On-Policy Dataset Synthesis for Learning Robot Grasping Policies Using Fully Convolutional Deep Networks

Vishal Satish[1], Jeffrey Mahler[1,2], Ken Goldberg[1,2]

***Abstract*— Rapid and reliable robot grasping for a diverse set of objects has applications from warehouse automation to home de-cluttering. One promising approach is to learn deep policies from synthetic training datasets of point clouds, grasps, and rewards sampled using analytic models with stochastic noise models for domain randomization. In this paper, we explore how the distribution of synthetic training examples affects the rate and reliability of the learned robot policy. We propose a synthetic data sampling distribution that combines grasps sampled from the policy action set with guiding samples from a robust grasping supervisor that has full state knowledge. We use this to train a robot policy based on a fully convolutional network architecture that evaluates millions of grasp candidates in 4-DOF (3D position and planar orientation). Physical robot experiments suggest that a policy based on Fully Convolutional Grasp Quality CNNs (FC-GQ-CNNs) can plan grasps in 0.625s, considering 5000x more grasps than our prior policy based on iterative grasp sampling and evaluation. This computational efficiency improves rate and reliability, achieving 296 mean picks per hour (MPPH) compared to 250 MPPH for iterative policies. Sensitivity experiments explore the effect of supervisor guidance level and granularity of the policy action space. Code, datasets, videos, and supplementary material can be found at http://berkeleyautomation.github.io/fcgqcnn.**

I. Introduction

Robots must be able to rapidly and reliably plan grasps for a wide variety of objects under inherent uncertainty in sensing, physics, and control. One approach is to compute grasps for a set of known 3D objects using analytic models and to plan grasps online by matching sensor data to known objects. However, this requires a perception system that can recognize object instances, making it difficult to scale to many novel objects. An alternative approach is to use machine learning to train a robot policy to predict the probability of success for candidate grasps based on sensor data such as images or point clouds. Recent results suggest that learned robot policies can generalize to a wide variety of novel objects on a physical robot. Learning-based grasp planning approaches require a data collection policy for collecting training examples. Empirical methods collect training data from human labeling [16], [25], [38], dataset aggregation from self-supervision [18], [28], or reinforcement learning [12]. However, these dataset collection approaches may be time-consuming and prone to mislabeled examples.

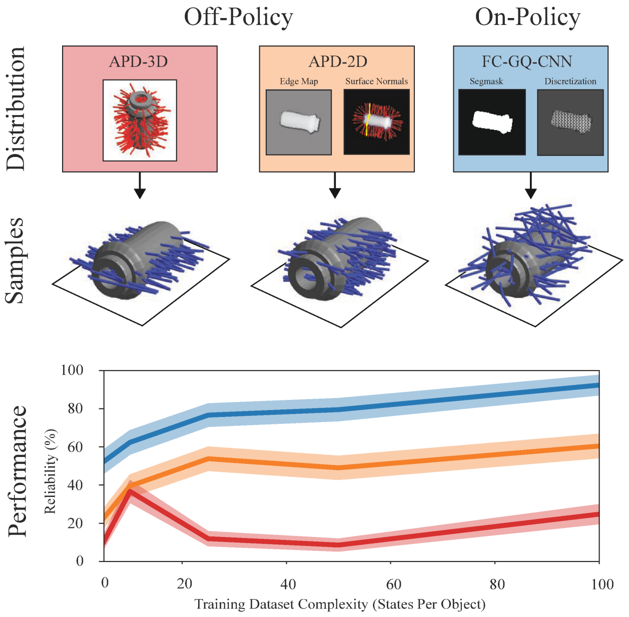

Fig. 1: Grasp action distributions for generating synthetic data to train robust grasping policies. (Top Left) 3D antipodal grasps sampled from object models (used by Dex-Net 2.0 [21]). (Top Middle) 2D antipodal grasps sampled from depth images. (Top Right) The proposed on-policy distribution where grasp actions are sampled from a 4-DOF dense discretization of actions based on a depth image and segmask. (Bottom) Performance in simulation on a singulated set of 10 objects with varying geometries of the FC-GQ-CNN policy trained on actions sampled from the off-policy APD-3D and APD-2D distributions and the on-policy FC-GQ-CNN distribution with supervisor guidance. The guided on-policy approach is able to reach performance close to that of a supervisor with full state knowledge.

An alternative is to rapidly generate massive synthetic training datasets using analytic metrics and structured domain randomization for robust transfer from simulation to reality [11], [20], [34], [35]. The synthetic data distribution can be tuned to the expected sensor, robot, and environment. However, many systems sample training grasps from a fixed distribution different from the set of actions that the policy must evaluate at runtime. For example, several approaches sample training grasp actions that are constrained to known 3D object surfaces while the learned policy samples grasps from observations [20], [21]. This may lead to reduced performance due to covariate shift [15], [32]. This raises the question: Is it possible to develop faster and more reliable grasping policies by modifying the data collection policy used for sampling synthetic training examples?

