2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

# Learning to Generate 6-DoF Grasp Poses with Reachability Awareness

Xibai Lou[1], Yang Yang[2] and Changhyun Choi[1]

***Abstract*— Motivated by the stringent requirements of unstructured real-world where a plethora of unknown objects reside in arbitrary locations of the surface, we propose a voxel-based deep 3D Convolutional Neural Network (3D CNN) that generates feasible 6-DoF grasp poses in unrestricted workspace with reachability awareness. Unlike the majority of works that predict if a proposed grasp pose within the restricted workspace will be successful solely based on grasp pose stability, our approach further learns a reachability predictor that evaluates if the grasp pose is reachable or not from robot's own experience. To avoid the laborious real training data collection, we exploit the power of simulation to train our networks on a large-scale synthetic dataset. This work is an early attempt that simultaneously learns grasping reachability while proposing feasible grasp poses with 3D CNN. Experimental results in both simulation and real-world demonstrate that our approach outperforms several other methods and achieves 82.5% grasping success rate on unknown objects.**

***Index Terms*— Grasping, Deep Learning in Robotics and Automation, Perception for Grasping and Manipulation**

## I. INTRODUCTION

Real-world applications demand robotic manipulation algorithms that are efficient in arbitrary workspace where objects may not be reachable. Fig. 1 illustrates a scenario where such an algorithm needs to 1) decide which of the sampled grasp pose candidates are more reachable and 2) grasp as many objects as possible from the dense clutter with minimal efforts.

The predominant top-down grasping is often restricted in narrowly prepared workspace [1], whereas practical problems are often in extended and obstacle-rich environments that require flexible 6-DoF grasp poses to reach objects. Albeit extensive researches have been conducted on this topic, the grasping reachability problem remains challenging. Current 6-DoF approaches grasp within restricted workspace and only predict successful grasps by analyzing grasp poses and object shapes [2]. When applied to unrestricted workspace, however, these approaches experience excessive planning failures that jeopardize grasping efficiency. We present a reachability aware 3D deep Convolutional Neural Network (3D CNN) that addresses these concerns by proposing feasible 6-DoF grasp poses that are both stable and reachable.

Our approach consists of a 3D CNN and a Reachability Predictor (RP). 3D CNN learns spatial information [3] which is effective in learning stable 6-DoF grasp poses and generalizing to novel objects [4]. Furthermore, the relatively small sim-to-real gap brought by depth is promising for direct real-world application. RP is effective in estimating the reachability of the sampled grasp poses without going through the computationally expensive motion planning algorithms. This attribute is jointly determined by the grasp pose and the kinematic constraints of a robot arm. Our approach discovers this intervened relationship and improves grasping efficiency by learning to approximate the grasping reachability from self-exploring experience. The immediate challenge here is how to train these models, which typically require hundreds of thousands of labeled data in order to generalize. Many learning-based grasping approaches suffered from insufficient training data since real robot data are notoriously expensive to collect. We exploit the power of a robot simulator and solve this problem with large-scale self-supervision.

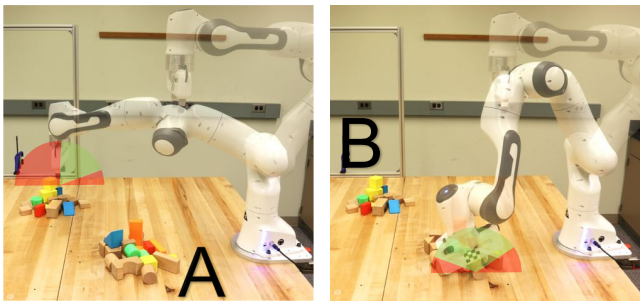



Fig. 1: **Example of searching for a feasible grasp pose.** Two clusters of objects are randomly arranged on the table. The green masks represent reachable approaching directions whereas the red mask is for unreachable ones. Note that in the left figure the robot has reached its limit, wherein the chance of finding a stable and reachable grasp in clutter A is much greater than that in clutter B.

Our work is inspired by human behavior; we naturally prefer to grasp closer objects with appropriate hand and arm poses, whereas for distant objects we either adjust our hand pose or abort grasping. Likewise, we propose an approach to mimic such a highly efficient grasping strategy. We conducted several experiments and ablation studies in simulation as well as real-world where our approach outperforms several comparable approaches in densely cluttered settings and generalizes to novel objects. To the best of our knowledge, our grasping strategy is the first attempt to generate reachability aware 6-DoF poses in dense clutter using a voxel-based 3D CNN. The main contributions of our

*This work was in part supported by the MnDRIVE Initiative on Robotics, Sensors, and Advanced Manufacturing.

