2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636012

# Simultaneous Semantic and Collision Learning for 6-DoF Grasp Pose Estimation

Yiming Li[1,2], Tao Kong[3], Ruihang Chu[4], Yifeng Li[3], Peng Wang[1,2] and Lei Li[3]

***Abstract*— Grasping in cluttered scenes has always been a great challenge for robots, due to the requirement of the ability to well understand the scene and object information. Previous works usually assume that the geometry information of the objects is available, or utilize a step-wise, multi-stage strategy to predict the feasible 6-DoF grasp poses. In this work, we propose to formalize the 6-DoF grasp pose estimation as a simultaneous multi-task learning problem. In a unified framework, we jointly predict the feasible 6-DoF grasp poses, instance semantic segmentation, and collision information. The whole framework is jointly optimized and end-to-end differentiable. Our model is evaluated on large-scale benchmarks as well as the real robot system. On the public dataset, our method outperforms prior state-of-the-art methods by a large margin (+4.08 AP). We also demonstrate the implementation of our model on a real robotic platform and show that the robot can accurately grasp target objects in cluttered scenarios with a high success rate. Project link: https://openbyterobotics.github.io/sscl.**

## I. Introduction

It has been a common ability for humans to grasp daily objects. By just taking a glance at the scene, we can easily localize the objects of interest and give a proper pose to grasp them. However, it remains quite challenging for robots. The grasping ability requires robots to comprehensively understand the scene and object information. One of the most challenging parts is being able to estimate the robust, accurate, and collision-free grasp pose given by visual sensors.

Grasping is a fundamental skill for most robotic manipulation systems, and has been widely studied over the last decades. Previous works on grasp pose estimation could be categorized into two groups: model-based methods and data-driven methods. Model-based methods assume the geometry information of an object is always available, and use mechanical analysis tools [1]–[3] to calculate the grasp poses of an object. However, it is still an open problem of how to grasp objects with various shapes and sizes in complex scenes. Data-driven methods that aim to address the generic grasp problem are attaching more and more research attention [4]–[6]. They usually adopt Deep Neural Networks (DNNs) for the prediction of feasible grasp poses. A simple manner is to project the 3D space into a 2D plane, transferring the 6-DoF grasping task to a 2D oriented rectangle detection problem [7], [8], where the gripper is forced to approach objects from above vertically. Although this top-down grasp representation can solve regular grasp tasks, it is still difficult to handle complex objects which are supposed to be grasped from diverse poses in cluttered scenes.

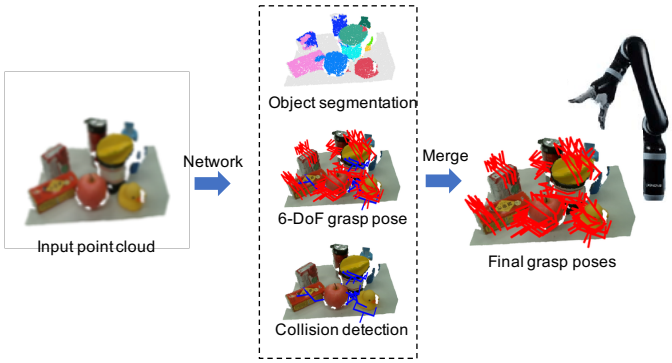


Fig. 1. Overview of our proposed simultaneous multi-task learning framework. Given a scene point cloud, our network jointly predicts instance labels, 6-DoF grasp poses and collisions simultaneously. Finally, we merge three branches and the instance-level, collision-free 6-DoF grasp poses are generated for the robot to execute.

Recent works tend to directly regress 6-DoF grasp poses, or evaluate grasp quality scores from raw 3D point cloud data [9]–[14]. They also utilize a collision detection module as post-processing to filter invalid grasps. Despite their impressive results, we observe several potential drawbacks in such methods. a) They can not learn instance-level grasps. The lack of instance information leads to the model can not carry out target-driven grasping, which is common in manipulation tasks. We also believe that the instance information could boost the grasp learning process. b) They always rely on collision detection as a post-processing module to filter invalid grasps, which is usually indispensable and time-consuming.

Ideally, a 6-DoF grasp pose estimation model should not only predict the feasible grasp poses but also be able to give the object level and collision information to guide the robot grasping. In this paper, we propose a joint learning framework for 6-DoF grasping, which simultaneously predicts the instance semantic segmentation, feasible grasp poses, and potential collisions, as shown in Fig. 1. Given a single-view point cloud as input, our model outputs object-level and collision-free 6-DoF grasps. More specifically, We adopt 3D PointNet++ [15] as our backbone network to extract point features, then jointly train these three target branches in a unified manner, as shown in Fig. 2. We observe that the position relationship between two points of a rigid object in 3D space is fixed, so the SE(3) grasps can be decomposed with two unit offsets, the approach direction and the close

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China.
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.
[3] Bytedance AI Lab
[4] The Chinese University of Hong Kong
Correspondence to: Tao Kong <taokongcn@gmail.com> and Peng Wang <peng_wang@ia.ac.cn>.

direction of a gripper. The grasp poses are merged with the instance prediction and collision detection modules to form the final accurate, robust, and collision-free grasp poses.

