Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics

Article · March 2017

CITATIONS
176

READS
582

8 authors, including:

Michael Laskey
University of California, Berkeley
23 PUBLICATIONS **535** CITATIONS

SEE PROFILE

Richard Doan
Georgia Institute of Technology
2 PUBLICATIONS **232** CITATIONS

SEE PROFILE

Kenneth Yigael Goldberg
University of California, Berkeley
421 PUBLICATIONS **10,637** CITATIONS

SEE PROFILE

Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics

Jeffrey Mahler*, Jacky Liang*, Sherdil Niyaz*, Michael Laskey*, Richard Doan*, Xinyu Liu*,
Juan Aparicio Ojea†, and Ken Goldberg*
*Dept. of EECS, University of California, Berkeley
Email: {jmahler, jackyliang, sniyaz, laskeymd, rdoan, xinyuliu, goldberg}@berkeley.edu
† Siemens Corporation, Corporate Technology
Email: juan.aparicio@siemens.com

***Abstract*—To reduce data collection time for deep learning of robust robotic grasp plans, we explore training from a synthetic dataset of 6.7 million point clouds, grasps, and robust analytic grasp metrics generated from thousands of 3D models from Dex-Net 1.0 in randomized poses on a table. We use the resulting dataset, Dex-Net 2.0, to train a Grasp Quality Convolutional Neural Network (GQ-CNN) model that rapidly classifies grasps as robust from depth images and the position, angle, and height of the gripper above a table. Experiments with over 1,000 trials on an ABB YuMi comparing grasp planning methods on singulated objects suggest that a GQ-CNN trained with only synthetic data from Dex-Net 2.0 can be used to plan grasps in 0.8s with a success rate of 93% on eight known objects with adversarial geometry and is 3× faster than registering point clouds to a precomputed dataset of objects and indexing grasps. The GQ-CNN is also the highest performing method on a dataset of ten novel household objects, with zero false positives out of 29 grasps classified as robust and a 1.5× higher success rate than a point cloud registration method.**

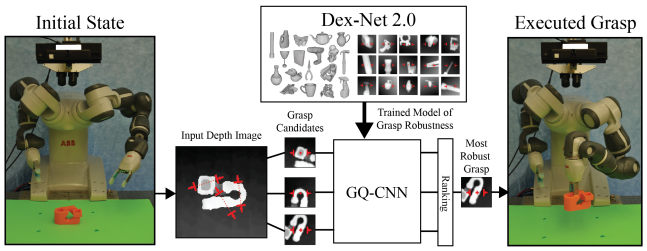

Fig. 1: Dex-Net 2.0 Architecture. (Center) The Grasp Quality Convolutional Neural Network (GQ-CNN) is trained offline to predict analytic robustness of candidate grasps from depth images using a dataset of 6.7 million synthetic point clouds, grasps, and associated analytic grasp metrics computed with Dex-Net 1.0. (Left) When an object is presented to the robot, a depth camera returns a 3D point cloud, where pairs of antipodal points identify a set of several hundred grasp candidates. (Right) The GQ-CNN rapidly determines the most robust grasp candidate, which is executed with the ABB YuMi robot. If the object is lifted, transported, and shaken without dropping, the trial is considered a success.

I. Introduction

Reliable robotic grasping is challenging due to imprecision in sensing and actuation, which leads to uncertainty about properties such as object shape, pose, material properties, and mass. Recent results suggest that deep neural networks trained on large datasets of human grasp labels [31] or physical grasp outcomes [43] can be used to learn to grasp from images or pointclouds across a wide variety of objects, similar to generalization results in computer vision [28]. However, data collection requires either tedious human labeling [25] or months of execution time on a physical system [33].

An alternative approach is to plan grasps using physics-based analytic metrics based on caging [50], grasp wrench space (GWS) analysis [48], or robust GWS analysis, which can be rapidly computed using Cloud Computing [27]. However, these methods typically assume a separate perception system that identifies objects and estimates properties such as object pose either perfectly [48] or according to known Gaussian distributions [36]. This is prone to errors, may not generalize well to new objects, and can be slow to match point clouds to known models during execution [15]. In this paper we consider estimating grasp robustness directly from noise in pixel values [20, 37] by training a deep Convolutional Neural Network (CNN) on a massive dataset of noisy rendered point clouds and grasps computed with robust quasi-static grasp analysis, building upon recent research on training deep networks to predict the outcomes of dynamic grasping simulations [24, 25, 58].

Our primary contributions are: 1) the Dexterity Network (Dex-Net) 2.0, a dataset associating 6.7 million point clouds, gripper poses, and grasp metrics, with parallel-jaw grasps planned using robust quasi-static analysis on a dataset of 1,500 3D object models, 2) a Grasp Quality Convolutional Neural Network (GQ-CNN) model trained to classify robust grasp locations in depth images using expected epsilon quality as supervision, where each grasp is specified as a planar pose and height above a table, 3) a method for online grasp planning on a physical robot that detects robust grasps by sampling antipodal grasp candidates from edges in depth images and ranking candidates with the GQ-CNN, and 4) experiments on a physical robot comparing the performance of CNN-based grasp detection with a point cloud registration system.

In over 1,000 blinded physical trials of grasping single objects on a tabletop with an ABB YuMi robot, we compare Dex-Net 2.0 to planar grasp quality metrics [7], a random forest [55], an SVM [56], and a baseline that recognizes

objects, registers their 3D pose [15], and indexes Dex-Net 1.0 [36] for the most robust grasp to execute. We find that the Dex-Net 2.0 grasp planner is $3\times$ faster than the registration-based method, 93% successful on objects seen in training (the highest of learning-based methods), and is 80% successful on novel objects with zero false positives out of 29 grasps classified as robust (the highest of all methods). Our grasp planner is also 93% successful in system for order fulfillment in which the ABB YuMi singulates a pile of objects, identifies target objects, and grasps objects to transport them to a shipping container.

II. Related Work

Grasp Planning. Given an object and reachability constraints due to the environment, grasp planning considers finding a gripper configuration that maximizes a success (or quality) metric. Methods fall into one of two categories based on success criteria: *analytic* methods [48], which consider performance according to physical models such as the ability to resist external wrenches [45], and *empirical* (or data-driven) methods [4], which typically use human labels [2] or the ability to lift the object in physical trials [43].

Analytic Methods. Analytic approaches typically assume that object and contact locations are known exactly and consider either the ability to resist external wrenches [48] or the ability to constrain the object's motion [50]. External wrenches may be modeled as arbitrary [45] or specific to a task [34]. As wrench space analysis can be computationally expensive, a common method is to precompute a database of known 3D objects labeled with grasps and the quality of each, such as GraspIt! [16] or OpenGRASP [32]. Using the database on a physical system typically involves registering the 3D poses of known objects to models in the database [4, 5, 8, 23, 27]. This can be extended to unknown objects by registering point clouds to similar objects in the database where similarity is based on point clouds [15] or images [62], and warping the point cloud to known shapes and replanning [46].

Robust grasp planning (RGP) methods maximize grasp robustness, or the expected value of an analytic metric under uncertainty in sensing and control. This typically involves labeling grasps on 3D object models [61] in a database with robust metrics such as probability of force closure [27] due to the computational complexity of sampling. To account for uncertainty on a physical system, the robustness of grasps may be updated online using a Bayesian model [5]. Recent research has demonstrated that the sampling complexity of RGP can be improved using Multi-Armed Bandits [29] and datasets of prior 3D objects and robust grasps, such as the Dexterity Network (Dex-Net) 1.0 [36]. In this work we make a major extension to Dex-Net 1.0 by associating synthetic point clouds with robust grasps and training a Convolutional Neural Network to map point clouds and grasps to estimated robustness for execution on a physical robot.