In this paper, we propose a dataset distribution that samples grasps from synthetic observations to approximate the distribution that the policy will evaluate with a learned quality function at runtime, as illustrated in Fig. 1. We refer to this as the *on-policy* distribution and distinguish this from

[1] Dept. of Electrical Engineering and Computer Science;
[2] Dept. of Industrial Operations and Engineering Research;
The AUTOLAB at UC Berkeley (automation.berkeley.edu).
{vsatish, jmahler, goldberg}@berkeley.edu

prior approaches that train and evaluate on two different distributions. To guide data collection towards successful grasps, the distribution samples a mixture of grasps from the action space of the robot policy and from an algorithmic robust grasping supervisor that leverages the known geometry and pose of 3D objects to index pre-computed grasps. We use this to train an efficient single-shot grasping policy based on Fully Convolutional Networks, an architecture introduced in computer vision for image segmentation [19] that has recently shown promising results for learning grasping policies from human labeled datasets [25], [38]. We develop a novel variant of this architecture that evaluates grasps in 4-DOF (3D position and planar orientation) by parallelizing standard Grasp Quality Convolutional Neural Networks (GQ-CNNs). The architecture can rapidly produce dense and reliable grasp predictions by evaluating millions of grasps in parallel.

This paper contributes:

1) A novel dataset collection policy for sampling synthetic training datasets that reflects the distribution of actions that the learned policy evaluates at runtime utilizing guidance from a robust supervisor.
2) Experimental data from a physical robot comparing the performance of this approach to current state-of-the-art approaches for training a policy based on a 4-DOF (3D position and gripper orientation) Fully Convolutional Grasp Quality CNN (FC-GQ-CNN).
3) Experimental data in simulation exploring the sensitivity of policy performance to supervisor guidance level and action space granularity.

Physical robot experiments suggest that a policy based on Fully Convolutional Grasp Quality CNNs (FC-GQ-CNNs) can plan grasps in 0.625s, considering 5000x more grasps than a policy based on iterative grasp sampling and evaluation. This computational efficiency improves rate and reliability, achieving 296 mean picks per hour (MPPH) compared to 250 MPPH for iterative policies.

II. Related Work

A. Learning for Grasp Planning

The goal of grasp planning is to find a gripper configuration that maximizes a quality metric. Initial approaches to the problem utilized analytic approaches (see [29] for a survey). However, the difficulty in using these approaches to generalize to novel objects has lead to the use of empirical and hybrid approaches, the latter of which utilizes massive synthetic training datasets generated with analytic models.

Combined with advances in deep learning, these approaches utilize policies that query a neural network to locate the highest quality grasp. These fall into two categories. *Discriminative* approaches utilize a neural network to rank grasps based on a quality metric and optimization techniques to search for high quality grasp candidates [21], [34]. *Generative* approaches instead directly generate a grasp set given sensor data, and may use heuristics to select the optimal grasp from this set. One popular approach is to regress to grasp coordinates in image space [16], [30].

These deep approaches have been trained on massive datasets of human-labeled [13], [16], [30], self-supervised [8], [18], [26], [28] or synthetic [3], [6], [11], [21], [36] grasps, images, and quality labels. A popular human-labeled dataset is the Cornell Grasping Dataset developed by Lenz et al. [16], which consists of 1k RGB-D images labeled with grasps parametrized by oriented bounding boxes. This has been extensively used to train CNN-based models for singulated objects [14], [25], [30]. Self-supervised datasets have been collected from grasp attempts on a physical robot. Pinto and Gupta [28] collected over 40k grasp attempts on a Baxter to train a CNN, whereas Levine et al. [18] expanded this approach even further by collecting over 800k datapoints using numerous robot arms. Synthetic datasets such as Dex-Net [21] have been used to train state-of-the-art hybrid approaches. We explore the effect of the distribution of synthetic training examples on the rate and reliability of learned policies.

B. Dense Predictions and Fully Convolutional Networks

Recent approaches to grasping have leveraged dense evaluations of the entire grasp action space instead of selectively choosing grasps to evaluate based on heuristics or iterative optimization techniques [21]. These approaches utilize deep neural networks to rapidly evaluate millions of grasps by offloading computation to specialized GPU hardware. Johns et al. [11] evaluated a dense set of output poses and applied a function to this output to make it robust to gripper pose uncertainty. However, the standard CNN architecture they chose limited them to a pre-determined image size and required the final layer of the network to scale with this image size, which can become large and computationally expensive.

The need for dense evaluations with smaller networks that scale to arbitrary image sizes led to the development of Fully Convolutional Networks (FCNs) in the field of computer vision for tasks requiring pixel-wise discrimination such as image segmentation [19], object detection [4], and visual tracking [37]. Several successful empirical grasping approaches have taken advantage of FCNs. Zeng et al. [38] trained FCNs on hundreds of human-labeled images to predict the probability of success for four grasp primitive actions. Morrison et al. [25] used FCNs to increase grasp planning frequency to 50Hz, using a discriminative head to predict the probability of grasp success and separate network heads to generate the grasp angle and gripper width.

The approaches used by [11], [38], [25] all evaluate grasps based on 3-DOF (planar position and orientation). We extend this approach to 4-DOF, including the grasp height in evaluation.