We study the 6-DoF grasping problem for a parallel gripper under a realistic yet challenging setting. Assuming various objects are scattered on a table, we only capture a single-view point cloud using a commodity camera. The camera pose is also unfixed so that a partial point cloud can be captured from a random viewpoint instead of the top-down view. This task is challenging both in perception and planning, caused by the scene clutters, multiple unknown objects, incomplete point cloud, and high grasp dimensions.

Extensive experiments on the public dataset [14] and real-world robot systems demonstrate the effectiveness of our approach. The results show that both segmentation and collision branches boost the performance of grasp pose estimation. Semantic information improves grasp pose learning by identifying which instance a point belongs to and collision attributes help to filter invalid grasps. Our method also outperforms current start-of-the-art methods both on the dataset and real robot experiments. In summary, our main contributions are as follows:

- We propose a simultaneous multi-task learning paradigm for 6-DoF grasp pose estimation in structured clutter with instance semantic segmentation and collision detection.
- With the proposed simultaneous multi-task grasping, we could directly predict the point-wise 6-DoF grasp poses.
- Our proposed method outperforms state-of-the-art grasp pose estimation methods both on the large-scale dataset and real robot experiments.

## II. Related Work

In this section, we introduce the previous works related to ours.

**Learning based Grasping Methods.** A large amount of prior grasp pose estimation methods focus on detecting a pair of graspable points or a rectangle on the planar surface based on RGB-D image input [4], [5], [7], [8], [16]–[20]. These simplified methods are constrained on a 2D plane since they only have 3-DoF at most. In order to improve the dexterity of grasping, some researchers focus on predict 6-DoF grasp poses. A straightforward way is to apply 6-DoF object pose estimation to grasp tasks [21]–[24]. Although such model-based approaches can acquire accurate 6-DoF grasps, it requires prior knowledge about the geometry information of the object. Recently, some works directly predict 6-DoF grasp pose from point clouds. GPD [10] proposes to generate grasp candidates based on the local geometry. A CNN-based network is also proposed to classify generated grasps. PointnetGPD [11] extends this work to 3D space and evaluates generated grasps via a PointNet-based network. $S^4G$ [13] uses a single-shot grasp detection proposal network to regress $SE(3)$ grasps in densely cluttered scenes, and [14] designs a two-stage network to decouple grasp poses as approach vectors and other grasp operations (angle, width, depth, etc.). GraspNet [12] adopts a variational autoencoder to sample grasps for the single object and introduces a grasp evaluate network to refine sampled grasps.

In [25], the authors present a target-driven grasp method to grasp a specific object in structured clutters. They use a cascaded pipeline to first crop the target object, then inference the grasp poses, and finally adopt a neural network to predict collision scores. The main difference between this work and ours is that we formulate the 6-DoF grasp task as a simultaneous multi-task learning problem, using a single shot grasp neural network, with jointly training segmentation and collision modules.

**Grasp Dataset Generation.** Some previous works [4], [16] use rectangle representation for grasp detection annotated by humans. In [18], [19], the authors collect grasp labels with real robot experiments. Dexnet [20] provides a universal grasp generation method by calculating force closure as grasp quality score from single object mesh. Some followed studies [13], [26] generate synthetic scene datasets by mapping 6-DoF object pose to single object grasps. [27] compares a detailed grasp sampling strategies for data generation, and [28] collects a large-scale grasp dateset on simulation. To deal with the gap between the virtual environment and the real world, [14] constructs a general grasp dataset in cluttered scenes, in which images are captured by regular commodity cameras.

**Deep Learning for Point Cloud.** Some researchers analyze point clouds by projecting points to multi-view or volumetric representations [29], [30]. To preserve complete geometry information, PointNet [31] and PointNet++ [15] directly process raw points through employing multi-layer perceptron (MLP) and yield point-level features. Such point-based framework has been widely used in the 3D domain such as classification, segmentation and detection [32]–[34]. Due to its inspiring performance, here we also adopt PointNet++ as our backbone network to extract features of the point cloud.

## III. Methods

Given a scene point cloud, we first use the backbone network PointNet++ [15] to encode features, then simultaneously attach three parallel decoders: instance segmentation, 6-DoF grasp pose, and collision detection. These three heads respectively output predicted point-wise instances, grasp configurations, and collisions. At inference phase, grasps from the same instance without collision are grouped, and a pose non-maximum suppression algorithm is proposed to form the final grasps.

