Context-Aware Grasp Generation in Cluttered Scenes

Dinh-Cuong Hoang[1,2], Johannes A. Stork[2], and Todor Stoyanov[2]

Abstract— **Conventional methods to autonomous grasping rely on a pre-computed database with known objects to synthesize grasps, which is not possible for novel objects. On the other hand, recently proposed deep learning-based approaches have demonstrated the ability to generalize grasp for unknown objects. However, grasp generation still remains a challenging problem, especially in cluttered environments under partial occlusion. In this work, we propose an end-to-end deep learning approach for generating 6-DOF collision-free grasps given a 3D scene point cloud. To build robustness to occlusion, the proposed model generates candidates by casting votes and accumulating evidence for feasible grasp configurations. We exploit contextual information by encoding the dependency of objects in the scene into features to boost the performance of grasp generation. The contextual information enables our model to increase the likelihood that the generated grasps are collision-free. Our experimental results confirm that the proposed system performs favorably in terms of predicting object grasps in cluttered environments in comparison to the current state of the art methods.**

I. Introduction

Vision-based robotic grasping has been an active area of research since the early days of computer vision. The conventional model-based grasp generation approaches apply a 6D object pose estimation algorithm [1], [2] to register a CAD model of the object to be grasped to measured data. A set of grasps is then selected from a database of pre-computed grasps [3]. However, synthesizing grasps for unknown objects is not possible, as for the model-based approaches we assume that the 3D model of objects is available and a grasp database is pre-defined.

An alternative approach is to generate the grasp configurations directly from sensor data without assuming a known 3D model of the object or pre-computed grasps [4]. Inspired by the success of convolutional neural networks (CNNs) in a broad range of computer vision tasks, recent works [4], [5], [6], [7] rely entirely or partially on deep learning. Some methods only employ deep CNNs for finding features of a good grasp from data [4], [5], while others employ end-to-end learning for grasp generation [6], [7]. The reported results from both are promising across a wide variety of objects, sensors, and robot end effectors. However, the current state of the art CNN-based grasp generation methods utilize 2D or 2.5D input without taking the 3D geometry information into consideration. This might lead to failure to perform a grasp due to the lack of geometric analysis. Therefore, a few approaches have been proposed to localize grasps from 3D point sets [8], [9], [10], [11]. Although grasp generation methods in point clouds have achieved remarkable results, many problems remain unsolved. Due to measurement noise, occlusions, and undesirable contacts with the environment, generating feasible and reliable grasps in cluttered scenes is difficult. Many existing methods require time-consuming multi-stage processing for sampling grasp candidates and evaluating the grasp quality. While several works proposed end-to-end models for 6-DOF grasp generation and achieved state-of-the-art results in benchmarks, most of these methods rely only on features extracted by a backbone network such as PointNet++ [12] to predict grasps without considering the relationship between objects in the scenes. Grasping in clutter requires both reasoning about object parts and potential collisions with the gripper. Therefore, the contextual information, encapsulating the geometry of the rest of the scene, is important and should be taken into consideration to boost the performance of collision-free grasp generation in cluttered environments.

In this paper, we propose an end-to-end deep learning approach for generating grasp configurations for a two-finger parallel jaw gripper, based on 3D point cloud observations of the scene. The core of our approach is to encode the positional relationship between objects in the scene into features by a context learning module. The contextual information enables our model to increase the likelihood that the generated grasps are collision-free. To make the developed system robust to occlusion, we built our approach on top of a deep Hough voting architecture [13]. The voting mechanism allows our method to perform reliable grasp generation under clutter and occlusion.

The main contributions of our work can be summarized as follows: (1) A new framework for 6-DOF grasp generation named VoteGrasp, that robustly generates grasp configurations in cluttered environments under severe occlusion using a voting mechanism. (2) A context learning module encoding the dependency of objects in the scene into features to boost the performance of collision-free grasp generation. (3) Demonstration of the generalization capability of our method to novel objects.

II. RELATED WORK

A. 3D Point Cloud Based Grasp Generation

Machine learning-based approaches have been introduced to detect grasps from 3D point clouds [8], [9], [10], [11], with promising results across a wide variety of objects, sensors, and robots. ten Pas et al. [8] proposed a grasp pose detection (GPD) algorithm in point clouds that first generates a large set of grasp hypotheses by a sampling process and then classifies them as good or bad grasps. Extending on the idea

[1] ICT Department, FPT University, Hanoi, Vietnam.

