Gripper-Aware GraspNet: End-Effector Shape Context for Cross-Gripper Generalization

Alina Sarmiento, Anthony Simeonov, Pulkit Agrawal
Massachusetts Institute of Technology

***Abstract*— In cluttered, unmodeled environments, many learned manipulation pipelines rely on some inherent knowledge of robot and end-effector extents to predict or solve for feasible grasp poses and motion plans. However, these models become specific to one robot geometry and cannot effectively generalize to other end-effectors that have different feasible grasp distributions. We present Gripper-Aware GraspNet, a learned pipeline for grasping and manipulating unknown objects in highly occluded environments, conditioned on gripper geometry. This method builds off of prior work in learned 6D grasp generation that was previously limited to specific gripper geometries and can predict grasps that utilize a wide range of gripper extents. We demonstrate results on cluttered tabletop picking from a single view pointcloud, and show results that utilize full gripper extents across different end-effectors in simulation. We also show a qualitative improvement on grasp diversity when using different grippers in the real world.**

I. Introduction

Robotic manipulation systems today are rather non-standardized. Constant developments across the stack of low-level controllers, high-level task and motion planners, and end-effector hardware diversify the range of possible system components and capabilities. Because of this, it can be challenging to quickly set up a manipulation system that is compatible with a specific mechanical or software module of focus - having to "reinvent the wheel" at every level of the stack can make iteration and collaboration difficult. As the field develops, there is clear merit in modular contributions that are "plug-and-play" compatible with a variety of workflows and associated full-stack components.

This challenge increases as manipulation tasks become more complex. When unmodeled environments and varying robot geometries are introduced, high-level task and motion planners necessarily become entangled with lower-level perception and control systems: where a robot should grasp an object depends on its task, where it could fit, where would grasp the object stably, and what it can see. For example, a home robot may be tasked with shelving a spice bottle in a cabinet. The robot's morphology may dictate which grasps are feasible, but this will be further constrained by limited perception and access to a cluttered shelf, as well as which grasps are compatible with the task and potential motion plan. The final solution that the robot arrives at will be highly tailored to all of these factors at once. One common approach is to create modules that are specialized to a certain type of gripper, task, object, or scene and then solve the task sequentially, iterating until each module can find a compatible solution to its part of the task.

However, a different approach is to learn the relationship between a robot's capabilities and its environment and task, and predict end-to-end compatible solutions. The bottle-shelving robot may imagine one grasp that works if it's allowed to place the bottle on its side, and another grasp that might work if it can open its fingers rather wide. This approach makes it particularly simple to generalize across different robot geometries and scenes, without needing to retrain or adapt the entire stack to handle changes.

Our aim is to create a manipulation module that is capable of synthesizing information from a perceived scene, robot morphology, and task to propose feasible and useful grasps. More specifically, we are interested in generating 6 degree of freedom (DoF) grasps in a generalized format that are feasible and compatible with any given end-effector and rigid-body rearrangement of a scene, given a single-view depth image of unmodeled clutter. We define "feasible" grasps as fulfilling two requirements: (1) end-effector placement must lead to a stable grasp of the object and (2) the grasp must enable the robot to execute a motion trajectory from start to goal placement without colliding with the surrounding scene.

We adapt the GraspNet framework, which can generate 6-DoF grasps for a specific gripper geometry on singulated object point clouds. A prominent variation of GraspNet is Contact-GraspNet, which adapts grasp generation to cluttered scenes by creating and scoring grasps that correspond to specific points on the perceived scene pointcloud. However, the grasps generated by Contact-GraspNet are still implicitly limited to the distribution of its training set, the ACRONYM dataset of feasible grasps for the Panda Hand.

We tackle three main issues with this line of prior work in 6-DoF grasp generation. (1) We generate a new dataset of 6-DoF grasps on synthetic objects that covers a significantly larger range of gripper shapes. (2) Using this data, we train a model to predict not only grasps but also the required end-effector geometry to make each grasp feasible. (3) We pair our grasp generation model with a convolutional occupancy network [1], which operates as a learned collision checker that generalizes to both different observed scene point clouds and different gripper geometries. Together, this combination of components allows us to apply the same model across different systems and filter grasp predictions at test time based on the geometry of the chosen end-effector and the observed scene geometry.

