we give the successful grasp with the reward 1, which means the total reward equals the total number of success grasps. So Average evaluation reward is the same as single object grasp success rate for single grasp. We evaluate our models with multiple simulator environment seeds and then calculate the average success rate among all the seeds.

5.2 Baseline Methods

Heuristic We define the Heuristic policy as follows: for each depth image captured, we rotated the image in 16 different angles. For each of the 16 rotated images, we shifted the image both vertically and horizontally by a certain amount. We found the heuristically best action grasping location where the original image differs the most from the rotated and shifted image.
GGCNN We use a convolutional Neural Network with 4 convolutional layers and 2 more dilated convolutional layers. Our data set was built with 7000 successful grasp collected by Heuristic method. 80% of the data set is training set and 20% is evaluation set.
Vanilla-PG Vanilla-PG shares the same network as our proposed method H-PG, but differs in ways of calculating reward, loss and update network.Vanilla-PG only uses positive rewards, which means it will not learn anything through the failure grasp.

5.3 Main Results

Baseline Comparisons We first compare H-PG to three baselines in terms of grasp success rate in single object grasping experiments. See Table 1, we show that among all the methods, H-PG achieves the highest average success rate, and the lowest standard deviation. The superior performance is consistent in settings of n = 50 and n = 100. The results support our hypothesis that the formulation of grasping as a hybrid RL problem is feasible to solve even using the basic optimization method policy gradient. More surprisingly, H-PG outperforms the important GGCNN baseline Morrison et al. (2018) by 7.4%.

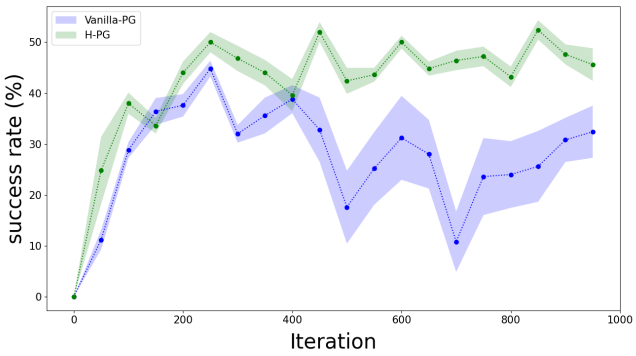


Figure 3: Success rate of Vanilla-PG and H-PG

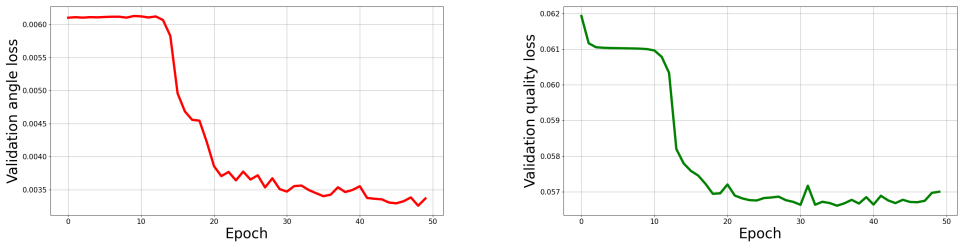
To seek the reason behind the phenomenon, H-PG can learn from both the success and failure grasp while GGCNN and Vanilla-PG only learn from success grasp samples. Learning from failure case means that H-PG minimizes $\pi(p|s)$ and $\pi(\phi|p,s)$ when the action (grasp) $a = (p, \phi)$ fails, but for other two baseline methods, the networks are not updated from failure cases. Therefore, H-PG is proven to be more sample efficient, and has better generalization ability on grasping scenarios.

	n=50	n=100
Heuristic	36.40 ± 7.12	36.67 ± 8.49
GGCNN (Morrison et al. (2018))	37.60 ± 5.90	39.40 ± 5.03
Vanilla-PG	38.80 ± 5.31	31.40 ± 2.94
H-PG	40.40 ± 4.27	40.80 ± 4.53

Table 1: Evaluation (mean $\pm$ std) of models with multiple random seeds and various number of grasp trials, Metrics are the average success rate for single grasps.

Training Dynamics As we can see from Figure 3,compared with Vanilla-PG, H-PG has less std and has a higher success rate. The disadvantage of Vanilla-PG mainly lies in the lack of the ability to learn from unsuccessful grasps. The loss is calculated as log probability times advantage. If all advantages are 1 or 0, it will only get the loss from the grasps with reward 1(successful grasp). But in H-PG, in the first 150 iterations, it is the same as Vanilla-PG.After 150 iterations, H-PG standardizes the advantage and stabilizes the feature network. From this time, H-PG's position and angle network can learn from unsuccessful grasps since some of the standardized advantages are negative.