[2] Centre for Applied Autonomous Sensor Systems (AASS), Orebro University, Sweden.

of GPD, PointNetGPD [10] replaces the CNN-based grasp quality evaluation model by a evaluation network using the architecture of PointNet [14]. Although both methods [8], [10] densely sample candidates, they are not able to generate grasps on regions such as rims of mugs or plates where they can not estimate surface normals correctly. To overcome this limitation, [9] considers grasp detection as sampling a set of grasps using a variational autoencoder, then assess and refines the sampled grasps using a grasp evaluator network. However, this approach only focuses on local features around the grasped object. To encode global information, [11], [15] abandon the conventional learning pipeline and takes the whole scene point clouds as input to regress the grasp poses. However, only relying on features extracted by a backbone network such as PointNet++ [12], these methods lack the consideration of the relationships between different objects, which limits their performance in cluttered scenes. As a result, they have not yet been demonstrated to be reliable under occlusion, which is common in manipulation domains. We address this challenge by leveraging a voting mechanism and contextual information to generate grasp configurations directly from 3D point clouds.

B. Hough Voting in Computer Vision

The Hough Transform has been widely used in computer vision for tasks like object detection [16], [17], motion detection [18], medical imaging [19], and robot navigation [20]. It was originally introduced to detect analytically defined shapes such as line, circle or ellipse [21], [22], [23] in 2D images. Today, Hough Transform or Hough voting usually refers to any detection process where evidence coming from local elements is accumulated to form a confident detection. Voting-based approaches [17], [13] demonstrated the ability to perform reliable detection under clutter and occlusion. This is due to the additive attribute of the Hough transform makes the method robust to partial occlusions. Tombari and Di Stefano [17] proposed a Hough Voting approach for object recognition in 3D scenes. Each corresponding feature can cast a vote to accumulate evidence for possible object centers. This permits simultaneous voting of all feature correspondences in the 3D Hough space. [24] presented a 3D object detection and pose estimation method by combining neural networks and a local voting-based approach. Recently, VoteNet [13] was introduced to detect objects via point feature grouping, sampling, and voting. VoteNet directly votes for virtual center points of objects from point clouds and generates a group of high-quality 3D object proposals by aggregating vote features. The experimental results show that the developed methods above perform well in 3D scenes with a significant degree of occlusion and clutter. Motivated by the success of Hough voting-based approaches in the object detection task, especially by VoteNet, we employ the deep Hough voting architecture in [13] to make our grasp generation system robust to occlusion.

C. Context and Attention in 3D point clouds

Much prior work has widely explored the use of the contextual information to improve the performance of 3D point matching [25], point cloud semantic segmentation [26], instance segmentation of 3D point clouds [27], and 3D scene layout prediction [28]. Motivated by the success in natural language processing, recent works have focused on leveraging the self-attention mechanism with contextual dependency to achieve more accurate results in various perception tasks. [29] connects the self-attention idea with shape context to propose ShapeContextNet that can be applied to the general point cloud classification and segmentation problems. In [30], the authors proposed a point contextual attention network for point cloud based retrieval. It takes the local point features and produces an attention map that enables the network to find more important features and produce a more discriminative global descriptor. Paigwar et al. [31] use a visual attention mechanism with point clouds to achieve accurate detection of objects. The core idea behind attention mechanisms is to pay more attention to the related parts of input. Grasping in clutter requires both reasoning about object parts and potential collisions with the gripper. Therefore, we stipulate that self-attention based context learning is well suited to our problem of interest.

III. VoteGrasp Approach

We introduce our end-to-end grasp generation network given scene point cloud inputs, which is illustrated in Fig. 1. In this work, we address the problem of generating grasps for any desired object in a cluttered scene from partial point cloud observations. The input to our approach is a point cloud of size $N \times 3$. The network aims to predict a ranked list of grasps, where each grasp $G = (p, R, w, q)$ specified by a center $p = (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{R}^3$, the gripper orientation $R \in SO(3)$, a width of the gripper $w \in \mathbb{R}$, and a grasp quality measure $q \in [0, 1]$. Due to the non-linearity of the rotation space as explained by [32], directly regressing the 3D orientation is difficult. Therefore, we reformulated the gripper orientation estimation as in [33]. We decouple orientation prediction into first recovering a viewpoint anchor (a discrete viewpoint classification task) and then estimating an in-plane rotation as a mixture of classification and regression formulations. In the rest of this section, we will examine each of the main components of our proposed architecture.

