

Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations

Zhenyu Jiang[†], Yifeng Zhu[†], Maxwell Svetlik[†], Kuan Fang[‡], Yuke Zhu[†]
[†]The University of Texas at Austin, [‡]Stanford University

Abstract—Grasp detection in clutter requires the robot to reason about the 3D scene from incomplete and noisy perception. In this work, we draw insight that 3D reconstruction and grasp learning are two intimately connected tasks, both of which require a fine-grained understanding of local geometry details. We thus propose to utilize the synergies between grasp affordance and 3D reconstruction through multi-task learning of a shared representation. Our model takes advantage of deep implicit functions, a continuous and memory-efficient representation, to enable differentiable training of both tasks. We train the model on self-supervised grasp trials data in simulation. Evaluation is conducted on a clutter removal task, where the robot clears cluttered objects by grasping them one at a time. The experimental results in simulation and on the real robot have demonstrated that the use of implicit neural representations and joint learning of grasp affordance and 3D reconstruction have led to state-of-the-art grasping results. Our method outperforms baselines by over 10% in terms of grasp success rate. Additional results and videos can be found at <https://sites.google.com/view/rpl-giga2021>

I. INTRODUCTION

Generating robust grasps from raw perception is an essential task for robots to physically interact with objects in unstructured environments. This task demands the robots to reason about the geometry and physical properties of objects from partially observed visual data, infer a proper grasp pose (3D position and 3D orientation), and move the gripper to the desired grasp configuration for execution. Here we consider the problem of 6-DoF grasp detection in clutter from 3D point cloud of the robot’s on-board depth camera. Our goal is to predict a set of candidate grasps on a clutter of objects from partial point cloud for grasping and decluttering.

Robot grasping is a long-standing challenge with decades of research. Pioneer work [12, 45] has cast it as a *geometry-centric* task, typically assuming access to the full 3D model of the objects. Grasps are thereby generated through optimization on analytical models of constraints derived from geometry and physics. In practice, the requirement of ground-truth models has impeded their applicability in unstructured scenes. One remedy is to integrate these model-driven methods with a 3D reconstruction pipeline that builds the object models from perception as the precursor step [2, 37]. However, it demands solving a full-fledged 3D reconstruction problem, an open challenge in computer vision. Motivated by the new development of machine learning, in particular deep learning, recent work on grasping has shifted focus towards a *data-driven* paradigm [3, 19, 30], where deep networks are trained end-

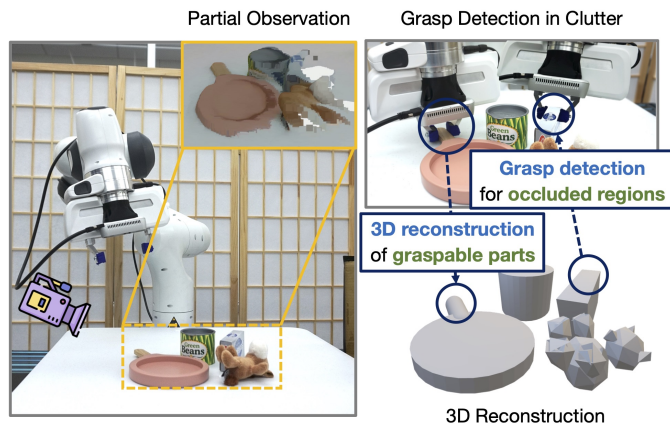


Fig. 1: We harness the synergies between affordance and geometry for 6-DoF grasp detection in clutter. Our model jointly learns grasp affordance prediction and 3D reconstruction. Supervision from reconstruction facilitates our model to learn geometrically-aware features for accurate grasps in occluded regions from partial observation. Supervision from grasp, in turn, produce better 3D reconstruction in graspable regions.

to-end on large-scale grasping datasets, either through manual labeling [18] or self-exploration [19, 41]. Data-driven methods have enabled direct grasp prediction from noisy perception. However, end-to-end deep learning for grasping often suffers from limited generalization within the training domains.

Inspired by the two threads of research on geometry-centric and data-driven approaches to grasping, we investigate the synergistic relations between geometry reasoning and grasp learning. Our key intuition is that *a learned representation capable of reconstructing the 3D scene encodes relevant geometry information for predicting grasp points and vice versa*. In this work, we develop a unified learning framework that enables a shared scene representation for both tasks of grasp prediction and 3D reconstruction, where grasps are represented by a landscape of grasp affordance over the scene. By grasp affordance, we refer to the likelihood of grasp success and the corresponding grasp parameters at each location.

The primary challenge here is to develop a shared representation that effectively encodes 3D geometry and grasp affordance information. Recent work from the 3D vision and graphics communities has shed light on the merits of implicit representations for geometry reasoning tasks [32, 38, 6, 40]. Deep implicit functions define a scene through continuous and differentiable representations parameterized by a deep network. The network maps each spatial location to a corresponding local feature, which can further be decoded to

geometry quantities, such as occupancy [32], signed distance functions [38], probability density, and emitted color [27, 33]. Implicit neural representations have demonstrated state-of-the-art results in a variety of 3D reconstruction tasks [32, 52, 7], due to their ability to represent smooth surfaces in high resolution. On top of the strengths of encoding 3D geometry, the implicit representations are desirable for our problem for two additional reasons: 1) They are differentiable and thus amenable to gradient-based multi-task training of 3D reconstruction and grasp prediction, enabling to learn a shared representation for both tasks; and 2) The representations parameterized by deep networks can adaptively allocate computational budgets to regions of importance. Hence, implicit neural representations can flexibly encode action-related information for the parts of the scene where grasps are likely to succeed.

To this end, we introduce our model: Grasp detection via Implicit Geometry and Affordance (GIGA). We develop a structured implicit neural representation for 6-DoF grasp detection. Our method extracts structured feature grids from the Truncated Signed Distance Function (TSDF) voxel grid fused from the input depth image. A local feature can be computed from the feature grids given a query 3D coordinate. This local feature is used by the implicit functions for estimating the grasp affordance (in the form of grasp quality, grasp orientation, and gripper width of a parallel jaw) and the 3D geometry (in the form of binary occupancy) at the query location. The model is jointly trained in simulation with known 3D geometry and self-supervised grasp trials. With multi-task supervision of affordance and geometry, our model takes advantage of the synergies between them for more robust grasp detection.