We also show the loss of GGCNN Training in Figure 4.We measure the loss for every epoch and find that the loss reduces rapidly at about 10-15 epoch. Then it almost does not change.



(a) (b)

Figure 4: The validation loss of GGCNN network (a) Validation loss of GGCNN angle network (b) validation loss of GGCNN quality network

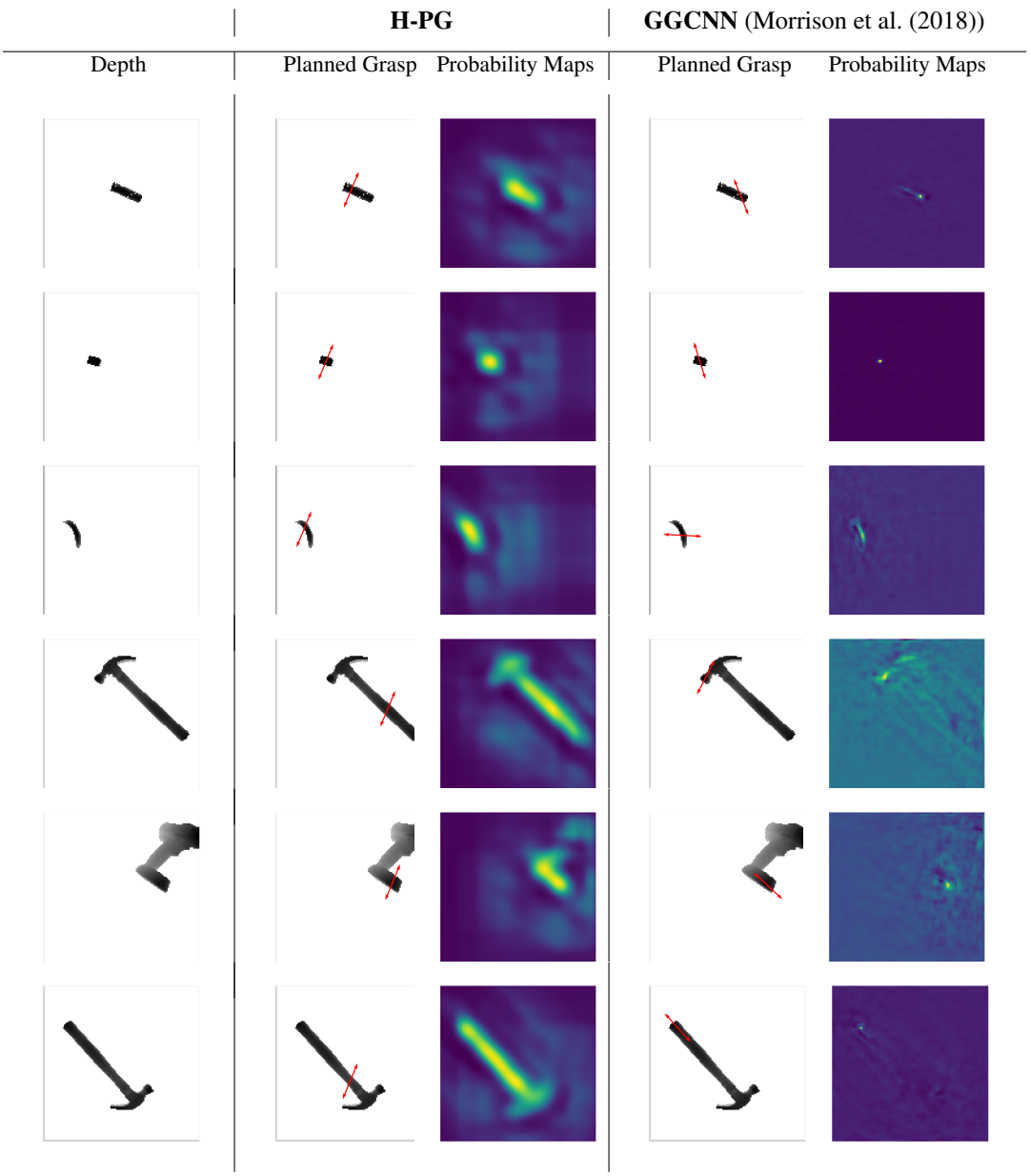


Figure 5: Comparing H-PG with supervised baseline GGCNN by visualizing the grasp annotation and predicted probability. The column titled with "Depth" shows the input depth image, the column titled with "Planned Grasp" shows the pose of parallel gripper when executing the grasp, and the column titled with "Probability Maps" shows the probability maps predicted by the two methods, the gripper position p is sampled based on the probability maps. The brighter color means the higher probability, while darker means smaller probability.

