Deep Learning Based 6-DoF Antipodal Grasp Planning From Point Cloud in Random Bin-Picking Task Using Single-View

Tat Hieu Bui, Yeong Gwang Son, Seung Jae Moon, Quang Huy Nguyen, Issac Rhee, Ju Yong Hong, and Hyouk Ryeol Choi, *Fellow, IEEE*

***Abstract*—Random bin picking is a crucial task in logistic centers, which is driven by E-Commerce growth. In this letter, we present an end-to-end method for 6-DoF antipodal grasps from cluttered scenes. Our approach includes two main steps: finding Potential Grasp Areas (PGAs) from depth image of the bin and detecting suitable parallel grasps in PGAs from point cloud data. To support our work, the training datasets are generated automatically in Pybullet simulation environment including 5000 depth images and above 30 000 point clouds of cluttered scenes with different number of objects, which save time significantly for collecting and labeling. We implemented real grasping experiments with a robot arm UR10, 2-finger gripper, depth camera L515, and 10 objects arranged randomly in the bin to evaluate the efficiency of this method. It is simple, fast, and efficient to deal with many kinds of object which are random in shape, dimension, pose, and material.**

***Index Terms*—Computer vision for automation, data sets for robotic vision, deep learning in grasping and manipulation.**

I. Introduction

GRASPING objects is a fundamental manipulation in robotics, and it plays a crucial role in automation processes. A robotic pick-and-place system that employs a robot arm, camera, and gripper is extensively utilized not only in industrial settings but also in everyday life. Our work focuses on addressing the challenge of random bin-picking tasks in logistics centers, which requires a system to automatically pick objects in a highly cluttered scene as quickly as possible.

There are two primary approaches to robotic object grasping: model-based and model-free. Model-based methods are classical approaches that involve estimating the object's pose beforehand and associating it with a pre-defined set of candidates to generate feasible grasp poses. However, these methods are inefficient when working with novel objects that are not within the limited number of 3D object models. Additionally, object pose estimation or segmentation can be challenging with noisy camera data and cluttered scenes. In contrast, model-free solutions do not require any geometric information about the target objects, making them a generalization for many types of objects and scenarios. However, recognizing grasp poses in 3D space can be difficult for machine learning. To simplify the problem, some works only focus on top-down grasping, which is perpendicular to the table [1], [2], [3], [26]. This approach has limitations in the kinematics of robots and its diversity of applications, especially when picking objects from a heap. For example, although Dex-Net 2.0 showed good performance by building 6.7 million synthetic data on single objects, we do not see its effectiveness in dealing with object overlap, even in the bin-picking version of Dex-net 4.0 [29]. In general, although top-down antipodal grasps are effective for upright objects on a flat surface, they may not be suitable for inclined objects and collision avoidance. On the other hand, we recognize that the full 6-DoF grasp space is unnecessary for the bin-picking task, as many grasp poses can be eliminated to avoid collisions with the bin, particularly those with approximately horizontal approaching vectors. To address these issues, we propose an end-to-end method that uses depth image and point cloud data to generate top-down biased 6-DoF antipodal grasps that are more flexible and dexterous for unknown objects in cluttered scenes. Our solution draws inspiration from Contact-GraspNet [4], which predicts 6-DoF parallel-jaw grasp distribution directly from a depth recording of a scene. However, Contact-GraspNet's training datasets were generated by simulating grasp trials on single 3D mesh models from ACRONYM datasets, and it did not present a clear method for eliminating collision grasps in cluttered scenes. Additionally, object segmentation was used to distinguish grasps belonging to a target object, which is challenging in overlapping scenes and noisy cameras. In contrast, our training dataset generation includes detailed calculations about collisions and free space for grasp approaches in object overlap situations. Our solution is divided into two main steps: finding PGAs to reduce operating time and detecting suitable antipodal grasps in PGAs from point cloud data. Our approach generates 6-DoF grasps directly from the point cloud of PGAs, independent of object segmentation errors (Fig. 1).

Manuscript received 28 February 2023; accepted 18 June 2023. Date of publication 7 July 2023; date of current version 14 July 2023. This letter was recommended for publication by Associate Editor Y. Hirata and Editor H. Moon upon evaluation of the reviewers' comments. This work was supported by the Ministry of Trade, Industry and Energy (MOTIE, Korea), through the Industrial Strategic Technology Development Program under Grant 20014558. *(Corresponding author: Hyouk Ryeol Choi.)*

The authors are with the School of Mechanical Engineering, Sungkyunkwan University, Suwon 16419, South Korea (e-mail: buitathieu1995@gmail.com; syoungk20@gmail.com; msj19@skku.edu; nqhuy@skku.edu; issacrh@skku.edu; juyong0000@skku.edu; choihyoukryeol@gmail.com).

Video of the real robotic experiments is available at https://www.youtube.com/watch?v=cx5THPyIKjA

Digital Object Identifier 10.1109/LRA.2023.3293314

Thus, our contribution is the following:

- A framework for recognizing the Potential Grasp Areas (PGAs) that contain objects possible for antipodal grasps in clutter scenes (step 1).
- A framework for detecting antipodal grasp poses in PGAs from point cloud data (step 2).
- An algorithm for automatically generating training datasets in pybullet simulation environment.
- An end-to-end method includes two steps for planning antipodal grasps in the random bin-picking application.

II. Related Works

The manipulation of pick-and-place has been a topic of research for many years, with various methods of classification based on object grasping methods, applications, or approaches [5], [6], [7]. In this section, we will focus on data-driven approaches.