Empirical Methods. Empirical approaches typically use machine learning to develop models that map from robotic sensor readings to metrics of success based on human labels or the outcomes of physical trials. Human labels have become popular due to empirical correlation with physical success [2]. Research in this area has largely focused on associating human labels with graspable regions in RGB-D images [31] or point clouds [10, 22]. Recently, Kappler et al. [25] collected a dataset of thousands of human grasp labels using Amazon Mechanical Turk and used the dataset to train a Convolutional Neural Network (CNN) to predict human labels from projections of point clouds. Lenz et al. [31] created a dataset of over 1k RGB-D images with human labels of successful and unsuccessful grasping regions, which has been used to train fast CNN-based detection models [49].

Another line of research on empirical grasp planning has attempted to optimize success in physical trials directly. The time cost of generating samples on a physical robot led to the development active methods for acquiring grasping experiences such as Multi-Armed Bandits using Correlated Beta Processes [40] or Prior Confidence Bounds [42]. Nonetheless, recently Pinto and Gupta [43] scaled up data collection by recording over 40k grasping experiences on a Baxter and iteratively training CNNs to predict lifting successes or to resist grasp perturbations caused by an adversary [44]. Levine et al. [33] scaled up dataset collection even futher, collecting over 800k datapoints with a set of continuously running robotic arms and using deep learning to predict end effector poses. However, this required over 2 months of training across 6-14 robots.

Robotic Perception for Grasping. A common approach to robotic perception of rigid objects is to segment, identify, and register the pose of 3D object models [8, 23, 53, 65] in order to index into a set of manipulation plans. However, this multi-stage approach can be prone to many hyperparameters that are difficult to tune and errors can compound across modules.

An alternate approach is to use deep learning to estimate 3D object shape and pose directly from color and depth images [66]. Wu et al. [64] learned to predict 3D model categories from a voxel grid representation using ModelNet, a dataset containing thousands of labeled 3D models. Gupta et al. [19] classified coarse object category and pose of objects in a scene by finetuning the AlexNet CNN [28] on depth and color renders from 3D models in ModelNet. Varley et al. [58] used a fully connected deep network to predict candidate fingertip locations on depth images for known 3D object models and later extended the method [59] to estimate the object shape from a single-view point cloud using a 3D CNN. Recent research in robotics has focused on how to improve accuracy in object recognition by structuring the way the neural network fuses the separate color and depth streams from images [47], adding synthetic noise to synthetic training images [11], and using 3D convolutions on an occupancy grid representation of point clouds [38].

Another approach is to detect graspable regions directly in images without explicitly representing object shape and pose [41], as it may not always be necessary to explicitly recognize objects and their pose. Girshick et al. [13] first demonstrated the use of deep CNNs for image detection by

combining region proposals with CNN-based image classifiers. Similarly, recent research has considered treating grasping as an image detection problem [31, 54], where a deep CNN is used to classify image regions as graspable based on an adaptive or fixed set of candidates. Since learning from real images may require a significant data collection time on a physical system, and alternative approach is to learn on simulated data [60] and to transfer the representation to real data. Methods to transfer representations from simulation to reality include finetuning the higher network layers [67], domain confusion [57], or progressive networks [51]. Recent research suggests that in some cases it may be sufficient to train on datasets generated using perturbations to the parameters of the simulator [17, 52]. Notably, Johns et al. [24] used rendered depth images with simulated noise to train a CNN-based detector for planar grasps using dynamic simulations with DART as supervision and successfully deployed the model directly on real depth images. We build upon these results by training a model using robust analytic grasp metrics as supervision, using the height of the gripper to predict robustness, and performing extensive evaluations on a physical robot.

III. Problem Statement

We consider the problem of planning a robust planar parallel-jaw grasp for a singulated rigid object resting on a table based on single-view point clouds from a depth camera. We learn a function that takes as input a candidate grasp and a depth image and outputs an estimate of robustness, or probability of success under uncertainty in sensing and control.

A. Assumptions

We assume a parallel-jaw gripper, rigid objects singulated on a planar worksurface, and single-view (2.5D) point clouds taken with a depth camera. For generating datasets, we assume a known gripper geometry and a single overhead depth camera with known intrinsics.

B. Definitions

Fig. 2 illustrates our coordinate frames, variables, and our graphical model for generating the Dex-Net 2.0 dataset (detailed in Section IV-A1).

States. Let $\mathbf{x} = (\mathcal{O}, T_o, T_c, \gamma)$ denote a state describing the variable properties of the camera and objects in the environment, where $\mathcal{O}$ specifies the geometry and mass properties of an object, T_o, T_c are the 3D poses of the object and camera, respectively, and $\gamma \in \mathbb{R}$ is the coefficient of friction between the object and gripper.

Grasps. Let $\mathbf{g} = (\mathbf{p}, \theta) \in \mathbb{R}^3 \times \mathcal{S}^1$ denote a parallel-jaw grasp in 3D space specified by a center $\mathbf{p} = (x, y, z) \in \mathbb{R}^3$ and angle in the table plane $\theta \in \mathcal{S}^1$.

Images. Let $\mathbf{y} = \mathbb{R}_+^{H \times W}$ be a real-valued depth image with height H and width W taken by a depth camera with known intrinsics [20] and let T_c be the 3D pose of the camera.

Grasp Robustness. Let $S(\mathbf{g}, \mathbf{x}) \in \{0, 1\}$ be a binary-valued grasp success metric, such as force closure or physical lifting.

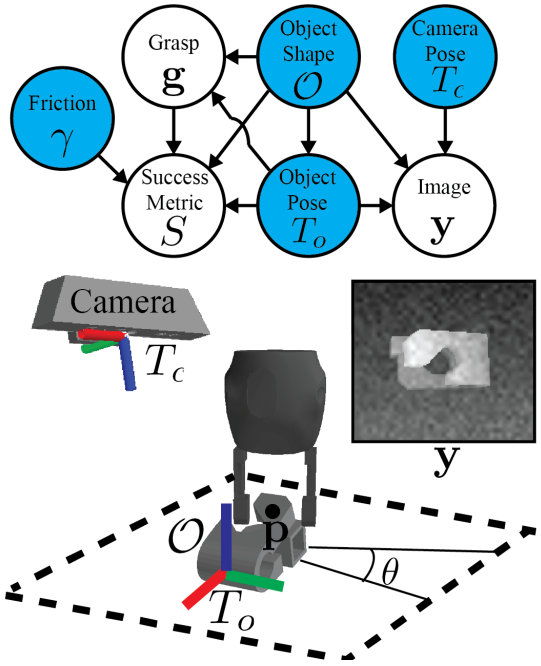

Fig. 2: Graphical model for robust parallel-jaw grasping of objects on a table surface based on point clouds. Blue nodes are variables included in the state representation. Object shapes $\mathcal{O}$ are uniformly distributed over a discrete set of object models and object poses T_o are distributed over the object's stable poses and a bounded region of a planar surface. Grasps $\mathbf{g} = (\mathbf{p}, \theta)$ are sampled uniformly from the object surface using antipodality constraints. Given the coefficient of friction γ we evaluate an analytic success metric S for a grasp on an object. A synthetic depth image $\mathbf{y}$ is generated from 3D meshes based on the camera pose T_c, object shape, and pose and corrupted with multiplicative and Gaussian Process noise.

Let $p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})$ be a joint distribution on grasp success, grasps, states, and images modeling imprecision in sensing and control. For example, p could be defined by noisy sensor readings of a known set of industrial parts coming down a conveyor belt in arbitrary poses. Let the *robustness* of a grasp given an observation [5, 61] be the expected value of the metric, or probability of success under uncertainty in sensing and control: $R(\mathbf{g}, \mathbf{y}) = \mathbb{E}_{p(S, \mathbf{x} | \mathbf{g}, \mathbf{y})} [S \mid \mathbf{g}, \mathbf{y}]$.