[1]X. Lou and C. Choi are with the Department of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, USA {lou00015, cchoi}@umn.edu

[2]Y. Yang is with the Department of Computer Science and Engineering, Univ. of Minnesota, Minneapolis, USA yang5276@umn.edu

work are bi-folded:

- **3D CNN-based grasp pose generation in 6-DoF** takes advantage of our grasp pose sampling algorithm that uniformly samples over the entire 3-dimensional space. In order to predict feasible 6-DoF grasp pose and generalize to novel objects, we exploit large-scale synthetic data collection via self-supervision. Furthermore, the domain-invariant nature of 3D CNN facilitates the direct application to real robot.
- **Reachability Predictor** learns the robot capability from extensive self-exploration and eliminates the need for human-imposed constraints such as workspace restrictions and approaching direction filtering. It is able to predict the reachability of the sampled grasp pose candidates, thus increase the grasping and planning success rate of 6-DoF grasp pose in unrestricted workspace. Since it is decoupled from grasp learning, RP is applicable to other manipulation learning.

## II. RELATED WORK

### A. Object Grasping

Though there are different taxonomies of robotic grasping [5], [6], the existing works of robotic grasping are commonly divided into two groups: traditional model-based and modern learning-based approaches.

Traditional model-based approaches often involve physical modeling of objects and thus require full knowledge of the objects such as shapes, weights, friction coefficients [7], [8]. More recent approaches utilize grasp quality metrics to select the force closure grasps from pre-planned sets by analyzing the contact wrench space [9]. These approaches grasp efficiently with accurate measurements and models, however, the prerequisite efforts scale-up fast when implemented in real unstructured environment where novel objects are prevalent.

Recent learning-based approaches apply deep neural networks to various robotic grasping problems [10], [11], [12], [13], [14], [15]. The majority of these approaches employ convolutional neural networks (CNNs) that take monocular RGB images or 2.5D depth image as input and map the extracted features to a less complicated 3-DoF grasp pose that includes a grasping point on 2D image plane and a corresponding wrist orientation [10], [12], [13], [1], [16]. [17] and [2] extend beyond the standard 3-DoF approaches. ten Pas et al. trained the networks with RGB-D images to evaluate a set of sampled grasp candidates. Since their sampling algorithm is based on geometric reasoning, they were able to find 6-DoF grasp poses. Due to the data hungry nature of learning-based approaches, generating large-scale training dataset is necessary. One approach is to label robot trails manually [16], [4]. A less laborious way is to collect data in simulation [12], [11], [13]. Although fast and scalable, the sim-to-real gap may render features learned in simulation inaccurate in real-world. Some re-train the network with real-world data [1], others solve the problem by either fine-tuning or domain adaptation [18], [19], [20].

Originated from computer vision tasks such as object recognition [3], 3D data segmentation [21], and scene completion [22], voxel-based 3D CNN have also been applied to robotic grasping [4]. Choi et al. showed 3D CNN trained with real-robot data is effective in classifying discrete grasp poses of a soft hand for a single object. Our works differ in two ways. First, our problem is much more difficult in that we search in a 6-DoF space, which is necessary for non-compliant grippers. Second, generalization of 6-DoF grasping requires exponentially more data that are only possible to collect in simulation, where we develop a data collection framework and show our approach can be directly applied to real robot.

### B. Grasp Pose Reachability

Robotic manipulation such as grasping needs to solve an inverse kinematic problem to find a path for the manipulation tool. The grasping range of a given robot arm is determined by its maximum manipulability of the closed kinematic chain [23]. This problem is nontrivial and an analytical solution is often not easy to find. Modern approaches employ fast motion planning algorithms such as RRT [24], and its variants [25]. Due to the random exploring nature of these algorithms, a solution for valid grasp poses is not guaranteed. To avoid excessive planning failures, the majority of approaches in robotic grasping restricts testing objects within a restricted workspace [2], [4], [1], trading operation capability with computation simplicity. Some previous works tried to address this limitation by an using offline database to estimate the grasping reachability [26]. Akinola et al. proposed an online grasp planning method that queries a large database of feasible grasp poses [27]. Distinguished from those approaches, our work is one of the early explorations that learns this reachability from synthetic dataset, which allows the robot to work at an increased capacity.