We conduct experiments on a clutter removal task [4] in simulation and on physical hardware. In the experiments, multiple objects are piled or packed in clutter, and the robot is tasked to remove the objects by grasping them one at a time. In each round, the robot receives a single-view depth image from the on-board depth camera and predicts a 6-DoF grasp configuration. The ability to reconstruct the 3D scene from a single view enables GIGA to achieve grasp performance on par with multi-view input as employed in prior work [4]. Empirical results have confirmed the benefits of implicit neural representations and the exploitation of the synergies between affordance and geometry. Our model achieved 87.9% and 69.2% grasp success rates on the packed and pile scenes, outperforming 74.5% and 60.7% reported by the state-of-the-art VGN model. We provide qualitative visualizations of the learned grasp affordance landscape over the entire scene, showing that our implicit representations have encoded scene-level context information for collision and occlusion reasoning. Meanwhile, grasp prediction guides the learned implicit representations to produce better reconstruction to the graspable regions of the scene.

We summarize the main contributions of our work below:

- We exploit the synergistic relationships between grasp affordance prediction and 3D reconstruction for 6-DoF grasp detection in clutter.

- We introduce structured implicit neural representations that effectively encode 3D scenes and jointly train the representations with simulated self-supervision.
- We demonstrate significant improvements of our model over the state-of-the-art in the clutter removal task in simulation and on the real robot.

II. RELATED WORK

A. Learning Grasp Detection

Grasping has been studied for decades. Pioneer work has developed analytical methods based on the object models [10, 28, 34, 43, 47, 56]. However, complete models of object geometry and physics are usually unavailable in real-world applications. To bridge these analytical methods with raw perception, prior works have resorted to various 3D reconstruction methods, such as by fitting known CAD models to partial observations [23, 25] or completing full 3D model from partial observations based on symmetry analysis [2, 44]. In recent years, learning methods, especially deep learning models, have gained increased attention for the grasping problem [2, 17, 19, 35, 36, 11]. Dex-Net [30, 31] introduced a two-stage pipeline for top-down antipodal grasping. It first samples candidate 4-DoF grasps. The grasp quality of each grasp candidate is then assessed by a convolutional neural network. 6-DoF GraspNet [36] extends grasp generation to the $SE(3)$ space with a variational autoencoder for grasp proposals on a singulated object. GPD [15, 50] and PointGPD [26] tackled 6-DoF grasp detection in clutter with a two-stage grasp pipeline. VGN [4] predicts 6-DoF grasps in clutter with a one-stage pipeline from input depth images. There is also a line of works that estimate affordance of an object or a scene first and then detect grasps based on estimated affordance [42, 24, 54]. In most of the prior works, deep networks are trained end-to-end with only grasp supervision. In contrast, our method learns a structured neural representation jointly with self-supervised geometry and grasp supervisions.

B. Geometry-Aware Grasping

The intimate connection between grasp detection and geometry reasoning has inspired a line of work on geometry-aware grasping. Bohg and Kragic [1] use a descriptor based on shape context with a non-linear classification algorithm to predict grasps. DGGN [53] regularized grasps through 3D geometry prediction. It learns to predict a voxel occupancy grid from partial observations and evaluates grasp quality from feature of the reconstructed grid. Most relevant to us is PointSDF [51], which learns 3D reconstruction via implicit functions and shares the learned geometry features with the grasp evaluation network. It used a global feature for the shared shape representation, and the quantitative results did not show that geometry-aware features improve grasp performances. On the contrary, our work demonstrates that through the use of structured feature grids and implicit functions, GIGA improves grasp detection and focuses more on the 3D reconstruction of graspable regions via joint training of both tasks.

C. Implicit Neural Representations

Our method is inspired by recent work from the vision and graphics communities on representing 3D objects and scenes with deep implicit functions [6, 32, 38]. Rather than explicit 3D representations such as voxels, point clouds, or meshes, these works have used the isosurface of an implicit function to represent the surface of a shape. By parametrizing these implicit functions with deep networks, they are capable of representing complex shapes smoothly and continuously in high resolution. The most common architecture for deep implicit functions is multi-layer perceptions (MLP), which encode the geometry information of the entire scene into the model parameters of the MLP. However, they have difficulty in preserving the fine-grained geometric details of local regions. To mitigate this problem, hybrid representations have been introduced to combine feature grid structures and neural representations [27, 40]. These structured representations are appealing for the grasping problem, which requires geometry reasoning in local object parts. Our model extends the architecture of convolutional occupancy network [40], the state-of-the-art scene reconstruction model, as the backbone for joint learning of affordance and geometry.

III. PRELIMINARIES

A. Implicit Neural Representations

Implicit neural representations [49] are continuous functions Φ that are parameterized by neural networks and satisfy equations of the form:

$$F(\mathbf{p}, \Phi_\theta) = 0, \quad \Phi_\theta : \mathbf{p} \rightarrow \Phi_\theta(\mathbf{p}) \quad (1)$$

where $\mathbf{p} \in \mathbb{R}^m$ is a spatial or spatio-temporal coordinate, Φ_θ is implicitly defined by relations defined by F , and θ refers to the parameters of the neural networks. As Φ_θ is defined on continuous domains of \mathbf{p} (e.g., Cartesian coordinates), it serves as a memory-efficient and differentiable representation of high-resolution 3D data compared to explicit and discrete representations.

B. Occupancy Networks

Here we introduce Occupancy Network (ONet) [32], one of the pioneer works that learn implicit neural representations for 3D reconstruction. Given an observation of a 3D object, ONet fits a continuous occupancy field defined over the bounding 3D volume of the object. The occupancy field is defined as:

$$o : \mathbb{R}^3 \rightarrow \{0, 1\}. \quad (2)$$

Following Equation (1), the *occupancy function* in ONet is defined as:

$$\Phi_\theta(\mathbf{p}) - o(\mathbf{p}) = 0. \quad (3)$$

The resulting occupancy function $\Phi_\theta(\mathbf{p})$ maps a 3D location $\mathbf{p} \in \mathbb{R}^3$ to the occupancy probability between 0 and 1.

To generalize over shapes, we need to estimate occupancy functions conditioning on a context vector $x \in \mathcal{X}$ computed from visual observations of the shape. A function that conditions on the context $x \in \mathcal{X}$ and instantiates an implicit

function $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ is equivalent to a function that takes a pair $(\mathbf{p}, x) \in \mathbb{R}^3 \times \mathcal{X}$ as input and has a real number as output. The latter function parameterized by a neural network f_θ is the Occupancy Network:

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \rightarrow [0, 1]. \quad (4)$$

Our model builds on top of the architecture of Convolutional Occupancy Networks (ConvONets) [40], which is designed for scene-level 3D reconstruction. ConvONets extend ONet with a convolutional encoder and a structured representation. ConvONets encode input observation (point cloud or voxel grid) into structured feature grids, and condition implicit functions on local features retrieved from these feature grids. They demonstrate more accurate 3D reconstructions of local details in large-scale environments.