C. Objective

Our goal is to find an estimated robustness function $R_{\theta^*}(\mathbf{g}, \mathbf{y}) \in [0, 1]$ over possible grasps, objects, and images that classifies grasps according to the binary success metric:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \, \mathbb{E}_{p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})} [\mathcal{L}(S, R_\theta(\mathbf{g}, \mathbf{y}))] \quad \text{(III.1)}$$

where $\mathcal{L}$ is the cross-entropy loss function and Θ defines the parameters of our function, which for us is the Grasp Quality Convolutional Network (GQ-CNN) parameters described in Section IV-B. This objective is motivated by that fact that $R_{\theta^*}(\mathbf{g}, \mathbf{y}) = R(\mathbf{g}, \mathbf{y})$ for all possible grasps and images when there exists $\theta \in \Theta$ such that $R_\theta(\mathbf{g}, \mathbf{y}) = R(\mathbf{g}, \mathbf{y})$ [39].

We wish to use the estimated function R_{θ^*} for robust grasp planning based on images. Given θ, we define a greedy grasping policy that selects the grasp with the highest estimated robustness: $\pi_\theta(\mathbf{y}) = \text{argmax}_{\mathbf{g} \in \mathcal{C}} R_\theta(\mathbf{g}, \mathbf{y})$, where $\mathcal{C}$ specifies constraints on the set of available grasps, such as collisions or kinematic feasibility.

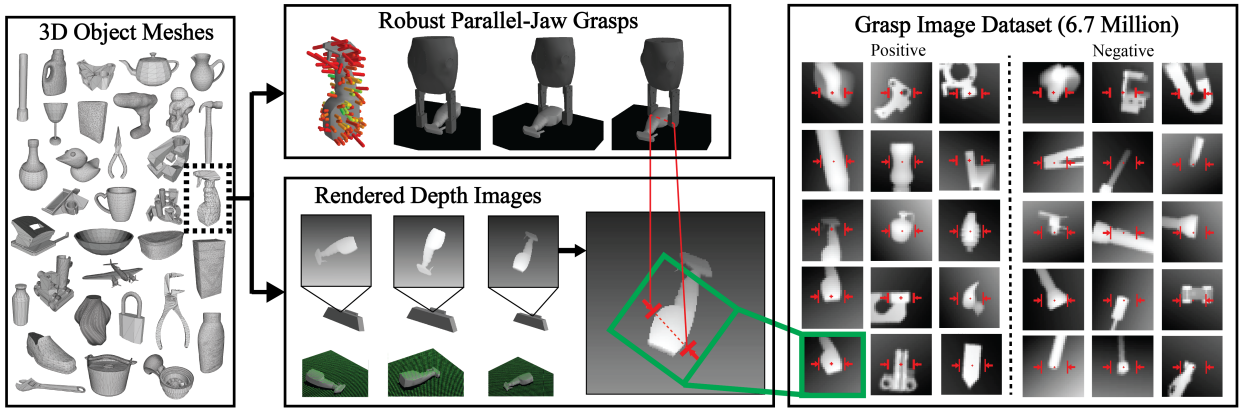

Fig. 3: Dex-Net 2.0 pipeline for training dataset generation. (Left) The database contains 1,500 3D object mesh models. (Top) For each object, we sample hundreds of parallel-jaw grasps to cover the surface of each 3D model and evaluate the analytic robustness using sampling. For each stable pose of the object we associate a set of grasps that are perpendicular to the table and collision-free for a given gripper model. (Bottom) We also render depth images of each object in each stable pose, with the planar object pose and camera pose sampled uniformly at random. Every grasp for a given stable pose is associated with a pixel location and orientation in the rendered image. (Right) Each image is rotated, translated, cropped, and scaled to align the grasp pixel location with the image center and the grasp axis with the middle row of the image, creating a 32×32 grasp image. The full dataset contains over 6.7 million grasp images.

IV. Learning a Grasp Robustness Function

Solving for the robustness function in objective III.1 is challenging for several reasons. First, we may need a huge number of samples to approximate the expectation over a large number of possible objects. We address this by sampling a training dataset of 6.7 million synthetic point clouds, parallel-jaw grasps, and analytic robustness metrics from Dex-Net 2.0 across 1,500 3D models according to the graphical model illustrated in Fig. 2. Second, the true robustness function may be complex and difficult to fit with linear or kernelized models. Consequently, we develop a Grasp Quality Convolutional Neural Network (GQ-CNN) model that classifies grasp robustness from a planar gripper pose, the height of the gripper above the table, and a point cloud specified as a depth image, and train it on data from Dex-Net 2.0.

A. Dataset Generation

We estimate R_{θ^*} using a sample approximation [12] of the objective in Equation III.1 using i.i.d samples $(S_1, \mathbf{g}_1, \mathbf{x}_1, \mathbf{y}_1), ..., (S_N, \mathbf{g}_N, \mathbf{x}_N, \mathbf{y}_N) \sim p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})$ from our generative graphical model for images, grasps, and success metrics:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(S_i, R_\theta(\mathbf{g}_i, \mathbf{y}_i)).$$

1) Graphical Model: Our graphical model is illustrated in Fig. 2 and models $p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})$ as the product of a state distribution $p(\mathbf{x})$, an observation model $p(\mathbf{y}|\mathbf{x})$, a grasp candidate model $p(\mathbf{g}|\mathbf{x})$, and a grasp success model $p(S|\mathbf{g}, \mathbf{x})$.

We model the state distribution as

$$p(\mathbf{x}) = p(\gamma)p(\mathcal{O})p(T_o|\mathcal{O})p(T_c)$$

where the distributions are detailed in Table I. Our grasp candidate model $p(\mathbf{g} \mid \mathbf{x})$ is a uniform distribution over pairs of antipodal points on the object surface that are parallel to the table plane. Our observation model is $\mathbf{y} = \alpha\hat{\mathbf{y}} + \epsilon$ where $\hat{\mathbf{y}}$ is a rendered depth image for a given object in a given pose, α is a Gamma random variable modeling depth-proportional noise, and ϵ is zero-mean Gaussian Process noise over pixel coordinates with kernel bandwidth ℓ and measurement noise σ chosen to approximate multiplicative and additive noise [37]. Our grasp success model is

$$S(\mathbf{g}, \mathbf{x}) = \begin{cases} 1 & E_Q > \delta \\ 0 & E_Q \leqslant \delta \end{cases}$$

where E_Q is the robust epsilon quality defined in [55], which we use to smooth the grasp success function on the object surface [24]. Appendix A details the parameters of these distributions.

Distribution	Description
$p(\gamma)$	truncated Gaussian distribution over friction coefficients
$p(\mathcal{O})$	discrete uniform distribution over 3D objects
$p(T_o\|\mathcal{O})$	continuous uniform distribution over the discrete set of object stable poses and planar poses on the table surface
$p(T_c)$	continuous uniform distribution over spherical coordinates for radial bounds $[r_\ell, r_u]$ and polar angle in $[0, \delta]$

TABLE I: Details of the distributions used in the Dex-Net 2.0 graphical model for generating the Dex-Net training dataset.

2) Database: Dex-Net 2.0 contains 6.7 million pre-generated samples of grasps, 3D models, and rendered depth images for model training. Fig. 3 illustrates our pipeline for generating the dataset.