## III. PROBLEM FORMULATION

We aim to find feasible 6-DoF grasp poses that are simultaneously stable and reachable. The problem is carried out in two stages. At the first stage, the robot finds a set of stable grasp candidates, some of which may not have valid motion plans due to invalid inverse-kinematic solution. At the second stage, the robot evaluates each grasp pose and excludes the unreachable ones. The problem can be formalized as follows:

**Definition 1.** *A grasp pose $\mathbf{X} \in SE(3)$ is **stable** if it is able to form a force closure grasp. This attribute is independent of kinematic constraints.*

**Definition 2.** *A grasp pose $\mathbf{X} \in SE(3)$ is **reachable** if the given robot arm is able to achieve the pose without violating the physical limits of itself and the environment.*

**Definition 3.** *Given a point cloud $\mathcal{P} \subset \mathbb{R}^3$, the goal is to find a **feasible** grasp pose $\mathbf{X_f} \in SE(3)$ that is both **stable** and **reachable**.*

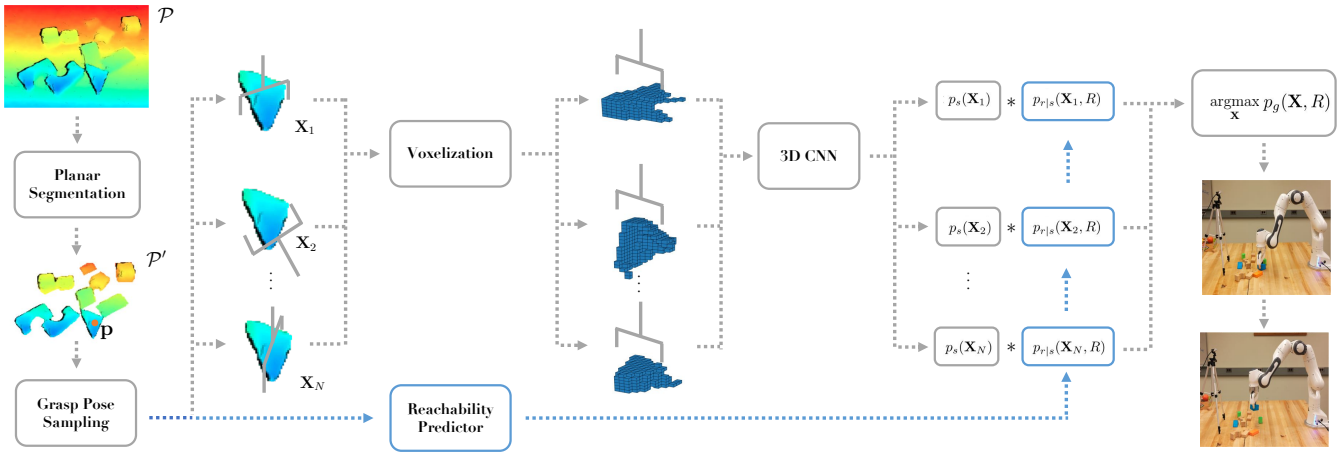



Fig. 2: **Grasping Pipeline.** Object point cloud $\mathcal{P}'$ is obtained from planar segmentation of point cloud $\mathcal{P}$. For each of the sampled grasp poses $\mathbf{X} \in \mathcal{X}$, the object point cloud $\mathcal{P}'$ is voxelized to voxel grid $\mathcal{V}$ and transformed by the corresponding grasp pose candidate $\mathbf{X}$. The input voxel grid is then passed to 3D CNN while the grasp candidate $\mathbf{X}$ is fed to RP for evaluation. The most probable grasp pose is chosen and executed by the robot manipulator.

The grasp pose $\mathbf{X}$ is defined with respect to the robot coordinate frame. The point cloud $\mathcal{P}$ is obtained via a depth sensor with known extrinsic parameters, by which the cloud $\mathcal{P}$ is transformed from the sensor coordinate frame to the robot coordinate frame. An important assumption is:

**Assumption 1.** *The set of 6-DoF grasp candidates $\mathcal{X}$ is randomly generated over the points $\mathbf{p}$ in the point cloud $\mathcal{P}$, i.e., $\mathbf{p} \in \mathcal{P}$.*

Given a point cloud $\mathcal{P}$ and a grasp candidate $\mathbf{X} \in \mathcal{X}$, where $\mathcal{X}$ denotes a set of $N$ uniformly generated 6-DoF grasp candidates, let $\mathcal{S}_s(\mathbf{X}) \in \{0,1\}$ denote a binary-valued stability metric where $\mathcal{S}_s = 1$ indicates that the grasp is stable according to Definition 1. Our goal is to estimate the grasping stability $p_s(\mathbf{X}) = Pr(\mathcal{S}_s = 1|\mathbf{X})$ by self-supervised learning. To train the network more efficiently, we constrain our grasp sampling algorithm as follows:

**Constraint 1.** *The grasp pose $\mathbf{X} \in SE(3)$ is constrained in that the sampled grasp candidates are limited so as to approach the target object from the top hemisphere.*

For an arbitrary grasp candidate $\mathbf{X}$, the wrist orientation has no influence over its reachability since the joint limit is $(-\pi, \pi)$. Therefore, a valid robot grasp $\mathbf{X}$ is determined by the combination of grasp location $(x, y, z) \in \mathbb{R}^3$ and approaching direction $(a_x, a_y, a_z) \in \mathbb{R}^3$. We construct the reachability determinant $\mathbf{a} = (x, y, z, a_x, a_y, a_z)$. To allow the robot fully explore the workspace, we assume that:

**Assumption 2.** *The robot workspace is unrestricted and can be anywhere within the camera observation space.*

Let $R$ denote the robot and $\mathcal{S}_r(\mathbf{X}, R) \in \{0,1\}$ a binary-valued reachability metric, where $\mathcal{S}_r = 1$ indicates that the grasp is reachable. We wish to learn to predict the grasping reachability $p_r(\mathbf{X}, R) = Pr(\mathcal{S}_r = 1|\mathbf{X}, R)$ from self-supervised exploration. It is important to notice that the stability metric of a pose $\mathbf{X}$ is independent of its reachability metric, i.e., $\mathcal{S}_s = 1$ does not indicate $\mathcal{S}_r = 1$ and vice versa. Let $S_g$ denote a binary-valued grasping feasibility metric, where $S_g = 1$ indicates the grasp pose is feasible, and therefore stable and reachable according to Definition 3.

## IV. PROPOSED APPROACH

### A. Generating Feasible Grasp Poses

The objective of our approach is to predict the most feasible grasp pose from a set of randomly sampled candidates in multiple settings. According to Definition 3, the selected grasp pose should be both stable and reachable. Training one generic model on this task leads to unsatisfactory performance, as the network may falsely ascribe the reason of a failed grasp to unstable grasp poses, while the true cause is poor reachability, and vice versa. As mentioned in the previous section, these two prerequisites of a feasible grasp pose are entirely independent of each other. This allows us to solve the credit assignment problem by decoupling the task into two independent sub-problems. The grasping success probability $p_g(\mathbf{X}, R)$ can be decomposed as follow:

$$\begin{aligned} p_g &= Pr(S_g = 1|\mathbf{X}, R) \\ &= Pr(S_r = 1, S_s = 1|\mathbf{X}, R) \\ &= Pr(S_r = 1|S_s = 1, \mathbf{X}, R) \times Pr(S_s = 1|\mathbf{X}) \\ &= p_{r|s}(\mathbf{X}, R) \times p_s(\mathbf{X}) \end{aligned} \tag{1}$$

We trained a 3D CNN grasp pose predictor and a reachability predictor to estimate the resulting grasping stability $p_s$ and grasping reachability $p_{r|s}$ respectively.

The grasping pipeline is described in Fig. 2. The system first obtains the object point cloud $\mathcal{P}'$ from planar segmentation of point cloud $\mathcal{P}$ and samples $N$ grasp candidates over $\mathcal{P}'$ via the sampling algorithm described in Section IV-B. For each sampled grasp candidate, objects point cloud is voxelized to a 3D voxel grid $\mathcal{V} \in \mathbb{Z}^{32\times32\times32}$, where each voxel in the grid is either 0 (not occupied) or 1 (occupied). The total physical edge length of the voxel

**Algorithm 1** Reachability Aware 3D CNN Grasping

**Input:** point cloud $\mathcal{P}$, 3D CNN Grasp model $\mathcal{N}_s$, reachability predictor $\mathcal{N}_r$

**Output:** feasible grasp pose $\mathbf{X_f} \in SE(3)$

$\mathcal{P}' \leftarrow$ `PlanarSegmentation`$(\mathcal{P})$
$\mathcal{X} \leftarrow$ `GraspPoseSampling`$(\mathcal{P}')$
**for** $\mathbf{X} \in \mathcal{X}$ **do**
  $\mathbf{a} \leftarrow$ `Extract`$(\mathbf{X})$
  $\mathcal{V} \leftarrow$ `Voxelization`$(\mathcal{P}', \mathbf{X})$
  $p_{r|s} \leftarrow \mathcal{N}_r$`.Feedforward`$(\mathbf{a})$
  $p_s \leftarrow \mathcal{N}_s$`.Evaluate`$(\mathcal{V})$
  $p_g(\mathbf{X}) \leftarrow p_{r|s} \times p_s$
$\mathbf{X_f} \leftarrow \arg\max_{\mathbf{X}} p_g(\mathbf{X})$
`Grasp`$(\mathbf{X_f})$

