2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561868

GPR: Grasp Pose Refinement Network for Cluttered Scenes

Wei Wei[1, 2, *], Yongkang Luo[1, *], Fuyu Li[1,2], Guangyun Xu[1,2], Jun Zhong[1], Wanyi Li[1], Peng Wang[1, 2, 3,✉]

Abstract— Object grasping in cluttered scenes is a widely investigated field of robot manipulation. Most of the current works focus on estimating grasp pose from point clouds based on an efficient single-shot grasp detection network. However, due to the lack of geometry awareness of the local grasping area, it may cause severe collisions and unstable grasp configurations. In this paper, we propose a two-stage grasp pose refinement network which detects grasps globally while fine-tuning low-quality grasps and filtering noisy grasps locally. Furthermore, we extend the 6-DoF grasp with an extra dimension as grasp width which is critical for collisionless grasping in cluttered scenes. It takes a single-view point cloud as input and predicts dense and precise grasp configurations. To enhance the generalization ability, we build a synthetic single-object grasp dataset including 150 commodities of various shapes, and a complex multi-object cluttered scene dataset including 100k point clouds with robust, dense grasp poses and mask annotations. Experiments conducted on Yumi IRB-1400 Robot demonstrate that the model trained on our dataset performs well in real environments and outperforms previous methods by a large margin.

I. INTRODUCTION

Robotic grasping is a fundamental problem in the robotics community and has many applications in industry and house-holding service. It has shown promising results in industrial applications, especially for grasping under structured environments, such as automated bin-picking [1]. However, it remains an open problem due to the variety of objects in complex scenarios. Objects have different 3D shapes, and their shapes and appearances are affected by lighting conditions, clutter and occlusions between each other.

Traditionally, the problem of object grasping in cluttered scenes is tackled by estimating 6D object pose [2], [3], [4] and selecting grasp from the grasp database. As a result, these approaches are not applicable to unseen objects. In order to generalize to unseen objects, many recent works [5], [6], [7], [8], [9] conduct grasp pose detection as rectangle detection in 2D space with CNNs, and their models perform well on novel objects. However, planar grasping with 3/4 DoF (degree of freedom) inevitably results in inflexibility, since the gripper is forced to approach objects vertically. Besides the DoF constraint, these works utilize the 2D images as input, which ignore gripper contact with the object in 3D space. Some recent works suggest that 3D geometry structure is highly relevant to grasp quality [10], [11]. PointNetGPD [11] evaluates grasp quality in 3D space with exhaustive searching in point clouds. S4G [12] and PointNet++Grasping [13] propose efficient single-shot grasp pose detection network architectures, while the results may be noisy and suffer collisions with surrounding objects. The main reason can be attributed to: 1) lack of shape awareness of the local contextual geometry of the gripper closing area; 2) grasping with max opening width is more likely to cause collisions with surrounding objects in dense clutter.

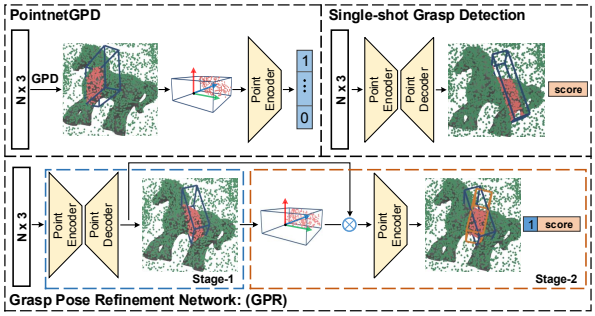


Fig. 1: Comparison with state-of-the-art methods. Instead of exhaustive searching and evaluating possible grasp candidates in the point cloud, our method generates possible grasp candidates efficiently in stage-1 as the single-shot grasp detection pipeline. Moreover, our model refines low-quality and classifies noisy grasp candidates in stage-2 base on discriminative feature representation of the local grasping area.

Considering the above problems, we propose to detect grasp poses globally and refine them locally. Single-shot feature representation helps to avoid exhaustive searching in the point cloud, while it is not able to learn discriminative local feature representation without further inspection of the local grasping area. For addressing the limitation, we turn to focus on the local grasping area and design a two-stage grasp pose refinement network (GPR) for estimating stable and collisionless grasps from point clouds. As illustrated in Fig.1, our model predicts coarse and noisy grasp proposals in the first stage. Then, points inside the proposals are cropped out and transformed into local gripper coordinate in the second stage. Finally, these points are used to encode discriminative local

This work was supported in part by the National Natural Science Foundation of China under Grants (91748131, 62006229 and 61771471), and in part by the Strategic Priority Research Program of Chinese Academy of Science under Grant XDB32050100.