One common research direction in pick-and-place operations is the use of reinforcement learning (RL) to learn an end-to-end strategy for similar tasks based on experiences with planning, grasping, and moving robots [8]. Teaching and rewarding procedures have been widely applied to bin-picking tasks, in particular, and object grasping in general [9], [10], [11], [12]. However, one main disadvantage of this method is its limited ability to generalize to new workspaces. Additionally, grasp pose selection remains an open problem in this field.

Another approach is to recognize or estimate the object's pose in advance and choose suitable grasps based on identified attributes [13], [14], [15]. These solutions often work with bins containing only one type of object or on objects with simple poses and shapes. Moreover, there is the issue of camera noise leading to errors in detecting key points or segmenting objects. A multi-view system [16] may be required to increase information about the object from many angles, which can be time-consuming for the algorithm and cumbersome to set up in a workspace.

In recent years, powerful methods that generate direct grasp poses by applying deep learning on RGB, depth, or point cloud data have shown promising results in object picking tasks. For example, works such as [1], [17], [18], [27], [29] have trained models using deep convolutional neural networks to classify or predict grasps from color and depth images. Even in the Amazon Picking Challenge, the MIT-Princeton team [19] achieved first place using multiple motion primitives for suction and grasping, classified by a trained model from RGB-D images. However, these methods only propose up-down parallel grasps or focus on suction to lift objects since most contact points of a two-finger grasp are often near the edges of the object, which lacks visibility from the camera. Some researchers have used point cloud data [20], [28] to have a more detailed view and search for more optimal grasps. PointNetGPD [20] employs a strategy [21] to generate parallel grasp candidates ranked by the model trained from point clouds within the closing area of the gripper. Contact-GraspNet [4] considers more about collisions in operation, but it is not entirely clear and still has potential problems when working with a high density of objects. Therefore, in this letter, we propose a solution that predicts feasible contact points on point clouds of highly cluttered scenes, independent of segmentation to restrict the effect of noise and with careful consideration of collisions in generating training datasets.

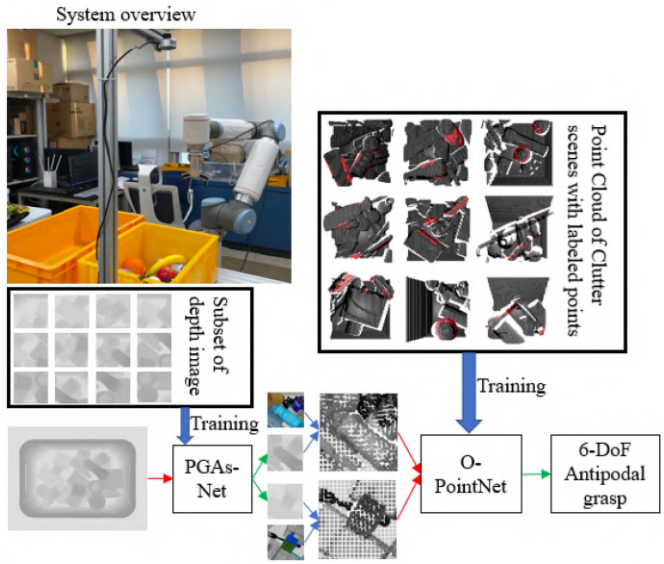

Fig. 1. An illustration for our bin picking system using a single up-view camera: The framework includes two models trained by synthetic datasets for grasp candidate generation.

III. Problem Statement

A. 6-DoF Grasp Representation

In single-view system, almost two-finger grasps only have one of two contact points visible with the camera. Hence, Contact-GraspNet [4] represents antipodal grasp by four parameters: coordinate of first contact point $p \in \mathbb{R}^3$, grasp width $w \in \mathbb{R}$, approaching vector $\mathbf{a} \in \mathbb{R}^3$ and grasp baseline vector $\mathbf{b} \in \mathbb{R}^3$, which is demonstrated in Fig. 3(a): g is defined by $(R_g, t_g) \in SE(3)$

$$t_g = p + (w/2) \cdot \mathbf{b} + d \cdot \mathbf{a} \tag{1}$$

$$R_g = \begin{pmatrix} | & | & | \\ \mathbf{b} & \mathbf{a} \times \mathbf{b} & \mathbf{a} \\ | & | & | \end{pmatrix} \tag{2}$$

$||\mathbf{a}|| = 1$ and $||\mathbf{b}|| = 1$; d is the constant distance from the gripper baseline to the gripper base.

B. Objective

Our research is implemented to solve the problem of planning 6-DoF antipodal grasps for bin-picking tasks based on point clouds from single-view. The main purpose is to learn a function that inputs point clouds from RGB-D images as a set of 3D points $\{c_i | i = 1, \ldots, n\}$, where each point c_i is a vector of its (x, y, z) and outputs $n \times 2$ scores and grasp widths respectively for each of the n points to reconstruct 6-DoF grasp poses. Earlier, we learn another function for finding PGAs to reduce the algorithm's time on the entire bin space.

