Learning to Detect Multi-Modal Grasps for Dexterous Grasping in Dense Clutter

Matt Corsaro[1], Stefanie Tellex[1], George Konidaris[1]

***Abstract*— We propose an approach to multi-modal grasp detection that jointly predicts the probabilities that several types of grasps succeed at a given grasp pose. Given a partial point cloud of a scene, the algorithm proposes a set of feasible grasp candidates, then estimates the probabilities that a grasp of each type would succeed at each candidate pose. Predicting grasp success probabilities directly from point clouds makes our approach agnostic to the number and placement of depth sensors at execution time. We evaluate our system both in simulation and on a real robot with a Robotiq 3-Finger Adaptive Gripper and compare our network against several baselines that perform fewer types of grasps. Our experiments show that a system that explicitly models grasp type achieves an object retrieval rate 8.5% higher in a complex cluttered environment than our highest-performing baseline.**

I. Introduction

Grasping is one of the most important open problems in robotics—it is the most essential skill for pick-and-place tasks and a prerequisite for many other manipulation tasks, such as tool use [1]. If robots are to one day perform the complex manipulation tasks humans are capable of in varying home and workplace environments, they must first master grasping.

Humans use multiple types of grasps, depending on the object, the task, and the scene [2]. A human may perform a large-diameter power grasp to stably grasp the handle of a heavy jug, but a precision sphere grasp to lift a golf ball off the ground. If the clutter around an object precludes one particular grasp type, humans simply switch to another. It is therefore natural that the ability to use multiple grasp modalities would substantially improve robots' ability to grasp a wide range of objects, especially in dense clutter. However, state-of-the-art grasp detection systems typically detect pincher grasps exclusively, and are evaluated using small objects and two-finger parallel-jaw grippers [3, 4, 5, 6]. Existing grippers are capable of executing multi-finger dexterous grasps better suited to stably grasping both small and larger, heavier objects; the grasps a Robotiq 3-Finger Adaptive Gripper is capable of executing are demonstrated in Figure 1. Several grasp detection approaches are applicable to multi-finger grippers, but are only capable of performing one type of grasp [7], or do not explicitly model grasp type [8]. Some have taken grasp type into consideration, but are evaluated on singulated objects [9], rely on human-labeled data [10], or return fingertip placement for fully actuated fingers [11]. Furthermore, these systems are not evaluated in dense clutter.

[1]Department of Computer Science, Brown University {mcorsaro, stefie10, gdk}@cs.brown.edu

We propose a data-driven grasp detection framework that, given partial depth data and a grasp pose, jointly predicts the grasp success probabilities of several types of grasps. We train a deep neural network to perform this joint classification using a dataset containing grasp candidates generated from real point clouds and grasp labels generated in simulation. Given a point cloud—captured from an arbitrary number of depth sensors in arbitrary poses—along with a grasp pose, our network outputs a probability for each available grasp modality. These values reflect the probability that the corresponding type of grasp would succeed at the given pose.

We evaluate our system both in simulation and experimentally on a Robotiq 3-Finger Adaptive Gripper. We first evaluate our system on a held-out test set from simulated data to show that our network efficiently learns to jointly predict grasp type when compared to a larger ensemble of networks. On a real robot, our system clears objects from cluttered tabletop piles containing objects of varying sizes. To show the usefulness of multiple grasp modalities in dense clutter, we compare against several ablations of our network capable of performing fewer grasp types, and find that a system capable of multiple grasp types clears more objects than baselines that use fewer.

II. Background

With recent advances in deep learning, data-driven grasp detectors have proven effective for generating parallel-jaw grasps for two-finger grippers. Given visual information, these systems return an end effector pose at which an executed grasp would likely be successful. Most state-of-the-art parallel-jaw grasp detectors, such as ten Pas et al. [3] and Mahler et al. [4], follow a two-step proposal-evaluation model. A proposal function $\mathbf{PROP} : P \rightarrow G$, implemented as a heuristic [3] or a generative network [12], first generates a large set of 6-DoF end effector poses $G \subseteq \mathbb{SE}(3)$ from a point cloud $P \subseteq \mathbb{R}^3$. A grasp evaluation neural network $\mathbf{EVAL} : g \in G \rightarrow [0, 1]$ then maps each g to a probability.