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China.
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China
* Authors contributed equally.
✉ Corresponding author: peng_wang@ia.ac.cn

feature representation for grasp proposals refinement and classification. Remarkably, our model takes a single-view point cloud as input and extends the 6-DoF grasp with an extra dimension as grasp width, which adjust gripper opening width and avoid unnecessary collisions. Furthermore, our two-stage network is trained in an end-to-end fashion.

For most data-driven methods, it is common to boost generalization performance with a large-scale dataset. However, manually annotated 6D grasps can be time-consuming [14]. Most current works generate grasp annotations based on traditional analysis methods [15], [16] or physics simulators [17], [18], [19]. In [11], [8], researchers had built datasets for individual objects, while ignoring multi-objects in cluttered scenes. [12], [13] propose to generate grasps in cluttered scenes. However, almost all of the object models come from the YCB object dataset [20] may lead to insufficient shape coverage. We collect 150 objects with various shapes and build large-scale synthetic datasets for both individual objects and objects in dense clutter. Experiment results show that the model trained on our dataset performs well in the real robot platform and gets promising results.

In summary, our primary contributions are:

- An end-to-end grasp pose refinement network for high-quality grasp pose prediction in cluttered scenes that detects globally while refines locally.
- Extend 6-DoF grasp with grasp width as a 7-DoF grasp for improvement of dexterous and collisionless grasping in dense clutter.
- A densely annotated synthetic single-object grasp dataset including 150 object models, and a large scale cluttered multi-object dataset with 100k point clouds with detailed annotations. We will release the dataset.

II. RELATED WORK

Deep Learning based Grasp Configuration Detection. [21] gives a thorough survey of robotic grasping based on deep learning. Given the object model and grasp annotations, [22], [2] tackle this problem as template matching, and the 6-DoF pose retrieving problem. While template matching methods show low generalization ability for unknown objects. [10] designs several projection features as the input of a CNN-based grasp quality evaluation model. [11] replaces input with direct irregular point cloud and train PointNet [23] for grasp classification. These methods rely on detailed local geometry for constructing both collision-free and force-closure grasps. [5], [6], [7], [24], [25] tackle this problem as grasp rectangle detection in 2D images from a single object to multi-object scenarios. While these methods just perform 3/4-DoF grasp. [12] proposes a single-shot grasp proposal framework to regress 6-DoF grasp configurations from point cloud directly. [13] follows a similar setting, while it generates grasp based on the assumption that the approaching direction of a grasp is along the surface normal of the objects. Worth noting that [8] collects numerous object models for GQ-CNN training and obtains state-of-the-art performance. Of all the above methods, GPD can also estimate grasp width with geometry prior. However, it relies on multi-view point clouds input. In this paper, we revisit grasp width as a critical element for grasp configuration and our model can directly predict high accuracy grasp width.

Grasping Dataset Synthesis. [26], [6], [7] annotate rectangle representation for grasping detection in images manually. [27], [28] collect annotations with a real robot. While an enormous amount of annotated data is needed for supervised deep learning, therefore manually grasp configuration annotating is unpractical due to time-consuming. Given an object with a gripper model and environment constraints, we are able to synthesize grasp configurations in two kinds of ways generally. One is based on analytic methods [29], which derive from force-closure [15] and Ferrari Canny metric [16]. [30] gives a detailed survey of these methods. [12], [13], [11], [31], [8], [14] generate dataset based on this way. Another is based on physical simulators, such as [18], [19], these simulators perform better than analytic methods in terms of force contacts. [32], [33], [34], [35], [36] generate their dataset using simulated environment.

Deep Learning on Point Cloud Data. PointNet [23] and PointNet++ [37] are two novel frameworks to directly extract feature representation from point cloud data. Many methods [38], [39], [40], [41], [42], [43] extend these frameworks to point cloud classification, detection and segmentation. In this paper, we utilize PointNet++ as the backbone.

III. PROBLEM STATEMENT

In this work, we focus on the problem of planning a robust two-fingered parallel-jaw grasping based on point clouds. Our two-stage refinement network takes the whole cluttered scene as input and outputs dense grasp poses with high quality and robustness. Some of the key definitions are introduced here:

Object States: Let $\boldsymbol{x}_i = (\mathcal{O}_i, \boldsymbol{T}_i, \boldsymbol{\gamma})$ describes state of an object in a grasp scene, where $\mathcal{O}_i$ specifies the surface model, mass and centroid properties of object i, $\boldsymbol{T}_i$ denotes 6D object pose, $\boldsymbol{\gamma}$ denotes friction coefficient.

Point Clouds: Let $\boldsymbol{y}_k \in \mathbb{R}^{N\times 3}$ represents the point cloud of the k^{th} scene captured by the depth camera.

