Graspness Discovery in Clutters for Fast and Accurate Grasp Detection

Chenxi Wang*, Hao-Shu Fang*, Minghao Gou, Hongjie Fang, Jin Gao, Cewu Lu†
Shanghai Jiao Tong University
wcx1997@sjtu.edu.cn, fhaoshu@gmail.com, {gmh2015, galaxies}@sjtu.edu.cn,
gao3944677@126.com, lucewu@sjtu.edu.cn

Abstract

*Efficient and robust grasp pose detection is vital for robotic manipulation. For general 6 DoF grasping, conventional methods treat all points in a scene equally and usually adopt uniform sampling to select grasp candidates. However, we discover that ignoring **where to grasp** greatly harms the speed and accuracy of current grasp pose detection methods. In this paper, we propose "**graspness**", a quality based on geometry cues that distinguishes graspable area in cluttered scenes. A look-ahead searching method is proposed for measuring the graspness and statistical results justify the rationality of our method. To quickly detect graspness in practice, we develop a neural network named graspness model to approximate the searching process. Extensive experiments verify the stability, generality and effectiveness of our graspness model, allowing it to be used as a plug-and-play module for different methods. A large improvement in accuracy is witnessed for various previous methods after equipping our graspness model. Moreover, we develop GSNet, an end-to-end network that incorporate our graspness model for early filtering of low quality predictions. Experiments on a large scale benchmark, GraspNet-1Billion, show that our method outperforms previous arts by a large margin (**30+ AP**) and achieves a high inference speed.*

1. Introduction

As a fundamental problem in robotics, robust grasp pose detection for unstructured environment has been fascinating our community for decades. It has a broad spectrum of applications in picking [10], assembling [40], home serving [11], etc. Advancing the generality, accuracy and efficiency is a long pursuit of researchers in this field.

For grasp pose detection in the wild, it can be regarded as a two-stage problem: given a single-view point cloud, we first find locations with high graspability (*where* stage) and then decide grasp parameters like in-plane rotation, approaching depth, grasp score and gripper width (*how* stage) for a local region.

* denotes equal contribution.
Cewu Lu is the corresponding author, member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China and Shanghai Qi Zhi institute.

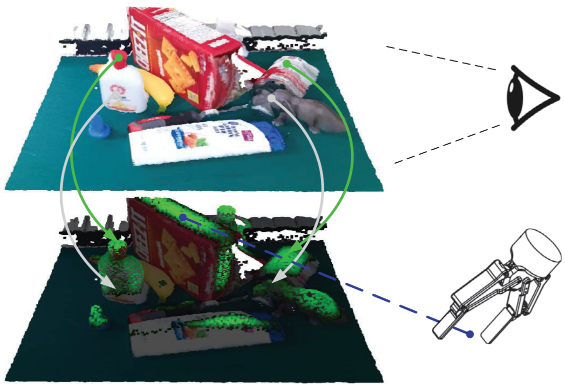


Figure 1. Graspness illustration for a cluttered scene. Brighter color denotes higher graspness. We prefer the points with high graspness for grasping.

Previous methods for 6-DoF grasp pose detection in cluttered scenes mainly focused on improving the quality of grasp parameter prediction, *i.e.*, the *how* stage, and two lines of research are explored. The first line [41, 27, 31] adopts a sampling-evaluation method, where grasp candidates are uniformly randomly sampled from the scene and evaluated by their model. The second line [36, 13, 32] proposes end-to-end networks to calculate grasp parameters for the whole scene, where point clouds are sampled before [32] or during [36, 13] the forward propagation. For all these methods, the *where* stage is not explicitly modeled (*i.e.*, they do not perform a filtering procedure in a first stage) and candidate grasp points distribute uniformly in the scene.

However, we find that such uniform sampling strategy greatly hinders the performance of the whole pipeline. There are tremendous points in 3D contiguous space, while positive samples are concentrated in small local regions. Take GraspNet-1Billion [13], the current largest dataset in grasp pose detection as an example. We statistically find that, *even with* object masks, the graspable points are less than 10% among all the samples, not to mention the candidate points in the whole scene. Such an imbalance causes a large waste of computing resources and degrades the efficiency.

To tackle the above bottleneck in grasp pose detection, we propose a novel geometrically based quality, ***graspness***, for distinguishing graspable area in cluttered scenes. One might think that we need complex geometric reasoning to obtain such graspness. However, we discover that a simple look-ahead search by exhaustively evaluating possible future grasp poses from a point can well represent its graspness. Statistical results demonstrate the justifiability of our proposed graspness measure, where the local geometry around points with high graspness are distinguished from those with low scores. Fig. 1 gives an illustration of our graspness for a cluttered scene.

Furthermore, we develop a graspness model that approximates the above process in practice. Given a point cloud input, it predicts point-wise graspness score, which is referred to as *graspable landscape*. Benefiting from the stability of the local geometry structures, our graspness model is object agnostic and robust to variation of viewpoint, scene, sensor, etc., making it a general and transferable module for grasp point sampling. We qualitatively evaluate its robustness and transferability in our analysis. Tremendous improvements in both speed and accuracy for previous sampling-evaluation based methods are witnessed after equipping them with our graspness model.

Based on our graspness model, we also propose Graspness-based Sampling Network (GSNet), an end-to-end two-stage network with a graspness-based sampling strategy. Our network takes a dense scene point cloud as input, which preserves the local geometry cues. The sampling layer firstly selects the points with high graspness. Remaining points are discarded from the forward propagation to improve the computation efficiency. Such two-stage design is beneficial to network convergence and also the final accuracy by providing more positive samples during training.