Another common approach is to train a neural network to predict optimal actions using a reinforcement-learning framework. Works such as Ibarz et al. [6] and Levine et al. [13] train their systems using real robot data, which is time-consuming to produce. Though such systems can achieve state-of-the-art grasp success rates, their reliance on reinforcement learning makes them brittle; the same camera configuration used while training is required at test time. Furthermore, modifying the system to grasp a specified object is not straightforward as it is in proposal-evaluation systems, where the proposal step can be easily modified

(a) Wide Power (b) Wide Precision (c) Basic Power (d) Basic Precision (e) Pincher

Fig. 1: The five main grasp types achievable using the Robotiq 3-Finger Adaptive Gripper. Our system detects a grasp type and pose that would lead to a successful grasp, enabling it to more effectively clear clutter.

without adjusting a reward function or retraining. Though both of these types of systems enable two-finger parallel-jaw grippers to grasp some objects, these grippers are capable of executing only simple pincher grasps.

Data-driven grasp detection frameworks have also been applied to perform multi-finger dexterous grasping. However, these systems are either capable of performing only fingertip or precision grasps [14, 7], or use supervised [8, 15, 16, 17, 18] or reinforcement learning [19] to evaluate or predict wrist poses and finger pre-grasp parameters, but do not explicitly model grasp type.

A few recent works predict grasp stability for multiple grasp types. Lu and Hermans [9] train two classifiers to predict power and precision success probabilities from a shared embedding for a 4-finger Allegro Hand. Each classifier is evaluated separately on singulated objects on which power grasps are always preferred. As our system jointly classifies candidates of each grasp type at a given position, it returns both a predicted optimal grasp pose and type. We evaluate our system in a cluttered real-world scenario where multiple types of grasps can be necessary to clear the scene. Deng et al. [10] and Santina et al. [20] use human-labeled data to train a neural network to predict grasp type, while our system learns to use grasp types from simulated grasping data, avoiding human bias or error. Both systems are not evaluated in dense clutter. Varley et al. [11] employ a hybrid approach, using a deep neural network to guide fingertip placement and a grasp planning simulator to localize gripper placement. They define a set of canonical grasp types based on the most common finger pre-poses in their simulated training set. Planning fingertip placement can be difficult for underactuated grippers; to perform a power grasp with an underactuated gripper, each finger's more proximal links would make contact with the object first, making the final distal link placement less relevant. Their system is also not evaluated in dense clutter. Osa et al. [21] use hierarchical reinforcement learning to select a grasp type and grasp location. They maintain a dataset of successful grasps for each grasp type and match new point clouds to this dataset using ICP. Their system is not evaluated in clutter. As they employ a reinforcement-learning framework, a higher-level controller could not request a grasp type or target object as it could with ours. Our approach is capable of executing multiple grasp types, allowing it to successfully grasp objects in a variety of real-world, cluttered experimental scenarios.

III. Learning to Detect Multi-Modal Grasps

Though parallel-jaw grasp detectors have proven successful, two-finger grasps can be insufficient when a robot deals with large, heavy objects. Grasp detectors designed for multi-finger grippers detect grasps of a single type, and those that can explicitly utilize multiple grasp types have not been proven to enable a robot to clear a pile of dense clutter. Our system demonstrates the usefulness of multiple grasp types when picking objects of varying sizes from piles of dense clutter using the proposal-evaluation paradigm commonly used in grasp detection systems [3]. The proposal function $\mathbf{PROP} : P \rightarrow G$ generates a set of 6-DoF end effector poses $G \subseteq \mathbb{SE}(3)$ from a partial point cloud $P \subseteq \mathbb{R}^3$. Unlike the grasp evaluators $\mathbf{EVAL} : g \in G \rightarrow [0, 1]$ used in related works that map a grasp pose to a single probability [3, 4], our grasp evaluation neural network $\mathbf{EVAL} : g \in G \rightarrow [0, 1]^n$ maps each g to a vector of n success probabilities, each corresponding to a different grasp type. This architecture enables us to jointly predict the probabilities of success for multiple grasp types at a given g.