As shown in Fig. 2, the input size of point cloud $\mathcal{P}$ is $N_p \times (3 + C)$, where $N_p$ is the number of points, and $C = 6$ is the extra channels about RGB colors and normalized coordinates. The PointNet++ encoder module is composed of four set abstraction (SA) layers, while decoder layers are three feature propagation (FP) modules[1] following with MLP layers, represents the segmentation head, grasp head, and collision head. The output size of these three heads are $N_p \times$

[1]For more details about the PointNet++, we refer readers to [15].

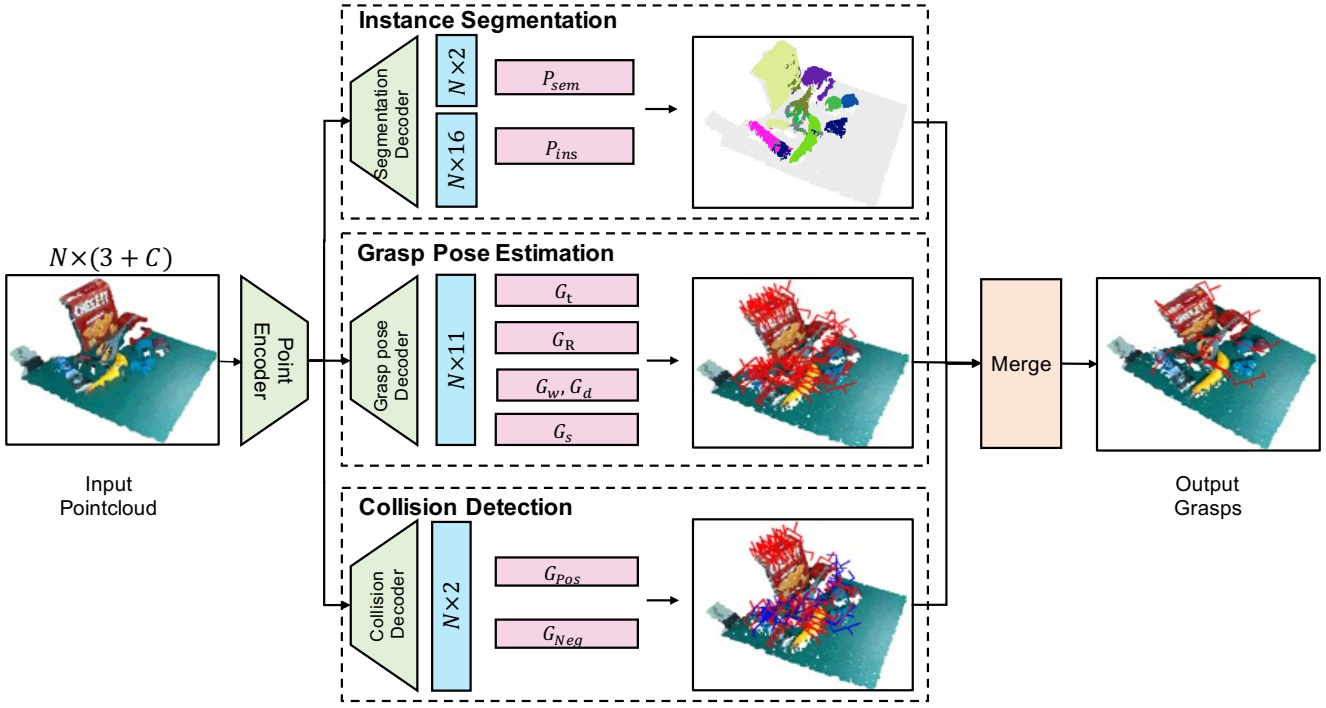


Fig. 2. Framework of our proposed grasp pose learning method. Given N points with (3+C)-dim features, we adopt a point encoder to extract features and three decoders are used to predict object instances, grasp poses and collisions.$P_{sem}, P_{ins}$ denote the predicted point-wise semantic, instance results, $G_t, G_R, G_w, G_d, G_s$ respectively mean center, rotation matrix, width, depth and score of predicted grasps, and $G_{pos}, G_{neg}$ are predicted collisions defined by 6-DoF grasps. An instance based pose-NMS algorithm is used to merge these three modules to form final instance-level, collision-free grasps.

$(2+16)$ (semantic mask and instance embedding), $N_p \times (2+6+1+1+1)$ (graspable mask, two rotation vectors, grasp depth, width and score) and $N_p \times 2$ (collision mask).