We conduct extensive experiments to evaluate the effectiveness of our proposed graspness measure, model and the end-to-end network. Several baseline methods equipped with our graspness model outperform their vanilla counterparts by a large margin in both speed and accuracy. Moreover, our GSNet outperforms previous methods to a large extent. Our code and models will be made publicly available to facilitate researches in related area.

2. Related Work

In this section, we first briefly review previous methods on grasping in cluttered scenes, followed by concluding the common strategies they have used to sample grasp candidates. Finally we surveyed some literature in cognitive science area where graspness recognition is witnessed in human perception.

Grasping in Cluttered Scenes For cluttered scene grasp pose detection, previous research can be mainly divided into two categories: plannar based grasp detection and 6-DoF based grasp detection. The research in the first category [1, 22, 25, 29, 30, 37, 24, 8] mainly took RGB images or depth images as inputs and output a set of rotated bounding boxes to represent the grasp poses. Due to the limitation of low DoF, their applications were usually restricted. Another line of research aimed to predict full DoF grasp poses. Among them, two different directions were explored. The first direction [42, 41, 27, 31] adopted the sampling-evaluation based two-step policy, where grasp candidates were densely uniformly sampled in the scene and evaluated using a deep quality model. The second direction [13, 36, 32, 4] adopted the end-to-end strategy, where point clouds of the scene were directly processed by end-to-end networks. For each input point, the network attempted to predict the most feasible grasp pose. All the mentioned methods focused on improving the quality of grasp parameters, and the problem of where to grasp was not investigated.

Grasp Sampling Strategies Several kinds of sampling strategies can be concluded from the above methods. The most common used strategy is the uniform sampling, which is adopted by [41, 27, 32, 36]. Specifically, GPD [41] and PointNetGPD [27] uniformly sampled grasp points in the scene point cloud and estimated the rotation by darboux frame. Some end-to-end models [36, 32] down-sampled the input point cloud by voxel grid to avoid memory explosion. A similar strategy, farthest point sampling, is adopted by other end-to-end model [13]. Some optimization based methods are also explored. Ciocarlie *et al*. [9] and Hang *et al*. [19] adopted the simulated annealing method, while Mahler *et al*. [29] proposed cross-entropy methods. In [31], a grasp sampler network first sampled possible grasp poses on partial object point cloud and conducted iterative refinement by a grasp evaluator based on its gradient. In a recent paper by Clemens *et al*. [12], several sampling methods for grasp dataset generation are reviewed. However, all the previous methods ignore the geometric cues for graspable point sampling. In this paper, we propose a novel graspness measure based on local geometry for graspable point sampling, which is much more efficient than previous uniform sampling and optimization based methods.

Graspness in Cognitive Science In cognitive area, researchers have studied the visual attention during grasping for a long period. Many literature [2, 3, 15, 20, 38] demonstrated that human bias the allocation of available perceptual resources, named as affordance attention, towards the region with the highest graspability. And such attention usually precedes the action preparation stage [2]. Such discovery corresponds to our graspness concept and motivates us to apply it in the grasp sampling strategy.

3. Graspness Discovery

3.1. Preliminary

As mentioned above, we decouple the grasp pose detection problem into two stages. Before the common practice in previous research that directly calculates the grasp parameters, we first sample points and views with high graspness. Computational resources will be allocated to these areas thereafter to improve computational efficiency.

To determine the suitable grasp locations and the feasible approach directions with high graspability, we define two kinds of graspness in a high dimensional space to represent parallel attention in point locations and approach directions. Before detailing our graspness measure, we first introduce some basic notations.

For a point sets $\mathcal{P} = \{p_i | i = 1, ..., N\}$, we assume V approach directions uniformly distributed in a sphere space $\mathcal{V} = \{v_j | j = 1, ..., V\}$.

Two kinds of graspness scores are discussed in this paper. The first is the point-wise graspness scores denoted as

$$\mathcal{S}^p = \{s_i^p | s_i^p \subset [0, 1], i = 1, ..., N\},$$

where $[0, 1]$ denotes that our graspness for each point ranges from 0 to 1. The second is the view-wise graspness scores denoted as

$$\mathcal{S}^v = \{s_i^v | s_i^v \subset [0, 1]^V, i = 1, ..., N\},$$

where $[0, 1]^V$ denotes V-dim graspness ranging in [0,1].

In the following section, we illustrate how we measure graspness for both single object and the cluttered scene.

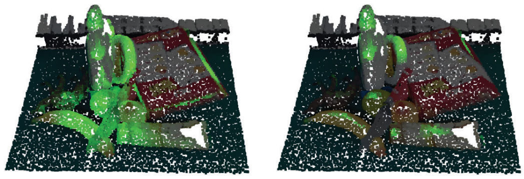
3.2. Graspness Measure

Single Object Graspness Given an object point cloud, we aim to generate graspness for each point where higher activation denotes larger possibility for successful grasping. Assuming there is an oracle $\mathbf{1}(\cdot)$ that tells whether an arbitrary grasp is successful, and $\mathbf{G}_{i,j}$ denotes the set of all feasible grasp poses for view v_j centered at point p_i, then the graspness score $\tilde{s}_i^{\mathrm{p}}$ and $\tilde{s}_i^{\mathrm{v}}$ can be obtained by an exhaustive look-ahead search:

$$\begin{aligned} \tilde{s}_i^{\mathrm{p}} &= \frac{\sum_{j=1}^{V} \sum_{g \in \mathbf{G}_{i,j}} \mathbf{1}(g)}{\sum_{j=1}^{V} |\mathbf{G}_{i,j}|}, i = 1, ..., N, \\ \tilde{s}_i^{\mathrm{v}} &= \Big\{ \frac{\sum_{g \in \mathbf{G}_{i,j}} \mathbf{1}(g)}{|\mathbf{G}_{i,j}|} \Big| 1 \le j \le V \Big\}, i = 1, ..., N. \end{aligned} \tag{1}$$

By doing so, we guarantee that higher graspness value always denote higher possibility of successful grasping.