3D Models. The dataset contains a subset of 1,500 mesh models from Dex-Net 1.0: 1,371 synthetic models from 3DNet [63] and 129 laser scans from the KIT object database [26]. Each mesh is aligned to a standard frame of reference using the principal axes, rescaled to fit within a gripper width of $5.0cm$ (the opening width of an ABB YuMi gripper), and assigned a mass of $1.0kg$ centered in the object bounding box since some meshes are nonclosed. For each

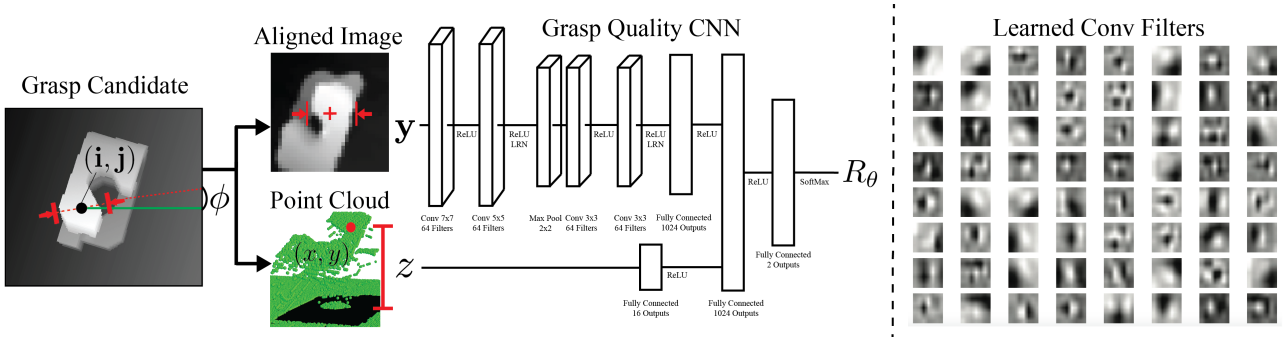

Fig. 4: (Left) Architecture of the Grasp Quality Convolutional Neural Network (GQ-CNN). Grasp candidates are generated from a depth image and transformed to align the image with by the grasp center and axis. The height of each grasp is sampled between the height at the grasp center pixel and the table surface. The architecture contains four convolutional layers in pairs of two separated by ReLU nonlinearities followed by 3 fully connected layers and a separate input layer for the gripper height. The use of convolutional layers was inspired by results indicating the relevance of depth edges as features for learning [3, 31, 36] and the use of ReLUs was inspired by classification results in computer vision [28]. The network outputs an estimate of grasp robustness R_θ, which can be used to rank grasp candidates. (Right) The first layer of convolutional filters learned by the GQ-CNN on Dex-Net 2.0. The filters appear to compute oriented image gradients at various scales.

object we also compute a set of stable poses [14] and store all stable poses with probability of occurence above a threshold.

Parallel-Jaw Grasps. Each object is labeled with a set of up to 100 parallel-jaw grasps. The grasps are sampled using the rejection sampling method for antipodal point pairs developed in Dex-Net 1.0 [36] with constraints to ensure coverage of the object surface [35]. For each grasp we evaluate the expected epsilon quality E_Q under object pose, gripper pose, and friction coefficient uncertainty. To evaluate the robust grasp metrics we use Monte-Carlo sampling from the graphical model described by Seita et al. [55]. We also associate each grasp with the set of stable poses for which the gripper and object geometry are collision-free.

Rendered Depth Images. Every object is also paired with a set of rendered depth images specifying a single-view point cloud for each object stable pose, with camera poses and planar object poses sampled according to the graphical model described in Section IV-A1. Images are rendered using a pinhole camera model and perspective projection with known camera intrinsics, and each rendered image is centered on the object of interest using pixel transformations. Noise is added to the images during training as described in Section IV-B3.

B. Grasp Quality Convolutional Neural Network

1) Architecture: We develop a Grasp Quality Convolutional Neural Network (GQ-CNN) model defining the set of parameters Θ used to represent the grasp robustness function R_θ. The architecture of the GQ-CNN is detailed in Fig. 4 with an example of deploying the model on a point cloud. The GQ-CNN takes as input the candidate gripper height and an image centered on the grasp center pixel and oriented along the grasp axis, both of which are extracted from a point cloud. The image-gripper alignment removes the need to learn rotational invariances that can be modeled by known, computationally-efficient image transformations [20] and allows us to evaluate grasp robustness at any orientation in the image rather than on a discrete set as in [24, 43]. Following standard preprocessing conventions, we normalize the input data by subtracting the mean and dividing by the standard deviation of the training data and then pass the image and gripper height through the network to estimate grasp robustness. The GQ-CNN has approximately 18 million parameters.

2) Training Dataset: Our pipeline for generating training datsets is illustrated in Fig. 3. GQ-CNN training datasets are generated by associating grasps and metrics with locations and angles in rendered depth images from Dex-Net 2.0. We project each grasp center to pixel coordinates $\mathbf{u}$ and grasp axis to an angle in the image plane φ using the rigid transformation T_c and a perspective projection with the camera intrinsics [20]. We then transform all pairs of images and gripper poses to a single grasp image centered on $\mathbf{u}$ and oriented along φ (see the left panel of Fig. 4 for a detailed illustration). After transforming the grasp and image pairs, we crop and resize the images. We also check collisions for each grasp given the gripper and object pose and geometry, and all grasps $\mathbf{g}_i$ in collision are assigned $S_i = 0$. The Dex-Net 2.0 training dataset contains 6.7 million datapoints and approximately 21.2% positive examples for the thresholded robust epsilon quality with threshold $\delta = 0.002$ [25] and our custom YuMi gripper.

3) Optimization: We optimize the parameters of our model using backpropagation with stochastic gradient descent and momentum [28]. We initialize the weights of the model by sampling from a zero mean gaussian with variance $\frac{2}{n_i}$, where n_i is the number of inputs to the i-th network layer [21]. We make use of several data augmentation techniques during training to further scale the training dataset. We reflect the image about its vertical and horizontal axes and rotate each image by 180° since these lead to equivalent grasps. We also adaptively sample the image noise from our noise model (see Section IV-A1) before computing the batch gradient for new samples during training to model imaging noise without explicitly storing multiple versions of each image. To speed up noise sampling we approximate the Gaussian Process noise by upsampling an array of uncorrelated zero-mean Gaussian noise

using bilinear interpolation. We set hyperparameters based on the performance on a randomize synthetic validation set as described in Section VI-C.

V. Grasp Planning

We use the GQ-CNN to plan robust grasps by sampling a set of antipodal grasp candidates [7] from a depth image and ranking the candidates by the GQ-CNN output, as illustrated in Fig. 1.

Our sampling method is designed to sample a set grasp candidates that are in 2D force closure [7] for surface normals defined by the depth image gradients. Detailed pseudocode on the sampling method can be found in Appendix B. We first threshold the depth image to find areas of high gradient and compute approximate outward-pointing surface normals by normalizing the depth image gradient. Then, we sample pairs of pixels uniformly at random and discard all grasps that are not in force closure for a given friction coefficient to generate a set of candidate force closure grasps. We incrementally increase the friction coefficient and repeat the sampling step until a desired number of grasps is reached in order to control for the number of candidates sampled on each run of the algorithm.

After sampling the set of candidate grasps, we transform each grasp candidate from image space to a 6-DOF end-effector pose in world frame. Each grasp is assigned a depth value between the depth measurement at the grasp center pixel and the depth of the table surface. Then the grasp center pixel and grasp axis are projected into 3D space using the depth value and camera intrinsics and transformed into world frame using a known registration of the overhead camera [20]. We rank the grasp candidates by passing each through the GQ-CNN, sorting by the estimated robustness, and executing the most robust grasp that is (a) kinematically reachable and (b) not in collision with the table.

VI. Experiments

We evaluated clasification performance on both real and synthetic data and performed extensive physical evaluations on an ABB YuMi to benchmark the performance of grasping a single object. 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.