IV. PROBLEM FORMULATION

We consider the problem of 6-DoF grasp detection for unknown rigid objects in clutter from a single-view depth image.

A. Assumptions

We assume a robot arm equipped with a parallel-jaw gripper in a cubic workspace with a planar tabletop. We initialize the workspace by placing multiple rigid objects on the tabletop. A single-view depth image taken with a fixed side view depth camera is fused into a Truncated Signed Distance Function (TSDF) and then passed into the model. The model outputs 6-DoF grasp pose predictions and associated grasp quality. We train our model with grasp trials generated in a self-supervised fashion from a physics engine. In simulated training, we assume the pose and shape of the objects are known.

B. Notations

Observations Given the depth image captured by the depth camera with known intrinsic and extrinsic, we fuse it into a TSDF, which is an N^3 voxel grid \mathbf{V} where each cell V_i contains the truncated signed distance to the nearest surface. We believe this additional distance-to-surface information provided by TSDF can improve grasp detection performance. Therefore we convert the raw depth image into TSDF volume \mathbf{V} and pass it into the model.

Grasps We define a 6-DoF grasp g as the grasp center position $\mathbf{t} \in \mathbb{R}^3$, the orientation $\mathbf{r} \in SO(3)$ of the gripper, and the opening width $w \in \mathbb{R}$ between the fingers.

Grasp Quality A scalar grasp quality $q \in [0, 1]$ estimates the probability of grasp success. We learn to predict the grasp quality of a grasp with binary success labels of executing the grasp trial in simulation.

Occupancy For an arbitrary point $\mathbf{p} \in \mathbb{R}^3$, the occupancy $b \in \{0, 1\}$ is a binary value indicating whether this point is occupied by any of the objects in the scene.

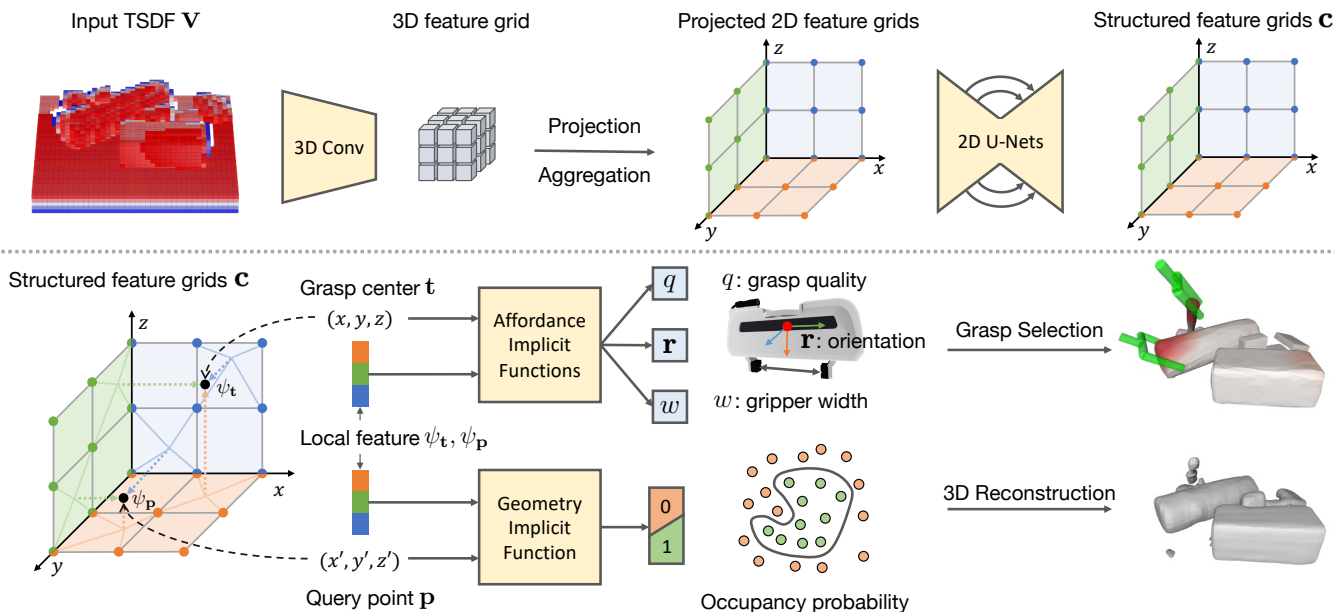


Fig. 2: Model architecture of GIGA. The input is a TSDF fused from the depth image. After a 3D convolution layer, the output 3D voxel features are projected to canonical planes and aggregated into 2D feature grids. After passing each of the three feature planes through three independent U-Nets, we query the local feature at grasp center/occupancy query point with bilinear interpolation. The affordance implicit functions predict grasp parameters from the local feature at the grasp center. The geometry implicit function predicts occupancy probability from the local feature at the query point.

C. Objectives

The primary goal is to detect 6-DoF grasp configurations that allow the robot arm to successfully grasp and remove the objects from the workspace. To foster multi-task learning of geometry reasoning and grasp affordance, we perform simultaneous 3D reconstruction. Given the input observation \mathbf{V} , our goal is to learn two functions:

$$\begin{aligned} f_a : \mathbf{t} &\rightarrow q, \mathbf{r}, w, \\ f_g : \mathbf{p} &\rightarrow b. \end{aligned} \quad (5)$$

The first function f_a maps from a grasp center to the rotation, gripper width, and grasp quality of the best grasp at that location. Once f_a is trained, we can select which grasp to execute based on the grasp quality at a set of grasp centers. The second function f_g maps any point in the workspace to the estimated occupancy value at that point. We can extract a 3D mesh from the learned occupancy function with the Marching Cube algorithm [29].

V. METHOD

We now present GIGA, a learning framework that exploits synergies between affordance and geometry for 6-DoF grasp detection from partial observation. We learn grasp affordance prediction and 3D occupancy prediction jointly with shared feature grids and a unified implicit neural representation. Figure 2 illustrates the overall model architecture.