Backbone Network: In order to extract geometric features, we utilize the PointNet++ architecture with multi-scale grouping as our backbone network. Thereby, we are able to capture fine geometric structures from the neighborhood of each point. The backbone network selects M interest points (called seed points) and enriches them with high-dimensional features $\{s_i\}_{i=1}^{M}$ where $s_i = [x_i; f_i]$ with $x_i \in \mathbb{R}^3$ being the seed location in 3D space and $f_i \in \mathbb{R}^F$ being a feature vector.

Vote and Cluster: The seed points $\{s_i\}_{i=1}^{M}$ are then fed into a multi-layer perceptron (MLP) to compute votes $\{\{v_{ij} = [y_{ij}; g_{ij}] \in \mathbb{R}^{3+F}\}_{i=1}^{M}\}_{j=1}^{J}$, J votes per seed. The

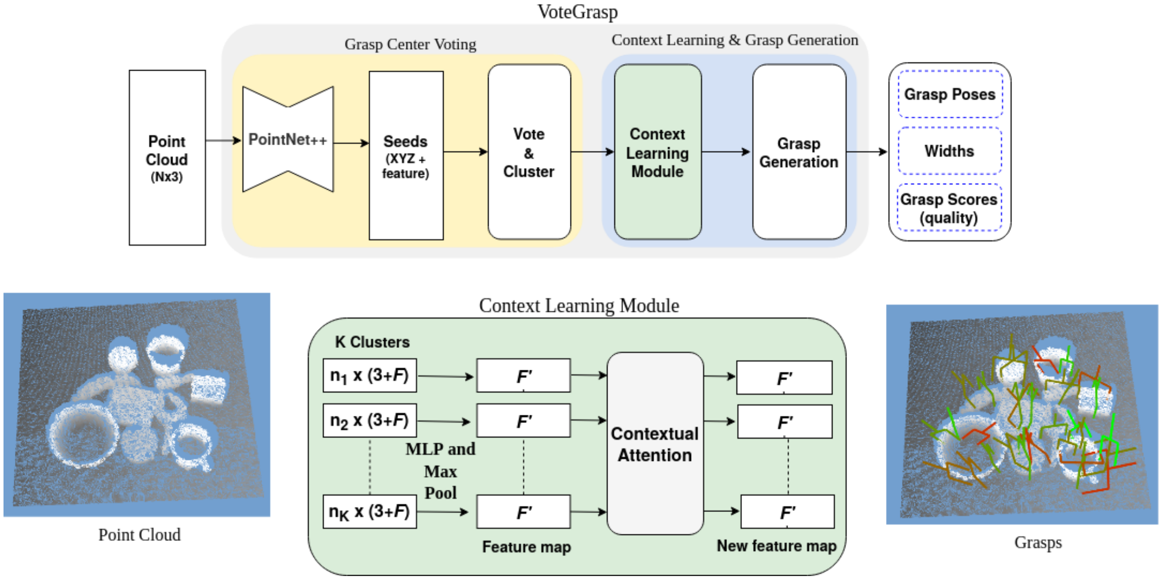


Fig. 1: The architecture of the proposed VoteGrasp for 6-DOF grasp generation in point cloud data. Our model builds on a deep Hough voting neural network [13] to vote grasps and is added a self-attention context learning module. Grasps are color-coded by the predicted quality scores. Green is the highest and red is the lowest.

MLP consists of fully connected layers, ReLU and batch normalization. Each vote v_{ij} is represented by a point y_{ij} in 3D space with its Euclidean coordinates supervised to be close to a grasp center, and a feature vector g_{ij} learned for the final grasp generation task (F-dimensions). VoteGrasp computes multiple votes per seed $\mathbf{V} = \{v_j\}$ with $j = 1, .., J$. This is because we aim to estimate more than one grasp pose for each object. The next step is to cluster the votes by uniform sampling and finding neighboring votes within a certain Euclidean distance. Given input votes $\{v_i = [y_i; g_i] \in \mathbb{R}^{3+F}\}_{i=1}^{M \times J}$, we use iterative farthest point sampling (FPS) based on $\{y_i\}$ to choose a subset of K votes $\{v_{i_k}\}_{k=1}^{K}$. To find neighboring votes, ball query finds all votes that are within a radius to the query vote v_{i_k}. The output are K groups of vote sets of size $K \times n_k \times (3 + F)$, where each group corresponds to a grasp center and n_k is the number of votes in the neighborhood of the vote v_{i_k}.