A. Physical Benchmark Description

We created a benchmark to evaluate grasp planning methods for a single object on a table surface with an ABB YuMi and custom silicone gripper tips designed by Guo et al. [18]. Our setup is illustrated in Fig. 5, and the experimental procedure is described in the caption and shown in the supplemental video[1]. Each method takes as input a color image, depth image, object bounding box, and camera parameters, and outputs a planned gripper pose. We required a human operator to reset the object in the workspace. Thus, to help remove bias we blinded the operators from knowledge of the tested method in all experiments.

[1]https://youtu.be/9eqAxk95I3Y

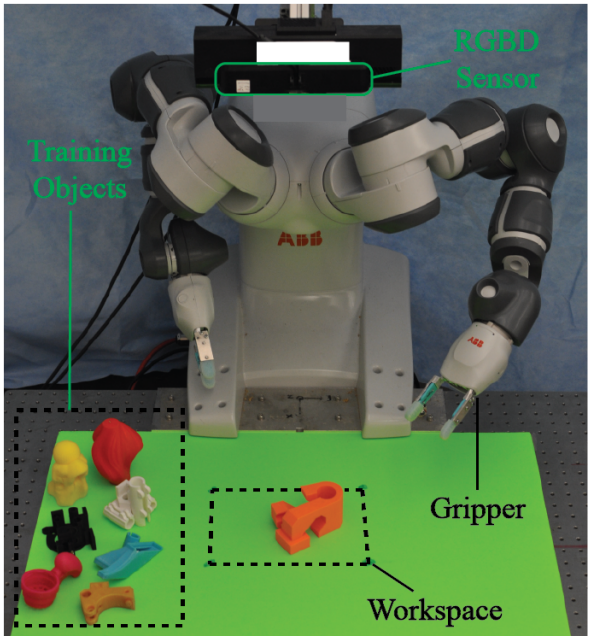

Fig. 5: Diagram of the Dex-Net 2.0 experimental platform for benchmarking grasping with the ABB YuMi. Before each run we register the camera to the robot with a chessboard. In each trial we set an object on the table by shaking the object in a box and placing the box upside down in the workspace to help remove placement bias. We take an image with a Primsense Carmine 1.08, fill in the image using inpainting [24], segment the object using color-based background subtraction, and center the image on the detected object. Each grasp planning method takes as input the original color image, original depth image, and the object detection and outputs a pose for the left gripper. A grasp is considered successful if the object remains within the gripper jaws after closing on the object, lifting, moving the arm to the side of the workspace, and shaking.

We compared performance on this benchmark with the following metrics:

1) **Success Rate:** The fraction of successes to total attempts.
2) **Precision:** The success rate on grasps that are classified as positive. The measures performance when the robot can decide not to grasp an object, which could be useful when the robot has other actions (e.g. pushing available).
3) **Planning Time:** The time in seconds betweem receiving a detection of the object bounding box and returning a planned gripper pose.

B. Datasets

Fig. 6 illustrates the datasets used in the benchmark:

1) **Train:** A set of 8 3D-printed objects with adversarial geometric features such as smooth, curved surfaces. This is used to set model parameters and to evaluate performance on known objects.
2) **Test:** A set of 10 household objects similar to models in Dex-Net 2.0 with various material, geometric, and specular properties. This is used to evaluate generalization to unknown objects.

We chose objects based on geometric features under three constraints: (a) small enough to fit within the workspace, (b)

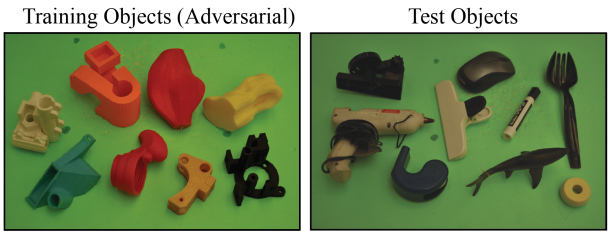

Fig. 6: (Left) Training set of 8 objects with adversarial geometric features such as smooth curved surfaces and narrow passageways for grasping on known objects. (Right) Set of 10 household and office objects not seen during training. The dataset was selected to test performance on challenging objects of varying material, geometry, and surface reflectance properties.

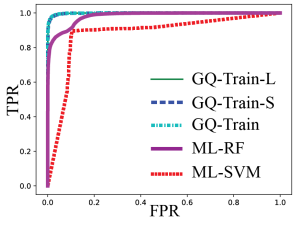

Fig. 7: Receiver operating characteristic comparing the performance of learning models on Train-Synth. The GQ-CNN models all perform similarly and have a significantly higher true positive rate when compared to ML-RF and ML-SVM.

Model	Accuracy (%)
ML-SVM	89.7
ML-RF	90.5
GQ-Train-S	97.8
GQ-Train-L	97.8
GQ-Train	98.1

TABLE II: The classification accuracy of each model on Train-Synth. We see that the GQ-CNN methods have less than 2.5% test error while ML-RF and ML-SVM are closer to 10% error. Pretraining does not appear to affect performance.

weight less than $0.25kg$, the payload of the YuMi, and (c) height from the table greater than $1.0cm$ due to a limitation of the silicone tip grippers.

We also use four different training datasets to study the effect on performance, each with a 80-20 image-wise training and validation split:

1) **Train-Synth:** Synthetic images and grasps for objects in Train (189k datapoints).
2) **Train-Phys:** Outcomes of executing random antipodal grasps with random gripper height and friction coefficient of $\mu = 0.5$ in 50 physical trials per object in Train (400 datapoints).
3) **Dex-Net-Small:** A subset of data from 150 models sampled uniformly from Dex-Net 2.0 (670k datapoints).
4) **Dex-Net-Large:** Data from all 1500 models in Dex-Net 2.0 (6.7m datapoints).

C. Grasp Planning Methods Used for Comparison

We compare a number of grasp planning methods on simulated and real data. We tuned the parameters of each method based on synthetic classification performance and physical performance on the training set of objects. All methods other than point cloud registration use our antipodal grasp sampling method with the same set of parameters (listed in Appendix A) to generate candidate grasps, and each planner executes the highest-ranked grasp according to the method.

Image-based Grasp Quality Metrics (IGQ). We sample a set of force closure grasp candidates by finding antipodal points on the object boundary and rank grasps by the distance from the center of the jaws to the centroid of the object segmentation mask. We set the gripper height as a fixed offset from the height of the pixel corresponding to the grasp center.

Point-Cloud Registration (REG). Our registration-based grasp planning system is inspired by [15, 19]. We first coarsely estimate pose using the object pose corresponding to the top 3 most similar synthetic images from Dex-Net 2.0, where similarity is measured as distance in a feature space [15] defined as the output of AlexNet conv5 [36]. When the object set is known (REG-K), we find similar images from the object instance and stable pose predicted by AlexNet [28] finetuned on hundreds of real color images of the objects. Our finetuned AlexNet had a classification accuracy of 93% on a held-out validation set of color images. When the object set is unknown (REG-U), we search for similar images over the entire database. Next, we finetune the pose of the object in the table plane using Iterated Closest Point [19, 27] with a point-to-plane cost. Finally, we index into Dex-Net 2.0 to retrieve the most robust gripper pose that is kinematically reachable and collision-free. We constrain gripper poses to the same four DOF as the learning-based methods. Our system has a median translational error of $4.5mm$ in the plane of the table and a median rotational error of 3.5° in the plane of the table for known objects.

Alternate Machine Learning Models (ML). We also compare the performance of a Random Forest with 200 trees of depth up to 10 (ML-RF) motivated by the results of [55] and Support Vector Machine with the RBF kernel and a regularization parameter of 1 (ML-SVM) motivated by the results of [3, 54, 56]. For the RF we used the raw images and height of the gripper normalized by the mean and standard deviation across all pixels as features. For the SVM we used a Histogram of Oriented Gradients (HOG) [9] feature representation. Both methods are trained using scikit-learn on the Train-Synth dataset.

