Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences

Jeffrey Mahler
EECS Department
UC Berkeley
jmahler@berkeley.edu

Ken Goldberg
EECS and IEOR Department
UC Berkeley
goldberg@berkeley.edu

Abstract: Recent results suggest that it is possible to grasp a variety of singulated objects with high precision using Convolutional Neural Networks (CNNs) trained on synthetic data. This paper considers the task of bin picking, where multiple objects are randomly arranged in a heap and the objective is to sequentially grasp and transport each into a packing box. We model bin picking with a discrete-time Partially Observable Markov Decision Process that specifies states of the heap, point cloud observations, and rewards. We collect synthetic demonstrations of bin picking from an algorithmic supervisor uses full state information to optimize for the most robust collision-free grasp in a forward simulator based on pybullet to model dynamic object-object interactions and robust wrench space analysis from the Dexterity Network (Dex-Net) to model quasi-static contact between the gripper and object. We learn a policy by fine-tuning a Grasp Quality CNN on Dex-Net 2.1 to classify the supervisor's actions from a dataset of 10,000 rollouts of the supervisor in the simulator with noise injection. In 2,192 physical trials of bin picking with an ABB YuMi on a dataset of 50 novel objects, we find that the resulting policies can achieve 94% success rate and 96% average precision (very few false positives) on heaps of 5-10 objects and can clear heaps of 10 objects in under three minutes. Datasets, experiments, and supplemental material are available at http://berkeleyautomation.github.io/dex-net.

Keywords: Grasping, Imitation Learning, Simulation

1 Introduction

Robots with parallel-jaw grippers can lift and transport a wide variety of rigid objects using deep learning when objects are singulated (sufficiently clear from obstacles) [1, 2, 3, 4]. However, objects are often in disorganized heaps in applications such as industrial bin picking, which is challenging due to sensor noise, obstructions, and occlusions that make it difficult to infer object shapes and poses from point clouds [5]. Furthermore, a robot must consider collisions with adjacent objects and cannot assume a finite set of stable resting poses for each object [6].

Recent research suggests that it is possible to grasp a diverse set of objects from clutter using deep Convolutional Neural Networks (CNNs) trained on large datasets of grasp attempts on a physical robot [7, 8]. However, the time cost of collecting physical data makes it difficult to collect clean and sufficiently large datasets to train different robots in different environments. An alternative is to train on synthetic datasets of grasps and point clouds labeled using geometric conditions related to grasp stability such as antipodality [9], but these methods typically require multiple viewpoints of the scene to be robust to sensor noise [10, 11].

In this paper, we consider explicitly modeling uncertainty during dataset generation in order to learn a robust policy for rapid bin picking from a single viewpoint. We formulate a discrete-time Partially Observed Markov Decision Process (POMDP) modeling bin picking as a sequence of 3D object poses in a heap with noisy point cloud observations and rewards for removing objects. Due to the difficulty of training POMDPs with continuous states and observations [12], we use imitation learning based on an algorithmic supervisor that synthesizes robust collision-free grasps using robust wrench space analysis and full knowledge of object shapes and poses in the environment [13].

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

This paper makes four contributions:

1. Formulating bin picking as a Partially Observable Markov Decision Process (POMDP) modeling the process of iteratively grasping and removing objects from a heap based on point clouds. This extends the single-object nonsequential robust grasping model from Dex-Net 2.0.
2. Dex-Net 2.1: A dataset of 10,000 rollouts in an implementation of the POMDP collected using noise injection on an algorithmic robust grasping supervisor that plans robust grasps with full state knowledge.
3. A study of transfer learning to learn a bin picking policy from pre-trained weights of a GQ-CNN policy for grasping singulated objects.
4. Experiments evaluating performance of the bin picking policies on heaps of up to 20 novel objects on an ABB YuMi robot.

Experiments suggest that a bin picking policy trained synthetic data from Dex-Net 2.1 can achieve up to 416 successful picks per hour with 96% average precision (very few false positives).

2 Related Work

Bin picking is a common task in material transfer automation [14]. The goal is to clear a pile of objects from an area by iteratively grasping objects and placing them in containers. A key subproblem is grasp synthesis, which considers the problem of finding a configuration of a robot gripper that can generate desired wrenches (forces and torques) and resist disturbing wrenches on an object through contact [15].