In summary, the contributions of our work are as follows:

- A 6-DoF grasp generation network capable of predicting both two-finger grasps and the associated end-effector

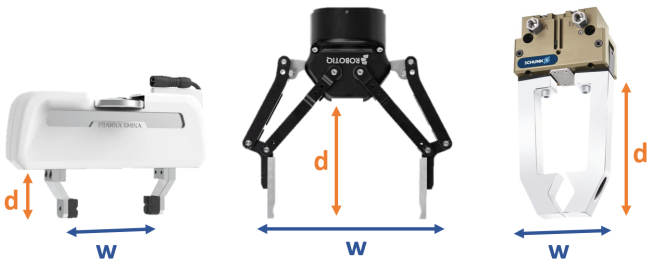


Fig. 1: Gripper parameters. Minimum required grasp width w is the distance between a contact point pair, and finger depth d is the length of finger needed to reach the contacts.

extents required to reach each grasp, given a cluttered, unmodeled scene.

- A manipulation pipeline using learned occupancy nets that determines 6-DoF grasps that are feasible at the start and end of a task.
- Demonstration and evaluation of this system across multiple end-effector geometries, clutter, objects, and tasks.

II. Method

A. Problem Setup

We consider a robot observing a cluttered scene described by pointcloud $\mathbf{P}_{\text{scene}} \in \mathbb{R}^{Nx3}$. Within the scene, the task is to move a rigid target object by a world-frame transformation $\mathbf{T}_{\text{target}} \in \text{SE(3)}$. It is assumed that both the transform and the segmentation mask of the object in $\mathbf{P}_{\text{scene}}$ are known, and the object may therefore be unmodeled and/or partially occluded without consequence. The robot is also equipped with a two-fingered end-effector defined by finger parameters depth d and grasp width range $[w_{\text{min}}, w_{\text{max}}]$. These parameters are visualized in Figure 1.

Given this information, we are interested in generating a set of 6-DoF end-effector poses $\mathbf{T}_{\text{grasp}} \in \text{SE(3)}$ that will achieve both successful grasp of the target object and execution of the task. That is to say, both initial pose $\mathbf{T}_{\text{initial}} = \mathbf{T}_{\text{grasp}}$ and final pose $\mathbf{T}_{\text{goal}} = \mathbf{T}_{\text{target}} \cdot \mathbf{T}_{\text{grasp}}$ must be collision free for the specified gripper in the observed cluttered scene.

B. Gripper-Agnostic Grasp Generation

We begin with an observation that most two-contact grasps are defined as a pair of antipodal contact points constrained by two main parameters: how far apart the contacts can be (the "grasp width" of the end-effector) and how far into the object they can be (the "depth" of the fingers). By jointly predicting grasp poses with the minimum gripper parameters needed to reach each pose, one can simply filter out poses associated to irrelevant parameter values in order to determine which grasp set is probably reachable with the current gripper geometry.

B1. Background: Contact-GraspNet

We modify the Contact-GraspNet (CGN) architecture [2] to generate stable 6-DoF grasps on the visible surface of the target object. We briefly review the architecture of CGN here.

Architecture and Feedforward Grasp Predictions. Given a point cloud $\mathbf{P}_{\text{scene}}$ consisting of 3D points $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^3$, CGN predicts a set of per-point gripper poses $\{\mathbf{T}_i\}_{i=1}^N$, $\mathbf{T}_i \in \text{SE(3)}$, and grasp-success probabilities $\{\hat{s}_i\}_{i=1}^N$, $\hat{s}_i \in [0, 1]$.

Training and Losses. CGN is trained on ground truth grasps obtained from the ACRONYM dataset [3]. Importantly, these are densely annotated grasps of object models which are then arranged into cluttered scenes and filtered for collisions. The loss for the success probability prediction for a given contact point $\mathbf{c}$ is based on whether or not $\mathbf{c}$ is within an ϵ Euclidean distance to any of the ground truth grasp labels for a given scene.