In practice, such an oracle $\mathbf{1}(\cdot)$ does not exist, and $\mathbf{G}_{i,j}$ can contain infinite grasp poses in a continuous space.



(a) without collision. (b) with collision.

Figure 2. Graspness scores. The left image shows the graspness without collision detection while the right image shows the graspness with collision detection

Thus, we make an approximation to the above process. For view v_j of point p_i, we generate L grasp candidates $\mathcal{G}_{i,j} = \{g_k^{i,j} | k = 1, ..., L\}$ by grid sampling along gripper depths and in-plane rotation angles. For each grasp $g_k^{i,j}$, we calculate a grasp quality score $q_k^{i,j}$ using a force analytic model [29]. A threshold c is manually set to filter out unsuccessful grasps. Then, the relaxation form of Eqn. 1 is:

$$\begin{aligned} \tilde{s}_i^{\mathrm{p}} &= \frac{\sum_{j=1}^{V} \sum_{k=1}^{L} \mathbf{1}(q_k^{i,j} > c)}{\sum_{j=1}^{V} |\mathcal{G}_{i,j}|}, i = 1, ..., N, \\ \tilde{s}_i^{\mathrm{v}} &= \Big\{ \frac{\sum_{k=1}^{L} \mathbf{1}(q_k^{i,j} > c)}{|\mathcal{G}_{i,j}|} \Big| 1 \le j \le V \Big\}, i = 1, ..., N. \end{aligned} \tag{2}$$

Scene-Level Graspness After defining the object-level graspness, we extend it to cluttered scenes by first discussing the gap between them and then redefining the graspness in cluttered scenes.

A cluttered scene contains multiple objects and the irrelevant background. As shown in Fig. 2(a), the simplest way to compute scene-level graspness is directly projecting the object-level graspness score to the scene by object 6D poses. However, this solution ignores the differences between an object model and a scene cloud captured from RGB-D camera. Firstly, a valid grasp of a single object may collide with background or other objects when placing in cluttered manner and becomes a negative grasp. Secondly, as the depth camera provides single-view partial point clouds, we need to associate the scene point cloud with the projected object point.

To deal with the collision problem, we follow [13] to reconstruct the scene using object 3D models and corresponding 6D poses. Each grasp $g_k^{i,j}$ is evaluated by a collision checking process and assigned a collision label $c_k^{i,j}$. Our graspness scores are then updated as:

$$\begin{aligned} \tilde{s}_i^{\mathrm{p}} &= \frac{\sum_{j=1}^{V} \sum_{k=1}^{L} \mathbf{1}(q_k^{i,j} > c) \cdot \mathbf{1}(c_k^{i,j})}{\sum_{j=1}^{V} |\mathcal{G}_{i,j}|}, i = 1, ..., N, \\ \tilde{s}_i^{\mathrm{v}} &= \Big\{ \frac{\sum_{k=1}^{L} \mathbf{1}(q_k^{i,j} > c) \cdot \mathbf{1}(c_k^{i,j})}{|\mathcal{G}_{i,j}|} \Big| 1 \le j \le V \Big\}, i = 1, ..., N. \end{aligned} \tag{3}$$

After that, we project the object points to the scene by object 6D poses. For each point in the scene, we obtain its graspness scores by nearest neighbor search and associate it with the nearest projected object point.

Finally, to obtain a coherent representation for the scene-level graspness scores, we perform a normalization for each scene:

$$
\begin{aligned}
\mathcal{S}^p &= \left\{ \frac{\tilde{s}^p_i - \min(\tilde{\mathcal{S}}^p)}{\max(\tilde{\mathcal{S}}^p) - \min(\tilde{\mathcal{S}}^p)} \middle| i = 1, ..., N \right\}, \\
\mathcal{S}^v &= \left\{ \frac{\tilde{s}^v_i - \mathbf{min}(\tilde{\mathcal{S}}^v)}{\mathbf{max}(\tilde{\mathcal{S}}^v) - \mathbf{min}(\tilde{\mathcal{S}}^v)} \middle| i = 1, ..., N \right\},
\end{aligned} \tag{4}
$$

where $\mathbf{min}(\cdot)$ denotes column wise minimum:

$$
\mathbf{min}(\tilde{\mathcal{S}}^v) = \left\{ \min_{i=1}^{N} \tilde{s}^v_{i(j)} \middle| j = 1, ..., V \right\},
$$

and so does $\mathbf{max}(\cdot)$. Fig. 2(b) shows an example of scene-level graspness scores.

3.3. Justification

In order to justify our graspness measure, we analyze the local geometry for regions with different graspness to find out whether they are really distinguishable geometrically. For a single-view point cloud, the cascaded graspness model detailed in Sec. 4.1 is used to extract the local feature vector of each point. The points with graspness more than 0.3 are treated as positive samples, and negative ones of the same size are sampled with graspness less than 0.1. Fig. 3 shows the t-SNE [28] visualization of the encoded local geometry (feature vectors of each point produced by backbone network) for all the scenes in GraspNet-1Billion [13] training/testing set respectively. We can observe that regions with different graspness are quite distinguishable. It demonstrates that our graspness measure is rational and reveals the potential of learning graspness from point cloud.