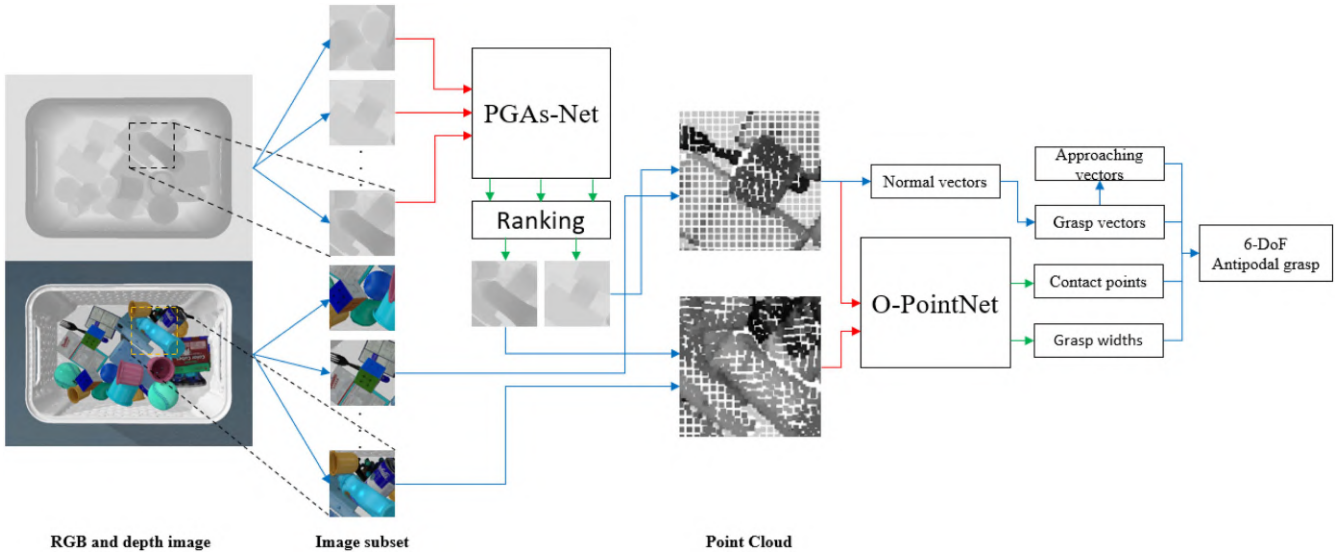

Fig. 2. Overview of our method: The RGB and depth image is split into small images representing regions in the bin. Then, the subsets of depth images are fed to PGAs-Net for ranking and recognizing Potential Grasp Areas. If the region is PGA, it will combine with the RGB image to render the point cloud which is fed to O-PointNet to acquire positive contact points and grasp widths respectively. Along with normal vectors calculated from point cloud and approaching vector inferred in specific conditions, 6-DoF antipodal grasps are reconstructed.

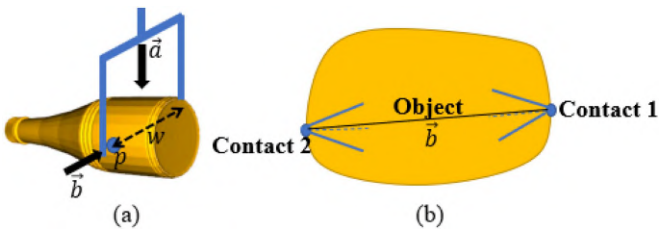

Fig. 3. (a) Parallel grasp representation in Contact-GraspNet. (b) Antipodal grasp principle with friction cones. The angle of cones depends on the friction coefficient μ at contact points. The grasp baseline vector has to lie on both friction cones at 2 contact points.

As mentioned in Section III-A, to reconstruct 6-DoF grasp poses, we need to recognize $g = (p, w, \mathbf{a}, \mathbf{b})$. We have: p and w are the output of O-PointNet, $\mathbf{b}$ is normal vectors estimated from point cloud. We need to find approaching vector $\mathbf{a}$. Assume: $p = (x_0, y_0, z_0)$ and $\mathbf{b}$ is normalized (x_b, y_b, z_b)

Approaching vectors $\mathbf{a}$ is inferred as follows:

- Build the plane Δ having normal vector $\overrightarrow{n} = [Oz, \mathbf{b}] = (y_b, -x_b, 0)$ and going through $p = (x_0, y_0, z_0)$

$$\Delta : y_b(x - x_0) - x_b(y - y_0) = 0 \tag{3}$$

- $\mathbf{a}(x_a, y_a, z_a)$ is recognized by conditions:

$$\begin{cases} \mathbf{a} \in \Delta : y_b(x_a - x_0) - x_b(y_a - y_0) = 0 \\ \mathbf{a} \perp \mathbf{b} : x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b = 0 \\ ||\mathbf{a}|| = 1 : \sqrt{{x_a}^2 + {y_a}^2 + {z_a}^2} = 1 \end{cases} \tag{4}$$

- Solve the system of (4), we have:

$$\begin{cases} y_a = \frac{y_b}{x_b} x_a - \frac{y_b x_0 + x_b y_0}{x_b} \\ z_a = -\frac{{x_b}^2 + {y_b}^2}{x_b z_b} x_a + \frac{{y_b}^2 x_0 + x_b y_b y_0}{x_b z_b} \\ A{x_a}^2 + Bx_a + C = 0 \end{cases}$$

$$A = 1 + \left(\frac{y_b}{x_b}\right)^2 + \left(\frac{{x_b}^2 + {y_b}^2}{x_b z_b}\right)^2$$
$$B = -2\frac{\left({x_b}^2 + {y_b}^2 + {z_b}^2\right)\left({y_b}^2 x_0 + x_b y_b z_0\right)}{{x_b}^2 {z_b}^2}$$
$$C = \left(\frac{y_b x_0 + x_b y_0}{x_b}\right)^2 + \left(\frac{{y_b}^2 x_0 + x_b y_b y_0}{x_b z_b}\right)^2 - 1$$

IV. Proposed Method

A. Overview

The bin-picking task presents challenges not only in identifying multiple grasp poses from various directions to avoid collisions but also in optimizing the algorithm's computing time. To address this, Step 1 of the proposed method involves finding Potential Grasp Areas (PGAs) in the bin to minimize the algorithm's computation time. The depth image of the bin is captured and partitioned into small subsets representing specific regions. These subsets are then evaluated using a trained model called PGAs-Net, which performs binary classification on each small image. If the region is a Potential Grasp Area (PGA) that contains objects that can be approached and grasped, the point cloud of the area is rendered and transferred to the next step. In the second step, another network called O-PointNet is used to predict positive contact points and grasp widths from point cloud data of PGAs. This is used to reconstruct 6-DoF antipodal grasp poses. Finally, grasp selection is based on specific criteria to avoid collisions with the bin and surrounding objects. The entire process is illustrated in Fig. 2.