Grasp Quality CNNs (GQ). We train our model (GQ) using the thresholded robust epsilon metric with $\delta = 0.002$ [25] for 5 epochs on Dex-Net-Large (all of Dex-Net 2.0) using Gaussian process image noise with standard deviation $\sigma = 0.005$. We trained our model using TensorFlow [1] using a batch size of 128, a momentum term of 0.9, and an exponentially decaying learning rate. Training took approximately 48 hours on an NVIDIA GeForce 1080. The first layer 7×7 convolution filters are shown in the right panel of Fig. 4, and suggest that the network learned coarse oriented gradients and finegrained vertical edge detectors. We believe the vertical edge detectors help to discriminate robust grasps because the antipodal edges map to vertical lines after our image transformation preprocessing step.

To benchmark our architecture outside of our datasets, we also trained on the Cornell Grasping Dataset [31] (containing 8,019 examples) and achieved a 93% recognition rate using grayscale images and an $80-20$ imagewise training-validation

	Comparions of Methods						GQ-CNN Parameter Sensitivity					
	Random	**IGQ**	**ML-RF**	**ML-SVM**	**REG-K**	**GQ-Train-L**	**GQ-Train-S**	**GQ-Train**	**GQ-Train-EMP**	**GQ-Train-FC**	**GQ-Train-LowU**	**GQ-Train-HighU**
Success Rate (%)	58±11	70±10	75±9	80±9	95±5	**93±6**	85±8	83±8	80±9	83±8	78±9	86±8
Precision (%)	N/A	N/A	100	100	N/A	**94**	90	91	80	89	90	92
Planning Time (sec)	N/A	1.9	0.8	0.9	2.6	**0.8**	0.9	0.8	0.8	0.7	0.8	0.9

TABLE III: Performance of grasp planning methods on our grasping benchmark with the Train dataset with 95% confidence intervals for the success rate. Each method was tested for 80 trials. Details on the methods used for comparison can be found in Section VI-C. We see that REG-K point cloud registration with known objects has the highest success rate at 95% but the GQ-CNN pretrained all using all of Dex-Net 2.0 performs comparably at 93% and is 3× faster. Performance of the GQ-CNN drops to 80% when trained on the Train-Phys dataset (GQ-Train-EMP), likely due to the small number of training examples, and drops to 78% when no noise is added to the images during training (GQ-Train-LowU).

split. We also trained several variants to evaluate parameter sensitivity.

Size of Prior Dataset. We train a version on Dex-Net-Small for 15 epochs (GQ-S) to study the effect of prior dataset size on performance.

Importance of Prior Data for Known Objects. To facilitate a comparison between methods that work on a set of known objects, we train three versions of the GQ model on the synthetic dataset of training objects (Train-Synth). The model GQ-Train is trained on only Train-Synth for 25 epochs. The models GQ-Train-L and GQ-Train-S are initialized with the weights of GQ and GQ-S, respectively, and finetuned for 5 epochs on Train-Synth.

Success Metric. We study the effect of different success metrics by training a GQ-CNN using labels from the probability of force closure thresholded at 25% (GQ-Train-FC) and and labels for 400 random grasp attempts on Train using a physical robot [33, 43] (GQ-Train-EMP).

Image Noise Levels. We study the effect of image noise levels by training with zero noise $\sigma = 0$ (GQ-Train-LowU) and high noise with $\sigma = 0.01$ (GQ-Train-HighU).

D. Classification of Synthetic Data

The GQ-CNN trained on all of Dex-Net 2.0 had an accuracy of 85.7% on a held out validation set of approximately 1.3 million datapoints. Due to the memory and time requirements of training SVMs, we compare synthetic classification performance across methods on the smaller Train-Synth dataset. Fig. 7 shows the receiver operating characteristic curve comparing the performance of GQ-Train-L, GQ-Train-S, GQ-Train, ML-SVM, and ML-RF on a held-out validation set. Table II details the classification accuracy for the various methods. The GQ-CNNs outperform ML-RF and ML-SVM, achieving near-perfect validation accuracy.

E. Physical Performance on Known Objects

We evaluated the performance of the grasp planning methods on known objects in the physical benchmark with the training objects. Each method was run for 10 trials per object, resulting in a total of 80 trials per method. The results are detailed in the left half of Table III. We see that the method with the highest success rate is REG-K with 95%. The GQ-CNN plans grasps 3× faster than REG-K and maintains a high 93% success rate and 94% precision. All other methods succeed on 80% or fewer attempts.

	IGQ	REG-U	GQ-Train-EMP	GQ-Train	GQ-S	GQ
Success Rate (%)	60±13	52±14	68±13	74±12	72±12	**80±11**
Precision (%)	N/A	N/A	68	87	92	**100**
Planning Time (sec)	1.8	3.4	0.7	0.7	0.8	0.8

TABLE IV: Performance of grasp planning methods on our grasping benchmark with the test dataset of 10 household objects with 95% confidence intervals for the success rate. Each method was tested for 50 trials, and details on the methods used for comparison can be found in Section VI-C. We see that the GQ-CNN trained on all of Dex-Net 2.0 performs best in terms of success rate and precision, with zero false positives among 29 positive classifications. Performance decreases with smaller training datasets, but the GQ-CNN method outperforms the image-based grasp quality metrics (IGQ) and registration of similar objects (REG-U).

We compare the performance of the GQ-CNN variations in the right half of Table III. The results suggests that using the full Dex-Net 2.0 in training increases performance as it is the only model to achieve higher than 90%, but the performance of the variations is still higher than the ML methods in most cases. This may be because the network learns features that are better at discriminating a wide variety of possible grasps from the larger datasets. GQ-CNN performance appears to be most negatively affected training without image noise and training on only physical outcomes, which drop performance to 78% and 80%, respectively.

F. Physical Performance on Test Objects

We also evaluate the performance of our models on the ten novel objects to study the generalization performance of our network. Each method was run for 5 trials per object, resulting in a total of 50 trials per method. The parameters were set based on the Train object performance without seeing the performance on novel objects. Table IV details the results. We see that our GQ-CNN performs best with an 80% success rate and 100% precision, which indicates that the GQ-CNN predicted zero false positives over 29 grasps classified as robust. Performance when training with either Dex-Net-Small or Train is similar at 72% and 74%, repectively, outperforming the non-GQ-CNN methods by 10%.

G. Failure Modes

Fig. 8 displays some common grasp failures of planning with the GQ-CNN on real sensor data. One failure mode occurs when the RGBD sensor fails to measure thin parts of the object geometry, making these regions seem accessible. A second type of failure occurs due to collisions with the object. It appears that the network cannot fully distinguish

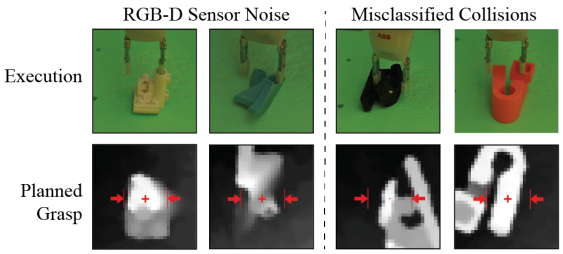

Fig. 8: Four examples of failed grasps planned using the GQ-CNN from Dex-Net 2.0. The most common failure modes appear to be related to: (left) missing sensor data for an important part of the object geometry, such a thin parts of the object surface, (right) collisions with the object are misclassified as robust.

collision-free grasps in narrow parts of the object geometry. This suggests that performance could be improved with more accurate depth sensing and using analytic methods to prune grasps in collsion. The result of predicting zero false positives on the novel objects also suggest that some failure occur due to undersampling candidate grasps, not mistakes made by the GQ-CNN.