We generate grasp pose candidates $G \subseteq \mathbb{SE}(3)$ using the 6-DoF candidate generation algorithm $\mathbf{GEN} : P \rightarrow G$ proposed by ten Pas and Platt [22]: given a point cloud of an object or cluttered pile of objects represented as a set of 3D points $P \subseteq \mathbb{R}^3$, sample a subset $C \subseteq P$ of k grasp candidate centroid positions. Each $c_s \in C$ is assigned a single orientation $o_s \in \mathbb{SO}(3)$ based on the normals and curvature estimated at c_s; the gripper approach direction is anti-parallel to the estimated normal, and the gripper closes along the curvature. Similar candidates can be sampled by rotating the sampled orientation about the approach direction; we rotate by 90° to generate one additional pose. Finally, candidates causing the gripper to collide with P are pruned. The candidate generation algorithm returns a set of

k proposed candidates $\mathbf{GEN}(P) = G$ where $g_s \in G$ and $g_s = \{c_s, o_s\}$.

The second phase of our system evaluates each of the k proposed candidates $g_s \in G$. A deep neural network estimates success probabilities for each grasp type at each g_s, taking P and an encoding of g_s as input. As several recent papers have shown [23, 24], grasp candidates can be efficiently encoded directly from point clouds recorded from arbitrary viewpoints using a PointNet-inspired architecture [25]. The encoding layers used in our network are based on the PointConv architecture [26]. We encode a candidate grasp pose by centering P at c_s and aligning P's orientation with o_s. We then crop all points outside the approximate grasping region, represented as a box around the fingers and the area they sweep through. This transformation $\mathbf{TF} : P, g_s \rightarrow P_s$ produces P_s, an encoding of a grasp pose and the object geometry local to it. This transformed, cropped cloud P_s representing a single g_s is then fed to the network $\mathbf{DNN}$, which is illustrated in Figure 2.

The encoding layers in our network consist of four PointConv feature encoding layers. Following der Merwe et al. [23], we reduce the first layer's number of points from 1024 to 512 and the third layer's final multi-layer perception from 128 to 64 units. The output from the fourth encoding layer is then fed through a series of five fully connected layers with ReLU activations. The final fully connected layer outputs a logit pair for each of the n grasp types the gripper is capable of: $\mathbf{DNN} : P_s \rightarrow X_s \in \mathbb{R}^{n \times 2}$. We output two logits per grasp type in order to train the network to perform binary classification on each grasp type and predict whether a grasp of each type would succeed or fail. These logit pairs are passed through n independent softmax functions. The n resulting probabilities corresponding to positive labels, $\sigma(X_s)_{*,1} = s_s \in [0, 1]^n$, can be interpreted as the probabilities that a grasp at g_s of the corresponding grasp type would succeed.

We train $\mathbf{DNN}$ to jointly perform n binary classifications using a summed cross entropy loss function. Joint binary classification is useful in cases where multiple entangled predictions are made from a single input source. By training a single network to perform joint binary classification, our system learns an embedding that efficiently encodes the information required to determine whether each grasp type succeeds given a cloud and grasp pose. Though joint binary classification has been proposed to solve problems such as emotion detection [27], ours is the first robotics application we are aware of that uses it. We define this summed cross entropy loss function (equation 2) as a modified form of the standard cross-entropy loss function for m-class classification,

$$- \sum_{c=0}^{m-1} y_c \; log(b_c), \tag{1}$$

where y_c is 1 if c is the correct label for a given exemplar and 0 otherwise and b_c is the estimated probability that the exemplar is of class c. Given a labeled grasp exemplar $e = \{g, P, l\}$ where $l \in \{0, 1\}^n$ and $P_g = \mathbf{TF}(P, g)$, we compute the summed cross entropy between l and $\sigma(\mathbf{DNN}(P_g)) = Z \in [0, 1]^{n \times 2}$. The summed cross entropy loss for our joint binary classification problem is:

$$- \sum_{i=0}^{n-1} \sum_{c=0}^{1} y_{i,c} \; log(Z_{i,c}), \tag{2}$$

where $y_{i,c}$ is 1 if $c = l_i$ and 0 otherwise. Training details are found in Section III-A.