A. Structured Feature Grids

To jointly learn the grasp affordance prediction and 3D reconstruction, we need to extract a shared feature from the input

TSDF. In previous implicit-function-based 3D reconstruction works [32, 38, 51], a flat global feature is extracted from the input observation and it is used to infer the implicit function. However, this simple representation falls short of encoding local spatial information, nor for incorporating inductive biases such as translational equivariance into the model [40]. For this reason, they lead to overly smooth surface reconstructions. For grasp detection, local geometry reasoning is even more important. The spatial distribution of grasp affordance is very sparse and highly localized: most viable grasps cluster around a few graspable regions. Therefore, we adopt the encoder architecture from ConvONets [40] and learn to extract structured feature grids from partial observation.

Our encoder takes as input a TSDF voxel field and processes it with a 3D CNN layer to obtain a feature embedding for every voxel. Given these features, we construct planar feature representations by performing an orthographic projection onto a canonical plane for each input voxel. The canonical plane is discretized into pixel cells. Then we aggregate the features of voxels projected onto the same pixel cell using average pooling, which gives us a feature plane. The projection operation greatly reduces the computation cost while keeping the spatial distribution of feature points. We apply this feature projection and aggregation process to all three canonical frames. The feature plane might have empty features due to the incomplete observation. We therefore process each of these feature planes with a 2D U-Net [46] which is composed of a series of down-sampling and up-sampling convolutions with skip connections. The U-Net integrates both local and global information and acts as a feature inpainting network. The output feature grids denoted as \mathbf{c} , are shared for affordance and geometry learning.

B. Implicit Neural Representations

A unified representation of affordance and geometry facilitates the joint learning of grasp detection and 3D occupancy prediction. Recent works in the 3D vision community have shown that deep implicit functions are capable of representing shapes in a continuous and memory-efficient way and they can adaptively allocate computational and memory resources to regions of importance [14, 27, 40]. Therefore, we are motivated to use implicit neural representations to encode both affordance and geometry. As both require reasoning about local geometry details, we condition the deep implicit functions on local features. We query the local feature from the shared feature planes \mathbf{c} . Given a query position \mathbf{p} , we project it to each feature plane and query the features at the projected locations using bilinear interpolation. Unlike ConvONets [40] where features from different planes are summed together, we concatenate them instead in order to preserve the spatial information. The local feature $\psi_{\mathbf{p}}$ can be formulated as:

$$\psi_{\mathbf{p}} = [\phi(\mathbf{c}_{xy}, \mathbf{p}_{xy}), \phi(\mathbf{c}_{yz}, \mathbf{p}_{yz}), \phi(\mathbf{c}_{xz}, \mathbf{p}_{xz})], \quad (6)$$

where $\mathbf{c}_{ij}, \mathbf{p}_{ij}$ ($i, j \in \{x, y, z\}$) are the plane feature and point projected onto the corresponding plane, and ϕ means bilinear interpolation of feature plane at the projected point.

a) *Affordance Implicit Functions*: The affordance implicit functions represent the grasp affordance field of grasp parameters and grasp quality. They map the grasp center \mathbf{t} to grasp parameters (\mathbf{r} and w) and the grasp quality metric q . These implicit neural representations enable learning directly from data with continuous grasp centers. In contrast, VGN has to snap grasp centers to the nearest voxel as it uses an explicit voxel-based grasp field. The snapping operation leads to information loss while our model does not.

We implement implicit neural representations as functions that map a pair of point and corresponding local feature ($\mathbf{t}, \psi_{\mathbf{t}}$) to target values. We parametrize the functions with small fully-connected occupancy networks with multiple ResNet blocks [16]. Grasp center, grasp orientation, and gripper width are output from three separate implicit functions:

$$\begin{aligned} f_{\theta_q}(\mathbf{t}, \psi_{\mathbf{t}}) &\rightarrow [0, 1], \\ f_{\theta_r}(\mathbf{t}, \psi_{\mathbf{t}}) &\rightarrow SO(3), \\ f_{\theta_w}(\mathbf{t}, \psi_{\mathbf{t}}) &\rightarrow [0, w_{max}]. \end{aligned} \quad (7)$$

Here w_{max} is the maximum gripper width. $\theta_q, \theta_r, \theta_w$ are neural networks parameters of each deep implicit function. We represent grasp orientation with quaternions.

b) *Geometry Implicit Function*: Our geometry implicit function maps from an arbitrary query point \mathbf{p} inside the bounded volume to the occupancy probability $o(\mathbf{p})$ at the point. Similar to affordance implicit functions, we learn a function that predicts occupancy based on input point coordinate and the corresponding local feature:

$$f_{\theta_o}(\mathbf{p}, \psi_{\mathbf{p}}) \rightarrow [0, 1]. \quad (8)$$

Notice that the query points of occupancy \mathbf{p} can be different from the grasp center points \mathbf{t} .

C. Grasp Detection

GIGA takes as input a TSDF voxel grid, a grasp center, and multiple occupancy query points and predicts grasp parameters corresponding to the grasp center and occupancy probabilities at the query points.

Given the trained GIGA model, we use a sampling procedure to select the final grasp pose. Grasp affordance is implicitly defined by the learned neural networks, so we need to query it from the learned implicit functions. To cover all possible graspable regions, we discretize the volume of the workspace into voxel grids and use the position of all the voxel cells as grasp centers. Then we query the grasp quality and grasp parameters corresponding to these grasp centers in parallel. Next, we mask out impractical grasps and apply non-maxima suppression as done in VGN [4]. Finally, we select a grasp with the highest quality if the quality is beyond a threshold. If no grasp has the quality above the threshold, we don't make grasp predictions and give up the current scene.

D. Training

The loss for training consists of two parts: the affordance loss and the geometry loss. For the affordance loss, we adopt the same training objective as VGN [4]:

$$\mathcal{L}_A(\hat{g}, g) = \mathcal{L}_q(\hat{q}, q) + q(\mathcal{L}_r(\hat{\mathbf{r}}, \mathbf{r}) + \mathcal{L}_w(\hat{w}, w)). \quad (9)$$

Here \hat{g} denotes predicted grasp parameters and g denotes ground-truth parameters. $q \in \{0, 1\}$ is the ground-truth grasp quality (0 for failure, 1 for success) and $\hat{q} \in [0, 1]$ is the predicted grasp quality. \mathcal{L}_q is a binary cross-entropy loss between the predicted and ground-truth grasp quality. \mathcal{L}_w is the ℓ_2 -distance between predicted gripper width \hat{w} and ground-truth one w . For orientation, \mathcal{L}_{quat} between predicted quaternion $\hat{\mathbf{r}}$ and target quaternion \mathbf{r} is given by $\mathcal{L}_{quat}(\hat{\mathbf{r}}, \mathbf{r}) = 1 - |\hat{\mathbf{r}} \cdot \mathbf{r}|$. However, the parallel-jaw gripper is symmetric, which means a grasp configuration corresponds to itself after rotated by 180° about the gripper's wrist axis. To handle this symmetry during training, both mirrored rotations \mathbf{r} and \mathbf{r}_π are deemed as ground-truth. Thus the orientation loss is defined as:

$$\mathcal{L}_r(\hat{\mathbf{r}}, \mathbf{r}) = \min(\mathcal{L}_{quat}(\hat{\mathbf{r}}, \mathbf{r}), \mathcal{L}_{quat}(\hat{\mathbf{r}}, \mathbf{r}_\pi)). \quad (10)$$

We only supervise the grasp orientation and gripper width when a grasp is successful ($q = 1$).