Grasps: Let $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots \boldsymbol{g}_m]$ denotes grasp configurations in a cluttered scene. Each grasp configuration is defined as $\boldsymbol{g}_i = (\boldsymbol{o}, \boldsymbol{n}, \boldsymbol{r}, \omega, c_1, c_2)$, where $\boldsymbol{o} = (o_x, o_y, o_z)$ represents the origin lies at the middle of the line segment connecting two finger tips, $\boldsymbol{n} = (n_x, n_y, n_z)$ and $\boldsymbol{r} = (r_x, r_y, r_z)$ denote approach direction and closing direction of a grasp, w describes grasp width, c_1 and c_2 denote contact points.

Grasp Metric: We adopt the widely used Ferrari Canny metric [16] for labelling grasp quality.

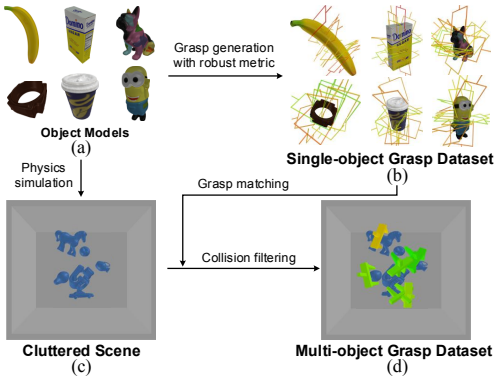


Fig. 2: Overview of our datasets generation procedure. (a) Example single object models. (b) Example single object grasps with label $\boldsymbol{Q}(\boldsymbol{g})$. For each object, 15 grasps are sampled for visualization. The colors from red to green represent $\boldsymbol{Q}(\boldsymbol{g})$ from low to high. (c) Illustration of a cluttered scene. (d) Example grasps in a cluttered scene.

IV. DATASET GENERATION

In this section, we introduce our dataset generation method for grasp poses annotation for both individual objects and objects in dense clusters. The overall pipeline is illustrated in Fig.2. We take the following procedure to obtain dense grasp annotations. Firstly, we label single-object grasp annotations and then match grasp annotations into cluttered scenes according to the 6D object pose. Finally, we apply collision filtering for all the grasp configurations.

A. Single-object grasp dataset Generation

For single-object grasp dataset generation, we collect 150 objects of various shapes and categories. Half of these objects come from the BOP-Challenge dataset and YCB-Video dataset [20], others are collected from the internet.

Given a specific object model $\mathcal{O}$, the target is to generate dense grasp annotations including grasp configuration g and corresponding grasp metric mentioned above. First, $\boldsymbol{N}$ candidate contact points $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots \boldsymbol{p}_N$ are sampled on object surface model with outward normals $\boldsymbol{n}$ calculated. Based on force-closure principle, $\boldsymbol{k}$ antipodal grasp directions are then sampled inside the friction cone of point $\boldsymbol{p}_i$. Each antipodal grasp candidate $\boldsymbol{g}_i$ will be classified as a positive grasp candidate $\boldsymbol{g}_i^T$, if satisfies rules as follows: 1) At least one antipodal contact point $\boldsymbol{c}_2$ is found on object backward surface; 2) Force-closure property. Otherwise, antipodal grasp candidate is classified as negative grasp $\boldsymbol{g}_i^F$.

Second, for each positive antipodal grasp candidate $\boldsymbol{g}_i^T$ of a contact point $\boldsymbol{p}_i^T$, collision check is applied between gripper and object. Those grasp candidates failed in collision check will be classified as negative grasps $\boldsymbol{g}_i^F$. If no positive antipodal grasp candidate is reserved, the corresponding sampled point is classified as a negative point $\boldsymbol{p}_i^F$, which means an unsuitable contact point.

Third, the grasp metric for each reserved positive grasp candidates is calculated by Ferrari Canny metric as $\boldsymbol{Q}(\boldsymbol{g})$.

Finally, we apply Non-maximum Suppression algorithm (NMS) for pruning redundant grasps. Distance between two sampled grasp g_1 and g_2 is calculated by following equation:

$$\begin{aligned}\boldsymbol{D}(g_1, g_2) = \beta_1 \cdot ||((c_1 + c_2)/2|g_1) - ((c_1 + c_2)/2|g_2)||_2 \\ + \beta_2 \cdot arccos(|(\boldsymbol{\gamma}|g_1) \cdot (\boldsymbol{\gamma}|g_2)|)/\pi \\ + \beta_3 \cdot arccos((\boldsymbol{n}|g_1) \cdot (\boldsymbol{n}|g_2))/\pi.\end{aligned} \tag{1}$$

$\boldsymbol{n}, \boldsymbol{\gamma}$ are set to 16384, 0.3. $\boldsymbol{\beta_1}, \boldsymbol{\beta_2}$, and $\boldsymbol{\beta_3}$ is set to 1, 0.03, and 0.03 in our experiments. For all the objects $\mathcal{O}_1, \mathcal{O}_2, \cdots \mathcal{O}_n$, the output annotations are denoted as $\{\boldsymbol{g}_i, \boldsymbol{Q}(\boldsymbol{g}_i) \mid \mathcal{O}_1, \mathcal{O}_2, \cdots \mathcal{O}_n\}$. Examples of our single-object grasp dataset are shown in Fig.2 (b).