4. GSNet Architecture

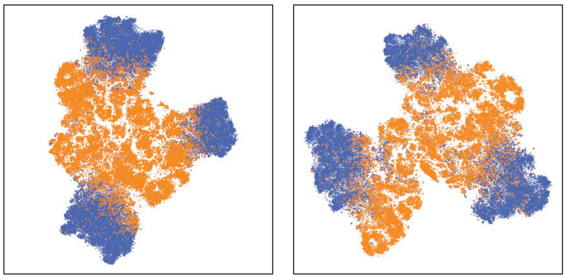
After defining the graspness measure, we introduce the end-to-end grasp pose detection network, GSNet, where our graspness is learned by an independent module and can be applied to other methods.

4.1. Cascaded Graspness Model

Given a dense single view point cloud $\mathcal{P}$, graspness model needs to learn two approximations: $f^p : \mathcal{P} \longrightarrow \mathcal{S}^p$ and $f^v : \mathcal{P} \longrightarrow \mathcal{S}^v$.

It is challenging to find a direct mapping from point coordinates to graspness scores due to the large domain gap between these two spaces. Instead, we decompose the whole process into two sub-functions. Consider a high dimensional feature set $\mathcal{F}$:

$$
\mathcal{F} = \{f_i | f_i \subset \mathbb{R}^C, i = 1, ..., N\},
$$



(a) Training set. (b) Testing set.

Figure 3. t-SNE visualization of encoded local geometry. Orange points denote the samples with high graspness, and blue points denote the samples with low graspness.

where $\mathbb{R}^C$ denotes C-dim feature space. The point set is firstly transformed to the feature set by $h^t : \mathcal{P} \longrightarrow \mathcal{F}$. Graspable landscapes are then generated by $h^p : \mathcal{F} \longrightarrow \mathcal{S}^p$ and $h^v : \mathcal{F} \longrightarrow \mathcal{S}^v$. Hence, we model the graspness scores by

$$
f^p = h^p \circ h^t, \quad f^v = h^v \circ h^t,
$$

where $\circ$ denotes function composition, and the feature set $\mathcal{F}$ is shared by both h^p and h^v.

Although h^p and h^v can be learned simultaneously, the computation overhead is quite expensive since $\mathcal{S}^v$ is in high dimensional space. Meanwhile, it is not necessary to compute the view-wise graspable landscapes for all points since most of the points are not graspable at point level. Hence, we propose cascaded graspness model to learn h^t, h^p and h^v step by step, where points are sampled by the output of h^p before learning h^v to reduce computation cost.

Backbone Network Approximation of h^t requires a strong backbone network for extraction of both global and local point features. We adopt ResUNet14 built upon MinkowskiEngine [6] because it can flexibly process point sets of any size with sparse convolution and has shown excellent performance in multiple tasks of 3D deep learning [7, 14, 18, 5]. The network can also be replaced by other point-wise networks, such as PointNet [34, 35], PointCNN [26] and SSCNs [17].

The network adopts a U-shape architecture with residual blocks, which obtains point features using 3D sparse (transposed) convolutions and skip-connections. For a point cloud of size $N \times 3$, it extracts a C-channel feature vector set, and outputs a point set of size $N \times (3+C)$ for graspable sampling and grasp generation.

Graspable Farthest Point Sampling The modeling for h^p is implemented with a multi-layer perceptron (MLP) network to generate point-wise graspable landscape. Specifically, the output contains a prediction for the graspable landscape of size $N \times 1$ and a binary objectness classification scores of size $N \times 2$, resulting a total output of size $N \times 3$. Graspness scores of non-object points are set to 0.

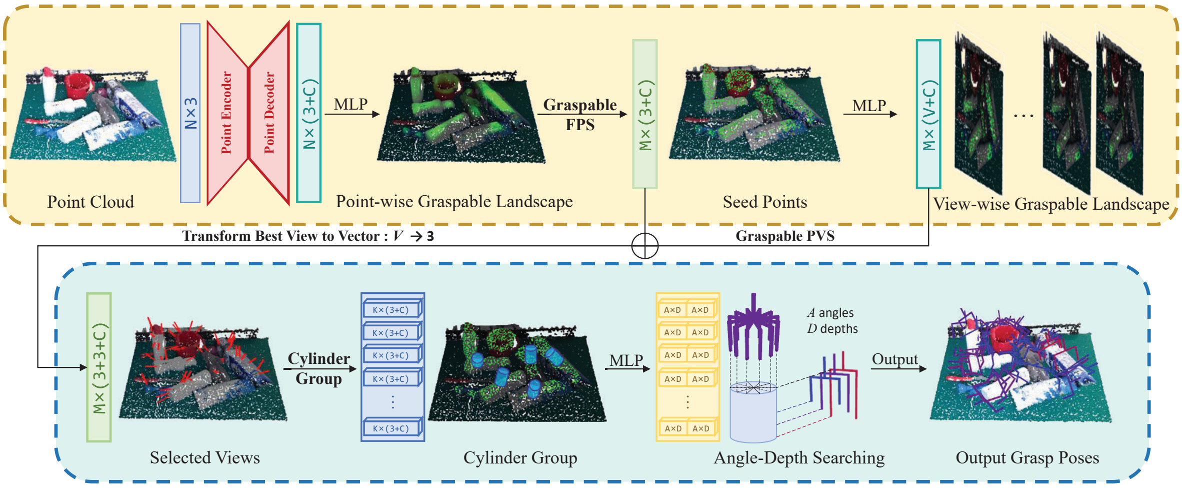


Figure 4. GSNet architecture. The two rows show the process of cascaded graspness model and grasp operation model respectively. In cascaded graspness model, point encoder-decoder outputs C-dim feature vectors for the input N points. A point-wise graspable landscape is generated and M seed points are sampled from it. The seeds are then used to generate view-wise graspable landscapes, and select the grasp view. In grasp operation model, the seeds are grouped in cylinder regions. The grasp scores and gripper widths are predicted for each group and used to output M grasp poses.