Many industrial bin picking robots pre-plan a set of grasps for known object CAD models and use a perception system to accurately estimate the shape and pose of objects to look up a grasp to execute [14, 16]. However, instance recognition and pose estimation in clutter is difficult due to sensor noise and occlusions. Research has focused on detecting 2D features and matching them to known 3D object geometries using methods such as Chamfer matching [17]. Approaches for unknown objects include finding antipodal grasps on object segementation masks [18] and using filter banks based on 2D projections of the gripper geometry to find collision-free grasps [5].

The difficulty of state estimation motivates methods the use of machine learning to plan grasps for cluttered environments directly from observations. One approach is to learn a classifier for successful grasps from 2D image features [19] and plan the grasp with the highest predicted probability of success. Recent research suggests that deep neural networks trained on large datasets of human labels [20, 2] or physical trials [8, 4] can be used to to predict grasp success directly from images or point clouds. However, data collection for these methods may take up to several months.

Recent research has proposed hybrid methods that use machine learning to classify grasps that satisfy geometric conditions such as antipodality from point clouds [9]. Gualtieri et al. [10] and Viereck et al. [11] used CNNs trained to predict antipodal grasps on dense 3D point clouds of a scene from synthetic datasets. These methods typically require multiple viewpoints of the scene to be robust to sensor noise, as antipodality has been observed to be sensitive to sensor noise [15, 21]. We learn a single-viewpoint bin-picking policy by explicity modeling sensor noise following the robust single-object grasping model of Mahler et al. [3] and extending it to model sequential grasping of objects from heaps.

3 Definitions and Problem Statement

We consider the problem of bin picking: clearing a heap of objects on a table by iteratively grasping a single object from the heap with a parallel-jaw gripper and transporting each object to a receptacle. Our goal is to learn a policy that takes as input point clouds from an overhead depth camera and outputs a robust grasp, or gripper pose to remove an object from the heap, along with a confidence value for the grasp.

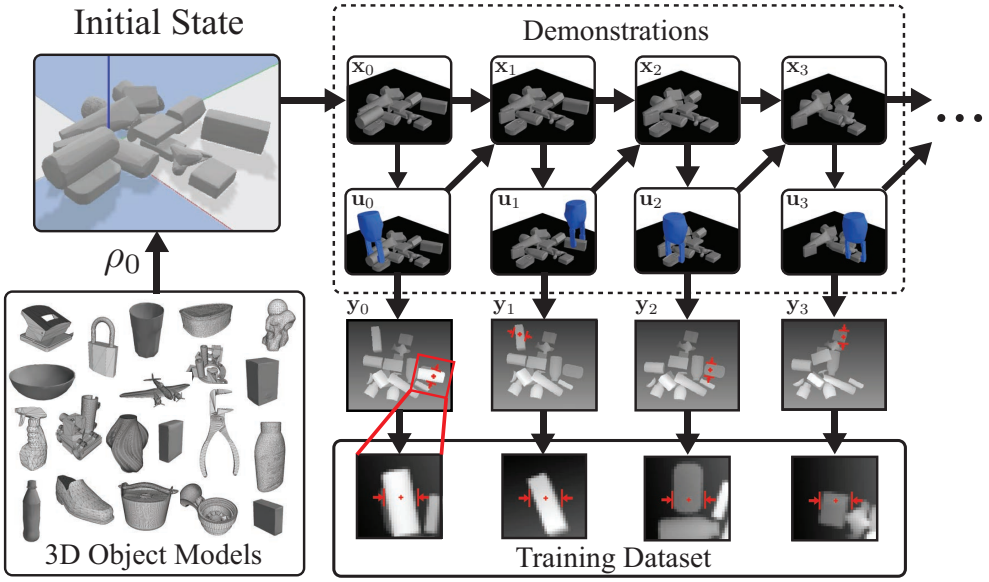

Figure 1: Overview of our POMDP model and simulator. We sample from the initial state distribution ρ_0 by uniformly sampling m 3D CAD object models from a dataset and dropping them in random poses in the pybullet dynamic simulator [23] to form a heap. The state $\mathbf{x}_t$ includes object shapes and poses in the heap. We generate demonstrations of robot grasping using an algorithmic supervisor π^* from Dex-Net 2.0 [3] that indexes the most robust collision-free parallel-jaw grasp $\mathbf{u}_t$ from a pre-planned grasp database using knowledge of the full state. We aggregate synthetic point cloud observations $\mathbf{y}_t$ and collected rewards R_t to form a labeled dataset for training a policy that classifies the supversiors actions on the partial observations using imitation learning. We preprocess training data by transforming the point clouds to align the grasp center and axis with the center pixel and middle row to improve GQ-CNN classification performance [2, 3].