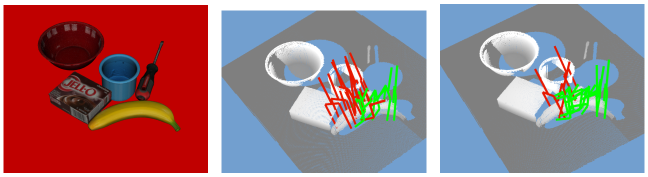

(a) (b) (c)

Fig. 2: An example to illustrate the effectiveness of context learning module on grasp generation: (a) simulated scene; (b) result without the context module; (c) result with context module. The red grasps are not collision-free. Here we only visualize grasps for the target object (banana).

Context Learning: Grasping in cluttered environments requires both reasoning about invisible object parts and potential collisions with the manipulator. Therefore, it is important to encode the relationship of objects in the scene into features or exploit contextual information outside of interest regions for detecting collision-free grasps. However, the VoteNet architecture is designed to detect each object individually. Indeed each cluster $\mathcal{C}_k$ is independently pushed through the MLP layer to regress its object class and bounding box. Context outside a cluster is crucial and could help make more informed grasp predictions. Therefore, instead of processing each cluster independently to predict grasps, our network computes a new feature map from all clusters to learn the context that considers the relationships between all clusters. We find inspiration from self-attention based models [34], [29], [35], [36] to add a contextual module into our framework to capture the contextual information in 3D points. By leveraging a self-attention mechanism, we can combine features from other clusters to give more information on the object relationships. More specifically, we first aggregate features from votes in each cluster. Votes $\{v_i = [y_i; g_i] \in \mathbb{R}^{3+F}\}_{i=1}^{n_k}$ in cluster k are fed into a MLP network before being max-pooled to a single feature vector $C_k \in \mathbb{R}^{F'}$. At this stage, we have a feature map $C = [\mathcal{C}_1; \mathcal{C}_2; ...; \mathcal{C}_K] \in \mathbb{R}^{K \times F'}$ from K clusters summarizing local context. In order to enable features to become aware of their global neighborhood, we explicitly model higher-order interactions between features in C, and it can be formulated as the non-local operation:

$$C_{context} = f(\theta(C)\phi(C))g(C) \qquad (1)$$

where $\theta(\cdot)$, $\phi(\cdot)$, $g(\cdot)$ are learnable transformations on the input feature map C, and $f(\cdot)$ encodes the relation between all positions. Following [35], we use the 1×1 convolution for the transformations:

$$\theta(C) = CW_{\theta} \in \mathbb{R}^{K\times F'} \tag{2}$$

$$\phi(C) = CW_{\phi} \in \mathbb{R}^{K\times F'} \tag{3}$$

$$g(C) = CW_{g} \in \mathbb{R}^{K\times F'} \tag{4}$$

parameterized by the weight matrices $W_{\theta}, W_{\phi}, W_{g} \in \mathbb{R}^{F'\times F'}$ respectively. The function $f(\cdot,\cdot) : \mathbb{R}^{K\times F'} \times \mathbb{R}^{K\times F'} \rightarrow \mathbb{R}^{K\times F'}$ computes the affinity between all positions. f is defined as a dot-product similarity:

$$f(\theta(C), \phi(C)) = \theta(C)\phi(C)^{\top} \tag{5}$$

As discussed in [35], the non-local operation is related to the self-attention [34] method. The self-attention mechanism allows the features from different clusters to interact with each other. The output is a new feature map of the same size $C_{context} = [\mathcal{C}_1^{ct}; \mathcal{C}_2^{ct}; ...; \mathcal{C}_K^{ct}] \in \mathbb{R}^{K\times F'}$. The effectiveness of the context learning module is visualized in Fig. 2. As we can see, when context is taken into account fewer of the grasps generated on a target object (banana) are in collision (shown in red) with neighboring objects.

Grasp Generation: Given a new feature map $C_{context} = [\mathcal{C}_1^{ct}; \mathcal{C}_2^{ct}; ...; \mathcal{C}_K^{ct}] \in \mathbb{R}^{K\times F'}$, a multi-layer perceptron network is applied to output a ranked list of grasps, where each grasp $G = (p, R, w, q)$ specified by a center $p = (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{R}^3$, the gripper orientation $R \in SO(3)$, a width of the gripper $w \in \mathbb{R}$, and a grasp quality measure $q \in [0, 1]$. To be specific, each $\mathcal{C}_k^{ct}$ is further processed by a multi-layer perceptron composed of 3 fully connected layers. All fully connected layers are followed by batch normalization and ReLU except for the last prediction layer. The prediction layer has $5 + V + 2A$ channels where the output consists of 3 grasp center regression values, 1 gripper width regression value, 1 grasp confidence regression value, V viewpoint scores, A angle scores (in-plane rotation), and A angle residual regression values (in-plane rotation). V and A denote the numbers of sampled viewpoints and in-plane rotations respectively.