In our experiments, given some P of an object or a set of objects, we generate a set of grasp candidates $G = \mathbf{GEN}(P, k)$ where $|G| = k$. With $\mathbf{DNN}$ trained to evaluate grasps for a specific gripper, we predict each candidate success probability vector $s_s = \sigma(\mathbf{DNN}(\mathbf{TF}(P, g_s)))_{*,1} \; \forall g_s \in G$ to get $S \in [0, 1]^{k \times n}$ where $s_s \in S$. Finally, we select the grasp pose $g_m = \{c_m, o_m\}$ and grasp type i_m corresponding to the maximum entry in S that is collision free. When executing this grasp, the gripper is first moved to a pre-grasp pose some distance d away from $\{p_m, o_m\}$ along the negative approach direction. Finally, the gripper is moved to $\{p_m, o_m\}$ and the fingers are closed to complete the grasp.

A. Dataset Generation & Network Training

In order to train $\mathbf{DNN}$, a dataset of grasp exemplars E where $e = \{g, P, l\} \in E$ and $l \in \{0, 1\}^n$ is required. The BigBIRD dataset [28] contains a set of real partial point clouds captured from 600 viewpoints on a set of common household products and a complete mesh for each object. BigBIRD is a popular dataset for training grasp detection systems since no simulation-to-real transfer is required with real point clouds. Our grasp dataset is generated from 20 BigBIRD objects and 12,000 point clouds. We generate a set of grasp candidates G from these clouds using GEN.

Each candidate is then assigned a label $l \in \{0, 1\}^n$. A label is generated by attempting each grasp type i at g in simulation and recording whether or not the grasp succeeds. We perform grasps in the Drake simulation environment [29] to accurately simulate the Robotiq 3-Finger Adaptive Gripper and its grasp types. The simulated graspable object models are based on the complete BigBIRD object models. Each object model is placed on a plane, the gripper model is placed at the grasp pose specified by the sampled candidate g and grasp type i. After the gripper executes a grasp on the object and the plane is removed from the scene, l_i is set to 1 if the object remains between the gripper's fingers or 0 if it falls out of the grasp. Candidate grasps at which no grasp type cause a collision between the gripper and object or table before execution are kept in the dataset, and those that cause collisions are removed. The resulting dataset E contains over 36,000 grasp candidates, each assigned labels l generated by simulating over 180,000 grasp attempts using $n = 5$ grasp types. This dataset is not perfectly balanced between positive and negative labels; the percentages of positive labels for each grasp type are 79%, 43%, 81%, 73%, and 58%. Though our labels are generated on singulated objects, ten Pas et al. [3] showed that systems trained with grasps on singular objects generalize well to real-world clutter.

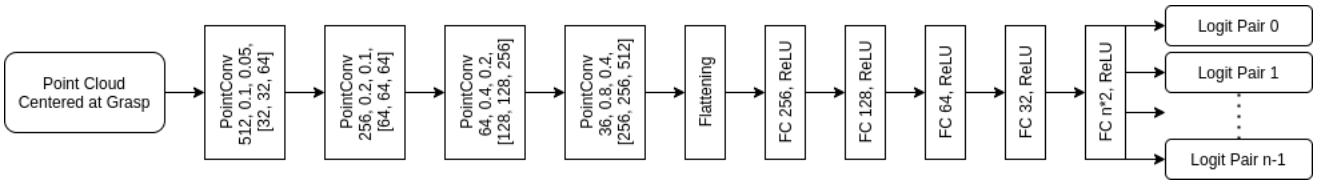

Fig. 2: Network diagram of **DNN**. PointConv layer parameters listed are number of points, radius, sigma, and MLP sizes. Fully connected layer parameters are output size.

The network architecture described in Section III is then trained to perform joint binary classification using the summed cross entropy loss function defined in equation 2. This network is implemented in TensorFlow and trained on a single GeForce GTX 1080 Ti. We use a learning rate of 10^{-5}, a batch size of 16, and the Adam optimizer to train the network. Like ten Pas et al. [3], the cloud input to our network is comprised of two point clouds; one cloud is the BigBIRD cloud used to generate a candidate, and the second is the cloud captured from the same viewing angle 54° away from the first. As the BigBIRD dataset is generated by capturing point clouds of objects from five fixed viewing angles and rotating the object in 3° increments on a turntable, these secondary clouds are available in the BigBIRD dataset. Since our network takes transformed and cropped point clouds P_s as input and classifies 6-DoF grasp candidates, the number of point clouds captured and their viewing angles are arbitrary. We choose to capture two point clouds at training and test time to sufficiently cover the workspace when generating grasp representations P_s, but the configurations need not be the same.