B2. Our Implementation

To adapt $\mathbf{T}_{\text{grasp}}$ to represent any gripper geometry, we recognize that translation $\mathbf{t}$ in $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ is dependent on depth d, which varies between grippers, and on the grasp width range $[w_{\text{min}}, w_{\text{max}}]$. Due to the nature of the ACRONYM dataset, CGN only predicts grasps with a maximum depth of 2.34 cm and predicts grasp widths that vary from 0 to 8 cm, which corresponds to the gripper extents of the Franka Emika Panda Hand. We modify this approach by increasing the possible grasp widths beyond the width of the Panda Hand and including d as an additional predicted value in the final multihead of the model.

This modification requires us to also expand the diversity of label grasps seen in training. We generate and sample a variety of antipodal grasp points on models across object categories, unconstrained by width or depth. From here, we are able to train on this data in the same manner that CGN was trained. More details about our data generation can be found in section III-E.

C. Modifications to CGN Architecture

Often, determining the required width and depth gripper parameters for a given grasp is most heavily reliant on a very local region to the queried contact point. Namely, the larger point neighborhoods encoded by PointNet++ may not be relevant to local parameter prediction, and it can prove useful to re-emphasize a smaller, more grasp relevant region of the pointcloud.

We observed that especially as the possible range of gripper parameters increased, predicting grasp orientation vectors, width, depth, and confidence in parallel could lead to disjoint or unrelated distributions between the outputs and generally make loss convergence rather difficult. Ultimately, we found that implementing a more sequential approach to certain parts of the final prediction outputs allowed for easier learning.

Our model architecture is as follows: As before, we begin by encoding the scene pointcloud with PointNet++ to generate pointwise feature vectors. We use these feature vectors to predict per-point baseline vectors and confidence scores. From here, we use the predicted baseline vectors to extract cylinders of points from the pointcloud along each vector direction. The corresponding pointwise features for each cylinder are combined using max pooling, and this new per-grasp feature is used to predict width, depth, and approach vector outputs.

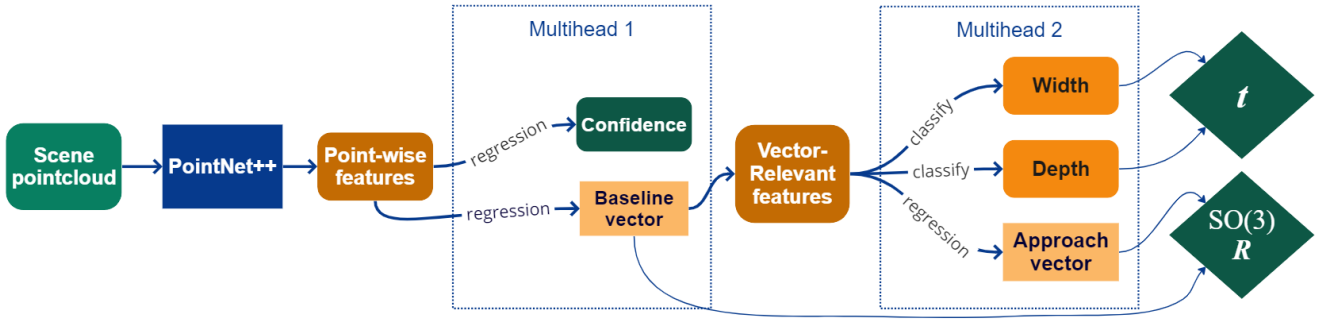


Fig. 2: Our modified architecture for Gripper-Aware GraspNet. We implement a tiered multihead to predict both grasp pose and necessary gripper parameters. For execution, grasps can be filtered to match current available gripper parameters.

We also move from scalar regression for approach vector prediction to discrete classification to better handle multi-modality in feasible grasping distributions (such as being able to grasp from both sides of a cereal box). A summary of this approach can be seen in Figure 2.

D. Collision Prediction

Because our method no longer includes collision checking implicitly in the forward pass (i.e. the gripper geometry is no longer assumed and learned), we include a Convolutional Occupancy Network [1] (CON) that predicts feasibility with a specific gripper geometry. For our purposes, we also modify model training to handle partial views of cluttered scenes.