B. Step 1 - Finding Potential Grasp Areas (PGAs)

1) Inference of PGAs in Training Datasets Generation: In cluttered environments, it is apparent that not every object can be grasped from any position. Therefore, it is necessary to reduce the time taken to calculate the entire bin space. This

Fig. 4. Examples of 3D mesh models in training datasets. The objects are from many categories, such as bottles, toys, fruit, and instrument.

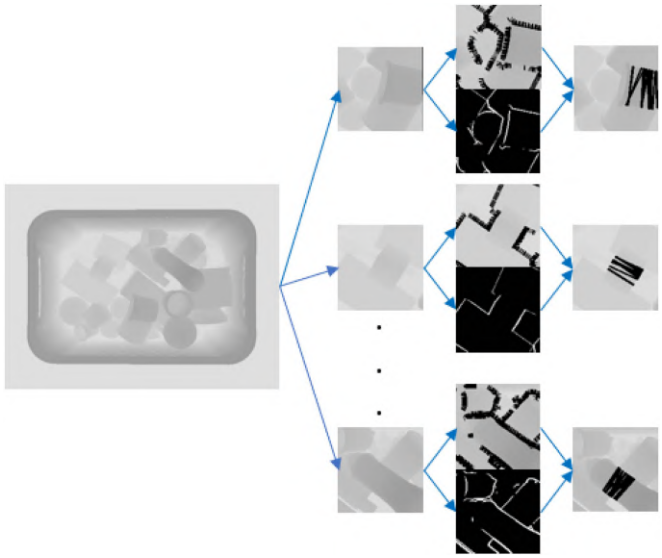

Fig. 5. Datasets generation process for Step 1: Finding PGAs. (Left to Right): Depth image, Subset, visualization of the contour of object and gradient orientation, visualization of antipodal points on the edge of object connected by black lines. From the gradient magnitude and orientation of depth images subset, antipodal points on the contour of object are calculated, which infers PGAs.

can be achieved by identifying PGA, which is defined as an area that includes at least one object that can be approached from two fingertips. As mentioned, the contact points of a parallel grasp often lie near the object's contour. Hence, we generate our datasets by calculating the gradient magnitude and orientation, which is a directional change in the intensity or color in an image. We set a gradient magnitude threshold to find the object's contour from the bin's gray image. Additionally, the gradient orientation is used to calculate the antipodal points on the edge of the object. The number of antipodal points determines whether the region is a PGA.

2) Dataset Generation: Based on our analysis of PGA, we automatically generate synthetic training datasets using the pybullet simulator. To begin with, we select 136 3D mesh models from approximately 1500 objects obtained from 3DNet [22] (Fig. 4). Objects that are transparent, duplicates, or do not fit the gripper's dimensions are eliminated. Once the objects are loaded into the bin in pybullet, a virtual camera with specific properties captures a cluttered scene from an up-down view. This process is repeated until the desired number of depth images is obtained. Each depth image is then divided into smaller images representing different areas with appropriate dimensions, which can display the biggest object in the bin. We label each small image positive (1) or negative (0) based on the gradient orientation on the contour of the objects. A small image is considered positive if it contains at least 5 antipodal point couples on the object's contour, and these points must be free for approach, meaning they are not near other objects. These areas are referred to as Potential Grasp Areas (PGAs) because they contain at least one object that can be grasped using antipodal grasping near the edge of the object. Negative labeling is applied to areas that have less than the 5 antipodal point couples threshold. The process is illustrated in Fig. 5.

3) Network and Training: We have developed a modified version of the Grasp Quality Convolutional Neural Network (GQ-CNN) [1] architecture called PGAs-Net for our work. The original GQ-CNN model has two inputs, the gripper depth and depth image. However, we modified this architecture by removing one of the inputs and using the entire backbone of the GQ-CNN to extract features from the depth image. The Dex-net research has shown that the network is proficient in learning rules related to edge and oriented gradients.

Our training dataset consists of 5000 depth images, each of which is divided into 60 smaller images, resulting in a total of 300,000 data points with approximately 30% positive samples. We adjusted some of the optimization parameters compared to Dex-net, such as training for 50 epochs, using a batch size of 64, maintaining an unchanged momentum term of 0.9, and eliminating an exponential decaying learning rate. We trained the model for approximately 5 hours on an NVIDIA GeForce 3090, achieving an accuracy of approximately 99%.