After obtaining the point-wise graspable landscape, we select points with graspness score larger than δ^{p} and adopt farthest point sampling (FPS) to maximize distances among sampled points. M seed points are sampled with $(3+C)$-dim features, where 3 denotes the point coordinates and C denotes the features output by the backbone network.

Graspable Probabilistic View Selection h^v is also modeled by an MLP. We apply it to the sampled seed points and output $M \times V$ vectors for view-wise graspable landscapes and $M \times C$ residual features for grasp generation. V views are sampled from a unit sphere using Fibonacci lattices [16].

After obtaining the view-wise graspness scores, we select the best view for afterward predictions during inference. For training, we adopt probabilistic view selection (PVS) that normalizes the graspness scores of all views on a seed point to (0,1) and regard them as probability scores, according to which the view is sampled. The M seed point-view pairs are then used to estimate grasp scores, gripper widths, approach distances and in-plane rotation angles.

4.2. Grasp Operation Model

Crop-and-refine has been proven effective in estimating candidate configuration in both 2D and 3D tasks [39, 21, 33]. We crop points in directional cylinder spaces which are generated by seed point-view pairs, transform them to gripper frames and estimate their grasp parameters.

Cylinder-Grouping from Seed Points The locations and directions of cylinder spaces are determined by seed point coordinates and view vectors respectively. For each of the M point-view pairs, we group and sample K points from M seed points using the cylinder with fixed height d and radius r. After aligning the cylinder with gripper frame as [13], the point coordinates are normalized by cylinder radius and concatenated with feature vectors which are the sum of features output by graspable FPS and graspable PVS. The grouped point sets of size $M \times K \times (3+C)$ are called grasp candidates, where K stands for the number of sampled points in each group.

Grasp Generation from Candidates We use a shared PointNet [34] for grasp generation. Grasp candidates are processed by an MLP network and a max-pooling layer, and be output as feature vectors of size $M \times C'$. Finally we get grasp configurations by a new MLP network.

The output of GSNet contains scores and widths for different (in-plane rotation)-(approach depth) combinations. We pick the combination with the highest score as the grasp prediction. The output size is $M \times (A \times D \times 2)$, where A denotes the number of in-plane rotation angles, D denotes the number of gripper depths and 2 denotes the score and the width.

Grasp Score Representation We use the minimum friction coefficient μ under which a grasp is antipodal to evaluate the quality of the grasp. Based on this, we define the grasp score as

$$q_i = \begin{cases} \frac{\ln(\mu_{\max}/\mu_i)}{\ln(\mu_{\max}/\mu_{\min})} & g_i \text{ is positive,} \\ 0 & g_i \text{ is negative.} \end{cases} \tag{5}$$

All scores are normalized to $[0,1]$. Smaller μ_i indicates higher score q_i and more probability to succeed.

4.2.1 Loss Function

Cascaded graspness model and grasp operation model are trained simultaneously with multi-task losses:

$$L = L_o + \alpha(L_p + \lambda L_v) + \beta(L_s + L_w), \tag{6}$$

where L_o is for objectness classification, L_p, L_v, L_s and L_w are for regressions of point-wise graspable landscape, view-wise graspable landscape, grasp scores and gripper widths respectively. L_p and L_s are calculated only if the related points are on objects, L_v is calculated for views on seed points and L_w is calculated for grasp poses with ground truth scores > 0. We use softmax for classification tasks and smooth-l_1 loss for regression tasks.

5. Experiments

5.1. Implementation Details

Benckmark Dataset GraspNet-1Billion [13] is a large-scale dataset for grasp pose detection, which contains 190 scenes with 256 different views captured by two cameras (RealSense/Kinect). The testing scenes are divided into three splits according to the object categories (seen/similar/novel). A unified evaluation metric is proposed to benchmark both image based methods and point cloud based methods. We adopt this benchmark as it aligns well with real-world grasping.

Data Processing and Augmentation The point cloud is downsampled with voxel size 0.005m before being fed into the network, and contains only XYZ in camera coordinates. Input clouds are augmented on the fly by random flipping along YZ plane and random rotation around Z axis in $\pm 30°$.

Implementations To obtain graspness for scenes in GraspNet-1Billion, we follow the process illustrated in Sec. 3.2 since it contains abundant grasp pose annotations. For each point, it densely labels grasp quality score for 300 different views and 48 grasps for each view. Thus, our approach directions V and grasp candidates per view L are set as 300 and 48.

For our network, the backbone network adopts an encoder-decoder architecture and outputs feature vectors of channel $C = 512$. In visual selection module, $M = 1024$ seed points and $V = 300$ views are sampled, and the threshold δ^{P} is set to 0.1. The size of MLP used for h^p is $(512, 3)$ and h^v is $(512, 512, 300)$. In cylinder-grouping, we sample $K = 16$ seed points in the cylinder space with radius $r = 0.05\mathrm{m}$ and height range of $[-0.02\mathrm{m}, 0.04\mathrm{m}]$. We divide in-plane rotation angles into $A = 12$ classes ($15°$ per class) and use $D = 4$ classes for approaching distances (0.01m, 0.02m, 0.03m, 0.04m). The two MLPs used to process attentional proposals and output grasp scores and gripper widths have the size of $(512, 256, 256)$ and $(256, 256, 96)$ respectively. Finally the network outputs grasp scores and gripper widths for $A \times D = 48$ classes. In loss functions, we set $\alpha, \beta, \lambda = 10, 10, 10$.