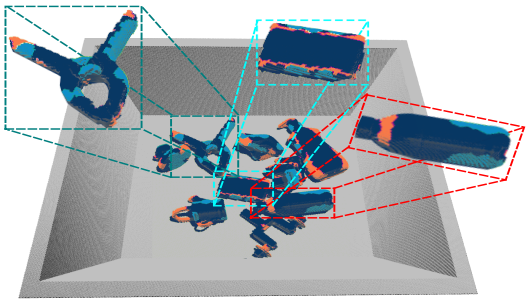


Fig. 3: An example shows points mask label $M(p)$ in our multi-object grasp dataset. Points in light blue denote positive grasp contact point. Points in dark blue denote negative contact point due to collision. Points in orange denote unsuitable contact points on foreground objects.

B. Multi-object Grasp Dataset Generation

To simulate densely cluttered scenes for the multi-object grasp dataset, we adopt the following procedures using [18]:

First, m objects are randomly sampled, then these sampled objects are initializing with random poses, and falling into a static bin successively in the simulator, as shown in Fig.2(c).

Then, the 6D object pose will be recorded after all sampled objects falling into the bin and reaching stable states. Each unsuitable grasp point p_i^F for each object $\mathcal{O}_i$ will be added into negative point set p_{neg}. Then we apply collision check for each grasp g_j of each object $\mathcal{O}_i$ obtained by single-object grasp generation. If no collision occurs, contact points c_1, c_2 of grasp g_i will be added into positive grasp contact points set p_{pos}, and the corresponding grasp annotation will be added into positive grasp set g_{pos}. Otherwise, the point will be added into negative grasp contact points set p_{neg}.

Point cloud y_k within the bin is cropped for generating points label and mask which is defined as follows:

$$\begin{aligned} L(p_i) &= [\boldsymbol{n}, \boldsymbol{\gamma}, \omega, Q(g_i)], \\ M(p_i) &= [\mathbb{I}(Q(g_i))], \\ \mathbb{I}(Q(g_i)) &= \begin{cases} 1 & \text{if } Q(g_i) > 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

Where $\mathbb{I}$ denotes Indicator function for generating points mask. For each point $p_i \in p_{pos}$, a KD-Tree search is applied to find the nearby points $p_i^R = [p_{i_1}, p_{i_2}, \cdots p_{i_k} | \boldsymbol{p}_i, y_k, R]$ among y_k with query radius R.

Moreover, each point in p_i^R will be broadcast with the same label $L(p_i)$ and mask $M(p_i)$. Finally, each point will only reserve the corresponding label and mask with the highest grasp score. For each point $p_i \in p_{neg}$, the similar process will be done. An Example is shown in Fig.3.

V. GRASP POSE REFINEMENT NETWORK

In this section, we present our proposed two-stage grasp pose refinement network (GPR) for grasp pose detection in cluttered scenes. The overall structure is illustrated in Fig.4.

A. Grasp Proposal Generation

Existing 6-DoF grasp pose detection methods could be classified into one-stage and two-stage methods. One-stage methods[24], [25], [31], [8], [12], [13] are generally faster but directly predict grasp pose without local geometry awareness. Two-stage methods[5], [6], [7] mostly depend on anchor mechanism[44] developed on 2D object detection, which generate proposals firstly and then refine the proposals and confidences in the second stage. However, directly applying anchor mechanism for predicting grasp pose in 3D space is non-trivial due to the huge search space and irregular format of the point cloud.

Therefore, we propose to directly estimate grasp pose in a bottom-up manner to avoid exhaustive searching in 3D space with 3D rotation inspired by [12], [13]. We predict mask and coarse 7-DoF grasp proposal for each point in the scene, as shown in stage-1 sub-network of Fig.4.

Feature representations and segmentation. We design the backbone network based on the PointNet++ [37], which is a robust learning model for dealing with sparse point cloud and non-uniform point density. We utilize the PointNet++ network with multi-scale grouping strategy as the backbone.

Given the point-wise feature encoded by the backbone network, we append two head ahead to our backbone: one segmentation head for predicting grasp contact points mask, and one grasp pose regression head for generating 7-DoF grasp proposals. We utilize focal loss [45] to handle the severe imbalance problem for grasp contacts segmentation, as shown in Fig.4.