C. Step 2 - Detecting Antipodal Grasps in PGAs

1) Computing Bases for Contact Points in Training Datasets Generation: In order to successfully grasp the target object, it is necessary to identify the antipodal points on the object, which are a pair of points whose normal vectors are collinear and in opposite directions [5]. The grasp vector must lie within both friction cones at the two contact points to ensure that the object can be lifted. The two-finger grasping principle is illustrated in Fig. 3(b). Additionally, the contact points must also satisfy certain conditions to avoid collisions during operation. There must be free space around the contact points to allow for the gripper's approach, and the angle between the grasp vectors as well as the approaching vectors and the table must be within specific limitations, ensuring that the kinematics of the gripper and robot remains in a safe region. To summarize, the positive contact points on 3D CAD models of objects in cluttered scenes must satisfy the following conditions:

- Grasp vector $\mathbf{b}$ has to lie in both friction cones at two contact points for antipodal grasp
- The angle between grasp vector $\mathbf{b}$ and $Oz \in [60^0, 120^0]$

- Around contact points with radius 3 cm is free space for gripper's approach

2) Dataset Generation: Before delving into this section, it is important to distinguish between the two types of point clouds used for computing and training.

- Full point cloud $P = \{p_1, p_2, \ldots, p_n\}$ is used for computing possible contact point couples on the entire surface of the objects in the cluttered scene. This point cloud is converted from the 3D mesh models in the pybullet environment using the Open3D library.
- Single-view point cloud $C = \{c_1, c_2, \ldots, c_m\}$ is used for training and is captured from an up-down view camera. It represents one side of the full point cloud of the scene.

3) Calculation in Full Point Cloud: We continue using simulation environment to build training datasets for this step. We utilize the network trained in step 1 to predict the Potential Grasp Areas (PGAs) from the cluttered scene in pybullet. Then, we identify objects in the PGAs and convert these 3D mesh models to point clouds using the Open3D library. Each point in the cloud is calculated based on the antipodal principle and distances to other points in PGAs. If a point meets the requirements for suitable contact points discussed above, such as having free space for the gripper's approach and avoiding collision, its coordinate is saved for the next calculation.

4) Labeling for Each Point in Single-View Point Cloud: A single-view point cloud for training $C = \{c_1, c_2, \ldots, c_m\}$. We assign a point-wise grasp success:

$$\forall i = 1, \ldots, m \quad s_i = \begin{cases} 1 & min_j \parallel c_i - p_j \parallel_2 < r \\ 0 & otherwise \end{cases} \tag{5}$$

where r is the radius to recognize positive labels around saved suitable contact points $p_j \in P$ from the full point cloud. Other words, the points that are near p_j in $r = 5$ mm limitation are labeled 1, and grasp widths w are computed respectively. The whole dataset generation process is automatic and reaches high accuracy. We use *pyfastnoisesimd* function to add noise for simulated point clouds. The visualization of training datasets is shown in Fig. 6.

5) Network and Training: In recent years, PointNet [23] is known as a common network consuming directly point cloud for classification or segmentation applications. PointNet++ [24] is an upgraded version that has three main layers: sampling, grouping, and using PoinNet to acquire features from local regions in a point cloud. In addition to learning under the unordered attribute and transformation, the network can adapt to different densities in point cloud because of learning local features with increasing contextual scales. In our work, the network that is called O-PointNet has two heads with per-point outputs $s \in \mathbb{R}$, $o \in \mathbb{R}^{10}$. After training, the model takes point cloud of PGA as input and predict parallel grasp poses via multi-outputs.

The training datasets generated within one week for this step include approximately 30.000 point cloud scenes of PGAs, positive points in each scene occupy an average of about 3.5%. We do not generate more samples because our model's accuracy is no longer improving with the number of samples, which is shown in Fig. 7 (Right). As shown in Contact-GraspNet [4],

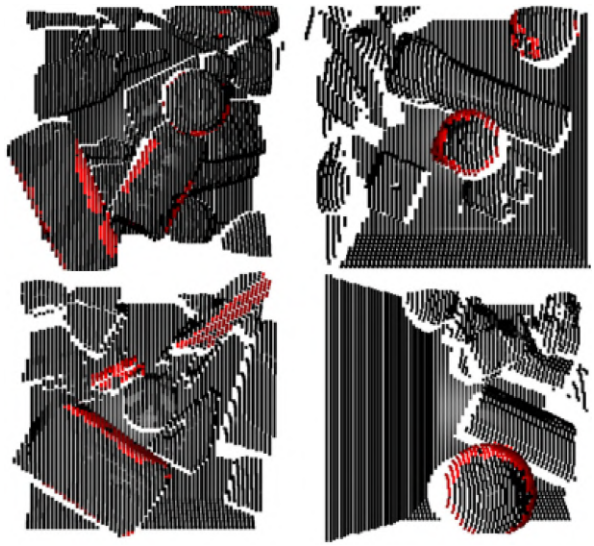

Fig. 6. Examples of synthetic training datasets in detecting positive points from point cloud. Red points represent possible contact points that are calculated automatically based on conditions to avoid collisions. These points tend to belong to objects that are above in clutter.

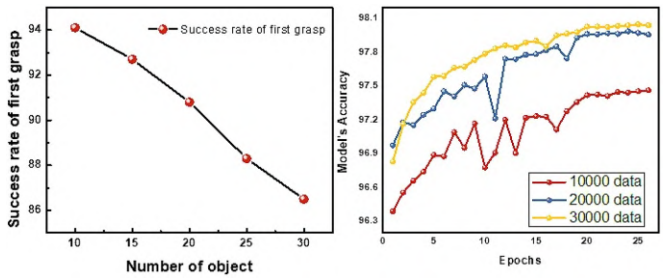

Fig. 7. (Left) The effect of number of objects on algorithms success rate; (Right) The effect of the size of synthesized datasets on the model's accuracy.

the grasp width $\hat{w}_i \in [0, w_{\max}]$ is divided into 10 equidistant ranges $\hat{o} \in \mathbb{R}^{10}$ to counteract data imbalance and the predicted results are center value in one of the ranges with the highest confidence. We use Adam optimizer with learning rate 0.001, betas 0.9–0.999, momentum 0.95, batch size 8 and training in 26 epochs. The training time is about 11 hours on NVIDIA GeForce 3090 card to reach roughly 98% model accuracy.