### A. Instance Segmentation Branch

We first attach a point-wise instance semantic segmentation module to distinguish multiple objects. Specifically, we formulate object instance segmentation as an embed-and-cluster task. Points belonging to the same instance should have similar features, while features for different instances should be dissimilar. During training, the semantic and instance label of each point are known, and we supervise the outputs semantic labels with a two-class cross entropy loss $L_{sem}(s_i, \hat{s}_i)$ to classify background and foreground, where $s_i$ and $\hat{s}_i$ represent predicted and ground truth binary semantic labels. The instance loss is optimized through a discriminative loss function $L_{ins}$ [35]:

$$L_{ins} = L_{var} + L_{dist} + \alpha L_{reg} \tag{1}$$

where

$$\begin{aligned} L_{var} &= \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}\left[\|x_i-\mu_c\|-\delta_{\mathrm{v}}\right]_+^2 \\ L_{\mathrm{dist}} &= \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1}^{C}\left[2\delta_{\mathrm{d}}-\|\mu_{c_A}-\mu_{c_B}\|\right]_+^2 \\ L_{reg} &= \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\| . \end{aligned} \tag{2}$$

Here $C$ is the number of objects, $N_c$ is the number of points in object $c$, $x_i$ is the feature embedding of point $i$ belong to $c$, $\mu_c = \frac{1}{N_c}\sum_{i=1}^{N_c} x_i$ is the object feature center of class $c$, $||\cdot||$ means the L2 distance. $[x]_+ = max(0,x)$ denotes the hinge. $\delta_v$ and $\delta_d$ are two margins for variance and distance $L_{var}, L_{dist}, L_{reg}$ respectively represent variance loss, distance loss and regularization loss, and $\alpha \ll 1$ is the regularization weights. Variance loss means an intra-cluster loss that draws embeddings towards the cluster center, while distance loss is an inter-cluster loss that increases the distance between different cluster centers. Regularization loss constraints that all clusters towards the origin, to keep the activations bounded.

The total segmentation loss $L_{seg}$ is defined as $L_{seg} = L_{sem} + L_{ins}$. After learning the embeddings, a Meanshift [36] clustering algorithm is applied to group points belong to the same instance.

### B. 6-DoF Grasp Pose Estimation Branch

As shown in Fig. 3 (a), the $SE(3)$ grasp configuration $g$ of the parallel gripper are formulated with a grasp center $g_t$, rotation matrix $g_R$, width $g_w$, depth $g_d$ and a grasp score $g_s$:

$$g = (g_t, g_R, g_w, g_d, g_s), \tag{3}$$

where $g_s$ is used to evaluate the quality of a grasp computed by a improved force-closure metric [11]. To directly regress the point-wise 6-DoF grasp pose, we introduce two assumptions for grasp training:

- For each point $p$ in the point cloud, there is at most one good grasp $g$, such that the mapping $f : p \rightarrow g$ is

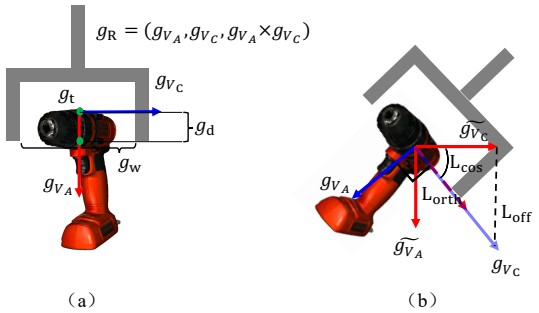


Fig. 3. Grasp illustration. (a) Representation of 6-DoF grasp in $SE(3)$. (b) Representation of decomposed rotation loss $L_{rot}$.

unique and fixed.

- The grasp center $g_t$ is defined on the object surface and refined by grasp depth $g_d$. That means it just needs to classify graspable points to predict grasp centers, instead of regressing 3D coordinates.

Directly learning the 6-DoF grasp pose through rotation representation such as quaternions or rotation matrices is challenging since they are nonlinear and discontinuous. To handle this problem, we propose a vector based method that decompose the rotation matrix with two orthogonal unit vectors.For a grasp $g$, The rotation matrix $g_R \in \mathcal{R}^{3\times3}$ is defined as:

$$g_R = [g_{v_A}, g_{v_C}, g_{v_A} \times g_{v_C}], \tag{4}$$

where $g_{v_*} \in \mathcal{R}^{3\times1}$ is a column vector. $g_{v_A}$ limits the approach direction of gripper, and $g_{v_C}$ is the direction of gripper closing.

For optimization, we divide the rotation loss $L_{rot}$ into three parts (Fig. 3(b)): offset loss $L_{off}$, cosine loss $L_{cos}$, and a relate loss $L_{orth}$, which respectively constrains the position, angle prediction, and the orthogonality:

$$L_{rot}(g_{v_*}, \hat{g}_{v_*}) = \beta_1 \cdot L_{off} + \beta_2 \cdot L_{cos} + \beta_3 \cdot L_{orth}, \tag{5}$$

$$\begin{aligned} L_{off}(g_{v_*}, \hat{g}_{v_*}) &= \frac{1}{G}\sum_{g\in G} \|g_{v_*} - \hat{g}_{v_*}\| \\ L_{cos}(g_{v_*}, \hat{g}_{v_*}) &= -\frac{1}{G}\sum_{g\in G} |g_{v_*} \cdot \hat{g}_{v_*}| \\ L_{orth}(g_{v_A}, g_{v_C}) &= -\frac{1}{G}\sum_{g\in G} |g_{v_A} \cdot g_{v_C}|, \end{aligned} \tag{6}$$

where $g$ is a predicted grasp in set $G$, $g_{v_*}, \hat{g}_{v_*}$ are ground truth and predicted vectors. $\beta_1, \beta_2, \beta_3$ are three coefficients to balance different terms. As the gripper close direction is symmetric, we limit $g_{v_C} = -g_{v_C}$ if $g_{v_C} \cdot (1,0,0)^T < 0$. We observe that this decomposed loss representation can improve the accuracy of grasp rotation compared with directly calculate the distance between two quaternions.