Training and Inference Our model is implemented with PyTorch and trained on Nvidia GTX 1080Ti GPUs for 10 epochs with Adam optimizer [23] and the batch size of 4. The learning rate is 0.001 at the first epoch, and multiplied by 0.95 every one epoch. The network takes about 1 day to converge. During training, we use one GPU for model updating and one GPU for label generation. In inference, we only use one GPU for fast prediction.

5.2. Performance of Cascaded Graspness Model

Cascaded graspness model is proposed to distinguish graspable areas in various scenes, thus the generality and stability across different domains are important for the model. Here we design an experiment to illustrate its generality and stability.

Evaluation Metric The *ranking error* is used to quantitatively evaluate the function approximation ability of the model. We divide the range of graspness score into K bins uniformly and convert the contiguous scores to discrete ranks. The ranking error is defined as the mean rank distances between predictions and labels:

$$e_{\mathrm{rank}} = \frac{1}{N_r}\sum_{i=1}^{N_r}\frac{|\hat{r}_i - r_i|}{K}, \tag{7}$$

where $r_i, \hat{r}_i \in \{0, 1, ..., K-1\}$ stand for the ranks for predictions and labels respectively, and N_r is the number of predictions. We set $K = 20$ in experiments. e^p_{rank} and e^v_{rank} are used to denote the ranking error of point-wise graspness score and view-wise graspness score respectively.

Inference in Different Domains We conduct three groups of experiments where the dataset is split by object categories, viewpoints and cameras respectively (detailed in Tab. 1). In the first group, we train the model on scene 0-99, and test it on scenes with three object categories (seen, similar and novel). The second group divides the 256 viewpoints into 3 sets, trains the model on viewpoint 0-127, and tests on three viewpoint sets respectively. The third group trains the model on Kinect captured data, and tests the performance on data captured by RealSense.

The results are summarized in Tab. 1. For point-wise graspness prediction, we can see that the difference between e^p_{rank} of seen and novel categories is not obvious. View variation also has a low impact on point-wise graspness prediction. The e^p_{rank} of RealSense is higher than Kinect, but the distance is still in an acceptable range. For view-wise graspness prediction, e^v_{rank} in all groups are nearly un-

	Object Variation				Viewpoint Variation				Camera Variation		
	Train	Test			Train	Test			Train	Test	
Scene	0-99	100-129	130-159	160-189	0-99	100-129	100-129	100-129	0-99	100-129	100-129
View	0-255	0-255	0-255	0-255	0-127	0-127	128-191	192-255	0-255	0-255	0-255
Camera	Kinect	Kinect	Kinect	Kinect	Kinect	Kinect	Kinect	Kinect	Kinect	Kinect	Realsense
e^p_{rank}	0.0485	0.0677	0.0856	0.0802	0.0484	0.0697	0.0725	0.0763	0.0485	0.0677	0.0984
e^v_{rank}	0.0451	0.0457	0.0459	0.0413	0.0458	0.0468	0.0473	0.0476	0.0451	0.0457	0.0461

Table 1. Ranking error of cascaded graspness model on different test setting. We can see that the graspness model is not sensitive to object/viewpoint/camera variations.

Methods	Seen			Similar			Novel		
	AP	$\mathrm{AP}_{0.8}$	$\mathrm{AP}_{0.4}$	**AP**	$\mathrm{AP}_{0.8}$	$\mathrm{AP}_{0.4}$	**AP**	$\mathrm{AP}_{0.8}$	$\mathrm{AP}_{0.4}$
GG-CNN [30]	15.48/16.89	21.84/22.47	10.25/11.23	13.26/15.05	18.37/19.76	4.62/6.19	5.52/7.38	5.93/8.78	1.86/1.32
Chu *et al.* [8]	15.97/17.59	23.66/24.67	10.80/12.74	15.41/17.36	20.21/21.64	7.06/8.86	7.64/8.04	8.69/9.34	2.52/1.76
GPD [41]	22.87/24.38	28.53/30.16	12.84/13.46	21.33/23.18	27.83/28.64	9.64/11.32	8.24/9.58	8.89/10.14	2.67/3.16
Liang *et al.* [27]	25.96/27.59	33.01/34.21	15.37/17.83	22.68/24.38	29.15/30.84	10.76/12.83	9.23/10.66	9.89/11.24	2.74/3.21
Fang *et al.* [13]	27.56/29.88	33.43/36.19	16.95/19.31	26.11/27.84	34.18/33.19	14.23/16.62	10.55/11.51	11.25/12.92	3.98/3.56
GPD + *CGM*	28.16/29.65	34.07/35.59	17.21/18.94	26.47/28.19	33.14/33.74	14.27/16.20	9.73/10.89	10.55/11.37	3.35/4.12
Liang *et al.* + *CGM*	33.86/33.17	41.50/40.85	22.93/23.18	28.91/29.06	34.70/35.96	16.95/17.33	11.97/12.47	13.52/13.31	4.01/4.64
Fang *et al.* + *CGM*	41.46/39.51	49.32/48.75	29.64/26.19	36.87/35.28	45.69/44.93	25.29/23.84	15.11/13.26	17.49/15.03	6.74/5.28
Ours	65.70/61.19	76.25/71.46	**61.08**/56.04	53.75/47.39	65.04/56.78	45.97/40.43	23.98/19.01	29.93/23.73	14.05/10.60
Ours + *CD*	**67.12/63.50**	**78.46/74.54**	60.90/**58.11**	**54.81/49.18**	**66.72/59.27**	**46.17/41.89**	**24.31/19.78**	**30.52/24.60**	**14.23/11.17**

Table 2. GraspNet-1Billion evaluation results on RealSense/Kinect. *CGM* is cascaded graspness model. *CD* is collision detection.

changed. These experiments prove the stability and generality of the cascaded graspness model when transferred to new domains.