We create a new dataset based off of the cluttered simulation-rendered scenes already used for training GACGN (section III-E contains further description of scene data generation). A training data sample consists of a single-view pointcloud of the scene and raster points that cover the volume surrounding the scene. Labels consist of binary occupancy scores for each raster point, only considering occupancy of objects that are visible from the camera.

At evaluation time, we create both (1) a "query pointcloud" of points sampled from the current gripper mesh, and (2) an encoding of the scene pointcloud. For every grasp scored highly by GACGN, we transform the gripper pointcloud to the predicted grasp pose and query the occupancy of these points using the CON decoder. Grasps that lead to the gripper points intersecting with the predicted occupancy of the scene are rejected as grasps in collision.

E. Data Generation

Our motivation for generating new training data is to expand the distribution of positively labeled grasps beyond the parameters of the Panda Hand. To do this, we used 67 objects across 19 classes from the ShapeNet object model dataset [4] to sample 100,000 antipodal contact pairs per object mesh. These object classes were selected to reflect common manipulation targets such as household objects. Namely, unusual or uncommon object classes such as "helicopter" were intentionally excluded from the object set – dense grasp annotation was prioritized over diverse object classes. Rather than labeling each contact pair with a gripper-specific Panda Hand 6-DoF pose, we label each pair of contact points with our intended model outputs: the associated vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, grasp width w, and finger depth d.

For training, we render each table scene in Pyrender from a randomly sampled camera angle and label pointcloud points with the closest grasp within a 1cm radius. Due to data storage constraints, our final dataset is comprised of 110 scenes of 4-8 objects, each rendered from 10 camera angles, for a total of 1,100 training data samples.

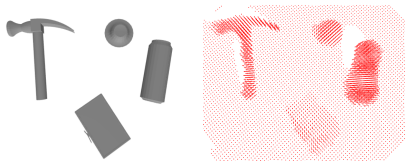
III. Experiments: Design and Setup

Robot, Environment, Task Setup. Our environment consists of a Franka Emika Panda robot arm mounted on a table with 4-7 unmodeled objects placed before it. A statically-mounted depth camera with known extrinsics observes the scene from one randomly sampled view. Across the experiments, the robot is equipped with various end-effectors, namely the Panda Hand, the Panda Hand with custom finger geometries, and the Robotiq 2F-140. For quantitative evaluation, we use PyBullet [5] for large-scale execution in physics simulation. We also qualitatively demonstrate the capabilities of the pipeline on a set of contextually grounded real world applications.

The evaluation task is to successfully execute pick and place of an unmodeled, partially occluded object in the scene. The target object is specified by a ground truth segmentation of the single-view pointcloud, and the object transform is specified as a rigid SE(3) spatial transform in the world frame.

For each method being evaluated, a set of feasible grasps (defined as those above a certain predicted success threshold) is generated for a given scene, object transform, and gripper geometry. Of this proposed grasp set, grasps are sampled and checked with PyBullet's ground truth collision checker. If the grasp pose is found to be feasible with the gripper and clutter for the start and end of a task, then the grasp is executed using off-the-shelf inverse kinematics and motion planning.

Because motion planning and collision avoidance during motion is not a main focus of this work, we use ground truth obstacle avoidance and motion plan with RRT* between fixed waypoints in the scene. For real world executions, we avoid collision with the observed pointcloud by querying the encoding of the scene that we already obtained via the convolutional occupancy net.



Ground truth Our training

Fig. 3: CON predictions. Object meshes are rendered from a single view, occupancy is predicted for dense raster points.

Evaluation Metrics. In evaluating GACGN, we seek to demonstrate the following:

- Capacity to generate feasible grasps for a specific task across a range of gripper geometries
- Improvement of grasp coverage from ACRONYM to better reflect how gripper geometry affects grasp affordances
- Efficacy of convolutional occupancy nets on single-view pointclouds in a collision-checking and motion planning application

We define the following metrics:

Success rate: The ratio of successfully executed grasps out of those proposed by the model, contingent on stable grasp and collision free placement.

Grasp coverage: The extent to which the generated grasps cover the graspable surface of the object. It is often valuable to identify a diverse, feasible grasp set over which other searches can be performed.