B. Grasp Types

A grasp taxonomy for any multi-finger robotic gripper could be derived to determine the number of grasp types n that it is capable of, much like the one Feix et al. [2] present to categorize 33 grasp types humans are capable of. Though our framework is compatible with any robot gripper and its n, we train and evaluate our system with the Robotiq 3-Finger Adaptive Gripper, a 4-DoF, 11-jointed underactuated gripper. Unlike the fully articulated grippers used in related work [9], the Robotiq 3-Finger Adaptive Gripper is designed specifically to perform different types of grasps [30]. Each 3-jointed finger is controlled by one motor, and an additional actuator adjusts the orientation of the non-thumb fingers. To avoid self-collision, the gripper's controller allows for three discrete operating modes. The non-thumb fingers are parallel in **basic** mode, spread apart in **wide** mode, and brought together in **pincher** mode. The gripper is capable of performing two types of grips: a **fingertip** or precision grip occurs when the distal links grip an object, while an **encompassing** or power grip occurs when the proximal links first contact an object, causing the fingers to wrap around it. The grip executed depends on the distance from the gripper's palm to the target object. In basic and wide mode, both encompassing and fingertip grips are possible, while in pincher mode, fingertip grips emulate parallel-jaw grippers. These operating modes and grip types enable the gripper to execute five types of grasps, illustrated in Figure 1: **basic power** and **wide power** are useful for firmly grasping large objects, **basic precision** and **wide precision** can grasp objects from a surface, and a precision **pincher** that emulates a parallel-jaw gripper is useful for precise tabletop grasps on small objects.

The mechanical design of the Robotiq 3-Finger Adaptive Gripper enables it to execute multiple types of grasps with a single simple control policy without the need for tactile sensors or joint encoders. It enables us to parameterize grasp type over the operating mode and distance from the object to the gripper. To execute a grasp of a given type i_m at candidate $g_m = c_m, o_m$, we assign the gripper a pose $\{p_m, o_m\}$. We define p_m to be the position of the point at the center of the gripper's palm. The orientation of this pose is the same as the candidate's pose. As c_m is a point sampled from the cloud P, assigning $p_m = c_m$ would result in a collision. p_m is instead set some distance d away from c_m along the negative approach direction depending on the grip type required to execute a grasp of type i_m: $p_m = c_m - o_m d$. To execute an encompassing grasp, p_m should be close to c_m so the fingers' proximal links, first make contact with the object and wrap around it. p_m should be sufficiently far from c_m during a fingertip grasp so the fingers' distal links would likely make contact with the object. We therefore set $d_e = 1.9$ cm to perform an encompassing grip when executing a basic or wide power grasp, and set $d_f = 8.22$ cm to perform a fingertip grip when executing a basic precision, wide precision, or pincher grasp.

IV. Measuring Network Generalization

We first evaluate our system by testing its performance on several held-out datasets. In the first scenario, the test set is comprised of 15% of the grasp candidates in our dataset, selected at random, while the remaining 85% are used to train the network. The second, more difficult scenario tests how well the system generalizes to unseen objects. Here, the test set contains all grasp candidates from 15% of the objects in the dataset, while the grasps on the remaining objects are used to train the network. In each scenario, we compare our architecture that outputs $n = 5$ predictions per pose from a shared embedding (COMBINED) to a similar architecture that uses an ensemble of n individual deep networks to predict grasp success for each grasp type (SEPARATE). These individual networks are a naive approach that uses n times as many parameters as the combined network, but

provide an upper bound to compare our system against. Table I shows the test-set classification accuracies of our system and the baseline trained and tested using the two dataset divisions. These results are each averaged over three random seeds used to divide the dataset.

Though test-set accuracy demonstrates how well the system learns, when executed on a real robot, selecting false positives can cause a low grasp success rate. Since the system chooses to execute the one grasp with the highest predicted success probability, the grasp may fail if an incorrectly classified false positive is selected. False negatives, however, are not as detrimental to the grasp success rate since the system will choose only one of the many grasps expected to be successful. It is, therefore, also important to verify the system's precision and F1 score.