C. Training Distribution

Learning-based approaches to grasp planning require a large labeled dataset of training data, and the distribution of the training data may affect the performance of the learned policy. Prior approaches have used a distribution based on human labels [16], [38], random exploration [18], [28], or the set of antipodal grasps on 3D mesh surfaces [21]. The fields of IL [32] and RL [33] have considered how to optimize the distribution of training data to improve learning efficiency and to reduce covariate shift. In IL, approaches are either on-policy, using supervisor labels on actions taken by the current learned policy [32], or off-policy [15], using actions taken by the supervisor. In RL, a common approach is to sample actions using epsilon greedy, which mixes random actions from the action set with actions preferred by the current trained policy [33]. Supervised actor-critic [31] approaches to RL, such as Actor-Mimic [27], use a supervisor policy to guide the distribution of actions taken to train a policy. Several methods incorporate similarity to supervisor actions into the RL reward function, such as Deep Learning from Demonstrations [10] and Guided Policy Search [17]. In comparison, we consider data collection for supervised learning and use a training dataset distribution based on a robust grasping supervisor that uses a database of 3D object models to index grasps.

III. Problem Statement

We consider the problem of learning a robot grasping policy for a wide variety of novel objects as measured by Mean Picks Per Hour (MPPH) [23], the number of objects that are successfully grasped per hour. This depends on rate, or frequency of grasp planning and execution, and reliability, or percentage of successful grasps.

The goal is for a robot to iteratively grasp and transport a single object from a bin to a receptacle based on point clouds from a depth camera. The *state* $\mathbf{x}$ includes the geometry, pose, and material properties of each object. The robot acquires a *point cloud observation* $\mathbf{y}$ represented as a depth image. Then the robot uses a *grasp policy* π_θ with parameters θ that takes as input an observation $\mathbf{y}$ and returns a grasp $\mathbf{u} = \pi_\theta(\mathbf{y})$. A grasp is specified as the 3D position and planar orientation of a parallel-jaw gripper. Upon executing the grasp, the robot receives a binary *reward* $R(\mathbf{x}, \mathbf{u}) \in \{0, 1\}$ based on whether or not an object is successfully grasped and transported to the receptacle. The grasp attempt has a duration consisting of the combined sensing time t_s, grasp computation time (GCT) t_c, and grasp execution time t_e, in fraction of hours, which we assume to be constant.

The objective is MPPH:

$$\max_{\theta \in \Theta} \mathbb{E}\left[\frac{R(\mathbf{x}, \pi_\theta(\mathbf{y}))}{t_s + t_c + t_e}\right] = \max_{\theta \in \Theta} \mathbb{E}\left[R(\mathbf{x}, \pi_\theta(\mathbf{y}))\right]$$

The expectation is taken with respect to the grasping environment, a distribution over possible states, observations, rewards, and actions based on the policy:

$$p(R, \mathbf{x}, \mathbf{y}, \mathbf{u} \mid \theta) = \underbrace{\pi(\mathbf{u} \mid \mathbf{y}, \theta)}_{\text{policy}} \underbrace{p(R \mid \mathbf{x}, \mathbf{u})}_{\text{reward}} \underbrace{p(\mathbf{y} \mid \mathbf{x})}_{\text{observation}} \underbrace{p(\mathbf{x})}_{\text{state}}$$

MPPH can be increased by improving rate, reliability, or both. In this paper we focus on improving rate by reducing GCT. Since MPPH depends on hardware, we individually measure GCT in our experiments to control for network, sensor and arm-movement speed.

IV. Learning Objective

We use supervised learning to train a policy based on a quality function Q_θ that predicts the probability of success for a given grasp using a deep neural network with parameters θ [2], [16], [11], [21]. The policy maximizes this function over all grasps in the action space $\mathcal{U}(\mathbf{y})$ to select a grasp:

$$\pi_\theta(\mathbf{x}) = \underset{\mathbf{u} \in \mathcal{U}(\mathbf{y})}{\operatorname{argmax}} \; Q_\theta(\mathbf{y}, \mathbf{u}) \qquad \text{(IV.1)}$$

To train the network, we minimize the cross-entropy loss between the predicted grasp quality and reward:

$$\min_{\theta \in \Theta} \mathbb{E}\left[\mathcal{L}(R, Q_\theta(\mathbf{y}, \mathbf{u}))\right]$$

Here the expectation is taken with respect to a dataset distribution defined by a dataset collection policy τ that may be independent of the policy parameters:

$$p(R, \mathbf{x}, \mathbf{y}, \mathbf{u} \mid \theta) = \underbrace{\tau(\mathbf{u} \mid \mathbf{x}, \mathbf{y})}_{\text{policy}} \underbrace{p(R \mid \mathbf{x}, \mathbf{u})}_{\text{reward}} \underbrace{p(\mathbf{y} \mid \mathbf{x})}_{\text{observation}} \underbrace{p(\mathbf{x})}_{\text{state}}$$

The distribution τ is designed to reflect a diverse set of actions that may be evaluated by the learned quality function at runtime. Note that this is distinct from the distribution of actions planned by the policy, as the quality function must evaluate a diverse set of grasp candidates and discard poor actions. In prior work, τ is sampled off-policy by collecting data from a human supervisor [25], [38], the current best policy [18], [28], or 3D antipodal grasps [21].

V. On-Policy Dataset Synthesis

The hybrid approach to learning robust grasping policies samples training datasets from a synthetic dataset distribution that is the product of a simulated training environment $\xi(R, \mathbf{x}, \mathbf{y} \mid \mathbf{u})$ and a data collection policy $\tau(\mathbf{u} \mid \mathbf{x}, \mathbf{y})$. The training environment ξ models the distribution of rewards, states, and point clouds using analytic models based on physics and geometry [21] with domain randomization for robust sim-to-real transfer [35]. The data collection policy τ attempts to sample a diverse set of actions that the learned quality function may need to evaluate at runtime. Nonetheless, several hybrid methods such as the Dexterity Network (Dex-Net) 2.0 [21], [20] use different distributions of grasp actions for training and policy deployment (See Fig. 1), which may reduce performance due to covariate shift [32], [15].