3.1 Assumptions

Our model assumes quasi-static physics, where inertial effects are negligible, to compute grasp robustness. Our model also assumes a parallel-jaw gripper, rigid objects, a depth sensor with bounds on the camera intrinsic parameters and pose relative to the robot, and bounds on friction across objects and their surfaces. These assumptions are common in industrial robotics [14]. We make the additional simplifying assumption that only one object is be grasped at a time. Our model also does not consider object identity when grasping.

3.2 POMDP Model

Due to the cost of learning a policy directly from data on a physical robot, we learn a policy in simulation using a model of iteratively grasping objects from a heap on an infinite planar worksurface based on models of quasi-static contact, image formation, and sensor noise. Specifically, we model the task of bin picking as a Partially Observable Markov Decision Process (POMDP) (see Fig. 1) specifed as a tuple $(\mathcal{X}, \mathcal{U}, \mathcal{Y}, R, \rho_0, p, q)$ consisting of a set of states $\mathcal{X}$ (object shapes and poses), a set of actions $\mathcal{U}$ (gripper poses), a set of observations $\mathcal{Y}$ (point clouds), a reward function R, an initial state distribution ρ_0 (object heaps), a next state distribution p, and a sensor noise distribution q [22]. Our POMDP uses a fixed maximum time horizon T. See the supplementary file [1] for numeric values of parameters for each distribution.

Initial State Distribution (ρ_0). The initial state distribution ρ_0 models the position and shape of objects in a heap as well as the parameters of the camera and friction which stay constant over an episode. We model ρ_0 as the product of independent distributions on:

1. *Object Count (m):* Poisson distribution with mean λ.
2. *Object Heap ($\mathcal{O}$):* Uniform distribution over a discrete set of m 3D triangular meshes $\{\mathcal{M}_0, ...\mathcal{M}_{m-1}\}$ and the pose from which each mesh is dropped into the heap.

[1] http://berkeleyautomation.github.io/dex-net

3. *Depth Camera ($\mathcal{C}$):* Uniform distribution over the camera pose and intrinsic parameters.

4. *Coulomb Friction (α):* Truncated Gaussian constrained to $[0, 1]$.

The initial state is sampled by (1) sampling an object count m and a set of m 3D CAD models, (2) sampling a planar pose for the heap center and planar pose offsets from the pile center for each of the objects, and (3) dropping the objects one by one from a fixed height h_0 above the table and running dynamic simulation until all objects come to rest (all velocities are zero). Any objects that roll beyond a distance W from the world center are removed.

States ($\mathcal{X}$). The state $\mathbf{x}_t$ at time t consists of the current set of 3D object meshes $\mathcal{O}_t$ and their poses.

Actions ($\mathcal{U}$). The robot can attempt to grasp and remove an object from the environment by executing an action $\mathbf{u}_t$ specified as a 4-DOF gripper pose $(\mathbf{p}, \theta, d)$ where $\mathbf{p}$ is the grasp center pixel, θ is the orientation of the gripper in image space, and d is the grasp depth, or distance of the 3D grasp center from the image plane [3]. The action is related to a grasp center in 3D space $\mathbf{c}$ by the formula $\mathbf{c} = (1/d)K^{-1}(\mathbf{p}_x, \mathbf{p}_y, 1)^T$ [24]. The robot executes an action by moving to the target 3D gripper pose along a linear approach trajectory, closing the jaws with constant force, and lifting upwards.

Observations ($\mathcal{Y}$). The robot observes a point cloud $\mathbf{y}_t$ specified as real-valued $H \times W \times 3$ matrix representing a set of 3D points imaged with a depth camera with $H \times W$ resolution.

Rewards (R). Binary rewards occur on transitions that remove a single object from the heap. Let $m_t = |\mathcal{O}_t|$ be the number of objects remaining in the heap. Then $R(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) = 1$ if $m_{t+1} < m_t$.

Next State Distribution (p). We use mechanical wrench space analysis to determine whether or not an object can be lifted from the heap under quasi-static conditions [13, 9], and we use multibody dynamic simulation with a velocity-based complementarity formulation implemented in pybullet [23] to determine the next state of the objects after an object is lifted.