As COMBINED is trained to predict success for all grasp types, all statistics are reported at the epoch at which average accuracy across all grasp types is maximum; the individual grasp type accuracies are not necessarily maximum at this epoch. For SEPARATE, since each network is trained with only one grasp type, each accuracy, precision, and F1 score are reported at the epoch at which that grasp type's accuracy is maximum. The reported average accuracy is the average of these maximum accuracies. Despite this, when learning these binary classifiers jointly from a shared embedding, the average classification accuracy decreases by only 0.4% when test objects have been seen and 3.0% when the training and test object sets are exclusive. Furthermore, our system achieves a higher precision in the case where test candidates are selected at random. These experiments show that jointly training individual classifiers from a shared PointConv embedding enables our system to more efficiently classify grasp poses with a negligible loss in performance compared to a similar set of networks with five times as many parameters.

V. Clearing a Cluttered Table

Fig. 3: Objects used in real-world experiments, none of which are found in our simulated training set.

We perform real-robot experiments to measure how much multi-modal grasps help to clear objects of varying sizes from a cluttered table. We capture two point clouds from fixed locations, then remove all points within a threshold of the known table plane. As described in Section III, we generate a set of 400 grasp candidate poses G using **GEN**, then check each for collisions, both in PyBullet with a simplified mesh and between the cloud and a simplified gripper model. Like ten Pas et al. [3], we also filter candidates with an insufficient number of points in the graspable region between the fingers. This is achieved by counting the number of points in the box each finger would sweep through as the gripper closes. We then evaluate the remaining candidates with our trained **DNN**. As detailed in Section III, we choose to execute grasp g_m of type i_m with predicted success probability S_m using our Robotiq 3-Finger Adaptive Gripper. If our motion planner fails to find a path to g_m, we execute the next most likely to succeed grasp.

Because related works are evaluated with a variety of datasets and often simpler experimental scenarios, and implemented on different robot hardware, it is difficult to compare our system with them directly. To examine the benefits of multiple grasp types when grasping in dense clutter, we compare our system to two baseline ablations representative of related work whose deep networks have not been trained to assess all five grasp types. The first, 1Type, predicts only the probability that a pincher grasp would succeed, and is representative of systems designed for parallel-jaw grippers [3, 4]. The second, 2Type, predicts whether $n = 2$ grasp types, basic power and basic precision, would succeed; this is representative of the framework defined by Lu and Hermans [9]. Our system, 5Type, models all $n = 5$ Robotiq grasp types.

Fig. 4: Ten small and medium objects in a cluttered pile surrounded by three large, upright objects.

In each of our experimental trials, three large, upright objects are placed around a pile of ten small and medium-sized objects. This scenario is designed to challenge the system, as it may depend on all five grasp types to clear the table. An example of the clutter our system clears is shown in Figure 4. The objects used in these experiments, an augmented segment of the YCB dataset [31] containing six large, 17 medium, and six small objects, did not appear in the training set and can be seen in Figure 3.

Our procedure follows that of ten Pas et al. [3]; a random selection of small and medium objects is placed in a box, shaken, and dumped into a cluttered pile on a table; large items are placed around the pile after dumping the box to ensure they remain upright. The system attempts to grasp objects until either 1) the same type of failure on the same object with the same grasp type fails three times in a row, 2) the system fails to generate reachable grasps in three consecutive attempts, or 3) all objects are removed from the table. If the system fails to find a feasible grasp or the

Split	Arch.	# Params	Avg Acc	Avg Prec	Avg F1	T1 Acc	T2 Acc	T3 Acc	T4 Acc	T5 Acc
Rand.	COMBINED	10.4M	0.867	0.879	0.897	0.913	0.802	0.908	0.824	0.886
	SEPARATE	51.9M	0.871	0.878	0.9	0.922	0.804	0.908	0.833	0.886
Obj.	COMBINED	10.4M	0.829	0.869	0.881	0.841	0.805	0.847	0.820	0.835
	SEPARATE	51.9M	0.859	0.876	0.902	0.915	0.818	0.854	0.838	0.869

TABLE I: Simulation test-set performance for our joint grasp type classifier (COMBINED) and an ensemble of individual networks (SEPARATE). # Params shows the number of learnable parameters. Average test-set accuracy, precision, and F1 score at convergence are followed by accuracy for all $n = 5$ grasp types (wide power, wide precision, basic power, basic precision, pincher).