Drawing inspiration from approaches in imitation learning [15], [32] and reinforcement learning [17], [31], we propose an on-policy dataset distribution. The distribution uses a data collection policy that uniformly samples grasps from the action space $\mathcal{U}(\mathbf{y})$ that the policy evaluates with the learned quality function at runtime (see Equation IV.1). To increase the percentage of successful grasp actions, the distribution uses guiding samples from a robust grasping supervisor that plans robust grasps analytically using full knowledge of 3D object geometry and pose.

Formally, the data collection policy is

$$\tau(\mathbf{u} \mid \mathbf{x}, \mathbf{y}) = (1 - \epsilon)\mathrm{U}(\mathcal{U}(\mathbf{y})) + \epsilon\Omega(x),$$

a mixture of a uniform distribution over the grasp action space $\mathcal{U}(\mathbf{y})$ and the Dex-Net 1.0 [24] robust grasping supervisor distribution $\Omega(\mathbf{x})$. The parameter ϵ controls the percentage of actions to sample from the supervisor. A larger value of ϵ may increase covariate shift as more actions are sampled from the supervisor, while smaller values of ϵ may skew the distribution toward many negative examples and require larger training datasets.

To increase the rate of grasp computation, we use this data collection policy to train a Fully Convolutional Grasp Quality CNN (FC-GQ-CNN) on a 4-DOF action space (3D position and planar orientation). Fig. 1 (Top Right) illustrates the dense discretization of the 4-DOF grasp action space that is evaluated at runtime and used to sample training data.

VI. Fully Convolutional 4-DOF Architecture

Prior to the use of FCNs, grasping policies could only evaluate a comparably limited number of grasps in a reasonable computational budget. With this constraint, many prior approaches used iterative optimization methods such as the cross-entropy method (CEM) [18], [21] to approximate (IV.1).

One drawback of these approaches is that they must be implemented in a serial fashion. In the particular case that they are implemented with a neural network quality function, they require significant computational overhead, such as copying data between device and host memory, every time the network must be queried for a new batch of predictions. Also, the iterative optimization itself often involves many parameters such as the ideal number of iterations, which may be difficult to tune.

As an alternative to these sparse approaches, Zeng [38] and Morrison [25] have proposed using Fully Convolutional Networks (FCNs) that can produce an extremely dense yet efficient set of predictions over the entire state space in a single-shot evaluation. This reduces the search over the action space to an argmax of the network output.

The denser we can make the FCN evaluation, the more efficiently we can cover the state space by offloading computation to neural network inference, which can be highly optimized on specialized GPU hardware. With this goal in mind, we extend an FCN to 4-DOF as opposed to prior 3-DOF approaches such as [38] and [25]. This is achieved by parameterizing the action space using 3D gripper position and planar orientation.

A. Architecture

We build the 4-DOF FC-GQ-CNN by:

1) Initially training a 4-DOF CNN.
2) At policy evaluation time converting all fully connected layers into fully convolutional layers, resulting in an FCN.

Although [25] and [38] choose to directly train the FCN, we choose to train a CNN and convert it to a FCN because this eliminates the need for densely-labeled ground-truth images during training. Instead, the CNN can be trained on much smaller crops of individual grasps.

1) 4-DOF GQ-CNN Architecture: We first design a 4-DOF CNN architecture. We take inspiration from the GQ-CNN [21], extending it to 4-DOF by incorporating the grasp angle θ. This takes as input a cropped thumbnail depth image of a single grasp centered on the grasp center pixel, $\mathbf{y}_{train}$, along with the corresponding grasp depth relative to the camera, $\mathbf{z}$. It computes a set of k success probabilities, each corresponding to a planar gripper angle.

Unlike in the original GQ-CNN, we cannot incorporate depth using a separate network stream. The separate stream presents a computational bottleneck during the FCN conversion because its output must be expensively tiled across the output of the final convolution layer, which can be fairly large for larger input sizes. We instead incorporate the depth $\mathbf{z}$ into the network by subtracting it from the depth image $\mathbf{y}_{train}$, thus transforming the depth image into the grasp frame of reference. Following standard conventions, we normalize the transformed depth images by subtracting the mean and dividing by the standard deviation of the training data.

2) Conversion to FCN: By converting each of the fully connected layers of the 4-DOF GQ-CNN into a convolution layer, we define the FC-GQ-CNN architecture. This is a valid transformation because of the one-to-one mapping between convolution and fully connected layer weights. The FC-GQ-CNN, illustrated in Fig. 2 (Top) and detailed in the caption, takes as input an arbitrarily sized depth image $\mathbf{y}$ and corresponding gripper depth relative to the camera $\mathbf{z}$, and evaluates a dense 4-DOF set of grasp quality predictions. This allows us to evaluate the 4-DOF GQ-CNN over the entire input image in an efficient manner, as if it were a giant convolution filter. The stride of the FC-GQ-CNN is determined by the amount of pooling present in the convolution layers of the 4-DOF GQ-CNN architecture, specifically each pooling by a factor of p will increase the stride by a corresponding factor.