D. Grasp Selection

This is the last step that selects the best grasp in a candidate list. We choose a specific number of PGAs that have the highest confidence from the prediction results of PGAs-Net to render point cloud. As for the second step, we use a confidence threshold of 0.5 to filter positive points in point cloud from the output of O-PointNet. We filter good candidates that have the approaching vector in the safe region to eliminate collisions with the bin. In conclusion, there are two priority attributes of grasp for selecting the best candidate:

- Approaching vectors $\mathbf{a}$ need to be in the limited region. This is the limitation for the points that are intersections between approaching vectors and the bin top plane $z = h$, where h is the height of bin.

The condition:

$$\begin{cases} |x - x_c| \leq x_{limit} \\ |y - y_c| \leq y_{limit} \end{cases} \quad (6)$$

where x, y is coordinate of intersection between approaching vectors $\mathbf{a}$ and the bin top plane; x_c, y_c is coordinate of the bin center.

The x, y intersection coordinate is calculated as follows: We have the line contains approaching vector and via grasp center:

$$d : \frac{x - x_g}{x_a} = \frac{y - y_g}{y_a} = \frac{z - z_g}{z_a} \quad (7)$$

and the bin top plane: $(P) : z = h$

$$\Rightarrow x = \frac{x_a(h - z_g)}{z_a} + x_g; \quad y = \frac{y_a(h - z_g)}{z_a} + y_g \quad (8)$$

where x_g, y_g, z_g is coordinate of grasp center

- The height of grasp: To grasp an object successfully in highly cluttered scenes, the robot must first avoid obstacles to approach the target object before lifting it. In our approach, we prioritize grasp selection based on the height of the grasp. Through experiments, we have observed that most failed grasps occur due to collisions. Hence, a higher grasp is better because it reduces the likelihood of collisions with the bin or other obstacles.

Firstly, the criteria related to the approaching vector will be applied to eliminate candidates that are outside the limited region. Once this filtering is done, we select the grasp with the highest height for manipulation.

V. Experiments

A. Experiments With Simulated Point Cloud

In the task of picking objects from a heap, the number of objects in the bin directly impacts the performance of the algorithm. Therefore, we investigated the *success rate of first grasp* by creating random bin picking scenes with varying *numbers of objects* in a 0.24 m^2 tray. The *success rate of first grasp* is the ratio of correctly predicted objects to the total number of predicted objects. The result is surveyed over 100 cluttered scenes and depicted in Fig. 7 (Left). An object is considered accurately predicted if it satisfies two conditions:

- The parallel grasp pose is predicted correctly on the object.
- The object is free to approach, which does not cause a strong collision with surrounding environment.

The graph shows the *success rate of first grasp* decreases if the *number of objects* increases. However, the model's prediction remains stable at around 90%. Based on our observations, the prediction of PGAs is not affected significantly by the number of object, but the overlap between many objects can lead to more errors in prediction from point cloud.

B. Real Robotic Experiments

1) Predefined Parameters:

a) The size of local regions (PGAs): This dimension should be appropriate for the maximum grasp width of the gripper or to cover the largest objects in the bin before resizing for input into CNNs.

Fig. 8. Our bin picking experiment system: depth camera L515, Universal Robots UR10, Parallel OnRobot gripper, and 2 bins for picking and placing.

b) The number of PGAs for computing: In theory, we can choose one PGA with the highest confidence for computing parallel grasp candidates. However, PGAs-Net sometimes can have wrong predictions leading to confusion. On the other hand, the number of PGAs should not be too high due to the computational burden. Therefore, we use the two PGAs with the highest confidence for calculations in the next steps.

c) The safe region for approaching vector: We calculate the coordinate of intersection between the approaching vectors of grasp and the upper plane of the bin (IV-D). The x_{limit} and y_{limit} depend on the specific dimensions of the bin.

d) The threshold for filter real point cloud: One of the typical errors of camera L515 in cloud is elongated points at the edge of objects, which can lead to non-existent contact points. To address this issue, we apply outlier filtering as described in [25] to eliminate noise at the edge by calculating local densities in point cloud.

2) Setup: The whole bin picking system consists of a camera L515, a robot arm UR10, a parallel OnRobot gripper and 2 bins for picking and placing. In our experiment, we choose 10 objects with arbitrary shapes and dimensions that fit with the gripper. The selection criteria for these objects were to emphasize the importance of antipodal grasp. Therefore, these objects have shapes not favorable for suction cup grasp used widely in the industry environment. They were arranged randomly in the bin to create a cluttered scene. A top-down camera L515 was mounted at about 1 m high respect to the bin for capturing RGB and depth images. The algorithm was run on a PC Intel Core i7, NVIDIA GeForce RTX 3060. Our results were surveyed on 20 different cluttered scenes. The experiment system and objects are shown in Fig. 8.