Let $\mathcal{M}_i \in \mathbf{x}_t$ be the first object to be contacted by the gripper jaws when executing action $\mathbf{u}_t$. Then we measure grasp sucesss with a binary-valued metric $S(\mathbf{x}_t, \mathbf{u}_t) \in \{0, 1\}$ that measures whether or not $\mathbf{u}_t$ is collision-free and can resist external wrenches on object $\mathcal{M}_i$ under state perturbations using point contact models [21]. Specifically, $S = 1$ if the robust epsilon metric is greater than a threshold $\delta = 0.002$ [3] and the gripper does not collide with the table or object along a linear approach trajectory [25, 21]. If $S(\mathbf{x}_t, \mathbf{u}_t) = 1$, then $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, i)$ where f returns the set of object meshes and poses resulting from a dynamic simulation of object heap $\mathcal{O}$ as object $\mathcal{M}_i$ is lifted until the remaining objects come to rest. Otherwise the state remains unchanged. If an object rolls beyond a distance W from the world center then it is re-dropped in the pile.

Observation Distribution (q). We model depth-proportional noise with a Gamma distribution modeling depth-proportional noise due to errors in disparity computation and a Gaussian Process to model correlated zero-mean noise in pixel space [3, 26].

3.3 Policy

A policy maps point cloud observations to actions $\pi_\theta(\mathbf{y}_t) = \mathbf{u}_t$, where policies are parametrized by a vector of neural network weights θ. We consider policies of the form $\pi_\theta(\mathbf{y}) = \text{argmax}_{\mathbf{u} \in \mathcal{C}} Q_\theta(\mathbf{u}, \mathbf{y})$ where $\mathcal{C}$ specifies constraints on the set of available grasps such as collisions and $Q_\theta(\mathbf{u}, \mathbf{y}) \in [0, 1]$ is a scoring function for actions given observations [1, 8, 3, 4]. A policy induces a distribution over trajectories given the initial state, next state, and perceptual distributions of the POMDP: $p(\tau \mid \theta) = \rho_0 \prod_{t=0}^{T-1} p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \pi_\theta(\mathbf{y}_t)) q(\mathbf{y}_t \mid \mathbf{x}_t)$, where τ is a trajectory of length $T + 1$ defined as a vector of states, actions, and observations: $\tau = (\mathbf{x}_0, \mathbf{u}_0, \mathbf{y}_0, ...\mathbf{x}_T, \mathbf{u}_T, \mathbf{y}_T)$.

3.4 Objective

The objective is to learn a policy π_θ that maximizes the sum of undiscounted rewards:

$$\theta^* = \underset{\theta \in \Theta}{\text{argmax}}\, \mathbb{E}_{p(\tau|\theta)} \left[\sum_{t=0}^{T-1} R(\mathbf{x}_t, \pi_\theta(\mathbf{y}_t), \mathbf{x}_{t+1}) \right].$$

In our POMDP, this corresponds to maximizing the number of objects removed from the heap.

4 Imitation Learning from an Algorithmic Supervisor

Optimal solutions to POMDPs are known to be computationally intractable [27]. Many approximate solution methods exist, however many assume closed-form dynamics [28], discrete or low-dimensional state spaces [12], or Gaussian distributions [29].

We explore the use of imitation learning (IL) [30] to learn the actions of an algorithmic supervisor that computes actions based on full state knowledge from the simulator. IL has been used to approximately solve POMDPs when there exists an algorithmic supervisor with access to full state information [31]. We first compute an algorithmic supervisor using the singulated object robust grasp planner from Dex-Net 2.0 [3] which computes grasps using mechanical wrench space analysis given known object shape and pose. We then collect a dataset of point clouds, actions, and rewards by rolling out the algorithmic supervisor with noise injection [32] to balance the distribution of positive and negative examples. Finally, we learn a CNN to classify the supervisor's actions and use the trained CNN as action scoring function [33, 34].