Loss function: We supervise the learning of modules jointly with a multi-tasks loss:

$$L_{votegrasp} = L_{vote} + L_{grasp} \tag{6}$$

The VoteGrasp loss $L_{votegrasp}$ includes a voting loss L_{vote} and a grasp estimation loss L_{grasp}. To supervise the learning of votes $\{v_i = [y_i; g_i] \in \mathbb{R}^{3+F}\}_{i=1}^{M\times J}$, we apply an regression loss:

$$L_{vote} = \frac{1}{M_o} \sum_{i} \|y_i - c_i^g\|_H \cdot \mathbb{1}(x_i) \tag{7}$$

where M_o is the count of the total number of seeds on the object surface, c_i^g is the closest ground truth grasp center, $\|\cdot\|_H$ is the Huber norm and $\mathbb{1}(\cdot)$ is a binary function indicating whether a seed point s_i belongs to an object. We define the grasp loss function as follows:

$$L_{grasp} = L_{center} + \alpha L_{rot} + \beta L_{width} + \gamma L_{score} \tag{8}$$

where α, β and γ are weights that scale the losses to similar scales. The grasp loss is composed of a grasp center loss L_{center} (regression), a rotation loss $L_{rot} = L_{viewpoint} + L_{in-plane}$, a gripper width loss L_{width} (regression), and a grasp confidence score L_{score} (regression). The loss $L_{viewpoint}$ is for viewpoint classification. Meanwhile, for the in-plane rotation estimation, we use a mixture of classification and regression formulations $L_{in-plane} = 0.1L_{angle-cls} + L_{angle-reg}$ as in [37]. For all regression loss components of L_{grasp} we use the robust L1-smooth loss [38], while for classification the standard cross entropy loss is employed.