H. Application: Order Fulfillment

We illustrate the modularity of our method we apply in to an application of order fulfillment with the ABB YuMi. The goal is to grasp and transport a set of three target objects to a shipping box in the presence of three distractor objects when starting with the objects in a pile on a planar worksurface. Rather than directly grasping objects in clutter [17, 33], the YuMi first singulates the objects [6] using a policy learned from human demonstrations mapping binary images to push locations [30]. When the robot detects an object with sufficient clearance from the pile, it identifies the object instance based on color and uses GQ-CNN-Train-L to plan a robust grasp. The robot then transports the object to either the shipping box or a reject box, depending on whether or not the object was a distractor. Dex-Net 2.0 successfully placed the correct objects in the box on 4 out of 5 attempts and was successful in grasping on 93% of 27 total attempts. The supplemental video shows a successful run[2].

VII. Discussion and Future Work

We present Dex-Net 2.0, a dataset containing 6.7 million point clouds, parallel-jaw grasps, and grasp success metrics. We developed a Grasp Quality Convolutional Neural Network (GQ-CNN) architecture that can estimate the robustness of a given grasp specified as a location and angle in the image and height above a table. We ran over 1,000 blinded physical evaluations with GQ-CNNs trained with various parameters and subsets of the Dex-Net 2.0 dataset, and found that our GQ-CNN grasp planner improves 300% in computation time and 150% in success rate over point cloud registration when grasping unknown objects. Furthermore, our method had zero false positives out of 29 grasps classified as robust on novel objects, suggesting the GQ-CNN prediction is conservative.

In future work, we plan to study hybrid methods to improve performance that use our GQ-CNN as an initialization to adaptively acquire physical grasping examples using active learning or Multi-Armed Bandits. We also plan to research how to include additional sensing modalities such as color images, point clouds from multiple viewpoints, and tactile data to take into account material properties, full object geometry, and object use cases. Furthermore, we plan to release a subset of our code and dataset in the future to facilitate further research and comparisons.

[2]https://youtu.be/9eqAxk95I3Y

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, 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, Google, Cisco, Autodesk, and IBM. 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 Pieter Abbeel, Brenton Chu, Roy Fox, David Gealy, Ed Johns, Sanjay Krishnan, Animesh Garg, Sergey Levine, Pusong Li, Matt Matl, Stephen McKinley, Andrew Reardon, Nan Tian, and Sammy Staszak.

Appendix A
Parameters of Graphical Model

Our graphical model is illustrated in Fig. 2 and models $p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})$ as the product of a state distribution $p(\mathbf{x})$, an observation model $p(\mathbf{y}|\mathbf{x})$, a grasp candidate model $p(\mathbf{g}|\mathbf{x})$, and a grasp success model $p(S|\mathbf{g}, \mathbf{x})$.

We model the state distribution as $p(\mathbf{x}) = p(\gamma)p(\mathcal{O})p(T_o|\mathcal{O})p(T_c)$. We model $p(\gamma)$ as a Gaussian distribution $\mathcal{N}(0.5, 0.1)$ truncated to $[0, 1]$. We model $p(\mathcal{O})$ as a discrete uniform distribution over 3D objects in a given dataset. We model $p(T_o|\mathcal{O}) = p(T_o|T_s)p(T_s|\mathcal{O})$, where is $p(T_s|\mathcal{O})$ is a discrete uniform distribution over object stable poses and $p(T_o|T_s)$ is uniform distribution over 2D poses: $\mathcal{U}([-0.1, 0.1] \times [-0.1, 0.1] \times [0, 2\pi))$. We compute stable poses using the quasi-static algorithm given by Goldberg et al. [14]. We model $p(T_c)$ as a uniform distribution on spherical coordinates $r, \theta, \varphi \sim \mathcal{U}([0.65, 0.75] \times [0, 2\pi) \times [0.05\pi, 0.1\pi])$, where the camera optical axis always intersects the center of the table.

Our distribution over grasps is a uniform distribution over pairs of antipodal points on the object surface that are parallel to the table plane. We sample from this distribution for a fixed coefficient of friction $\mu = 0.6$ and reject samples outside the friction cone or non-parallel to the surface.

We model images as $\mathbf{y} = \alpha * \hat{\mathbf{y}} + \epsilon$ where $\hat{\mathbf{y}}$ is a rendered depth image created using OSMesa offscreen rendering. We

model α as a Gamma random variable with shape$= 1000.0$ and scale=0.001. We model ϵ as Gaussian Process noise drawn with measurement noise $\sigma = 0.005$ and kernel bandwidth $\ell = \sqrt{2}px$.

We compute grasp robustness metrics using the graphical model and noise parameters of [55].

APPENDIX B
ANTIPODAL GRASP SAMPLING

Our antipodal grasp sampling method is designed to sample antipodal grasps specified as a planar pose, angle, and height with respect to a table. The algorithm is detailed in Algorithm 1. We first threshold the depth image to find areas of high gradient. Then, we use rejection sampling over pairs of pixels to generate a set of candidate antipodal grasps, incrementally increasing the friction coefficient until a desired number of grasps is reached in case the desired number cannot be achieved with a smaller friction coefficient. We convert antipodal grasps in image space to 3D by assigning discretizing the gripper height between the height of the grasp center pixel relative and the height of the table surface itself.

This grasp sampling method is used for all image based grasp planners in the paper. We used $M = 1000$, K set to the intrinsics of a Primesense Carmine 1.08, T_c determined by chessboard registration, $g = 0.0025m$, $\mu_\ell = 0.4$, $\delta_\mu = 0.2$, $N = 1000$, and $\delta_h = 0.01m$.

Input: Depth image $\mathbf{y}$, Number of grasps M, Camera Intrinsics Matrix K, Camera pose T_c, Depth gradient threshold g, Min friction coef μ_ℓ, Friction coef increment δ_μ, Max samples per friction coef N, Gripper height resolution δ_h
Result: $\mathcal{G}$, set of candidate grasps
`// Compute depth edges`
$G_x = \nabla_x \mathbf{y}, G_y = \nabla_y \mathbf{y}$;
$\mathcal{E} = \{\mathbf{u} \in \mathbb{R}^2 : G_x(\mathbf{u})^2 + G_y(\mathbf{u})^2 > g\}$;
`// Find antipodal pairs`
$\mathcal{G} = \{\}$, $i = 0$, $j = 0$;
while $|\mathcal{G}| < M$ *and* $\mu <= 1.0$ **do**
  $\mathbf{u}, \mathbf{v}$ =UniformRandom($\mathcal{E}, 2$);
  if *Antipodal(*$\mathbf{u}, \mathbf{v}, \mu$*)* **then**
    `// Compute piont in world coordinates`
    $\mathbf{c} = 0.5 * (\mathbf{u} + \mathbf{v})$;
    $\mathbf{p}_c$ =Deproject($K, \mathbf{y}, \mathbf{c}$);
    $\mathbf{p} = T_c * \mathbf{p}_c$;
    $h = \mathbf{p}.z$;
    `// Add all heights`
    while $h > 0$ **do**
      $\mathcal{G} = \mathcal{G} \cup \{\mathbf{g}(\mathbf{u}, \mathbf{v}, h)\}$;
      $h = h - \delta_h$;
    end
  end
  $i = i + 1$, $j = j + 1$;
  `// Update friction coef`
  if $j >= N$ **then**
    $\mu = \mu + \delta_\mu$;
    $j = 0$;
  end
end
return $\mathcal{G}$;

Algorithm 1: Antipodal Grasp Sampling from a Depth Image

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Ravi Balasubramanian, Ling Xu, Peter D Brook, Joshua R Smith, and Yoky Matsuoka. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *Robotics, IEEE Transactions on*, 28(4):899–910, 2012.