motion planner fails to find paths to the top 25 grasps, we repeat the candidate generation process up to two more times, each time proposing twice as many candidates. A grasp is successful if one or more objects are lifted from the scene and moved towards a box, and do not fall from the gripper until the fingers are opened. If an object leaves the workspace during an unsuccessful grasp or during a grasp on a different object, it is placed back in the scene near where it left, abutting as many other objects as possible. If an unforeseen collision occurs during a grasp or placement execution, the disturbed objects are reset and the attempt is not counted. Each system is presented with the same objects as the other systems in each of the ten trials, but in a different random configuration. We report both the grasp success rate (number of successful grasps divided by number of attempted grasps) with number of attempts and overall object removal rate (number of objects removed from table at the end of a trial divided by initial number of objects); results are in Table II, and an example trial is shown in the accompanying video.

	Success Rate	#Attempts	Removal Rate
5Type	0.808	125	**0.808**
2Type	0.692	120	0.654
1Type	**0.825**	114	0.723

TABLE II: Real-world performance on the experimental scenario averaged across ten trials.

VI. Discussion and Conclusion

As seen in Table II, our system with access to all five grasp types, 5Type, outperforms the ablations of our system, 2Type and 1Type, when grasping items from cluttered scenes surrounded by large objects. 5Type achieved the highest object removal rate by successfully clearing nearly all scenes. In some trials, though **GEN** generated grasps on the small Lego or duck, the motion planner failed to find a path to these grasps. A common failure mode of the system, which is a common failure mode in other grasp detection systems [3], was that it attempted to grasp multiple objects at once. The 5Type system suffers from this issue more than the baselines because it has access to wide-type grasps that spread the fingers out and are more likely to contact multiple cluttered objects. The issue could be alleviated with an off-the-shelf depth image segmentation algorithm and an additional filtering step in GEN. Other failure modes of our system include false positives and objects moving as the fingers close. The system also struggled with the heavy shampoo bottle in two of the five trials it appeared in. Because our grasp evaluation network makes predictions based only on local geometry, it incorrectly predicted that unstable grasps would succeed. Since our system has no notion of grasp history, it became stuck in a local minima and unsuccessfully attempted similar grasps three times in a row, ending the trials. However, in three of the trials, it correctly used power grasps to stably lift the heavy bottle.

Since 2Type is incapable of performing pincher grasps, it often fails to find feasible grasp candidates on small objects that fit between the gripper's spread fingers. The system encountered difficulty with the rice box in one trial and the pear in another, objects that the 5Type system never failed to grasp. However, because this ablation did not have access to the weak but precise pincher grasps, it was able to successfully grasp the heavy shampoo bottle in three of the five trials it appeared in. The 1Type system outperformed the 2Type system in these experiments because pincher grasps succeeded on the small or medium objects that made up the majority of our object set. Even most of the large objects were light enough that they could be lifted with a pincher grasp. 1Type attempted to grasp multiple objects at once less frequently because the fingers are not spread apart during a pincher grasp. The 1Type system successively failed to lift the heavy shampoo bottle by its cap three times in three different trials, ending the trials prematurely and decreasing the object removal rate.

Our 5Type system was able to choose applicable grasp types for the situation to clear each scene more completely. It used a wide power grasp to stably grasp the top of the drill, and another to perform a spherical grasp on the soccer ball, one of the largest medium-sized objects. It used basic power grasps when faced with the heavy shampoo. The 2Type system also selected basic power grasps for these objects. Overall, our 5Type system executed three wide power, 33 wide precision, six basic power, 38 basic precision, and 45 pincher grasps. Collision-free power grasps were rarely generated on the small and medium-sized objects because they lie close to the tabletop. As the large objects were often partially occluded by the adjacent clutter pile, their larger graspable areas were hidden. Though our 5Type

system chose to use fingertip grasps in the many applicable scenarios, it used power grasps to lift the heavy objects the 1Type system often could not.

The modular nature of our system will allow us to include our proposed grasp success prediction network as a useful module in other systems, such as an object retrieval system [32]. The inputs or outputs of our system could be partially masked to target specific objects or grasp types without dataset regeneration or network retraining, enabling our system to evaluate grasps in a robot's high-level planner.