5.3. Comparing with Representative Methods

We compare our method with previous representative methods. GG-CNN [30] and Chu *et al.* [8] are rectangle based methods which take images as input. GPD [41] and Liang *et al.* [27] classify grasp candidates generated by rule-based point cloud sampling. Fang *et al.* [13] propose an end-to-end network which predicts grasp poses directly from scene point clouds.

We test our method in three object categories respectively and report the results in Tab. 2. The models for RealSense and Kinect are trained separately. Our method outperforms previous methods by a large margin on both cameras without any post-processing. Compared with Fang *et al.*, the previous state-of-the-art method, GSNet improves the performance by $\sim$2x on **AP** metric [13]. Notably, on the most difficult metric $\mathrm{AP}_{0.4}$, GSNet still achieves a great relative improvement ($>$ 140%) on all categories. Fig. 5 presents the qualitative results of our network. The top-1 grasp accuracy on three categories are 78.22/76.49, 62.88/57.64 and 28.97/24.04 for Realsense/Kinect input.

We also report the results after simple collision detection using a parallel-jaw gripper model, where all grasps collided with scene points are removed. The results are improved by 1.42/2.31 **AP**, 1.06/1.79 **AP** and 0.33/0.77 **AP** on the three categories respectively.

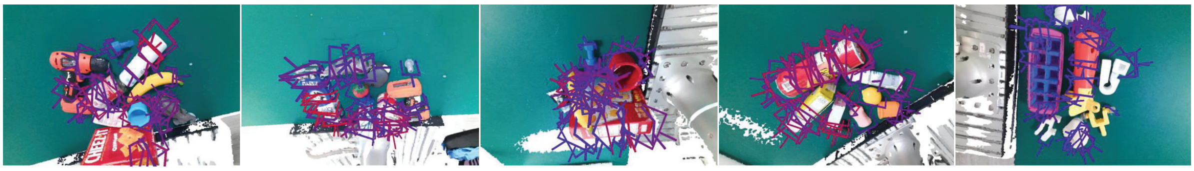
5.4. Boosting with Cascaded Graspness Model

We apply the cascaded graspness model(CGM) to GPD, Liang *et al.* and Fang *et al.* directly and compare the results with the original methods. For Fang *et al.*, we simply replace ApproachNet with our module. For GPD and Liang *et al.*, we first determine the grasp candidate points using our predicted point-wise graspable landscape, followed by their post processing of Darboux frame estimation and grasp images/clouds classification.

In the middle of Tab. 2, we show the results after adding the CGM. Both the two-step methods and the end-to-end method achieve significant performance gains, proving the effectiveness of cascaded graspness model. Graspable landscapes can not only improve candidate qualities, but also reduce the huge computation time caused by densely sampling.

5.5. Analysis

Effects of Graspable FPS/PVS In Sec. 4.1 we use graspable FPS to sample seed points from graspable landscapes, while other sampling methods can also be applied to the network. We compare our sampling method with three alternatives: a) random sampling from the whole point cloud; b) FPS from the whole point cloud; c) random sampling from graspable landscapes. Tab. 3 shows the results of the models trained using different sampling methods. FPS outperforms random sampling by at least 4.98 **AP** and sampling with graspable landscapes improves the results by over 7 **AP** for both FPS and random sampling, which proves the effectiveness of graspable FPS.



Scene 108 | Scene 121 | Scene 125 | Scene 148 | Scene 175

Figure 5. Qualitative results of GSNet. Top 50 grasps after grasp-NMS[13] are displayed.

Point Sampling	View Sampling	AP
random	graspable PVS	46.17
FPS	graspable PVS	51.15
graspable random	graspable PVS	53.32
graspable FPS	normal	55.63
graspable FPS	top-1 score	58.34
graspable FPS	graspable PVS	**59.70**

Table 3. Comparison of different sampling methods."top-1 score" stands for selecting the view with the highest graspness score.

Landscape	AP
object-level	55.33
scene-level	**59.70**

Table 4. Landscape types.

View Graspness	AP
mean score	50.62
max score	56.95
feasible ratio	**59.70**

Table 5. View graspness types.

Camera	CGM	GOM	Total
RealSense	0.08s	0.02s	0.10s
Kinect	0.10s	0.02s	0.12s

Table 6. Inference speed on GraspNet-1Billion. "CGM" is cascaded graspness model and "GOM" is grasp operation model.

For view selection, we compare graspable PVS with two methods: a) selecting views by surface normal; b) selecting the view with the highest graspness score during training. The results in Tab. 3 show that our method outperforms both alternative strategies. Graspable PVS dynamically selects approach vectors, which provides richer data for model training than other methods.

Selection of Landscape Representations In Sec. 3.2 we extend object-level graspness scores to cluttered scenes. Tab. 4 shows that sampling from scene-level graspness performs better than object-level counterpart. The representation for graspness score also has multiple choices. We replace the original definition *ratio of feasible grasps* with mean and maximum grasp quality scores respectively in the calculation of view-wise graspness scores, and the results in Tab. 5 shows that feasible grasp ratio performs the best.

Model Speed Tab. 6 shows the inference time of our method. Cascaded graspness model achieves a high speed on RealSense/Kinect data, which can also provide accurate sampling for various grasp detection methods. GPD and PointNetGPD take $>$1s while ours takes only $\sim$0.1s.