3) Evaluation Metrics: During the operation of bin picking task, we survey *Success rate*, *Collision-free* and *Completion* on

TABLE I
COMPARISON OF REAL ROBOT EXPERIMENTS IN HIGHLY CLUTTERED SCENES

	without Bin			with Bin			Computing Time
	Success rate	Collision-free	Completion	Success rate	Collision-free	Completion	(sec)
PointNetGPD [20]	77.77 %	41.17 %	97.5 %	None	None	None	∼17.69
Contact-GraspNet	None	None	None	None	None	None	∼3.05
S4G [28]	77.1 %	53.32 %	92.5 %	None	None	None	∼5.8
Dex-Net 4.0	86.5 %	74.3 %	100 %	76.44 %	66.67 %	89.58 %	**∼1.37**
FC-GQ-CNN	87.42 %	82.78 %	100 %	79.91 %	75.34 %	90.62 %	∼3.01
Our method	**92.31** %	**87.17** %	100 %	**88** %	**83.33** %	**92** %	∼2.2

'None': There is no information or the method is not suitable for testing in highly cluttered scenes of bin-picking task.

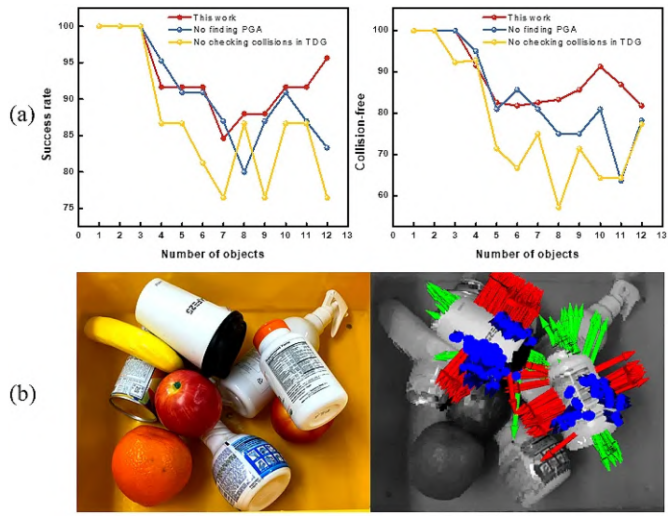

Fig. 9. (a) Survey the algorithm's performance in two ablation studies: no finding PGA and no checking collisions in training datasets generation (TDG); (b) The predicting results in the real experiment: red, green, blue axis represent parallel grasp pose, red axis is grasp vector, blue axis is approaching vector.

several cluttered scenes. The *Success rate* is the percentage of successful grasp, *Collision-free* is the percentage of collision-free grasp, and *Completion* is the fraction of the number of objects transported to another bin and the total number. The termination criteria is that all objects in the bin are emptied or the algorithm can not predict any more possible grasp poses.

4) Results: An example for predicting results on the real point cloud is shown in Fig. 9(b). Red, green, blue axis depicts grasp pose of gripper, where red axis is grasp vector and blue axis is approaching vector. From the comparison results presented in Table I, our method has significant progress for all of the evaluation metrics compared with top-down grasp method Dex-net 4.0 [29] and FC-GQ-CNN [30]. The superiority is also shown when compared with other methods using point cloud data, such as PointNetGPD [20] and S4G [28]. Besides, because of using step 1 for finding PGAs, our method shows quick computing time. In some manipulation cases with bin, all objects are not cleared completely because of the noise of real point cloud or target objects in difficult positions such as corner of the bin.

5) Ablation Study: To better illustrate the effectiveness of our method, we conducted two ablation studies, as depicted in Fig. 9(a). These studies focused on evaluating the impact of eliminating step 1 in finding PGA (blue line) and omitting collision checks during training dataset generation (TDG) on both the *Success rate* and *Collision-free* metrics. The complete approach outperforms the ablation studies. This clearly demonstrates that the success of our method stems from the inclusion of step 1, which reduces incorrect predictions across entire scenes, as well as the meticulous collision checking during dataset labeling.

6) Discussion: In general, the weakness of the principally cited and compared methods is their sampling process, which informs the candidate set of grasps tested or evaluated for grasp selection. For example, Dex-net 2.0 and Dex-net 4.0 rely on a random sampling process and use the cross-entropy method to select the grasps with the highest estimated quality. Therefore, the model's performance depends on the number of samples fed to GQ-CNN. On the other hand, Contact-GraspNet uses point cloud segmentation and prediction on entire scenes, which is also slow and prone to false positive/false negative issues. To address these limitations, our method employs PGAs-Net, which uses detection with a deep network as an alternative "sampling process." This not only reduces incorrect predictions on entire scenes but also leads to reduced computing time.

Moreover, the primary difference between our training dataset generation for step 2 and point cloud-based other methods such as Contact-GraspNet, S4G, and PointNetGPD is that, instead of simulating grasp trials on single objects to label contact points, we perform detailed calculations regarding collision and the antipodal principle for each point in the point clouds. This approach ensures that our data labeling is more detailed and consistent, making it adaptable to varying densities, ranging from individual objects to highly cluttered scenes. This has been demonstrated by the superior performance of our method in terms of both *Success rate* and *Collision-free* metrics. Although FC-GQ-CNN [30] achieves a *Completion* rate that is only 1.38% lower than our method, the use of top-down approaches to deal with many situations makes it more susceptible to collisions and unsuccessful grasps during execution. This highlights the benefits of our top-down biased 6-DoF approach, which generates more flexible and dexterous grasp poses to adapt to different object poses and avoid collisions.

It is worth noting that PointNetGPD, Contact-GraspNet, and S4G are not suitable for testing with a bin since these methods are full 6-DoF approaches that do not sufficiently consider collision conditions in highly cluttered scenes and in a bin. Some reference data is available at https://drive.google.com/drive/folders/1ybsDmErNLz7YYPvYiAk2_W4qtLh08ylK

7) Difficult Cases: In terms of predicting PGAs, PGAs-Net utilizes the edge and gradient orientation of objects in 2.5D depth images. This means that the network's predictions are limited to

potential objects viewed from an up-down perspective, without considering other angles. As a result, objects located near the edge of the bin may be overlooked, even if they are approachable from a 6-DoF grasp. Additionally, the single-view system lacks comprehensive information about an object's geometry, particularly the vertical edges of box-shaped objects. This can result in difficulties detecting contact points, leading to failures in the grasping process.