One can also quantify *start and end feasibility* of the predicted grasps within the surrounding clutter. This is defined as the percentage of grasps that are defined in the ground truth as collision-free.

Baselines. We compare GACGN against two other methods. CGN-CON refers to grasps that were generated using Contact-GraspNet trained on ACRONYM scenes, and then filtered using a convolutional occupancy net. GACGN generates grasps using our modified version of Contact-GraspNet, filters grasps to a specific gripper parameter set, and then filters this set with the convolutional occupancy net. As an oracle comparison, Labels-CON refers to execution of grasps in the new dataset created to train GACGN. There is merit in evaluating this dataset because it is based on a small set of ray-casting heuristics to determine feasibility, rather than the more rigorous way that ACRONYM grasps were evaluated. GACGN(>Panda) refers to the execution of grasps solely beyond the width and depth feasible by the Panda Hand.

IV. Experiments: Results

Pick and Place. We first evaluate the performance of GACGN on 6-DoF pick and place tasks.

Four to seven object models from the ACRONYM dataset are placed in front of a Panda robot, which is fitted with either the Franka Emika Panda Hand or a larger parallel jaw gripper with double the grasp width and height. Predicted grasps are generated across a single-view pointcloud of the scene, and these grasps are filtered by CON to remove colliding grasps. Of these, the six grasps with highest predicted success are executed by the robot. Success rates of collision free grasps are recorded in Figure 4, and overall execution success rates are recorded in Figure 5.

	Panda Hand	Large Hand
Labels + CON	83%	72%
GACGN	81%	87%
GACGN (>Panda)	–	95%
CGN + CON	100%	–

Fig. 4: Execution success rate for collision free grasps in simulation (30 trials).

	Panda Hand	Large Hand
Labels + CON	65.2%	65.7%
GACGN	78.6%	67.7%
GACGN (>Panda)	–	73.7%
CGN + CON	67.9%	–

Fig. 5: Overall execution success rate across two different end-effectors in simulation (30 trials).

Dataset Evaluation. We evaluate the quality of our training dataset as it compares to ACRONYM. This can be seen in the "Labels + CON" lines of Figures 4 and 5.

Real World. We demonstrate GACGN quantitatively in the real world. As shown in Figure 6, our model is capable of generating grasps that fully utilize the extents of the Robotiq-2F140 gripper, which has almost double the grasp width and depth of the Panda Hand.

Future Work. Thus far, we have only fully evaluated execution success rate across grippers. Future evaluation would include grasp coverage and feasibility across a motion plan and object transformation, as well as across a wider range of grippers.

V. Conclusion

We propose a method for learning grasp generation on single-view, cluttered, novel scenes by adapting the GraspNet framework to a more generalized grasp output format. By using modern occupancy nets as collision checkers, grasps generated in this format are able to effectively generalize grasp generation to a wide range of end-effectors and tasks while only proposing grasps that are relevant and viable.

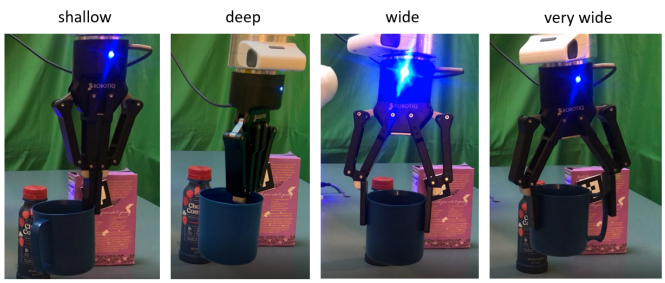


Fig. 6: Various grasps generated for use with the Robotiq-2F140 gripper. Grasps are beyond the extents of the Panda Hand in both width and depth.

REFERENCES

[1] Songyou Peng et al. "Convolutional occupancy networks". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer. 2020, pp. 523–540.

[2] Martin Sundermeyer et al. "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13438–13444.

[3] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. "ACRONYM: A Large-Scale Grasp Dataset Based on Simulation". In: *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*. 2020.

[4] Angel X Chang et al. "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012* (2015).

[5] Erwin Coumans and Yunfei Bai. "Pybullet, a python module for physics simulation for games, robotics and machine learning". In: *GitHub repository* (2016).