For the geometry loss, we apply the standard binary cross-entropy loss between the predicted occupancy $\hat{o} \in [0, 1]$ and the ground-truth occupancy label $o \in \{0, 1\}$. The loss is denoted as \mathcal{L}_G . The final loss is simply the direct sum of the affordance loss and geometry loss:

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_G. \quad (11)$$

We implement the models with Pytorch [39] and train the models with the Adam optimizer [22] and a learning rate of 2×10^{-4} and batch sizes of 32.



Fig. 3: Visualization of packed (left) and pile (right) scenarios. In the packed scenario, objects are placed on the table at their canonical poses. In the pile scenario, objects are dropped on the workspace with random poses.

TABLE I: Quantitative results of clutter removal. We report mean and standard deviation of grasp success rates (GSR) and declutter rates (DR). HR denotes high resolution.

Method	Packed		Pile	
	GSR (%)	DR (%)	GSR (%)	DR (%)
SHAF [13]	56.6 ± 2.0	58.0 ± 3.0	50.7 ± 1.7	42.6 ± 2.8
GPD [15]	35.4 ± 1.9	30.7 ± 2.0	17.7 ± 2.3	9.2 ± 1.3
VGN [4]	74.5 ± 1.3	79.2 ± 2.3	60.7 ± 4.2	44.0 ± 4.9
GIGA-Aff	77.2 ± 2.3	78.9 ± 1.7	67.8 ± 3.0	49.7 ± 1.9
GIGA	83.5 ± 2.4	84.3 ± 2.2	69.3 ± 3.3	49.8 ± 3.9
GIGA (HR)	87.9 ± 3.0	86.0 ± 3.2	69.8 ± 3.2	51.1 ± 2.8

VI. EXPERIMENTS

We study the efficacy of synergies between affordance geometry on grasp detection in clutter. Specifically, we would like to investigate three complementary questions: 1) Can the structured implicit neural representations encode action-related information for grasping? 2) Does joint learning of geometry and affordance improve grasp detection? 3) How does grasp affordance learning impact the performance of 3D reconstruction?

A. Experimental Setup

Our model is trained in a self-supervised manner with ground-truth grasp labels collected from physical trials in simulation and occupancy data obtained from the object meshes. The use of TSDF enables zero-shot transfer of our model from simulation to a real Panda arm from Franka Emika.

a) Simulation Environment: Our simulated environment is built on PyBullet [8]. We use a free gripper to sample grasps in a $30 \times 30 \times 30 \text{cm}^3$ tabletop workspace. For a fair comparison, we use the same object assets as VGN, including 303 training and 40 test objects from different sources [5, 20, 21, 48]. The simulation grasp evaluations are all done with the test objects, which are excluded from training.

We collect grasp data in a self-supervised fashion in two type of simulated scenes, *pile* and *packed* as in VGN [4]. In the pile scenario, objects are randomly dropped to a box of the same size as the workspace. Removing the box leaves a cluttered pile of objects. In the packed scenario, a subset of taller objects is placed at random locations on the table at their canonical pose. Examples of these two scenarios are shown in Figure 3. Once the scene is created, we randomly sample grasp centers and grasp orientations near the surface of

the objects and execute these grasp samples in simulation. We store grasp parameters and the corresponding outcomes of the grasp trials and balance the dataset by discarding redundant negative samples.

We collect the occupancy training data in the same scenes where grasp trials are performed. Upon the creation of a simulation scene, we query the binary occupancy of a large number of points uniformly distributed in the cubic workspace as the training data.

b) Camera Observations: To evaluate the model’s robustness against noise and occlusion in real-world clutter, we assume that the robot perceives the workspace by a single depth image from a fixed side view. If we denote the length of the workspace as l and use the spherical coordinates with the workspace center as the origin, the viewpoint of the virtual camera points towards the origin at $r = 2l, \theta = \frac{\pi}{3}, \phi = 0$. To expedite sim-to-real transfer, we add noise to the rendered images in simulation using the additive noise model [30]:

$$\mathbf{y} = \alpha \hat{\mathbf{y}} + \epsilon, \quad (12)$$

where $\hat{\mathbf{y}}$ is a rendered depth image, α is a Gamma random variable with $k = 1000$ and $\theta = 0.001$ and ϵ is a Gaussian Process noise drawn with measurement noise $\sigma = 0.005$ and kernel bandwidth $l = \sqrt{2}px$. The input to the our algorithm is a $40 \times 40 \times 40$ TSDF [9] fused from this noisy single-view depth image using the Open3D library [55].

c) Grasp Execution: We select top grasps to execute by querying grasp parameters from the learned implicit functions with a set of grasp centers. For a fair comparison with VGN [4], our **GIGA** model samples $40 \times 40 \times 40$ uniformly distributed grasp centers in the workspace and query the grasp parameters. However, our implicit representations are continuous, so we can query grasp samples in arbitrary resolutions. In **GIGA (HR)**, we query at a higher resolution of $60 \times 60 \times 60$.

We use a set of clutter removal scenarios to evaluate GIGA and other baselines. Each round, a pile or packed scene with 5 objects is generated. We take a depth image from the same viewpoint as training. The grasp detection algorithm generates a grasp proposal given the input TSDF. We execute the grasp and remove the grasped object from the workspace. If all objects are cleared, two consecutive failures happen, or no grasp is detected, we terminate the current scene. Otherwise, we collect the new observation and predict the next grasp. In our experiments, grasp proposals with a predicted grasp quality below 0.5 are discarded.

B. Baselines

We compare the performance of our method and the following baselines:

- **SHAF:** We use the highest point heuristic [13] by classic work of grasping in clutter, rather than the learned grasp quality, for grasp selection. Among all grasps of quality over 0.5 predicted by our model, we select the highest one to grasp from the clutter.