VI. Conclusion and Future Work

In this letter, we present an end-to-end method for generating the 6-DoF antipodal grasp pose in bin picking task, which are challenging due to cluttered scenes, overlapping objects, and potential collisions during operation. The grasp pose is recognized by the outputs of model, which include the first contact points, grasp widths, normal vectors, and approaching vectors calculated within defined constraints to avoid collisions. The algorithm does not depend on the segmentation, hence it restricts the effect of real point cloud's noise. Our solution has been shown to be effective and adaptable to a variety of objects in a heap.

However, using a single-view system limits our ability to render point clouds from multiple sides of objects, probably leading to missed potential grasps from other sides. Additionally, the two-fingered gripper may not be suitable for grasping all shapes or objects in certain positions, such as cones or objects in corners. Therefore, we are researching a gripper that combines suction cups and parallel grasps, offering three modes for grasping: only suction, only parallel grasp, and a combination of both. It is essential to reduce real point cloud noise and investigate sim-to-real domain adaptation to improve our algorithm's performance.

References

[1] J. Mahler et al., "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot.: Sci. Syst.* 2017, doi: 10.15607/RSS.2017.XIII.058.

[2] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9626–9633.

[3] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multi-object, multi-grasp detection," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.

[4] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13438–13444.

[5] I-M. Chen and J. W. Burdick, "Finding antipodal point grasps on irregularly shaped objects," *IEEE Trans. Robot. Automat.*, vol. 9, no. 4, pp. 507–512, Aug. 1993.

[6] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia Proc.*, 2000, pp. 348–353.

[7] Y. Sun, J. Falco, M. A. Roa, and B. Calli, "Research challenges and progress in robotic grasping and manipulation competitions," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 874–881, Apr. 2022.

[8] A. Lobbezoo, Y. Qian, and H.-J. Kwon, "Reinforcement learning for pick and place operations in robotics: A survey," *Robotics*, vol. 10, 2021, Art. no. 105.

[9] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp 515–524.

[10] M. Gualtieri and R. Platt, "Learning 6-DoF grasping and pick-place using attention focus," in *Proc. 2nd Conf. Robot Learn.*, 2018, pp 477–486.

[11] Y. Xiao, S. Katt, A. t. Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8241–8247, doi: 10.1109/ICRA.2019.8793494.

[12] B. Wu, I. Akinola, and P. K. Allen, "Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1789–1796.

[13] A. Mallick, A. P. del Pobil, and E. Cervera, "Deep learning based object recognition for robot picking task," in *Proc. 12th Int. Conf. Ubiquitous Inf. Manage. Commun.*2018, pp. 1–9.

[14] H.-J. Jo, C.-H. Min, and J.-B. Song, "Bin picking system using object recognition based on automated synthetic dataset generation," in *Proc. IEEE 15th Int. Conf. Ubiquitous Robots*, 2018, pp. 886–890.

[15] X. Li et al., "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3961–3968, Apr. 2022, doi: 10.1109/LRA.2022.3149026.

[16] A. Zeng et al., "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge," in *Proc. IEEE Int. Conf. Robot. Automat.*2017, pp. 1386–1383.

[17] S. Araia, Z. Fenga, F. Tokudaa, A. S. Z. Purnomoa, Y. Xua, and K. Kosuge, "Deep learning-based fast grasp planning for robotic bin-picking by small data set without GPU," 2021, doi: 10.36227/techrxiv.14384864.v1.

[18] W. Liu et al., "Deep learning for picking point detection in dense cluster," in *Proc. IEEE 11th Asian Control Conf. Gold Coast Conv.*, 2017, pp. 1644–1649.

[19] A. Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.*, vol. 41, no. 7, pp 690–705, 2022, doi: 10.1177/0278364919868017.

[20] H. Liang et al., "PointNetGPD: Detecting grasp configurations from point sets," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3629–3635.

[21] A. ten Pas, M. K. Gualtieri, and R. S. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, pp. 1455–1473, 2017.

[22] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3DNet: Large-scale object class recognition from CAD models," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 5384–5391, doi: 10.1109/ICRA.2012.6225116.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[25] R. B. Rusu, "Dissertation: Semantic 3D object maps for everyday manipulation in human living environments," *Künstl Intell*, vol. 24, pp. 345–348, 2010, doi: 10.1007/s13218-010-0059-6.

[26] H. Tachikake and W. Watanabe, "A learning-based robotic bin-picking with flexibly customizable grasping conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9040–9047, doi: 10.1109/IROS45743.2020.9340904.

[27] C. Graf et al., "Learning dense visual descriptors using image augmentations for robot manipulation tasks," in *Proc. Conf. Robot Learn.*, 2023, pp. 871–880.

[28] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4G: Amodal single-view single-shot SE(3) grasp detection in cluttered scenes," in *Proc. Conf. Robot Learn.*, 2020, pp. 53–65.

[29] J. Mahler et al., "Learning ambidextrous robot grasping policies," *Sci. Robot*, vol. 4, 2019, Art. no. eaau4984, doi: 10.1126/scirobotics.aau4984.

[30] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1357–1364, Apr. 2019, doi: 10.1109/LRA.2019.2895878.