The simulation experiments presented in Section IV show that our system is able to efficiently learn to jointly evaluate multiple grasp types for a given grasp candidate. Our real-world experiments show that this architecture enables a robot equipped with a multi-finger gripper to more successfully clear a scene of cluttered objects of various sizes from a tabletop than systems that use fewer grasp types.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant IIS-1717569 and NSF CAREER Award 1844960 to Konidaris, and by the ONR under the PERISCOPE MURI Contract N00014-17-1-2699. The authors wish to thank Ben Burchfiel, Ben Abbatematteo, and Barrett Ames for their helpful feedback and support.

REFERENCES

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, 2021.

[2] T. Feix, J. Romero, H. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 66–77, 2016.

[3] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, pp. 1455–1473, 2017.

[4] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A., and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proceedings of Robotics: Science and Systems*, July 2017.

[5] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation*, June 2020, pp. 6232–6238.

[6] J. Ibarz, D. Kalashnikov, P. Pastor, M. Kalakrishnan, D. Quillen, A. Herzog, S. Levine, V. Vanhoucke, E. Holly, E. Jang, and A. Irpan, "Qt-opt: Scalable deep reinforcement learningfor vision-based robotic manipulation," in *2018 Conference on Robot Learning*, Oct 2018.

[7] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Deep Differentiable Grasp Planner for High-DOF Grippers," in *Proceedings of Robotics: Science and Systems*, July 2020.

[8] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE International Conference on Robotics and Automation*, May 2015, pp. 4304–4311.

[9] Q. Lu and T. Hermans, "Modeling Grasp Type Improves Learning-Based Grasp Planning," *IEEE Robotics and Automation Letters*, vol. 4, pp. 784–791, 2019.

[10] Z. Deng, G. Gao, S. Frintrop, F. Sun, C. Zhang, and J. Zhang, "Attention based visual analysis for fast grasp planning with a multi-fingered robotic hand," *Frontiers in Neurorobotics*, vol. 13, p. 60, 2019.

[11] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2015, pp. 4415–4420.

[12] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Oct 2019.

[13] 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, pp. 421–436, 2018.

[14] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, pp. 2286–2293, 2020.

[15] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning Multi-Fingered Grasps as Probabilistic Inference in a Learned Deep Network," in *International Symposium on Robotics Research*, 2017, pp. 455–472.

[16] P. Schmidt, N. Vahrenkamp, M. Wächter, and T. Asfour, "Grasping of unknown objects using deep convolutional neural networks based on depth images," *2018 IEEE International Conference on Robotics and Automation*, pp. 6831–6838, May 2018.

[17] F. Song, Z. Zhao, W. Ge, W. Shang, and S. Cong, "Learning optimal grasping posture of multi-fingered dexterous hands for unknown objects," in *2018 IEEE International Conference on Robotics and Biomimetics*, Dec 2018, pp. 2310–2315.

[18] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Generating Grasp Poses for a High-DOF Gripper Using Neural Networks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2019, pp. 1518–1525.

[19] B. Wu, I. Akinola, and P. Allen, "Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2019, pp. 1789–1796.

[20] C. D. Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Letters*, pp. 1533–1540, 2019.

[21] T. Osa, J. Peters, and G. Neumann, "Experiments with hierarchical reinforcement learning of multiple grasping policies," in *2016 International Symposium on Experimental Robotics*, 2016.

[22] A. ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," in *2015 International Symposium on Robotics Research*, 2015.

[23] M. V. der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation*, May 2020, pp. 11 516–11 522.

[24] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation*, May 2019, pp. 3629–3635.

[25] C. Qi, H. Su, K. Mo, and L. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul 2017.

[26] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[27] H. He and R. Xia, "Joint binary neural network for multi-label learning with applications to emotion classification," in *Natural Language Processing and Chinese Computing*, 2018, pp. 250–259.

[28] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel, "Bigbird: (big) berkeley instance recognition dataset," in *2014 IEEE International Conference on Robotics and Automation*, May 2014, pp. 509–516.

[29] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[30] Robotiq, *3-Finger Adaptive Robot Gripper Instruction Manual*. [Online]. Available: https://assets.robotiq.com/website-assets/support_documents/document/3-Finger_PDF_20190221.pdf

[31] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, pp. 261–268, 2017.

[32] T. Nguyen, N. Gopalan, R. Patel, M. Corsaro, E. Pavlick, and S. Tellex, "Robot Object Retrieval with Contextual Natural Language Queries," in *Proceedings of Robotics: Science and Systems*, July 2020.