VII. Policy Learning

A. Policy

Given an arbitrarily sized depth image $\mathbf{y}$, the FC-GQ-CNN policy discretizes the action space based on grasp center pixel, angular bin, and gripper depth. The granularity of the former two are determined by the architecture, whereas the latter is a policy parameter. Once we have formed this action space, we can efficiently evaluate it with the FC-GQ-CNN and take the argmax (IV.1).

B. FC-GQ-CNN Training

We train the 4-DOF GQ-CNN on 96x96 depth image thumbnails, $\mathbf{y}_{train}$, of individual grasps. We optimize the parameters of the network using backpropagation with stochastic gradient descent and momentum. The network output consists of all k angular predictions, however each training sample corresponds to only one specific angle. Given a depth image with grasp angle θ, we first map θ to the corresponding angular bin, then we backpropagate only through the network output corresponding to that particular angular bin. The network weights are initialized using a Kaiming initializer [9]. The network architecture and optimization framework are

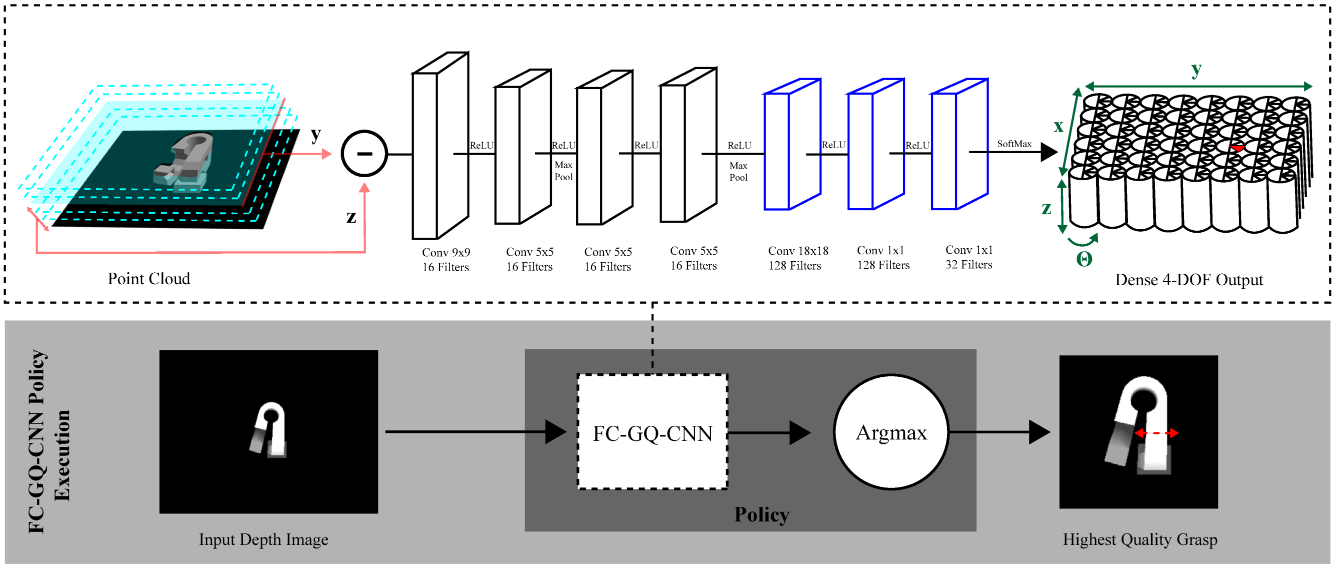

Fig. 2: Grasping policy based on a Fully Convolutional Grasp Quality Convolutional Neural Network (FC-GQ-CNN). (Top) 4-DOF evaluation of an arbitrarily sized depth image $\mathbf{y}$ by the network. Convolutional layers in the latter half of the network (highlighted in blue) were originally fully connected layers in the 4-DOF GQ-CNN architecture. (Bottom) Given a depth image, the policy queries the network and takes the argmax to return the highest quality grasp, highlighted in red.

written in Python using Tensorflow. All training was done on Ubuntu 16.04 with an NVIDIA Titan Xp and an Intel Core i7-6850K clocked at 3.6 GHz.

VIII. Experiments

To characterize the effect of training distribution on FC-GQ-CNN policy performance, we perform experiments on singulated objects both in a quasi-static simulator and on a physical robot. We also probe the effect of action space granularity and choice of ϵ on policy performance in simulation. Finally, to test generalization and performance in clutter, we perform experiments on a physical robot with 25 novel objects placed in a bin.

All experiments were performed on a desktop running Ubuntu 16.04 with an NVIDIA Titan Xp and an Intel Core i7-6850k clocked at 3.6GHz. Physical experiments were performed on an ABB YuMi with custom silicone fingertips [7] and a high-res Photoneo PhoXi S depth sensor (See Fig. 3 Top Left). In this setup, GCT comprises 26% of the total grasping cycle.

A. Object Sets

We use 4 different object sets in our experiments:

1) *Thingiverse_large* is a cleaned and pruned set of 1,600 3D CAD models from Thingiverse [1] used by Danielczuk et al. [5].
2) *Thingiverse_mini* is a sub-set of 10 objects from *thingiverse_large* with varying geometry.
3) *Adv* is the adversarial object set proposed by Mahler et al. [21] (Fig. 3 Top Right).
4) *Novel* is a set of 25 physical objects with diverse geometries used to test generalization and performance in dense clutter (Fig. 3 Bottom Right).