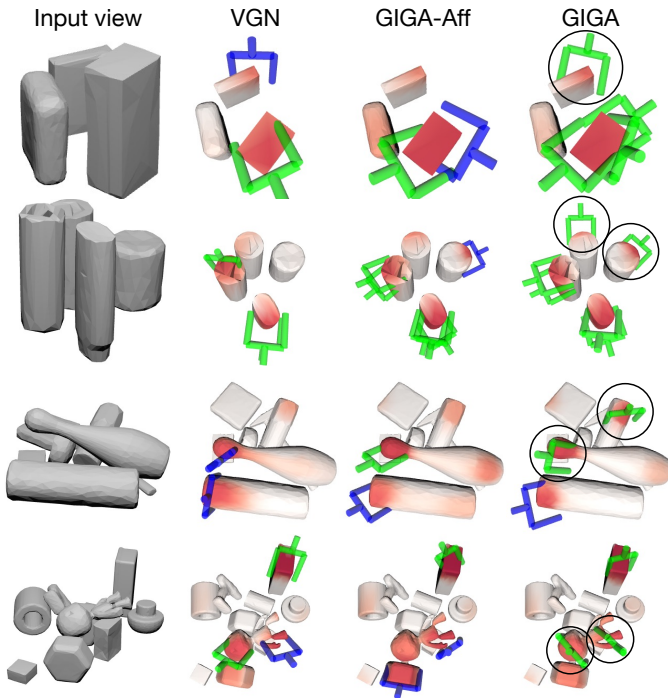


Fig. 4: Visualization of the grasp affordance landscape and predicted grasps. **Red** indicates that the method predicts high grasp affordance near the corresponding area. **Green** indicates successful grasps and **Blue** failures. The circles highlight interesting examples, such as asymmetric affordance heatmaps and highly occluded objects.

- **GPD** [15]: Grasp Pose Detection, a two-stage 6-DoF grasp detection algorithm that generates a large set of grasp candidates and classifies each of them.
- **VGN** [4]: Volumetric Grasping Network, a single-stage 6-DoF grasp detection algorithm that generates a large number of grasp parameters in parallel given input TSDF volume.
- **GIGA-Aff**: An ablated version of our method with only affordance implicit function branch. The network is trained with only grasp supervision but no reconstruction.

Performance is measured using the following metrics averaged over 100 simulation rounds: 1) Grasp success rate (GSR), the ratio of success grasp executions; and 2) Declutter rate (DR), the average ratio of objects removed. The original VGN uses multi-view inputs, we re-train the VGN model on the same single-view data we used for training GIGA for fair comparisons.

C. Grasp Detection Results

We report grasp success rate and declutter rate for different scenarios in Table I. We can see that GIGA and GIGA-Aff outperform other baselines in almost all scenarios and metrics. Even though GIGA-Aff does not utilize the synergies between affordance and geometry and is trained without geometry supervision, it still outperforms the state-of-the-art VGN baseline. We attribute this to the high expressiveness of our implicit neural representations. VGN snaps the grasp center to voxel grid cells during training. In contrast, our implicit

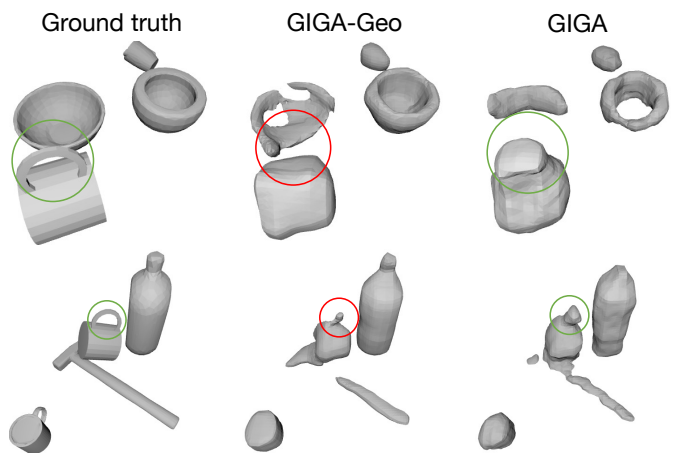


Fig. 5: Qualitative 3D reconstruction results of a scene rendered from the top view. The circles highlight the contrast.

neural representations learn to fit grasp affordance field with continuous functions. It allows us to query grasp parameters at a higher resolution as done in GIGA (HR). GIGA (HR) gives the highest performance in all cases.

Next, we compare the results of GIGA-Aff with GIGA. In the pile scenario, the gain from geometry supervision is relatively small (around 2% grasp success rate). However, in the packed scenario, GIGA outperforms GIGA-Aff by a large margin of around 5%. We believe this is due to the different characteristics of these two scenarios. From Figure 3, we can see that in the packed scene, some tall objects standing in the workspace would occlude the objects behind them and the occluded objects are partially visible. We hypothesize that in this case, the geometrically-aware feature representation learned via geometry supervision facilitates the model to predict grasps on partially visible objects. Such occlusion is, however, less frequent in pile scenarios. These results demonstrate that synergies between affordance and geometry improve grasp detection, especially in the presence of occlusion.

We show examples of the learned grasp affordance landscape (grasp quality at different locations of the scene) and predicted top grasps in Figure 4. Our first observation is that our method is able to take into account the context of the scene and predict collision-free grasps. It often predicts asymmetric affordance heatmap over symmetric objects (*e.g.*, boxes and cylinders), where one part of the object is graspable (red) while the symmetric part is not (grey), and grasping these grey regions is likely to lead to a collision with the neighboring objects. This indicates that our model encodes scene-level information from training on self-supervised grasp trials and takes into account practical constraints when making grasp predictions.

Our second observation is that GIGA produces more diverse and accurate grasp detections compared with the baselines. In the first two rows, we visualize the affordance and grasp of two packed scenes. In regions marked by the black circle, GIGA predicts high affordance and more accurate grasps. The objects in these regions are largely occluded from the view but

TABLE II: Quantitative results of 3D reconstruction. We see that models trained with grasp supervision tend to have better reconstruction performance near graspable regions than average by a larger margin.

Method	IoU (%)	IoU-Grasp (%)	$\Delta\%$ (IoU-Grasp-IoU)
GIGA-Detach	53.2	68.8	+15.6
GIGA	70.0	78.1	+8.1
GIGA-Geo	80.0	84.0	+4.0

are easy to grasp given full 3D information. For example, the thin box in the circle of the first row affords a wide range of grasp poses. These easy grasps are not detected in GIGA-Aff or VGN due to occlusion but are successfully predicted by GIGA. The last two rows show the affordance landscape and top grasps for two pile scenes. We see that baselines without the multi-task training of 3D reconstruction tend to generate failed or no grasp, whereas GIGA produces more diverse and accurate grasps due to the learned geometrically-aware representations.