4.1 Algorithmic Robust Grasping Supervisor

The algorithmic supervisor π^* precomputes a set of robust grasps for each 3D object in a dataset using full state knowledge. For computational efficiency, the supervisor is implemented by precomputing a database of robust grasps (such as Dex-Net) for each 3D object by evaluating the grasp success metric $S(\mathbf{u}, \mathbf{x})$ using Monte-Carlo sampling [3]. Given a state $\mathbf{x}_t$, π^* plans a robust grasp by pruning the set of possible actions for every object in the heap using collision checking and returning an action uniformly at random from the remaining set of robust grasps, if one exists. We note that π^* maximizes reward for the only current timestep and may not be optimal for the full time horizon.

4.2 Learning a Bin Picking Policy

We learn a bin picking policy by learning a classifier for the supervisor's actions [33, 34] on rollouts of π^* with noise injection [32]. Noise injection balances the distribution of positive and negative examples for the classifier, as rolling out the algorithmic supervisor results in all positive examples.

Our policy learning algorithm consists of two steps: (1) collect demonstrations by executing π^* with probabliity ϵ and a random action from the grasp database with probability $1 - \epsilon$, and (2) use supervised learning to classify actions taken by the supervisor with a Grasp Quality CNN (GQ-CNN). Specifically, given K demonstrations from the noise-injected supervisor we optimize:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{j=1}^{K} \sum_{t=1}^{T} \mathcal{L}\left(Q_\theta(\mathbf{u}_{j,t}, \mathbf{y}_{j,t}), R_{j,t}\right)$$

where $R_{j,t} = 1$ if the supervisor agrees with the action on timestep t in rollout j and $\mathcal{L}$ is the cross entropy classification loss. Given a point cloud, we use the robust grasping policy of Dex-Net 2.0 [3] that samples and ranks a set of antipodal grasp candidates according to $Q_{\hat{\theta}}$ using the Cross Entropy Method.

4.2.1 Transfer Learning

Research in computer vision suggests that features from deep CNNs performs well as generic features when classifying images in new domains [35]. Since simulating object heaps is slower than simulating singulated objects due to object interactions [23], we explore optimizing the neural network weights of the bin picking policy by transfer learning from features of a GQ-CNN trained to grasp singulated objects on millions of examples from Dex-Net 2.0 [3]. Specifically, we fine-tune features from the Dex-Net 2.0 GQ-CNN by using the weights as an initialization for optimization with SGD and only updating the fully connected layers, leaving the conv layers fixed. We update the network for 10 epochs using SGD with momentum of 0.9, a base learning rate of 0.01, and a staircase exponential learning rate decay with a decrease of 5% on each epoch.

5 Experiments

We evaluated classification performance on synthetic data from the simulator and performed extensive physical evaluations on an ABB YuMi with a Primesense Carmine 1.08 depth sensor and custom silicone gripper tips designed by Guo et al. [36]. All experiments ran on a Desktop running Ubuntu 14.04 with a 2.7 GHz Intel Core i5-6400 Quad-Core CPU and an NVIDIA GeForce 980, and we used an NVIDIA GeForce GTX 1080 for training large models.

5.1 Synthetic Training

We generated three versions of the Dex-Net 2.1 training dataset with noise levels $\epsilon = \{0.1, 0.5, 0.9\}$ using 10,000 rollouts of the noisy supervisor policy in our POMDP with an average of $\lambda = 5$ objects per heap sampled from the 1,500 object models of Dex-Net 2.0 [3]. Each dataset contained approximately 100k datapoints. The mean number of objects in the initial state distribution was $\lambda = 5$.

We trained the following models with a 80-20 image-wise split on each dataset:

- **SVM.** A bagging classifier composed of 50 SVMs trained on the first 100 principal components of the GQ-CNN fc4 feature space [3].
- **Random Forest (RF).** A set of 50 trees of max depth 10 trained on the first 100 principal components of the GQ-CNN fc4 feature space [3].
- **Dex-Net 2.1 (Scratch).** A GQ-CNN [3] trained only on Dex-Net 2.1 for 25 epochs.
- **Dex-Net 2.1 (Fine-tuned).** A GQ-CNN [3] initialized with pretrained weights from Dex-Net 2.0 and fine-tuned on Dex-Net 2.1 for 10 epochs with fixed conv layers.

The parameters of each model were set based on performance on a randomized validation set. We swept over model parameters and data featurizations for the SVM and RF. For the GQ-CNN models we swept over learning rate, dropout, fully connected reinitializations, and fully connected layer sizes. More details can be found in the supplemental file.