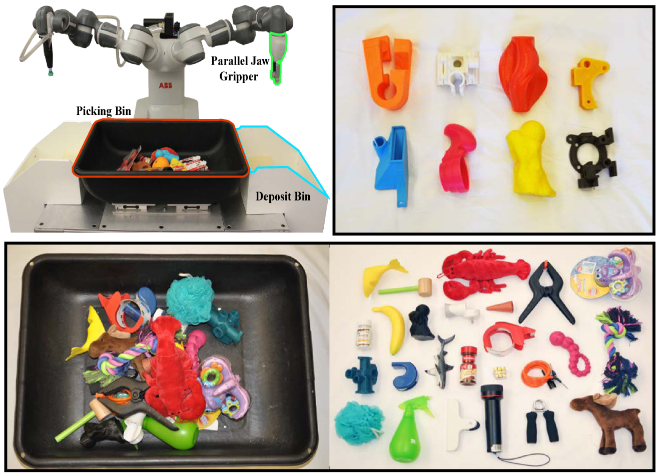

Fig. 3: (Top Left) The experimental setup consisting of an ABB YuMi with custom silicone fingertips [7] and a high-res Photoneo PhoXi depth sensor. The experimental objective is to move objects from the picking bin to the deposit bin. (Top Right) The 8 adversarial objects used by Mahler et al. [21]. (Bottom Right) The 25 novel objects with diverse geometries used to test generalization. (Bottom Left) The 25 novel objects arranged in a bin to simulate dense clutter.

B. Training Distribution

We characterize the effect of training distribution on policy performance in simulation by training and testing an FC-GQ-CNN policy on singulated objects from *thingiverse_mini* with datasets of varying size (measured in unique states per object) sampled with the following data collection policies:

1) Uniform 3D Antipodal Action Space (APD-3D) [24]
2) Uniform 2D Antipodal Action Space with Supervisor Guidance (APD-2D+SUP) [21]
3) 4D Discrete Action Space with Supervisor Guidance (FC-GQ-CNN+SUP)

We choose these specific distributions as a spectrum from fully off-policy (1) to our proposed guided on-policy ap-

Training Distribution	Reliability(%)	AP(%)
APD-3D	72.5	91.2
APD-2D+SUP	65.0	69.0
FC-GQ-CNN+SUP	**87.5**	**97.7**

TABLE I: Performance on a physical robot of the FC-GQ-CNN policy versus training distribution measured on a set of 8 known adversarial objects in singulation over 80 evaluations, 10 per object. For comparison, GQ-CNN is able to reach 83% and 91%, accordingly [21].

proach (3). Policy (2) is chosen as an intermediate because it contains grasps closer to those evaluated by the learned FC-GQ-CNN policy, but still constrained by antipodality.

We evaluate the reliability of the resulting learned policies for 250 evaluations of grasping each object. On each evaluation, the object is placed in a stable resting pose in a given 2D position on a planar worksurface, the policy plans a parallel-jaw grasp, and the grasp is evaluated with robust quasi-static wrench resistance [22] for a known direction of gravity. To test whether or not performance differences are due to sample approximation error, we evaluate reliability over increasing dataset sizes by varying the number of unique positions and orientations of each object from 5 to 100.

Fig. 1 (Bottom) shows the results. Across all dataset sizes, the policy trained with the FC-GQ-CNN+SUP guided on-policy training distribution performs significantly better than the other policies, suggesting more efficient learning. The policies trained on APD-3D and APD-2D+SUP have significantly lower performance than the supervisor even with 100 states per object. We hypothesize that APD-2D+SUP outperforms APD-3D because it is a larger subset of the FC-GQ-CNN+SUP training distribution, but lacks sufficient coverage of the policy action space.

Next we extend experiments on training distribution to a physical robot by training and testing an FC-GQ-CNN policy on singulated objects from *adv* using the three different distributions. We evaluate the resulting policies with 10 grasp attempts per object. Table I shows the results. As we hypothesized, the FC-GQ-CNN+SUP distribution performs significantly better than the off-policy approaches. However, we do not find the same trend as in simulation where the APD-2D+SUP distribution performs in-between the off-policy supervisor and our proposed on-policy method. In fact, the off-policy supervisor performs surprisingly well. Mahler et al. [21] found similar performance.

C. Sensitivity Experiments

The granularity of the policy action space can have a significant impact on the speed and reliability of dense approaches, in particular the trade-off between the two. A very high granularity will result in a very precise policy, however producing a dense-enough output for this granularity will be computationally expensive and require a large grasp computation time (GCT). On the other extreme, a low granularity will result in a policy that is quick to evaluate due to significantly reduced computation, but is imprecise because it never evaluates many grasps, some of which could be robust.

Policy	Rel.(%)	AP(%)	GCT(s)	MPPH
PJ Heuristic	53.4	77.1	2.0	162
GQ-CNN	75.8	**96.0**	1.5	250
GQ-CNN(*)	81.2	93.8	3.0	236
FC-GQ-CNN	**85.6**	95.2	**0.6**	**296**

TABLE II: Performance of PJ Heuristic, GQ-CNN, and FC-GQ-CNN on bin picking with 25 novel objects on a physical robot. GQ-CNN(*) is a version of GQ-CNN with increased CEM samples, increasing the performance of CEM at the cost of rate. The FC-GQ-CNN outperforms the GQ-CNN, GQ-CNN(*), and PJ heuristic in rate and reliability. The higher AP of GQ-CNN suggests that it fails to find a grasp on failures rather than better predicting grasp quality.