The prediction of graspable point is treated as a two-class classification task and a cross-entropy loss function $L_{gp}(g_{p_i}, \hat{g}_{p_i})$ is applied with weight $w_1 = 1.0$ and $w_2 = 5.0$. $g_{p_i}, \hat{g}_{p_i}$ donate the binary ground truth and prediction of graspable or not for point $i$. In addition, we regress grasp width, depth and score for each grasp $g$ with mean squared error (MSE) loss, record as $L_w(g_w, \hat{g}_w)$, $L_d(g_d, \hat{g}_d)$ and $L_s(g_s, \hat{g}_s)$. $g_*$ and $\hat{g}_*$ respectively mean grasp labels and predicted grasps. The total grasp loss is written as:

$$L_{grasp} = \sum_{P_s} L_{gp} + \sum_{P_g}(L_{rot} + \gamma_1 L_w + \gamma_2 L_d + \gamma_3 L_s), \tag{7}$$

where $P_s$, $P_g$ represent the whole scene point cloud and point set with graspable points in ground truth, respectively. $\gamma_1, \gamma_2, \gamma_3$ are coefficients for grasp configurations.

### C. Collision Detection Branch

Although the methods above are able to predict 6-DoF grasps at the instance level, it is still necessary to determine whether generated grasps are valid and executable in scenes. For this purpose, we attach a collision detection branch to infer potential collisions for each grasp.

We provide a learn-able collision detection network to directly predict collisions for generated grasps. Binary collision labels are generated by an off-the-shelf collision detection module according to the grasp configuration. During training, we sample both positive (without collisions) and negative (with collisions) grasps and supervise the collision as a classification task with a two-class cross-entropy loss $L_{coll}(c_i, \hat{c}_i)$:

$$L_{coll} = \sum_{P_i} -[c_i \cdot \log \hat{c}_i + (1 - c_i) \cdot \log(1 - \hat{c}_i)], \tag{8}$$

where $c_i$ is the collision label and $\hat{c}_i$ denotes the predicted collision probability for point $i$.

**Algorithm 1:** Instance based Pose-NMS

**Input:** Prediction $P = (S, G, C)$
S, G, C are sets of predicted semantic, grasp pose and collision labels, respectively.
**Export:** Executable grasps set $V = \{\}$.
Select collision-free grasps $G_C = G \cap \tilde{C}$.
**for** $(G_i \subset G_C)$ *where* $(S_i \subset instance\ N)$ **do**
  **while** $G_i ! = \emptyset$ **do**
    Sort $g \in G_i$ *by* $g_s$
    $g = g_0$
    Add $g$ *to* $V$, Delete $g_0$
    **for** $g_k \in G_i$ ) **do**
      **if** $SE(3) Distance(g, g_k) < \epsilon$ **then**
        Delete $g_k$
**Output:** Executable grasps set $V = \{g_1, g_2, \dots g_n\}$

### D. Forming Final Grasps

For training, the total loss of the whole multi-task learning framework is written as:

$$L = L_{seg} + L_{grasp} + L_{coll}. \tag{9}$$

After three branches being jointly optimized, our network outputs the instance label, grasp, and collision for each point.

Then we run the instance-based pose-NMS algorithm to merge three modules and form final executable grasps, as shown in Algorithm 1. Points belong to the same instance are grouped and we calculate the SE(3) distance between two grasps to suppress non-maximum grasps. The distance threshold $\epsilon$ is set to $10mm, 30°$.

## IV. Experiments

In this section, we first introduce the experimental setup, including dataset, metrics, and implementation details. Then we analyze the main results and perform ablation studies to evaluate the effectiveness of different modules. Finally, robot experiments are conducted to validate the performance of our method in real-world robot grasping.

### A. Experimental Setup

*1) Dataset and Metrics:* We evaluate the proposed method on both public benchmarks and real robot systems. We adopt GraspNet-1Billion dataset for benchmark evaluation [14], which is a large benchmark for general object grasping. The dataset contains 97,280 RGB-D images with over one billion 6-DoF grasp poses for 88 objects, captured by two popular cameras, Intel Realsense and Kinect in the real world. It is split into the train, test seen, test similar, and test novel sets with 100, 30, 30, 30 scenes, and each scene contains 256 images and millions of grasps for 10 objects randomly sampled. We adopt average *Precision@k* as the evaluation metric [14] , which measures the precision of top-$k$ ranked grasps. $AP_\mu$ denotes the average precision for $k$ ranges with a force closure parameter lower than $\mu$, where $\mu \in \{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$. Here we use the default value of parameter $k = 50$. In real robot experiments, we adopt grasp success rate and completion rate to evaluate the performance. Success rate denotes the percentage of successful grasps, and completion rate represents the percentage of objects that is successfully grasped.