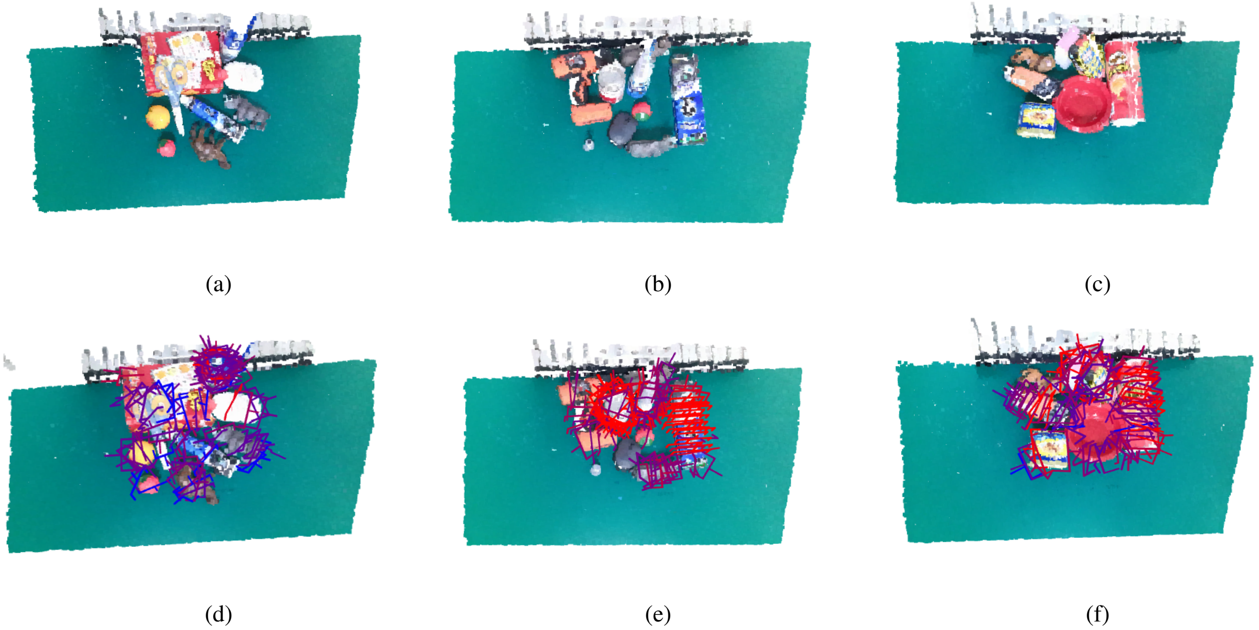
IV. Implementation Details

Network Architecture. In our implementation, we randomly choose N=50k points from each raw point cloud and set α=β=γ=1.0 in Eq. 8. We then apply the PointNet++ [12] based feature learning network, which has 4 set abstraction layers (SA) and 2 feature propagation layers (FP). The FP2 outputs $M = 1024$ seeds with $F = 256 - dim$ features and 3D coordinates that will be transformed to votes. The voting module generates $J = 10$ votes per seed with an MLP layer spec: $[256, 256, 259 \times 10]$. In the context module, we form $K = 1024$ clusters and output a new feature map $C_{context} \in K \times F'$ where $K = 1024, F' = 128$. In the last step, 1024 grasps are generated from the new feature map. The prediction layer has $5 + V + 2A$ channels where $V = 120$, and $A = 6$.

V. Evaluation

In this section we aim to determine to what extent our proposed aproach utilizes the available training data to generate feasible grasp candidates. We are particularly interested to evaluate how well the learned model generalizes to novel object categories and in what way it compares to current state of the art. Finally, we evaluate the robustness of our approach to clutter and explore to what extent the use of context learning can mitigate the negative effects of occlusions.

To answer the above questions, we evaluate our method and compare with other state of the art methods on the public dataset GraspNet-1Billion [11]. This is a large-scale grasp dataset collected from cluttered scenes considering multi-object-multi-grasp setting. The objects in GraspNet-1Billion have varying shapes, textures, sizes, materials and under different occlusion conditions. Hence, it can be used to evaluate robustness to occlusion and the generalization ability of our trained model. The proposed network is trained from scratch in an end-to-end manner. We train the entire network with the batch size 8 and use Adam for optimization, with a learning rate of 0.001 over 200 epochs. It takes around 80 hours for training on one Nvidia GeForce RTX 2080 Ti 10GB GPU. For inference, the forward-pass time of VoteGrasp for a single scene with size 50k points is 150ms.


(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3: Examples of input point clouds and predicted grasps from our proposed method; (a-c) input point clouds in GraspNet-1Billion dataset [11]; (d-f) grasps generated by VoteGrasp. Grasps are color-coded by the confidence score. Red is the highest and blue is the lowest.

TABLE I: The table shows the results on GraspNet-1Billion test set captured by RealSense/Kinect sensors respectively. $Ours^{-}$ denotes our proposed network without context learning module.

	Seen			Unseen (but similar)			Novel		
	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$
GG-CNN [39]	15.5/16.9	21.8/22.5	10.3/11.2	13.3/15.1	18.4/19.8	4.6/6.2	5.5/7.4	5.9/8.8	1.9/1.3
Chu et al. [40]	16.0/17.6	23.7/24.7	10.8/12.7	15.4/17.4	20.2/21.6	7.1/8.9	7.6/8.0	8.7/9.3	2.5/1.8
GPD [8]	22.9/24.4	28.5/30.2	12.8/13.5	21.3/23.2	27.8/28.6	9.6/11.3	8.2/9.6	8.9/10.1	2.7/3.2
PointNetGPD [10]	26.0/27.6	33.0/34.2	15.4/17.8	22.7/24.4	29.2/30.8	10.8/12.8	9.2/10.7	9.9/11.2	2.7/3.2
Fang et al. [11]	27.6/29.9	33.4/36.2	17.0/19.3	26.1/27.8	34.2/33.2	14.2/16.6	10.6/11.5	11.3/12.9	4.0/3.6
Gou et al. [41]	28.0/32.1	33.5/39.5	17.8/20.9	27.2/30.4	36.3/37.9	15.6/18.7	12.3/13.1	12.5/13.8	5.6/6.0
$Ours^{-}$	29.2/33.8	34.7/41.0	19.1/22.2	28.3/31.7	37.2/39.1	16.8/19.9	13.6/15.0	13.7/15.1	6.8/7.2
Ours	**34.1/37.5**	**38.9/45.6**	**24.0/27.7**	**33.0/35.9**	**40.8/43.3**	**20.5/24.7**	**16.9/18.5**	**17.0/18.5**	**10.0/10.6**

A. Dataset

The GraspNet-1Billion [11] consists of 97,280 RGB-D images captured from 190 cluttered scenes. The dataset provides over one billion grasp poses for 88 objects presented in the scenes. An accurate 3D mesh model of each object is available as well. Besides, they also provide camera poses, 6D object poses, object masks and bounding boxes for all frames. The rich annotations allow us to generate ground truth votes and grasp configurations easily. Following [11] we split the dataset into 100 scenes for training and 90 scenes for testing. To evaluate model generalizability, the test sets are divided into 30 scenes with novel objects, 30 for unseen but similar objects, and the rest for seen objects.