Another advantage of GIGA is the improvement of the grasping efficiency. The planning time (time between receiving depth image(s) and returning a list of grasps) of GIGA and original multi-view VGN is similar, which are 46ms (GIGA) and 46ms (VGN) on an NVIDIA Titan RTX 2080Ti. However, GIGA uses single-view depth input, which can be collected from a fixed camera instantaneously. Meanwhile, the original VGN model has to collect multi-view inputs by moving the wrist-mounted camera along a scanning trajectory. The estimated time cost of the scanning process before each grasp is about 16-20s. Therefore, GIGA grasps at least one order of magnitude faster than the VGN.

To further verify GIGA’s ability to predict grasps from partial observation, we retrain GIGA on a fused TSDF of multi-view depth images taken from six randomly distributed viewpoints along a circle in the workspace, the same setup as VGN [4]. Compared to the single-view setup, acquiring multi-view observations is more time-consuming and subject to environmental constraints. Yet with multi-view inputs, GIGA achieves 88.8% and 69.6% grasp success rates in packed and pile scenarios, which is on par with the performances of GIGA on single-view input reported in Table I. We hypothesize that supervision from 3D reconstruction plays a key role for GIGA to reason about the occluded parts of the scene.

D. 3D Reconstruction

In the previous section, we have demonstrated that geometry supervision benefits grasp affordance learning. We now examine how grasp affordance learning affects 3D reconstruction. To this end, we evaluate the performance of 3D reconstruction on pile scenes. We further compare two ablated versions of our method: **GIGA-Geo**, which is trained solely for 3D reconstruction without the affordance implicit functions branch, and **GIGA-Detach**, which is trained to reconstruct the scene on the fixed feature grids from GIGA-Aff.

We extract mesh from the learned implicit function through marching cube algorithm [29]. We use the Volumetric IoU metric for evaluation, which is the intersection over the union

TABLE III: Quantitative results of clutter removal in real world. We report GSR, DR, the number of successful grasps, and the number of total grasp trials (in bracket).

Method	Packed		Pile	
	GSR (%)	DR (%)	GSR (%)	DR (%)
VGN [4]	77.2 (61/79)	81.3	79.0 (64/81)	85.3
GIGA	83.3 (65/78)	86.6	86.9 (73/84)	97.3

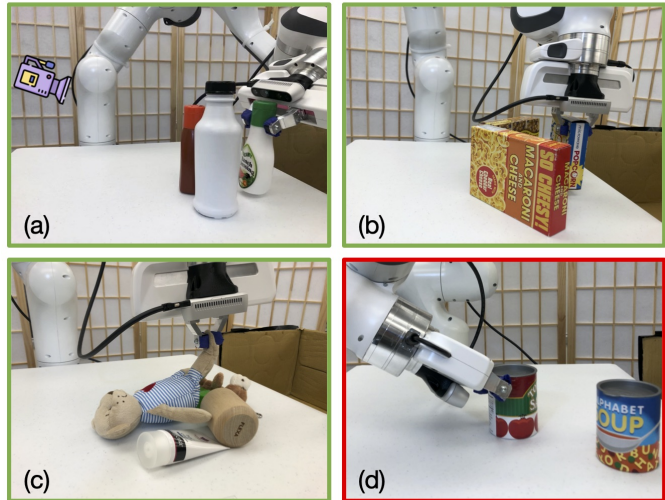


Fig. 6: Examples of real-world grasps by GIGA. (a) and (b) show two examples of grasps of partially occluded objects. The tilted camera looks at the workspace from the left. (c) shows an example where GIGA picks up the bear doll from a localized graspable part. (d) illustrates a typical failure where the gripper slips off the object due to the small contact surface.

between predicted mesh and ground-truth one. For comparison, we also report the IoU near graspable parts given by grasp trials, denoted as IoU-Grasp. Specifically, we sample the successful grasps and evaluate the IoU in the regions between the two fingers of these grasps.

Table II shows the results of 3D reconstruction. GIGA-Geo gives the highest overall IoU and the highest IoU in graspable parts. This result is unsurprising since GIGA-Geo is trained to specialize in this task. In contrast, both GIGA and GIGA-Detach are trained to predict grasps, of which the spatial distribution is highly localized. Thus, we expect them to be biased their representational resources towards these graspable regions.

Interesting observations arise when we look at the performance gaps between IoU-Grasp and IoU for the same method. GIGA-Detach gives the largest delta, then comes the GIGA. It indicates that scene features learned from grasp supervision dynamically allocate the representational budget to actionable regions. In Figure 5, GIGA-Geo shows more accurate overall reconstruction results than GIGA, consistent with the quantitative results. In comparison, some parts are missing in the GIGA results, such as the bottom of the bowl. However, the missing parts are mostly non-graspable parts, while graspable parts such as the edges of the bowls are clearly reconstructed. In addition, we see that GIGA successfully reconstructs the handle of the cup, while GIGA-Geo either

ignores the handle (in the first row) or reconstructs a small proportion of the handle. These results imply that GIGA attends to the action-related regions of high grasp affordance, driven by the grasp supervision.

E. Real Robot Experiments

Finally, we test our method in the clutter removal experiments on the real hardware. 15 rounds of experiments are performed with GIGA and VGN for both the Packed and Pile scenarios respectively. In each round, 5 objects are randomly selected and placed on the table. In each grasp trial, we pass the TSDF from a side view depth image to the model and execute the predicted top grasp. A grasp trial is marked as a success if the robot grasps the object and places it into a bin next to the workspace. We repeatedly plan and execute grasps till either all objects are cleared or two consecutive failures occur.

Table III reports the real-world evaluations. GIGA achieves higher success rates and clears more objects. We show some grasp examples in Figure 6. Compared with VGN, GIGA is better at detecting grasps for partially occluded objects and finding graspable object parts such as edges and handles. We also show a failure case. In Figure 6(d), the grasp fails due to the insufficient contact surface. We believe this is attributed to the unrealistic contact and friction models in the simulated environments that we used for generating training data. Videos of clutter removal can be found in the supplementary video and on our website.