Bin-based grasp pose regression. It is difficult to regress 7-DoF grasp configuration directly, which has been proved in previous literature [46], [42], [43]. Therefore, we develop bin-based regression method similar as [42]. Specifically, a 7-DoF grasp is represented as $\boldsymbol{g} = (\boldsymbol{o}, \boldsymbol{n}, \boldsymbol{r}, \omega)$, where $\boldsymbol{o} = (x, y, z)$ denotes the grasp center, $\boldsymbol{n}$ and $\boldsymbol{r}$ denote approach and closing directions of the gripper, ω denotes gripper opening width. Gripper direction regression is converted to angle prediction, as show in Fig.5. For angle prediction, gripper approach vector is denoted by $\theta_1 \in [0, 2\pi]$ and $\theta_2 \in [0, \pi/2]$ jointly, while finger closing direction is projected onto X-Y plane, and denoted by $\theta_3 \in [-\pi/2, \pi/2]$.

We divide a target angle of point p, e.g. θ_1^p, into n bins with uniform angle δ_{θ_1}, and calculate the bin classification target $\text{bin}_{\theta_1}^p$ and residual regression target $\text{res}_{\theta_1}^p$ within the classified bin. The angle loss for θ_1, θ_2 and θ_3 consists of two terms, one term for bin classification and another for residual regression within the classified bin. The target angle could be formulated as follows:

$$\begin{aligned} \underset{\theta\in\{\theta_{1,2,3}\}}{\text{bin}_\theta^p} &= \left\lfloor \frac{\theta^p - \theta_s}{\delta_\theta} \right\rfloor, \\ \underset{\theta\in\{\theta_{1,2,3}\}}{\text{res}_\theta^p} &= \frac{1}{\delta_\theta}\left(\theta^p - \theta_s - \left(\text{bin}_\theta^p \cdot \delta_\theta + \frac{\delta_\theta}{2}\right)\right). \end{aligned} \tag{3}$$

Where θ^p $(\theta \in \{\theta_1, \theta_2, \theta_3\})$ is the target grasp angle of a specific grasp contact point p, θ_s denote the starting angle, bin_θ^p is the ground-truth bin assignment, res_θ^p is the residual value for further angle regression within the assigned bin, and δ_θ is the unit bin angle of θ for normalization.

For grasp center and grasp width prediction, we adopt the following formulation:

$$\begin{aligned} \underset{u\in\{x,y,z,\omega\}}{\text{bin}_u^p} &= \left\lfloor \frac{u^p - u^{p_c} + \mathcal{S}_u}{d_u} \right\rfloor, \\ \underset{u\in\{x,y,z,\omega\}}{\text{res}_u^p} &= \frac{1}{d_u}\left(u^p - u^{p_c} + \mathcal{S}_u - \left(\text{bin}_u^p \cdot d_u + \frac{d_u}{2}\right)\right). \end{aligned} \tag{4}$$

Where (x^p, y^p, z^p) is the coordinates of an interest grasp contact point, ω^p is the grasp width, $(x^{p_c}, y^{p_c}, z^{p_c})$ and ω^{p_c} is the grasp center coordinates and grasp width of its corresponding grasp configuration. The bin_u^p and res_u^p $(u \in \{x, y, z, \omega\})$ are ground-truth bin assignment and residual location within the assigned bin, and d_u is the bin length for normalization. $\mathcal{S}_u$ denotes the corresponding search range.

The overall loss of grasp proposal generation sub-network could be formulated as follows:

$$\begin{aligned} \mathcal{L}_{\text{grasp}}^p = &\sum_{u\in\{x,y,z,\omega,\theta_{1,2,3}\}} (\mathcal{F}_{\text{cls}}(\widehat{\text{bin}}_u^p, \text{bin}_u^p) \\ &+ \mathcal{F}_{\text{reg}}(\widehat{\text{res}}_u^p, \text{res}_u^p)), \\ \mathcal{L}_{\text{stage-1}} = &\frac{1}{N_{\text{pos}}} \sum_{p\in\text{pos}} \mathcal{L}_{\text{grasp}}^p + \sum \mathcal{L}_{\text{focal}}^p(y_t). \end{aligned} \tag{5}$$

The loss $\mathcal{L}_{\text{stage-1}}$ includes two terms, $\mathcal{L}_{\text{grasp}}$ for grasp poses prediction and $\mathcal{L}_{\text{focal}}$ for grasp contact points segmentation. Where N_{pos} is the number of positive grasp contact points, y_t is the probability of point p as a positive grasp contact point. Where $\widehat{\text{bin}}_u^p$ and $\widehat{\text{res}}_u^p$ are the predicted bin assignment and residual of point p, bin_u^p and res_u^p are corresponding ground-truth. $\mathcal{F}_{\text{cls}}$ denotes the classification loss of bin assignment, and $\mathcal{F}_{\text{reg}}$ denotes regression loss for residual prediction.