B. Evaluation Metric

We follow prior work [11] and evaluate our result on the dataset using $Precision@k$. This metric measures the precision of top-k ranked grasps. We first check whether a predicted grasp (G_p) is true positive or not. It is considered a true positive only if the grasp satisfies three conditions: (i) there is an object inside the gripper; (ii) it is collision-free; (iii) the grasp is antipodal under a given friction coefficient μ. The third condition is computed based on the prior works [8], [11]. We let AP_μ denote the average $Precision@k$ for k ranges from 1 to 50 given a friction coefficient μ. Besides AP_μ, we also report the average of AP_μ with $\mu = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, denoted as AP.

C. Results

Fig. 3 shows qualitative results of our predicted grasp poses. Table I shows the performance of our approach compared to state of the art methods. We evaluated our trained model using the implementation of the evaluation metric shared by the authors of [11] enabling a direct comparison with the results of related works reported in [11], [41]. From the results presented in the table, we found that all the methods overall perform better in scenes with seen objects

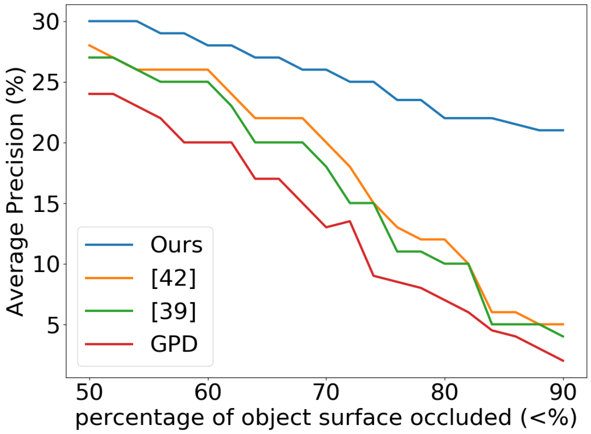


Fig. 4: Performance of different approaches under increasing levels of occlusion on the synthetic test set.

than scenes for novel objects. Notably, the AP score of our method surpasses the others in all the test sets by a large margin. Even on the scenes with novel objects, the proposed model still has an averaged 5.0% improvement over the best baseline [41]. This implies that our model is able to generalize and perform well on novel objects. Moreover, in order to evaluate the robustness of algorithms towards occlusion, we perform grasp generation under increasing levels of occlusion. To estimate the levels of occlusion, we calculate the visible surface ratio of each object instance. Fig. 4 illustrates how methods are influenced by different levels of occlusion. As shown, VoteGrasp performs well even when objects are heavily occluded, while the results of the previous approaches indicate high sensitivity to occlusion.

D. Ablation Study

TABLE II: Effects of number of votes per seed to the performance of our model. Evaluation metric is AP on GraspNet-1Billion [11].

	Seen	Unseen (but similar)	Novel
J=1	**34.8/38.0**	32.6/35.5	13.2/15.6
J=5	34.5/37.8	32.8/35.6	14.5/16.9
J=10	34.1/37.5	**33.0/35.9**	**16.9/18.5**
J=15	32.4/35.1	31.3/33.7	14.0/16.6
J=20	29.1/32.2	28.0/31.1	12.2/14.0

We modified VoteNet architecture to directly synthesize grasps from the vote aggregated features and used it as the baseline method ($VoteNet^*$ or $Ours^-$). Our results in Table I show that the modified VoteNet performs favorably in terms of grasp generation in cluttered environments in comparison to other state-of-the-art methods. This confirms that voting mechanism is well suitable for the problem of interest. Furthermore, we validate the effectiveness of the self-attention contextual module of our network by comparing it with the model that directly generates grasps without context learning. According to results in Table I, we see an improvement of 4.3%, from 31.5% to 35.8% for seen objects (averaged AP from both cameras). We also observe marked improvements, from 30.0% to 34.5% with unseen (but similar) objects and from 14.3% to 17.7% for novel objects. It confirms that our method greatly benefits from the use of contextual information. Furthermore, we evaluate the effects of the number of votes per seed J on the performance of our model. Our model tends to perform better on seen objects with a smaller value of J, but does not generalize well for novel objects. We find that $J = 10$ votes per seed achieve the best results in scenes with novel objects, as shown in Table II.