*2) Implementation Details:* The input point cloud is generated from an RGB-D image, and preprocessed with workspace filtering, random sampling, and normalization. The number of points $N_p$ is set to 20,000. Because a large amount of grasps for each image is too dense for the network to learn, we select object grasps with $score > 0.5$ by approach-based grasp sampling [27] during training. The point-grasp mapping is formulated by calculating the minimum distance between points and grasp centers within $5mm$. Points without any ground-truth grasps are ignored, which contributes nothing to the training objective. Our neural network is trained for 80 epochs with Adam optimizer [37]. The initial learning rate is 0.05 and decreases by a factor of 2 for every 10 epochs with the batchsize of 64. For the loss function, we set $\alpha = 0.01$, $\delta_d, \delta_v = 1.5, 0.5$, $\beta_1, \beta_2, \beta_3 = 5.0, 1.0, 1.0$, and $\gamma_1, \gamma_2, \gamma_3 = 100, 1000, 10$.

At inference, our network outputs point-wise semantic, instance embeddings, grasps, and collisions. A MeanShift algorithm is adopted to group points belong to the same instance.s For each predicted graspable point, a grasp configuration composed of grasp pose, width, depth, and score is generated. Collision detection network outputs the probability of whether a grasp is collision-free.

The proposed simultaneous 6-DoF grasp pose estimation model consists of a point encoder module and three decoders: segmentation head, grasp head and collision head. We describe the network architecture of our model and summarize the details as Tab. I. Following the same notation in PointNet++, $SA(K, r, [l_1, \cdots, l_d])$ is a set abstract layer with $K$ local regions in radius $r$, and $[l_1, \cdots, l_d]$ are fully connected layers with $l_i (i = 1, \cdots, d)$ output channels. FP($[l_1, \cdots, l_d]$) is a feature propagation layer with $d$ fully connected layers. MLP($l_1, \cdots, l_d$) means a multi-layer perceptron with output layer sizes $l_1, \cdots, l_d$ on each point.

TABLE I
Detailed network architecture of our proposed method.

| Network | Architecture |
|---|---|
| Point encoder | SA(1024,0.1,[32,32,64]) → SA(256,0.2,[64,64,128]) → SA(64,0.4,[128,128,256]) → SA(None,None,[256,256,512]) |
| Segmentation head | FP[256,256]→FP[256,256]→ FP[256,128]→FP[128,128,128]→ MLP(128,128,2) & MLP(128,128,16) |
| Grasp head | FP[256,256]→FP[256,256]→ FP[256,128]→FP[128,128,128]→ MLP(128,128,2) & MLP(128,128,6)& MLP(128,128,3) |
| Collision head | FP[256,256]→FP[256,256]→ FP[256,128]→FP[128,128,128]→ MLP(128,128,2) |

### B. Main Results

We compare the performance of different methods on GraspNet-1Billion, which is based on physical analysis with calculating the grasp quality by force closure metric (Tab. II). It can be seen that our simultaneous learning approach achieves the best performance. Specifically, compared with rectangle-based grasp methods which execute top-down grasps on the 2D plane [5], [38], the 6-DoF grasp representation has obvious advantages. Besides, point-based networks also outperform 2D convolution networks [10]. Compared with other deep 6-DoF grasp methods [11], [14], our approach also shows obvious gains (*e.g.* +4.08 AP on seen split).

Despite our single-shot end-to-end network predicts grasps without any refinement, it still has the best performance of mAP on all of the test datasets. In addition, we observe that our model always performs better on $AP_\mu$ with a large $\mu$, indicating that it tends to generate more proper predictions and avoids wrong candidates to ensure an executable grasp instead of learning several best grasps only, which is significant in real robot grasping tasks.

For qualitative valuation, we visualize the outputs of grasp pose results and compare them with the benchmark method. For a fair comparison, we choose 20 grasps with the highest scores in each point cloud, which is shown in Fig. 4.