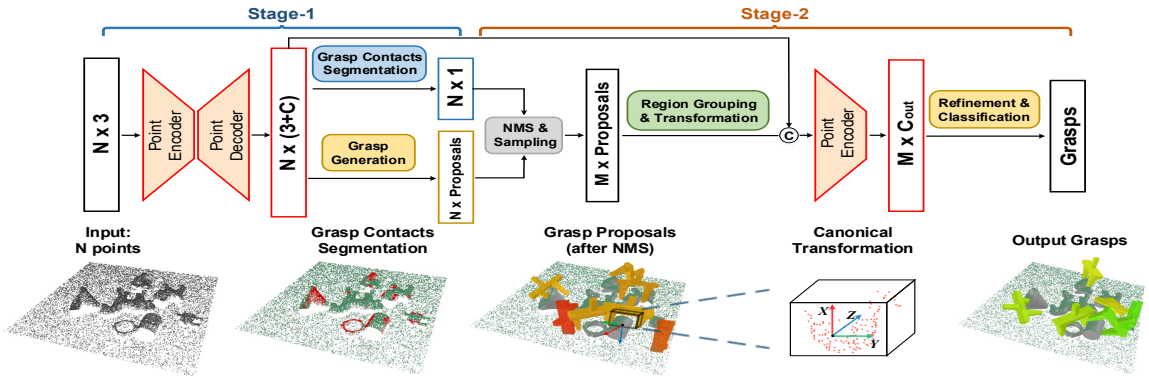
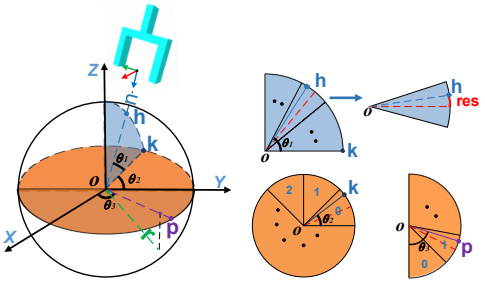


Fig. 4: Overview of our GPR network for grasp pose detection and refinement in point cloud. Stage-1 for generating 7-DoF grasp proposals. Stage-2 for refining grasp proposals with further geometry awareness of local grasping area.



(a) (b)

Fig. 5: An illustration of bin-based angle regression. (a) The grasp approach vector n is denoted by azimuth angle θ_1 and elevation angle θ_2, closing vector r is projected to X-Y plane and denoted by azimuth angle θ_3. (b) Examples show range of azimuth and elevation angle are split into a series of bins, where res denotes normalized residual value within the bin.

B. Grasp Proposal Refinement

Non-maximum suppression and Grasp proposal sampling. Since sub-network for stage-1 generates one proposal per point, there are a larger number of proposals around ground-truth grasps. Non-maximum suppression (NMS) is applied to select the local maximum.

Region grouping and grasp canonical transformation. Given the grasp proposals generated by stage-1, point clouds within the gripper closing area are cropped out for further feature representation learning. Unified local coordinates are utilized to eliminate the ambiguity caused by absolute coordinate for objects with various poses and locations. Specifically, we adopt canonical transformation for points within the gripper closing area as shown in Fig.4. We set Approaching, Closing, and Orthogonal directions of the gripper as X, Y, and Z axes respectively, and the origin locates at the gripper bottom center. In experiments, the gripper closing area is enlarged by a scalar ϵ to capture more contextual information, which helps for proposal refinement.

Feature learning for grasp proposal refinement. After proposal canonical transformation, fine-grained local features within the proposals will be learned with the following steps.

First, for each point within the enlarged 3D grasp proposal, we obtain its canonical coordinate $\tilde{p} = \mathcal{T}(p) = (x^{\tilde{p}}, y^{\tilde{p}}, z^{\tilde{p}})$ and corresponding global semantic feature learned by stage-1. Then, each inside point $\tilde{p}$ and corresponding feature f^p of each grasp proposal are combined. Finally, the concated feature of each point inside the proposal are fed into a point cloud encoder to fuse both the global and local feature. Thus, we can obtain discriminative feature representation for grasp proposal refinement with grasp width and confidence.

The overall loss for training grasp proposal refinement sub-network is similar as depicted in grasp proposal generation sub-network.

VI. EXPERIMENTS

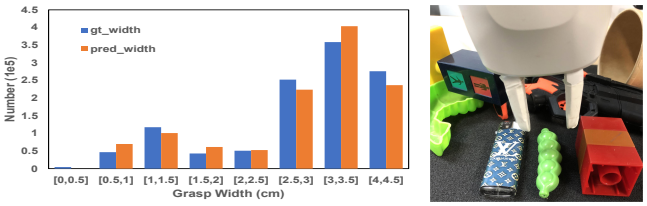
We evaluate our GPR network both in simulation and the Yumi IRB-1400 Robot platform. In simulation experiments, ablation studies show our model predicts high precision grasp configurations. In the real robot platform, experimental results show that our model has good generalization ability.