Table 1 details classification performance on the Dex-Net 2.1 dataset with $\epsilon = 0.9$. The Dex-Net 2.0 GQ-CNN model fine-tuned on Dex-Net 2.1 performed best in terms of classification accuracy and average precision (AP). This suggests that GQ-CNN weights pretrained on Dex-Net may be useful as generic features for parallel-jaw grasping from point clouds.

Model	Learning	Acc (%)	AP
SVM	Supervised	91.6	0.56
RF	Supervised	92.0	0.59
GQ-CNN	Supervised	91.3	0.58
GQ-CNN	Transfer	**92.4**	**0.64**

Table 1: Classification performance on the Dex-Net 2.1 dataset with 90% noise. The GQ-CNN trained with transfer learning from a set of network weights pretrained on Dex-Net 2.0 performs better than the models trained using standard supervised learning on Dex-Net 2.1, suggesting that there is shared structure between the datasets.

5.2 Bin Picking on an ABB YuMi

To study performance on a physical robot, we designed a bin picking benchmark where the robot was presented a subset of objects from a dataset of 50 test objects in a bin and the goal was to iteratively move objects from the bin to a receptacle as illustrated in Fig. 2. First, we sampled N objects from the validation set sampled uniformly at random. Then, each of the N objects was placed in a box and the box was shaken and placed upside-down in the center of the bin to randomize object poses. On each timestep the grasping policy received as input a depth image, bounding box containing the object, and camera intrinsics, and output a target pose of the gripper in the robot's coordinate frame. The robot then approached the target grasp along a linear approach trajectory and closed the jaws. Grasp success was defined by whether or not the grasp transported the target object to the receptacle. The system iterated until either (a) no objects remain or (b) the robot has 5 consecutive failed grasps on the same object.

5.2.1 Performance Metrics

We compared performance on this benchmark with the following metrics:

Figure 2: (Let) For each experiment, a subset of N validation objects are randomly dropped into a bin (green rim, center), at which point the YuMi iteratively plans grasps from point clouds and attempts to lift and transport the objects to a packing box (blue rim, right side).(Middle) A set of 50 test objects with various shapes, sizes, and material properties. A subset of 25 are rigid and opaque, and 25 others have transparency (e.g. goggles), moving parts (e.g. can opener), or deformable material (e.g. cloth). (Right) Example color and depth images from the physical setup with example grasp planned with the Dex-Net 2.1 $\epsilon = 0.9$ policy.

1. **Success Rate:** The percentage of grasp attempts that moved an object to the packing box.
2. **Average Precision (AP):** The area under the precision-recall curve. In some applications a robot can decide whether or not to execute grasps based on a threshold for the classifier confidence. AP measure the average success rate over all possible thresholds for these scenarios.
3. **Percent Cleared:** The fraction of objects that were moved to the receptacle.
4. **Picks per Hour (PPH):** The estimated number of bin picks per hour of runtime computed by multiplying the average number of grasp attempts per hour by the success rate. Human performance is approximately 600 PPH.

5.2.2 Datasets

Fig. 2 illustrates the test set of physical objects used in the benchmark, which includes 50 objects of various sizes, shapes, and materials. The objects all satisfy three criteria to be graspable by the YuMi: (1) the jaws can fit around the object in at least one configuration, (2) the object weighs less than $0.25kg$ due to the YuMi payload, (3) some part of the object is opaque and non-specular (can be sensed with a depth camera), and (4) the min diameter of the object is greater than $1cm$.

We break the test set up into two partitions:

- **Basic.** A subset of 25 test objects that are rigid, weigh less than $0.25kg$ (the payload of the YuMi), and are fully visible with a depth camera, which tests generalization to novel shapes when assumptions of the simulator are satisfied.
- **Typical.** The entire 50 object dataset, which additionally tests generalization to novel object properties (e.g. transparency, deformation, moving parts).

5.2.3 Model Performance

Table 2 compares the performance of five policies on the bin picking benchmark with $N = \{5, 10\}$ test objects from the Basic subset for 20 and 10 trials, respectively. This measures performance on a heap size and set of objects that match the assumptions of the simulator. We compared the following policies:

1. **Image-Based Force Closure.** Executes a random planar force grasp with friction coefficient $\mu = 0.8$ computed from edge detection in depth images inspired by [10, 9].
2. **Dex-Net 2.0.** Ranks grasps using the GQ-CNN model of [3] trained on Dex-Net-Large.
3. **Dex-Net 2.1 ($\epsilon = 0.1, \epsilon = 0.5, \epsilon = 0.9$).** Ranks grasps using the Dex-Net 2.1 (Fine-tuned) classifier for varying levels of noise injection in the training dataset.