VII. CONCLUSION

We introduced GIGA, a method for 6-DoF grasp detection in a clutter removal task. Our model learns grasp detection and 3D reconstruction simultaneously using implicit neural representations, exploiting the synergies between affordance and geometry. Concretely, we represent the grasp affordance and occupancy field of an input scene with continuous implicit functions, parametrized with neural networks. We learn shared structured feature grids on multi-task grasp and 3D supervision. In experiments, we study the influence of geometry supervision on grasp affordance learning and vice versa. The results demonstrate that utilizing the synergies between affordance and geometry can improve 6-DoF grasp detection, especially in the case of large occlusion, and grasp supervision produces better reconstruction in graspable regions.

Our method can be extended and improved in several fronts. First, in the training process, we assume only a single ground-truth grasp pose is provided for each grasp center. We hope to extend the current work and learn to predict the full distribution of viable grasp parameters with generative modeling. Second, though our model implicitly learns to avoid collisions from the training data where collided grasps are marked as negative, we do not explicitly reason about collision-free paths to grasps. We plan to utilize the reconstructed 3D scene to constrain the grasp prediction to be collision-free. We also plan to adapt this learning paradigm to close-loop grasp planning by integrating real-time feedback.

ACKNOWLEDGMENTS

We would like to thank Zhiyao Bao for efforts on affordance visualization. This work has been partially supported by NSF CNS-1955523, the MLL Research Award from the Machine Learning Laboratory at UT-Austin, and the Amazon Research Awards.

REFERENCES

- [1] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377, 2010.
- [2] Jeannette Bohg, Matthew Johnson-Roberson, Beatriz León, Javier Felip, Xavi Gratal, Niklas Bergström, Danica Kragic, and Antonio Morales. Mind the gap-robotic grasping under incomplete observation. In *IEEE International Conference on Robotics and Automation*, pages 686–693, 2011.
- [3] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2013.
- [4] Michel Breyer, Jen Jen Chung, Lionel Ott, Siegwart Roland, and Nieto Juan. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on Robot Learning*, 2020.
- [5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics*, pages 510–517. IEEE, 2015.
- [6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.
- [7] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020.
- [8] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>, 2016–2019.
- [9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996.
- [10] Dan Ding, Yun-Hui Liu, and Shuguo Wang. Computing 3-d optimal form-closure grasps. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3573–3578. IEEE, 2000.
- [11] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, 39(2-3):202–216, 2020.
- [12] Carlo Ferrari and John F Canny. Planning optimal grasps. In *ICRA*, volume 3, pages 2290–2295, 1992.

- [13] David Fischinger, Markus Vincze, and Yun Jiang. Learning grasps for unknown objects in cluttered scenes. In *2013 IEEE international conference on robotics and automation*, pages 609–616. IEEE, 2013.
- [14] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020.
- [15] Marcus Gualtieri, Andreas Ten Pas, Kate Saenko, and Robert Platt. High precision grasp pose detection in dense clutter. In *International Conference on Intelligent Robots and Systems*, pages 598–605, 2016.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [17] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019.
- [18] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgb-d images: Learning using a new rectangle representation. In *IEEE International Conference on Robotics and Automation*, pages 3304–3311, 2011.
- [19] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.
- [20] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In *IEEE International Conference on Robotics and Automation*, pages 4304–4311, 2015.
- [21] 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.
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [23] Ulrich Klank, Dejan Pangercic, Radu Bogdan Rusu, and Michael Beetz. Real-time cad model matching for mobile manipulation and grasping. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 290–296. IEEE, 2009.
- [24] Mia Kokic, Johannes A Stork, Joshua A Haustein, and Danica Kragic. Affordance detection for task-specific grasping using deep learning. In *International Conference on Humanoid Robotics*, pages 91–98. IEEE, 2017.
- [25] Danica Kragic, Andrew T Miller, and Peter K Allen. Real-time tracking meets online grasp planning. In *International Conference on Robotics and Automation*, volume 3, pages 2460–2465, 2001.
- [26] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *International Conference on Robotics and Automation*, pages 3629–3635, 2019.
- [27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020.
- [28] Yun-Hui Liu. Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15(1):163–173, 1999.
- [29] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987.
- [30] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.
- [31] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.
- [32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
- [33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.
- [34] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-d and 3-d. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 739–747. IEEE, 1994.
- [35] Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*, 2018.
- [36] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *IEEE International Conference on Computer Vision*, pages 2901–2910, 2019.
- [37] Chavdar Papazov and Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Asian Conference on Computer Vision*, pages 135–148, 2010.
- [38] Jeong Joon Park, Peter Florence, Julian Straub, Richard

- Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.
- [39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.
- [40] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2020.
- [41] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.
- [42] Christoph Pohl, Kevin Hitzler, Raphael Grimm, Antonio Zea, Uwe D Hanebeck, and Tamim Asfour. Affordance-based grasping and manipulation in real world applications. In *International Conference on Intelligent Robots and Systems*, pages 9569–9576, 2020.
- [43] Jean Ponce, Steve Sullivan, J-D Boissonnat, and J-P Merlet. On characterizing and computing three-and four-finger force-closure grasps of polyhedral objects. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 821–827. IEEE, 1993.
- [44] Ana Huamán Quispe, Benoît Milville, Marco A Gutiérrez, Can Erdogan, Mike Stilman, Henrik Christensen, and Heni Ben Amor. Exploiting symmetries and extrusions for grasping household objects. In *International Conference on Robotics and Automation*, pages 3702–3708, 2015.
- [45] Alberto Rodriguez, Matthew T Mason, and Steve Ferry. From caging to grasping. *The International Journal of Robotics Research*, 31(7), 2012.
- [46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and Computer-Assisted Intervention*, pages 234–241, 2015.
- [47] Anis Sahbani, Sahar El-Khoury, and Philippe Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.
- [48] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *IEEE International Conference on Robotics and Automation*, pages 509–516, 2014.
- [49] Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020.
- [50] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14): 1455–1473, 2017.
- [51] Mark Van der Merwe, Qingkai Lu, Balakumar Sundaralingam, Martin Matak, and Tucker Hermans. Learning continuous 3d reconstructions for geometrically aware grasping. In *International Conference on Robotics and Automation*, 2020.
- [52] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019.
- [53] Xinchun Yan, Jasmined Hsu, Mohammad Khansari, Yunfei Bai, Arkanath Pathak, Abhinav Gupta, James Davidson, and Honglak Lee. Learning 6-dof grasping interaction via deep geometry-aware 3d representations. In *IEEE International Conference on Robotics and Automation*, pages 3766–3773, 2018.
- [54] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *IEEE International Conference on Robotics and Automation*, pages 7286–7293. IEEE, 2020.
- [55] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
- [56] Xiangyang Zhu and Han Ding. Planning force-closure grasps on 3-d objects. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1258–1263, 2004.