A. Implementation Details

For each point cloud grasp scene, 16384 points are sampled as input. The learning rate is set to 0.02 at start, and it is divided by 10 when the error plateaus. During the training phase, 256 proposals are sampled after proposals NMS for stage-2, while 100 proposals for inference. Of all the 150 object models, 120 objects are selected for training. Of all the 100k point clouds, 80k point clouds as training data.

B. Simulation Experiments

1) Extend 6-DoF Grasp with Grasp Width: We first evaluate our proposed method in terms of grasp width. To demonstrate the high precision prediction of grasp width, we show a quantitative analysis of over 20k scene with around 2M synthetic grasps. In our experiments, we define the measurement for grasping width as the absolute difference between the predicted grasp width and the ground-truth grasp width $|\omega - \widehat{\omega}|$. We set 4 groups threshold for a comprehensive evaluation of grasp width prediction. For evaluation of each threshold, each absolute grasp width difference smaller than the threshold is classified as positive, otherwise negative.



(a) (b)

Fig. 6: (a) Ground-truth and predicted grasp width distribution. (b) An example shows adaptive grasp width in a clutter scene.

TABLE I: Comparison of Grasp Width Accuracy

Grasp Width Threshold(mm)	Accuracy (%)	
	stage-1	stage-2
2.5	42.0	52.5
5.0	76.0	82.2
7.5	87.5	90.2
10.0	92.9	93.1

We select 100 proposals after NMS operation and filter out the negative samples. Experimental results shown in Tab.I demonstrate that our model can estimate high precision grasp width, and achieves 82.2% accuracy under 5 mm threshold. Fig.6(a) shows the overall grasp width distribution in our dataset. Grasp width is uniformly divided into 8 groups with an interval of 5 mm. While only 1/4 of all lie in the range $[3.5, 4]$ cm. Grasping with max opening width can be problematic in cluttered scenes, because it may lead to collisions with surrounding objects. Fig.6(b) shows an example that adaptive grasp width is critical for dexterous grasping in cluttered scenes.

2) One-stage VS. Two-stage: To illustrate the effectiveness of our proposed grasp pose refinement network, we evaluate the generated grasp proposals quality for both the two stages.

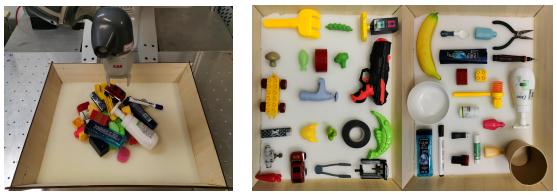
As shown in Tab.I, grasp width accuracy after refinement has 25% and 8% improvement respectively over stage-1 under threshold 2.5 mm and 5 mm. The improvement gets saturated with higher tolerances. For grasp pose accuracy, we adopt the distance measurement of grasp pose as in Eq.1. For evaluation of predicted grasp g_p, g_p is classified as positive, when $D(g_p, g_t)$ is smaller than the predefined threshold, otherwise negative. Experimental results shown in Tab.II demonstrate that proposals after refinement outperform stage-1 by a large margin.

C. Robotic Experiments

We validate the reliability and efficiency of our proposed GPR network in ABB Yumi IRB-1400 robot and a PhoXi industrial sensor. Objects are presented to the robot in dense clutter as shown in Fig.7(a). We keep a similar setting as in the simulation environment: 1) Camera is placed on top of the bin about 1.3 m; 2) Point cloud within the bin is cropped out for input data. 20 similar and 20 novel objects are selected for testing the generalization ability of our proposed network, as shown in Fig.7(b).

TABLE II: Comparison of Grasp Pose Accuracy

Grasp Pose Threshold	Accuracy (%)	
	stage-1	stage-2
0.005	25.3	52.5
0.01	29.1	61.2
0.015	31.8	63.9
0.02	33.5	65.2



(a) (b)

Fig. 7: Real setting of our robotic grasping experiments. (a) Cluttered scene grasping experiment setup with ABB Yumi robotic arm. (b) Objects used in our robotic experiments. Left one shows novel objects which are absent in the training dataset, right one shows similar objects.

We compare GPR to two state-of-the-art, open-sourced 6D grasp baselines, GPD [10] and PointNetGPD [11]. We train GPD and PointNetGPD with their default setting on our dataset with the code they released.

The experiment procedure is as follows: 1) 10 of 20 objects are random sampled out, and then poured into the bin; 2) The robot attempts multiple grasps until all objects are grasped or 15 grasps have been attempted; 3) 10 times testing for each algorithm. The result is shown in Tab.III. Success Rate (SR) and Completion Rate (CR) are used as the evaluation metrics.