TABLE II

6 DOF GRASP POSE ESTIMATION RESULTS ON GRASPNET-1BILLION DATASET.

| Methods | Grasp type | Seen | | | 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 [38] | 2D planar | 15.48 | 21.84 | 10.25 | 13.26 | 18.37 | 4.62 | 5.52 | 5.93 | 1.86 |
| Chu et al. [5] | 2D planar | 15.97 | 23.66 | 10.80 | 15.41 | 20.21 | 7.06 | 7.64 | 8.69 | 2.52 |
| GPD [10] | 6-DoF | 22.87 | 28.53 | 12.84 | 21.33 | 27.83 | 9.64 | 8.24 | 8.89 | 2.67 |
| PointnetGPD [11] | 6-DoF | 25.96 | 33.01 | 15.37 | 22.68 | 29.15 | 10.76 | 9.23 | 9.89 | 2.74 |
| GraspNet [14] | 6-DoF | 27.56 | 33.43 | 16.95 | 26.11 | 34.18 | 14.23 | 10.55 | 11.25 | 3.98 |
| GraspNet re-implementation [14] | 6-DoF | 32.47 | 37.72 | **29.21** | 26.78 | 31.96 | **23.19** | 10.27 | 12.94 | **5.41** |
| Ours | 6-DoF | **36.55** | **47.22** | 19.24 | **28.36** | **36.11** | 10.85 | **14.01** | **16.56** | 4.82 |

TABLE III

ABLATION STUDY ON INSTANCE SEGMENTATION AND COLLISION DETECTION MODULES6.

| Branches | Seen | | | Similar | | | Novel | | |
|---|---|---|---|---|---|---|---|---|---|
| | **AP** | $AP_{0.8}$ | $AP_{0.4}$ | **AP** | $AP_{0.8}$ | $AP_{0.4}$ | **AP** | $AP_{0.8}$ | $AP_{0.4}$ |
| Pose | 25.57 | 31.46 | 13.75 | 16.01 | 19.52 | 6.97 | 7.55 | 8.34 | 2.40 |
| Pose+Seg | 26.62 | 33.94 | 14.72 | 16.93 | 21.27 | 7.64 | 8.43 | 9.68 | 2.93 |
| Pose+Coll | 34.58 | 43.96 | 18.42 | 27.35 | 33.86 | **12.30** | 12.07 | 14.08 | 4.00 |
| Pose+Seg+Coll | **36.55** | **47.22** | **19.24** | **28.36** | **36.11** | 10.85 | **14.01** | **16.56** | **4.82** |


Fig. 4. A visual comparison for GraspNet [14] and our proposed grasp method. It can be seen that our model could generate high quality, more diverse grasp poses on object level.

### C. Ablation Study

Our multi-task learning method introduces instance segmentation and collision detection modules to improve grasp pose estimation. To investigate the influence of these components and the proposed rotation loss, we conduct a set of ablation studies.

*Instance Segmentation.* The instance segmentation information boosts the naive grasp pose baseline by 1.05%, as shown in Tab. III. The improved performance demonstrates that the segmentation module boosts grasp pose learning by providing more semantic information about instance geometry and relation. Note that different objects are usually occluded with each other, the segmentation plays a crucial role in separating different objects in the cluttered scenario.

*Collision Detection.* Another important feature of our framework is that it could simultaneously predict the collision probability of each potential grasp pose. The collision module prevents grasps with invalid poses which are misjudged by the network. Including collision detection branch improves the pose AP by 9.01%, which demonstrates its effectiveness.

*Rotation Loss.* Our proposed $L_{rot}$ loss also benefits grasp pose learning by 1.59%, as shown in Tab. IV. We compare it with directly calculating the related angle between quaternion label $g_q$ and prediction $\hat{g}_q$, where the loss function $L_{quat}$ is defined by:

$$Angle(g_q, \hat{g}_q) = \arccos(0.5 \times (\text{trace } [g_R \hat{g}_R^T] - 1)$$
$$L_{quat} = \frac{1}{G} \sum_{g \in G} Angle(g_q, \hat{g}_q), \tag{10}$$

where $g_R, \hat{g}_R$ are two rotation matrices equal to $g_q, \hat{g}_q$. By decomposing rotation matrix with two specific directions, our network learns the nonlinear representation much easier.

*Running Time.* The inference time for the forward path of our network on Nvidia RTX 2070 is 21ms, which could largely satisfy the most application requirements. The post-processing time, including point MeanShift clustering and grasp pose-NMS, respectively cost 762ms and 851ms on

TABLE IV

ABLATION STUDY ON ROTATION LOSS.

| Loss | AP | $AP_{0.8}$ | $AP_{0.4}$ |
|---|---|---|---|
| Quaternion angle | 34.96 | 45.39 | 18.53 |
| Ours | **36.55** | **47.22** | **19.24** |

Intel Core i5-8500 CPU. The total time efficiency of our method is 1634ms. We believe both clustering and pose-NMS could be implemented in more time-efficient ways [39], [40], which will be further explored in the future work.

### D. Robot Experiment

In order to evaluate the performance of our proposed methods in the real world, we set up robot experiments on a Kinova Gen2 Robot with a parallel gripper Jaco2. A commercial RGB-D camera, Realsense D435i is mounted on the robot wrist to capture the input point cloud from an oblique perspective (Fig. 5).


Fig. 5. Our robot grasp system and workspace.