	5 Objects				10 Objects			
Policy	**Success** (%)	**AP** (%)	% **Cleared**	**PPH**	**Success** (%)	**AP** (%)	% **Cleared**	**PPH**
Force Closure	54	N/A	97	271	55	N/A	92	276
Dex-Net 2.0	92	96	**100**	407	83	84	98	367
Dex-Net 2.1 ($\epsilon = 0.1$)	91	91	**100**	402	86	89	99	380
Dex-Net 2.1 ($\epsilon = 0.5$)	85	89	98	376	66	69	96	292
Dex-Net 2.1 ($\epsilon = 0.9$)	**94**	**97**	**100**	**416**	**89**	**93**	**100**	**394**

Table 2: Performance of grasping policies on the Basic dataset containing 25 opaque and rigid test objects with heaps of size $N = \{5, 10\}$ averaged over 20 and 10 independent trials, repsectively. Human performance is approximately 600 PPH.

	10 Objects				20 Objects			
Policy	**Success** (%)	**AP** (%)	% **Cleared**	**PPH**	**Success** (%)	**AP** (%)	% **Cleared**	**PPH**
Force Closure	64	N/A	98	321	50	N/A	77	251
Dex-Net 2.0	81	88	98	358	70	79	**97**	310
Dex-Net 2.1 ($\epsilon = 0.9$)	**85**	**93**	**100**	**376**	**78**	**86**	**97**	**345**

Table 3: Generalization performance of grasping policies on the Typical dataset containing 50 test objects with hinged parts, deformability, and some material transparency on heaps of size 10 and 20 with 5 independent trials of each.

The Dex-Net 2.1 ($\epsilon = 0.9$) variant performed best across all metrics. The increase in perfromance over the other noise levels may be because the training dataset was heavily skewed toward negative examples, encouraging the learned policy to predict grasp failure when uncertain.

5.2.4 Generalization

We also evaluated the Dex-Net 2.1 $\epsilon = 0.9$ policy on larger heaps of size $N = \{10, 20\}$ with all 50 test objects for 5 independent trials each to evaluate generalization to large piles and different object properties that were not encountered in the simulator. The results are detailed in Table 3. While performance decreases across all categories, the $\epsilon = 0.9$ policy outperforms the Dex-Net 2.0 and antipodal baseline across all metrics. The performance appears to be more sigificantly affected by the heap size than the addition of deformable objects. Qualitative failure modes of the Dex-Net 2.1 policy included collisions where the gripper pressed into another object in the heap and an inability to find robust grasps on thin, curved objects such as the measuring spoon and scissors.

6 Discussion and Future Work

We formulate bin picking as a POMDP and train a GQ-CNN to predict grasps with high reward using a simulator of robust grasping and dynamic object interactions in a heap. We used the model to sample the Dex-Net 2.1 dataset of tens of thousands of demonstrations across 1,500 3D object models from an algorithmic supervisor with noise injection that used full state information to index precomputed robust grasps for 3D models in the simulation.

We find that a policy trained using behavior cloning with high levels of noise injection (90% probability of selecting a random action) has the highest performance across all metrics when transferred to a physical robot, suggesting that conservative policies which favor false negatives over false positives may transfer better from simulation to reality. Furthermore, the high average precision score of GQ-CNNs from Dex-Net 2.1 suggest that performance could be improved by introducing alternative actions, such as probing or using a suction cup [37], when the model does not have high confidence. In future work, we will develop hierarchical policies for bin picking.

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS People and Robots (CPAR) Initiative. The authors were supported in part by the U.S. National Science Foundation under NRI Award IIS-1227536: Multilateral Manipulation by Human-Robot Collaborative Systems, by the Scalable Collaborative Human-Robot Learning (SCHooL) Project, NSF National Robotics Initiative Award 1734633, the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG)

Program, the Berkeley Deep Drive (BDD) Program, and by donations from Siemens, Knapp, Google, Cisco, Autodesk, IBM, Amazon Robotics, Toyota Robotics Institute, and Loccioni, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Sponsors. We thank our colleagues who provided helpful feedback, code, and suggestions, in particular Frederik Ebert, Roy Fox, Animesh Garg, Menglong Guo, Sanjay Krishnan, Fritz Kuttler, Michael Laskey, Sergey Levine, Pusong Li, Jacky Liang, Matt Matl, Michael Peinhopf, Peter Puchwein, and Vishal Satish.