TABLE III: Results of Clutter Removal Experiments

Method	Similar objects		Novel objects	
	SR	CR	SR	CR
GPD (3 channels) [10]	60%	84%	50%	66%
GPD (15 channels) [10]	52.7%	78%	36%	54%
PointNetGPD (3 classes)[11]	64.6%	84%	54.8%	80%
Ours	78.3%	94%	69.2%	90%

As shown in Tab.III, our method outperforms baseline methods in terms of Success Rate, Completion Rate, which demonstrates the superiority of our methods. In our observation, our algorithm can get better performance in terms of collisions with surrounding objects and stable grasp configuration.

VII. CONCLUSIONS

In this paper, we proposed an end-to-end grasp pose refinement network for fine-tuning low-quality and filtering noisy grasps, which detects globally and refines locally. Meanwhile, we build a single-object grasp dataset which consists of 150 objects with various shapes, and a large-scale dataset for cluttered scenes. Experiments show that our model trained on the synthetic dataset performs well in real-world scenarios and achieves state-of-the-art performance.

References

[1] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," The International Journal of Robotics Research (IJRR), 2012.

[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 IEEE international conference on robotics and automation (ICRA), 2017.

[3] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in Robotics: Science and Systems (RSS), 2018.

[4] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," The International Journal of Robotics Research (IJRR), 2015.

[6] F. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," IEEE Robotics and Automation Letters (RAL), 2018.

[7] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng, "Roi-based robotic grasp detection for object overlapping scenes," in IEEE International Conference on Intelligent Robots and Systems (IROS), 2019.

[8] 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," in Robotics: Science and Systems (RSS), 2017.

[9] D. Morrison, J. Leitner, and P. Corke, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in Robotics: Science and Systems (RSS), 2018.

[10] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," The International Journal of Robotics Research (IJRR), 2017.

[11] 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 IEEE International Conference on Robotics and Automation (ICRA), 2019.

[12] 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 Conference on robot learning (CoRL), 2019.

[13] 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 IEEE International Conference on Robotics and Automation (ICRA), 2020.

[14] H. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[15] V. Nguyen, "Constructing force-closure grasps," The International Journal of Robotics Research (IJRR), 1988.

[16] C. Ferrari and J. F. Canny, "Planning optimal grasps," in IEEE International Conference on Robotics and Automation (ICRA), 1992.

[17] A. T. Miller and P. K. Allen, "Graspit! A versatile simulator for robotic grasping," IEEE Robotics & Automation Magazine, 2004.

[18] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2020.

[19] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

[20] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in IEEE International conference on advanced robotics (ICAR), 2015.

[21] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," Multimodal Technologies and Interaction, 2018.

[22] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," The international journal of robotics research (IJRR), 2011.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in IEEE conference on computer vision and pattern recognition (CVPR), 2017.

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

[25] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in IEEE International Conference on Intelligent Robots and Systems (IROS), 2017.

[26] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in IEEE International conference on robotics and automation (ICRA), 2011.

[27] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in IEEE international conference on robotics and automation (ICRA), 2016.

[28] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," The International Journal of Robotics Research (IJRR), 2018.

[29] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in IEEE International Conference on Robotics and Automation (ICRA), 2000.

[30] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - A survey," IEEE Transactions on Robotics (TRO), 2014.

[31] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in IEEE international conference on robotics and automation (ICRA), 2016.

[32] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in IEEE International Conference on Computer Vision (ICCV), 2019.

[33] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in IEEE International Conference on Intelligent Robots and Systems (IROS), 2018.

[34] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in IEEE International Conference on Intelligent Robots and Systems (IROS), 2019.

[35] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in IEEE International Conference on Robotics and Automation (ICRA), 2018.

[36] X. Yan, M. Khansari, J. Hsu, Y. Gong, Y. Bai, S. Pirk, and H. Lee, "Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks," arXiv preprint arXiv:1906.08989, 2019.

[37] 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 (NIPS), 2017.

[38] W. Wu, Z. Qi, and F. Li, "Pointconv: Deep convolutional networks on 3d point clouds," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[39] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in IEEE International Conference on Computer Vision (ICCV), 2019.

[40] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in Advances in neural information processing systems (NIPS), 2018.

[41] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "Densepoint: Learning densely contextual representation for efficient

point cloud processing," in IEEE International Conference on Computer Vision (ICCV), 2019.
[42] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[43] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in IEEE International Conference on Computer Vision (ICCV), 2019.
[44] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in Advances in neural information processing systems (NIPS), 2015.
[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in IEEE international conference on computer vision (ICCV), 2017.
[46] 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 IEEE International Conference on Computer Vision (ICCV), 2017.