Qualitative Results We show qualitative results to compare H-PG with the baseline method GGCNN. See Figure 5. First, given the same depth image shown by the two columns titled with "Planned Grasp" under H-PG and GGCNN. H-PG predicts more successful antipodal grasps, shown as the red lines (pose of parallel gripper) are mostly perpendicular to flat plane structure. In particular, H-PG outperforms GGCNN in predicting grasps for large objects with less graspable parts like hammer and power drill (last three rows). For these hard objects, both two methods have a tendency to predict grasp points on the handles, but H-PG predicts better rotations to avoid the collision and satisfy the force closure. Second, the probability maps is useful to analyze the underlying mecha-

nism of deep grasping models. We can see that H-PG considers the most of areas of objects have high probability to be execute as success grasps, shown as bulk of areas around objects with brighter colors. It is reasonable for human beings because we can grasp an object from most of its parts by adjusting hand pose (planar rotation). However, for GGCNN, the graspable areas are pretty sparse. That might cause from the fact that the training dataset is sparsely labeled and only one successful grasp is labeled for one depth image. Therefore, we see the limitation of supervised learning, as the its accuracy and robustness would vary depending on whether the training data is densely labeled or not. RL-based method like H-PG is able to resolve this issue, and achieves stable performance invariant to data quality.

6 Conclusion and Discussion

This paper proposes a Hybrid Policy Gradient (H-PG) method to solve the robotic grasping tasks with daily objects. The proposed method can achieve a 41% success rate of grasping daily objects which shows a noticeable performance increase compared with the baseline models: Heuristic and Supervised Learning Method, e.g., GGCNN and the vanilla Policy Gradient method.

For future works, we plan to train the policy to maximize the Q value as Deep Deterministic Policy Gradient(DDPG) or other variants: Multi-Pass Deep Q-Networks(MP-DQN) Bester et al. (2019b) or a Hybrid Maximum a Posteriori Policy Optimisation (MPO) Abdolmaleki et al. (2018)

Team Member Contributions

Yefan Zhou mainly contributed to the Hybrid Policy Gradient (H-PG) part. Xiangyu Zhou mainly contributed to the vanilla Policy Gradient sections. Jerry Ge is mainly responsible for the Heuristic part and the Supervised Learning (GGCNN) parts. All team members share equal contributions.

Acknowledgments

We thank Prof. Levine from UC Berkeley for suggestions on motivation of ideas. We thank Xinyu Chen from UC Berkeley for all the guidance and instructions in the course of the final project. Finally, we thank Xinghao Zhu from UC Berkeley for the mentorship and technical support during the course of this project.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation, 2018.

Craig J Bester, Steven D James, and George D Konidaris. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *arXiv preprint arXiv:1905.04388*, 2019a.

Craig J. Bester, Steven D. James, and George D. Konidaris. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces, 2019b.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics Automation Magazine*, 22(3):36–52, 2015. doi: 10.1109/MRA.2015.2448951.

Olivier Delalleau, Maxim Peter, Eloi Alonso, and Adrien Logut. Discrete and continuous action representation for practical rl in video games. *arXiv preprint arXiv:1912.11077*, 2019.

Zhou Fan, Rui Su, Weinan Zhang, and Yong Yu. Hybrid actor-critic reinforcement learning in parameterized action space. *arXiv preprint arXiv:1903.01344*, 2019.

Carlo Ferrari and John F. Canny. Planning optimal grasps. *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 2290–2295 vol.3, 1992.

Edward Johns, Stefan Leutenegger, and Andrew J. Davison. Deep learning a grasp function for grasping under gripper pose uncertainty. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4461–4468, 2016. doi: 10.1109/IROS.2016.7759657.

Dmitry Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, E. Holly, Mrinal Kalakrishnan, V. Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *ArXiv*, abs/1806.10293, 2018.

Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. *Robotics: Science and Systems (RSS)*, 2018.

Michael Neunert, Abbas Abdolmaleki, Markus Wulfmeier, Thomas Lampe, Tobias Springenberg, Roland Hafner, Francesco Romano, Jonas Buchli, Nicolas Heess, and Martin Riedmiller. Continuous-discrete reinforcement learning for hybrid control in robotics. In *Conference on Robot Learning*, pp. 735–751. PMLR, 2020.

Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6284–6291. IEEE, 2018.

Vishal Satish, Jeffrey Mahler, and Ken Goldberg. On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks. *IEEE Robotics and Automation Letters*, 2019.

Jiechao Xiong, Qing Wang, Zhuoran Yang, Peng Sun, Lei Han, Yang Zheng, Haobo Fu, Tong Zhang, Ji Liu, and Han Liu. Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space. *arXiv preprint arXiv:1810.06394*, 2018.

Andy Zeng and et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 ICRA*, pp. 3750–3757, 2018. doi: 10.1109/ICRA.2018.8461044.

Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *IROS*, 2018.