Policy	GCT(s)	# Evaluations	CTPG(ms)
GQ-CNN	1.485	400	3.7125
FC-GQ-CNN	0.625	**2,008,064**	**0.0003**

TABLE III: Comparison of the number of grasps evaluated, GCT, and computation time per grasp (CTPG) for GQ-CNN and FC-GQ-CNN on a 386x516 depth image.

We characterize the effect of the policy action space granularity on performance in simulation by training and testing an FC-GQ-CNN policy on singulated objects from *thingiverse_mini* using the FC-GQ-CNN+SUP distribution. However, now we independently vary the number of angular bins k and stride s in the FC-GQ-CNN architecture, and the number of depth bins d used in the FC-GQ-CNN policy. Fig. 4 shows the results of experiments. The goal is to maximize reliability while minimizing GCT. We find that the best choice of these parameters is $s = 4, d = 16, k = 16$, that is we evaluate the image in pixel-wise strides of 4, bin the depth into 16 bins, and have angular bins of size $180/16 = 11.25\,\text{deg}$.

D. Novel Objects in Clutter

Robots in warehouses must be able to pick not only singulated objects, but more importantly objects in dense clutter. In order to test generalization and performance in clutter, we train an FC-GQ-CNN on the FC-GQ-CNN+SUP distribution with objects placed in heaps (which simulates real-world clutter) sampled from *thingiverse_large*. We then test the policy's ability to completely clear a bin consisting of all the objects from *novel* (Fig. 3 Bottom Left) by picking only a single object at a time. If more than one object is picked, we arbitrarily choose a single object to count and place the rest back into the bin. We compare performance on 5 rollouts against a carefully tuned parallel jaw heuristic and GQ-CNN [21] trained using the APD-2D training distribution, which is on-policy for the standard GQ-CNN. We find that the FC-GQ-CNN policy performs best overall, achieving 296 MPPH. Results are shown in Table II.

E. Efficiency of FC-GQ-CNN Policy

We can quantify the efficiency of the proposed FC-GQ-CNN policy with the millions of grasps it evaluates in a single pass of the policy and the amortized time per

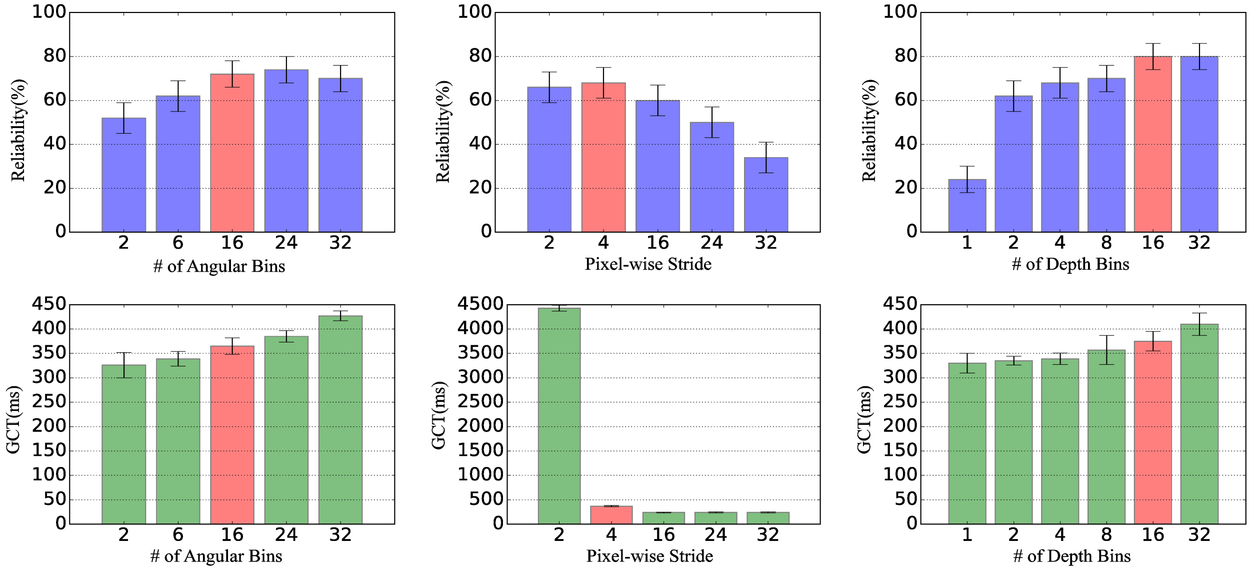

Fig. 4: Sensitivity of FC-GQ-CNN performance to granularity of action space, in particular varying # of angular bins, pixel-wise stride, and # of depth bins, measured in simulation on *thingiverse_mini*. The objective is to minimize GCT while maximizing reliability. Highlighted in red are the best choice of values. Note: The image size in simulation is slightly smaller than in physical experiments, which is why the best GCT here is lower than that in Table II.