We prepare over 20 objects absent in the training dataset with various shapes for robot grasping. For each experiment procedure, we randomly choose 5, 8, 10 objects, and put them on a table to form a structure cluttered scene. The point cloud of the scene is firstly collected by the Realsense camera and sent to the trained model to get the final grasp parameters. The robot attempts multiple grasps until all of the objects are grasped, and the max number of grasping attempts is set to 15. We repeat the experiments 3 times for each method. We calculate the success rate in Tab. V. It can be seen that compared with 2D planar grasp (GQ-CNN [20]), 6-DoF grasp from diverse directions performs better. Our model shows a 76.0% success rate with an 82.6% completion rate and outperforms baseline methods by a large margin. It proves that our grasp prediction model can successfully transfer to real robot grasp tasks.

Since 2D planar grasp methods capture vision inputs from a top-down view, we also compare our proposed model with GQ-CNN [20] from the same view for a fair comparison.

TABLE V

THE MAIN RESULTS OF REAL ROBOT EXPERIMENT.

| Methods | Success rate | Completion rate |
|---|---|---|
| GQ-CNN [20] | 58.1% | 62.3% |
| GPD (3-channel) [10] | 61.5% | 69.6% |
| GPD (12-channel) [10] | 60.0% | 65.2% |
| GraspNet [14] | 72.4% | 79.7% |
| Ours | **76.0%** | **82.6%** |

TABLE VI

TOP-DOWN VIEW GRASPING EXPERIMENT RESULTS.

| Methods | Success rate | Completion rate |
|---|---|---|
| GQ-CNN [20] | 64.1% | 72.5% |
| Ours | **70.1%** | **78.3%** |

Experiment results are recorded in Tab. VI. Although the drastic gradient change in depth image boosts the 2D planar grasping, our simultaneous grasp learning method still has a better performance. We also observe that the top-down view brings severe self-occlusion which leads to a worse point cloud input and causes more collision between grippers and the unseen part of the object, reducing the success rate and completion rate of grasping. It also proves that capturing visual inputs from an oblique viewpoint could improve the performance of 6-DoF grasping.

## V. CONCLUSION

In this paper, we formalize the 6-DoF grasp pose estimation in clutters as a simultaneous multi-task learning problem. We jointly predict the feasible 6-DoF grasp poses, instance semantic segmentation, and collisions. The whole framework is end-to-end trainable and jointly optimized in a unified network. On the large scale public dataset, our method outperforms prior state-of-the-art methods by a large margin. We also demonstrate the implementation of our model on a real robotic platform. Besides, it is convenient to extend our simultaneous grasp learning model to several target-driven robotic manipulation tasks, such as picking and placement, object rearrangement, and interactive grasping. In the future work, we will (a) improve the pose-NMS efficiency by parallel implementations, and (b) utilize multi-view reconstruction techniques to further boost the performance.

## REFERENCES

[1] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004. 1

[2] 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. 1

[3] H. Dang and P. K. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1311–1317. 1

[4] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3511–3516. 1, 2

[5] 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. 1, 2, 5, 6

[6] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 769–776. 1

[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. 1, 2

[8] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2038–2043. 1, 2

[9] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 598–605. 1

[10] 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. 1, 2, 5, 6, 7

[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 *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635. 1, 2, 3, 5, 6

[12] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910. 1, 2

[13] 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*. PMLR, 2020, pp. 53–65. 1, 2

[14] 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. 1, 2, 5, 6, 7

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017. 1, 2

[16] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 3304–3311. 2

[17] 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. 2

[18] 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*, vol. 37, no. 4-5, pp. 421–436, 2018. 2

[19] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3406–3413. 2

[20] 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," *arXiv preprint arXiv:1703.09312*, 2017. 2, 7

[21] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017. 2

[22] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352. 2

[23] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3665–3671. 2

[24] W. Wei, Y. Luo, F. Li, G. Xu, J. Zhong, W. Li, and P. Wang, "Gpr: Grasp pose refinement network for cluttered scenes," *arXiv preprint arXiv:2105.08502*, 2021. 2

[25] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6232–6238. 2

[26] 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)*. IEEE, 2020, pp. 3619–3625. 2

[27] C. Eppner, A. Mousavian, and D. Fox, "A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set," *arXiv preprint arXiv:1912.05604*, 2019. 2, 5

[28] E. Clemens, A. Mousavian, and D. Fox, "ACRONYM: A large-scale grasp dataset based on simulation," *arXiv preprint arXiv:2011.09584*, 2020. 2

[29] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5648–5656. 2

[30] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920. 2

[31] 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. 2

[32] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2088–2096. 2

[33] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232. 2

[34] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779. 2

[35] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017. 3

[36] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002. 3

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 5

[38] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018. 5, 6

[39] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "Pointgroup: Dual-set point grouping for 3d instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4867–4876. 7

[40] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural Information Processing Systems*, 2020. 7