IDs	#Objects	#Attempts	Success Rate
4, 10, 22, 32, 36, 57	6	6	100%
2, 38, 58, 59, 61, 69	6	7	85.7%
34, 37, 64, 66, 68, 72, 77	7	9	77.8%
0, 2, 23, 29, 39, 56, 62	7	7	100%
1, 10, 40, 41, 44, 48, 65, 69	8	8	100%
3, 9, 10, 23, 33, 42, 63, 68	8	9	88.9%
Total	42	46	91.3%

Table 7. Results of cluttered scene grasping. #Objects denotes the number of objects, and so does #Attempts.

5.6. Real Grasping Experiments

We also conduct grasping experiments for cluttered scenes in the real-world setting. The configuration of our experimental setup is illustrated in supplementary materials. The experiments are conducted on a UR-5 robotic arm with an Intel RealSense D435 camera and a Robotiq two-finger gripper. During experiments, we only keep the points on table workspace for speed up.

We conduct grasping experiments in six cluttered scenes. Each scene contains 6-8 objects selected from GraspNet-1Billion. Objects are put together randomly and we repeat the grasping pipeline until the table are cleaned. The success rate is defined as the ratio of object number and attempt number. Tab. 7 reports the grasping performance, which proves the effectiveness of our method. A comparison with other baselines is detailed in supplementary materials.

6. Conclusion

In this paper, we propose a novel geometrically based quality named graspness. A look-ahead searching method is adopted as our graspness measure and we statistically demonstrate its effectiveness and rationality. An end-to-end network is developed to incorporate graspness into grasp pose detection problem, wherein an independent model learns the graspable landscapes. We conduct extensive experiments and demonstrate the stability, generality, effectiveness and robustness of our graspness model. Large margin of improvements are witnessed for previous methods after equipping with our graspness model, and our final network sets a high record for both accuracy and speed.

Acknowledgement This work is supported in part by the National Key R&D Program of China, No.2017YFA0700800, National Natural Science Foundation of China under Grants 61772332 and Shanghai Qi Zhi Institute, SHEITC(2018-RGZN-02046).

References

[1] Umar Asif, Jianbin Tang, and Stefan Harrer. Densely supervised grasp detector (dsgd). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8085–8093, 2019. 2

[2] Daniel Baldauf and Heiner Deubel. Attentional landscapes in reaching and grasping. *Vision research*, 50(11):999–1013, 2010. 2, 3

[3] Lindsay E Bamford, Nikola R Klassen, and Jenni M Karl. Faster recognition of graspable targets defined by orientation in a visual search task. *Experimental Brain Research*, pages 1–12, 2020. 2

[4] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. *arXiv preprint arXiv:2101.01132*, 2021. 2

[5] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523, 2020. 4

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 4

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8958–8966, 2019. 4

[8] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. Real-world multiobject, multigrasp detection. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3355–3362, 2018. 2, 7

[9] Matei Ciocarlie, Corey Goldfeder, and Peter Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275. IEEE, 2007. 2

[10] Nikolaus Correll, Kostas E Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M Romano, and Peter R Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 15(1):172–188, 2016. 1

[11] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3530–3537. IEEE, 2017. 1

[12] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. *arXiv preprint arXiv:1912.05604*, 2019. 2

[13] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11444–11453, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[14] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1759–1769, 2020. 4

[15] Michael A Gomez, Rafal M Skiba, and Jacqueline C Snow. Graspable objects grab attention more than images do. *Psychological Science*, 29(2):206–218, 2018. 2

[16] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49, 2010. 5

[17] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 4

[18] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. *arXiv preprint arXiv:2006.12356*, 2020. 4

[19] Kaiyu Hang, Miao Li, Johannes A Stork, Yasemin Bekiroglu, Florian T Pokorny, Aude Billard, and Danica Kragic. Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on robotics*, 32(4):960–972, 2016. 2

[20] Aave Hannus, Frans W Cornelissen, Oliver Lindemann, and Harold Bekkering. Selection-for-action in visual search. *Acta psychologica*, 118(1-2):171–191, 2005. 2

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 2961–2969, 2017. 5

[22] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International conference on robotics and automation (ICRA)*, pages 3304–3311. IEEE, 2011. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776. IEEE, 2017. 2

[25] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research (IJRR)*, 34(4-5):705–724, 2015. 2

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 4

[27] 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 *2019 International Conference on Robotics and Automation (ICRA)*, pages 3629–3635. IEEE, 2019. 1, 2, 7

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4

[29] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio, and Ken 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 (RSS)*, Cambridge, Massachusetts, July 2017. 2, 3

[30] Douglas Morrison, Peter Corke, and Jurgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *Robotics: Science and Systems XIV*, pages 1–10, 2018. 2, 7

[31] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2901–2910, 2019. 1, 2

[32] P. Ni, W. Zhang, X. Zhu, and Q. Cao. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3619–3625, 2020. 1, 2

[33] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 5

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 652–660, 2017. 4, 5

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems (NeurIPS*, pages 5099–5108, 2017. 4

[36] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes. In *Conference on robot learning (CoRL)*, pages 53–65. PMLR, 2020. 1, 2

[37] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. *Proceedings IEEE International Conference on Robotics & Automation (ICRA)*, 2014. 2

[38] Catherine L Reed, Ryan Betz, John P Garza, and Ralph J Roberts. Grab it! biased attention in functional hand and tool space. *Attention, Perception, & Psychophysics*, 72(1):236–245, 2010. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*, pages 91–99, 2015. 5

[40] Francisco Suárez-Ruiz, Xian Zhou, and Quang-Cuong Pham. Can robots assemble an ikea chair? *Science Robotics*, 3(17), 2018. 1

[41] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research (IJRR)*, 36(13-14):1455–1473, 2017. 1, 2, 7

[42] Jacob Varley, Jonathan Weisz, Jared Weiss, and Peter Allen. Generating multi-fingered robotic grasps via deep learning. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4415–4420. IEEE, 2015. 2