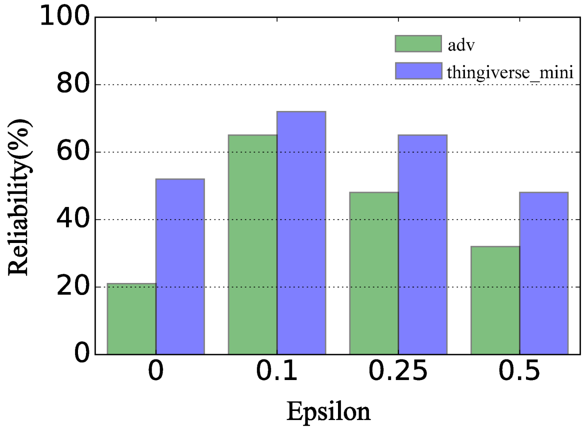

Fig. 5: Reliability of FC-GQ-CNN policy versus choice of ϵ in simulation on *thingiverse_mini* and *adv*. Too low of an ϵ lacks sufficient guiding examples, whereas too large of an epsilon causes covariate shift. We find that the ideal choice is $\boldsymbol{\epsilon = 0.1}$.

individual grasp as shown in Table III. This 1,200x speedup in computation time per grasp significantly outperforms previous iterative sampling and ranking policies such as the cross-entropy method (CEM).

F. Choice of ϵ

We explore the choice of epsilon and its effect on reliability in simulation by training and testing an FC-GQ-CNN policy on singulated objects from *thingiverse_mini* and *adv* using the FC-GQ-CNN+SUP distribution with 20 evaluations of grasping each object. Results are shown in Fig. 5. We find the highest reliability when $\boldsymbol{\epsilon = 0.1}$, which we use in all experiments. If we set $\epsilon = 0$, resulting in a fully on-policy distribution, reliability drops significantly. We hypothesize that there are not enough positive guiding examples in the training distribution. This is corroborated by the steeper drop on *adv*, since challenging objects benefit more from supervisor guidance. As we increase ϵ past 0.1, reliability starts to drop, suggesting covariate shift.

IX. Discussion and Future Work

In this paper, we present a novel on-policy data collection policy that combines grasps sampled from the policy action space with guiding samples from a supervisor. We use this distribution to train a 4-DOF Fully Convolutional Grasp Quality CNN (FC-GQ-CNN). Physical robot experiments show that the FC-GQ-CNN policy significantly increases the number of grasps considered by 5000x in 0.625s to achieve up to 296 mean picks per hour (MPPH) compared to 250 MPPH for policies based on iterative grasp sampling and evaluation.

In future work, we will analyze the differences between the FC-GQ-CNN and iterative approaches such as CEM, in particular whether or not our FC-GQ-CNN is able to find a more optimal grasp. We hypothesize that this might be the case considering the high AP in Table II, which suggests that the GQ-CNN is confident in its grasp ranking, meaning that the decrease in reliability is because the CEM is unable to properly cover the action space to find the best grasp. We will also explore 6-DOF grasps and suction approaches such as Dex-Net 3.0 [22].

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. The authors were supported in part by donations from Siemens, Google, Amazon Robotics, Toyota Research Institute, Autodesk, ABB, Samsung, Knapp, Loccioni, Honda, Intel, Comcast, Cisco, Hewlett-Packard and by equipment grants from PhotoNeo, Nvidia, and Intuitive Surgical. 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 and suggestions, in particular

Matthew Matl, Bill DeRose, Mike Danielczuk, Jonathan Lee, Michelle Lu, David Tseng, Daniel Seita.

References

[1] Thingiverse online 3d object database. [Online]. Available: https://www.thingiverse.com/

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis–a survey," *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[3] 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," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[5] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic point clouds," *arXiv preprint arXiv:1809.05825*, 2018.

[6] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[7] M. Guo, D. V. Gealy, J. Liang, J. Mahler, A. Goncalves, S. McKinley, J. A. Ojea, and K. Goldberg, "Design of parallel-jaw gripper tip surfaces for robust grasping," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2831–2838.

[8] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," in *Advances in Neural Information Processing Systems*, 2018, pp. 9111–9121.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[10] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, G. Dulac-Arnold *et al.*, "Deep q-learning from demonstrations," in *AAAI Conference on Artifical Intelligence*, 2018.

[11] 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)*. IEEE, 2016, pp. 4461–4468.

[12] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

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

[14] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 769–776.

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

[16] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 705–724, 2015.

[17] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013, pp. 1–9.

[18] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conference on Robot Learning (CoRL)*, 2017, pp. 515–524.

[21] 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.

[22] J. Mahler, M. Matl, X. Liu, A. Li, D. V. Gealy, and K. Y. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2018.

[23] J. Mahler, R. Platt, A. Rodriguez, M. Ciocarlie, A. Dollar, R. Detry, M. A. Roa, H. Yanco, A. Norton, J. Falco *et al.*, "Guest editorial open discussion of robot grasping benchmarks, protocols, and metrics," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1440–1442, 2018.

[24] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2016.

[25] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robotics: Science and Systems (RSS)*, 2018.

[26] J. Oberlin and S. Tellex, "Autonomously acquiring instance-based object models from experience," in *Int. S. Robotics Research (ISRR)*, 2015.

[27] E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Actor-mimic: Deep multitask and transfer reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2016.

[28] 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.

[29] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer handbook of robotics*. Springer, 2008, pp. 671–700.

[30] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[31] M. T. Rosenstein and A. G. Barto, "Reinforcement learning with supervision by a stable controller," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 5. IEEE, 2004, pp. 4517–4522.

[32] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.

[33] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[34] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[35] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[36] 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.

[37] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127.

[38] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.