[3] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377, 2010.

[4] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesisa survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

[5] Peter Brook, Matei Ciocarlie, and Kaijen Hsiao. Collaborative grasp planning with multiple object representations. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2851–2858. IEEE, 2011.

[6] Lillian Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3875–3882. IEEE, 2012.

[7] I-Ming Chen and Joel W Burdick. Finding antipodal point grasps on irregularly shaped objects. *IEEE transactions on Robotics and Automation*, 9(4):507–512, 1993.

[8] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer, 2014.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] Renaud Detry, Carl Henrik Ek, Marianna Madry, and Danica Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 601–608. IEEE, 2013.

[11] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[14] Ken Goldberg, Brian V Mirtich, Yan Zhuang, John Craig, Brian R Carlisle, and John Canny. Part pose statistics: Estimators and experiments. *IEEE Transactions on Robotics and Automation*, 15 (5):849–857, 1999.

[15] Corey Goldfeder and Peter K Allen. Data-driven grasping. *Autonomous Robots*, 31(1):1–20, 2011.

[16] Corey Goldfeder, Matei Ciocarlie, Hao Dang, and Peter K Allen. The columbia grasp database. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1710–1716. IEEE, 2009.

[17] Marcus Gualtieri, Andreas ten Pas, Kate Saenko, and Robert Platt. High precision grasp pose detection in dense clutter. *arXiv preprint arXiv:1603.01564*, 2016.

[18] Menglong Guo, David V Gealy, Jacky Liang, Jeffrey Mahler, Aimee Goncalves, Stephen McKinley, and Ken Goldberg. Design of parallel-jaw gripper tip surfaces for robust grasping. In *Proc.*

IEEE Int. Conf. Robotics and Automation (ICRA), 2017.

[19] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4731–4740, 2015.

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[22] Alexander Herzog, Peter Pastor, Mrinal Kalakrishnan, Ludovic Righetti, Jeannette Bohg, Tamim Asfour, and Stefan Schaal. Learning of grasp selection based on shape-templates. *Autonomous Robots*, 36(1-2):51–65, 2014.

[23] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865. IEEE, 2011.

[24] Edward Johns, Stefan Leutenegger, and Andrew 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.

[25] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015.

[26] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.

[27] Ben Kehoe, Akihiro Matsukawa, Sal Candido, James Kuffner, and Ken Goldberg. Cloud-based robot grasping with the google object recognition engine. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4263–4270. IEEE, 2013.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[29] Michael Laskey, Jeffrey Mahler, Zoe McCarthy, Florian T Pokorny, Sachin Patil, Jur van den Berg, Danica Kragic, Pieter Abbeel, and Ken Goldberg. Multi-armed bandit models for 2d grasp planning with uncertainty. In *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*. IEEE, 2015.

[30] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2017.

[31] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[32] Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, et al. Opengrasp: a toolkit for robot grasping simulation. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 109–120. Springer, 2010.

[33] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *arXiv preprint arXiv:1603.02199*, 2016.

[34] Yun Lin and Yu Sun. Grasp planning to maximize task coverage. *The International Journal of Robotics Research*, 34(9):1195–1210, 2015.

[35] Jeffrey Mahler, Brian Hou, Sherdil Niyaz, Florian T Pokorny, Ramu Chandra, and Ken Goldberg. Privacy-preserving grasp planning in the cloud. In *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*, pages 468–475. IEEE, 2016.

[36] Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken 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.

[37] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. Characterizations of noise in kinect depth images: A review. *IEEE Sensors journal*, 14(6):1731–1740, 2014.

[38] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[39] John W Miller, Rod Goodman, and Padhraic Smyth. On loss functions which minimize to conditional expected values and posterior probabilities. *IEEE Transactions on Information Theory*, 39(4):1404–1408, 1993.

[40] Luis Montesano and Manuel Lopes. Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions. *Robotics and Autonomous Systems*, 60(3):452–462, 2012.

[41] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Detecting object affordances with convolutional neural networks. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2765–2770. IEEE, 2016.

[42] John Oberlin and Stefanie Tellex. Autonomously acquiring instance-based object models from experience. In *Int. S. Robotics Research (ISRR)*, 2015.

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

[44] Lerrel Pinto, James Davidson, and Abhinav Gupta. Supervision via competition: Robot adversaries for learning tasks. *arXiv preprint arXiv:1610.01685*, 2016.

[45] Florian T Pokorny and Danica Kragic. Classical grasp quality evaluation: New algorithms and theory. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3493–3500. IEEE, 2013.

[46] Florian T Pokorny, Kaiyu Hang, and Danica Kragic. Grasp moduli spaces. In *Proc. Robotics: Science and Systems (RSS)*, 2013.

[47] Lorenzo Porzi, Samuel Rota Bulo, Adrian Penate-Sanchez, Elisa Ricci, and Francesc Moreno-Noguer. Learning depth-aware deep representations for robotic perception. *IEEE Robotics and Automation Letters*, 2016.

[48] Domenico Prattichizzo and Jeffrey C Trinkle. Grasping. In *Springer handbook of robotics*, pages 671–700. Springer, 2008.

[49] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1316–1322. IEEE, 2015.

[50] Alberto Rodriguez, Matthew T Mason, and Steve Ferry. From caging to grasping. *The International Journal of Robotics Research*, page 0278364912442972, 2012.

[51] Andrei A Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*, 2016.

[52] Fereshteh Sadeghi and Sergey Levine. (cad)2 rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[53] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-*

tion, pages 1352–1359, 2013.

[54] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[55] Daniel Seita, Florian T Pokorny, Jeffrey Mahler, Danica Kragic, Michael Franklin, John Canny, and Ken Goldberg. Large-scale supervised learning of the grasp robustness of surface patch pairs. In *Proc. IEEE Int. Conf. on Simulation, Modeling, and Programming of Autonomous Robots (SIMPAR)*. IEEE, 2016.

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

[57] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, and Trevor Darrell. Adapting deep visuomotor representations with weak pairwise constraints. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.

[58] Jacob Varley, Jonathan Weisz, Jared Weiss, and Peter Allen. Generating multi-fingered robotic grasps via deep learning. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4415–4420. IEEE, 2015.

[59] Jacob Varley, Chad DeChant, Adam Richardson, Avinash Nair, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. *arXiv preprint arXiv:1609.08546*, 2016.

[60] Matthew Veres, Medhat Moussa, and Graham W Taylor. Modeling grasp motor imagery through deep conditional generative models. *arXiv preprint arXiv:1701.03041*, 2017.

[61] Jonathan Weisz and Peter K Allen. Pose error robust grasping from contact wrench space metrics. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 557–562. IEEE, 2012.

[62] Walter Wohlkinger and Markus Vincze. Shape-based depth image to 3d model matching and classification with inter-view similarity. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4865–4870. IEEE, 2011.

[63] Walter Wohlkinger, Aitor Aldoma, Radu B Rusu, and Markus Vincze. 3dnet: Large-scale object class recognition from cad models. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5384–5391. IEEE, 2012.

[64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[65] Ziang Xie, Arjun Singh, Justin Uang, Karthik S Narayan, and Pieter Abbeel. Multimodal blending for high-accuracy instance recognition. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2214–2221. IEEE, 2013.

[66] Jincheng Yu, Kaijian Weng, Guoyuan Liang, and Guanghan Xie. A vision-based robotic grasping system using deep learning for 3d object recognition and pose estimation. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 1175–1180. IEEE, 2013.

[67] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *arXiv preprint arXiv:1609.05143*, 2016.