E. Robotic Grasping Experiment

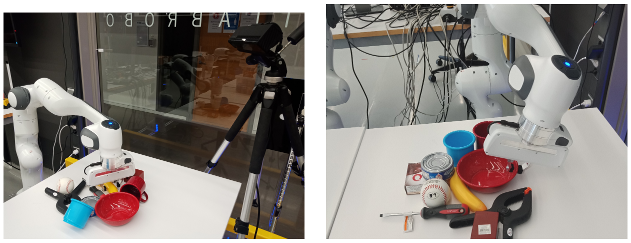


Fig. 5: Real-world grasping experiment.

The experiments were conducted with a Franka Emika Panda robot arm with 7-DOF, equipped with a parallel-jaw gripper as shown in Fig. 5. To capture the input point clouds, we used either ASUS Xtion PRO LIVE sensor or Microsoft Kinect sensor v2. The whole system is implemented using the ROS and MoveIt! frameworks. The objects are randomly placed within the workspace of the robot arm and the camera. A grasp was considered a success if the robot could grasp and lift the object within one attempt. The robot succeeds on 70% of the grasps using our proposed approach.

VI. Conclusions

In this work we introduced VoteGrasp — an end-to-end 6-DOF grasp generation network operating on 3D point clouds. The main contribution of this paper is to show that by taking advantage of the deep Hough voting mechanism and contextual information we are able to improve the performance of grasp generation compared to the previous state of the art methods. Through experiments, we demonstrate that VoteGrasp is highly robust to clutter and occlusions. Importantly, the results confirm that our proposed model is able to generalize and perform well on novel objects. Interesting future work is to consider adding a reachability predictor to the grasping network and explore the use of our approach in task planning applications.

VII. Acknowledgment

This work was supported by the European Union's Horizon 2020 research and innovation program under agreement no. 101017274 (DARKO).

References

[1] E. Muñoz, Y. Konishi, V. Murino, and A. Del Bue, "Fast 6d pose estimation for texture-less objects from a single rgb image," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5623–5630.

[2] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1386–1383.

[3] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

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

[5] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[6] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[7] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[8] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[9] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2901–2910.

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

[11] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 444–11 453.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[13] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9277–9286.

[14] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[15] 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)*, 2020, pp. 3619–3625.

[16] T. M. Silberberg, L. Davis, and D. Harwood, "An iterative hough procedure for three-dimensional object recognition," *Pattern Recognition*, vol. 17, no. 6, pp. 621–629, 1984.

[17] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *2010 Fourth Pacific-Rim Symposium on Image and Video Technology*. IEEE, 2010, pp. 349–355.

[18] H. Kälviäinen, "Motion detection using the randomised hough transform: exploiting gradient information and detecting multiple moving objects," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 143, no. 6, pp. 361–369, 1996.

[19] S. Golemati, J. Stoitsis, T. Balkizas, and K. Nikita, "Comparison of b-mode, m-mode and hough transform methods for measurement of arterial diastolic and systolic diameters," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2006, pp. 1758–1761.

[20] L. Iocchi, D. Mastrantuono, and D. Nardi, "A probabilistic approach to hough localization," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 4. IEEE, 2001, pp. 4250–4255.

[21] P. V. Hough, "Machine analysis of bubble chamber pictures," in *Proc. of the International Conference on High Energy Accelerators and Instrumentation, Sept. 1959*, 1959, pp. 554–556.

[22] P. Hough, "Method and means for recognizing complex patterns us patent 3,069,654, 1962," *Appendix i Development of Equation*, vol. 5.

[23] R. Duda and P. Hart, "Use of the hough trans-form to detect lines andl curves in pictures," *Coin-mun. ACM*, vol. 15.

[24] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 205–220.

[25] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 195–205.

[26] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3d recurrent neural networks with context fusion for point cloud semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 403–417.

[27] S.-M. Hu, J.-X. Cai, and Y.-K. Lai, "Semantic labeling and instance segmentation of 3d point clouds using patch context analysis and multiscale processing," *IEEE transactions on visualization and computer graphics*, 2018.

[28] Y. Shi, A. X. Chang, Z. Wu, M. Savva, and K. Xu, "Hierarchy denoising recursive autoencoders for 3d scene layout prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1771–1780.

[29] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4606–4615.

[30] W. Zhang and C. Xiao, "Pcan: 3d attention map learning using contextual information for point cloud based retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 436–12 445.

[31] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, "Attentional pointnet for 3d-object detection in point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[32] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[33] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[35] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[36] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[37] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[39] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[40] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[41] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgbd images," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 459–13 466.