References

[1] E. Johns, S. Leutenegger, and A. J. Davison. Deep learning a grasp function for grasping under gripper pose uncertainty. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4461–4468. IEEE, 2016.

[2] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *Int. Journal of Robotics Research (IJRR)*, 34(4-5):705–724, 2015.

[3] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Proc. Robotics: Science and Systems (RSS)*, 2017.

[4] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016.

[5] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai. Fast graspability evaluation on single depth maps for bin picking with general grippers. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1997–2004. IEEE, 2014.

[6] K. Goldberg, B. V. Mirtich, Y. Zhuang, J. Craig, B. R. Carlisle, and J. Canny. Part pose statistics: Estimators and experiments. *IEEE Trans. Robotics and Automation*, 15(5):849–857, 1999.

[7] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *arXiv preprint arXiv:1709.07857*, 2017.

[8] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *arXiv preprint arXiv:1603.02199*, 2016.

[9] A. ten Pas and R. Platt. Using geometry to detect grasp poses in 3d point clouds. In *Intl Symp. on Robotics Research*, 2015.

[10] M. Gualtieri, A. t. Pas, K. Saenko, and R. Platt. High precision grasp pose detection in dense clutter. *arXiv preprint arXiv:1603.01564*, 2016.

[11] U. Viereck, A. t. Pas, K. Saenko, and R. Platt. Learning a visuomotor controller for real world robotic grasping using easily simulated depth images. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[12] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, pages 1–51, 2013.

[13] D. Prattichizzo and J. C. Trinkle. Grasping. In *Springer handbook of robotics*, pages 671–700. Springer, 2008.

[14] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff. Industrial robotics. In *Springer handbook of robotics*, pages 1385–1422. Springer, 2016.

[15] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesisa survey. *IEEE Trans. Robotics*, 30(2):289–309, 2014.

[16] K. Ikeuchi, B. K. Horn, S. Nagata, T. Callahan, and O. Feingold. Picking up an object from a pile of objects. Technical report, DTIC Document, 1983.

[17] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, 31(8):951–973, 2012.

[18] V. N. Christopoulos and P. Schrater. Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1557–1563. IEEE, 2007.

[19] D. Fischinger, M. Vincze, and Y. Jiang. Learning grasps for unknown objects in cluttered scenes. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 609–616. IEEE, 2013.

[20] D. Kappler, J. Bohg, and S. Schaal. Leveraging big data for grasp planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015.

[21] J. Weisz and P. K. Allen. Pose error robust grasping from contact wrench space metrics. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 557–562. IEEE, 2012.

[22] S. Thrun. Monte carlo pomdps. In *Proc. Advances in Neural Information Processing Systems*, pages 1064–1070, 2000.

[23] E. Coumans et al. Bullet physics library. *Open source: bulletphysics. org*, 15:49, 2013.

[24] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[25] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2290–2295, 1992.

[26] T. Mallick, P. P. Das, and A. K. Majumdar. Characterizations of noise in kinect depth images: A review. *IEEE Sensors Journal*, 14(6):1731–1740, 2014.

[27] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[28] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[29] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[30] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[31] S. Choudhury, A. Kapoor, G. Ranade, S. Scherer, and D. Dey. Adaptive information gathering via imitation learning. In *Proc. Robotics: Science and Systems (RSS)*, 2017.

[32] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation learning. In *Conf. on Robot Learning (CoRL)*, 2017.

[33] E. Klein, M. Geist, B. Piot, and O. Pietquin. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems*, pages 1007–1015, 2012.

[34] N. Ratliff, J. A. Bagnell, and S. S. Srinivasa. Imitation learning for locomotion and manipulation. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 392–397. IEEE, 2007.

[35] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *Computer Vision–ECCV 2016 Workshops*, pages 435–442. Springer, 2016.

[36] M. Guo, D. V. Gealy, J. Liang, J. Mahler, A. Goncalves, S. McKinley, and K. Goldberg. Design of parallel-jaw gripper tip surfaces for robust grasping. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017.

[37] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg. Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning. *arXiv preprint arXiv:1709.06670*, 2017.