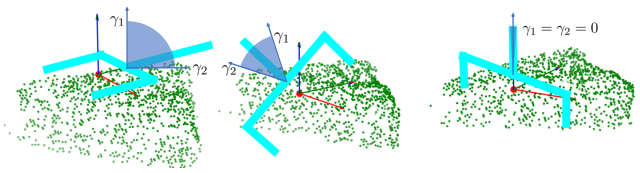



Fig. 3: **Examples of randomly sampled grasp candidates**. The grasp candidates are sampled over a triangle shape. For each candidate, the wrist orientation is sampled in $[0, 2\pi]$ while approaching direction bounded by $[\gamma_1, \gamma_2]$. Different thresholds of the approaching direction are visualized here. Left figure is searching in full 6-DoF by setting $\gamma_1$ = 0°and $\gamma_2$ = 90°, the middle one shows a search constrained by $\gamma_1$ = 30°and $\gamma_2$ = 60° and on the right is a top grasp pose by setting both $\gamma_1$ and $\gamma_2$ to 0.

grid is $0.1m$, equivalent to crop the object point cloud by a $0.001m^3$ cubic box that centered at grasping point $\mathbf{p}$. We choose cropping rather than segmenting an object from the point cloud as it preserves surrounding geometry that helps avoid collisions in dense clutter. The object voxel grid $\mathcal{V}$ may partially contain voxels of adjacent objects, which contributes to lower grasping stability predictions. Therefore, less surrounded objects, i.e., objects on the peripherals, are prioritized, resulting in an onion-peeling grasping pattern.

Our 3D CNN architecture is similar to those in [3] and [4]. We embed the grasp pose within the voxel grid by transforming it with grasp candidate $\mathbf{X}$. The 3D CNN will predict the grasping stability $p_s(\mathbf{X})$ based on the transformed voxel grid. The reachability predictor then extracts the reachability determinant $\mathbf{a} \in \mathbb{R}^6$ from each randomly generated pose $\mathbf{X}$ and estimates the grasping reachability $p_{r|s}(\mathbf{X}, R)$. The grasping success probability $p_g$ is calculated by multiplying the two terms, Algorithm 1 explains this prediction procedure in detail.

### B. Grasp Pose Sampling Algorithm

Unlike the geometric reasoning-based sampling algorithm proposed in [2], to ensure that the robot comprehensively explores the 6-DoF action space, we present a flexible algorithm that uniformly samples grasp candidates over the target objects within two threshold values. Given a desired number of samples $N$ and approaching vector thresholds $\gamma_1, \gamma_2 \in [0°, 90°]$, where $\gamma_1 < \gamma_2$, the algorithm uniformly selects $N$ grasp points from the input point cloud $\mathcal{P}$. For each grasp point, a random pose is associated. Fig. 3 shows three examples of the sampled grasp pose candidates. The sampling algorithm gives little constraints to the generated grasp candidates, so the network is able to learn from the failed grasps, such as not to grasp a corner or collide with the object.

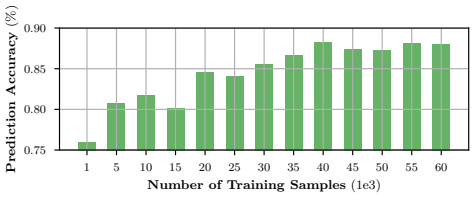



Fig. 4: **Prediction accuracy w.r.t. the size of dataset.** 3D CNN demands large-scale labeled data, tailored to the task, and such requirement is only attainable in simulation.

### C. Network Architecture

We borrow the network structure from [4]; the first layer has 32 filters of size $5 \times 5 \times 5$, the second layer has 32 filters of $3 \times 3 \times 3$. The features are condensed by a Max Pooling layer of $2 \times 2 \times 2$, followed by two dense layers of 128 and 1. A given grasp pose can either be 1 (stable grasp) or 0 (unstable grasp), so we use binary cross-entropy as the loss function. We use the sigmoid activation function in the final layer to predict the grasping stability for the voxel grid $\mathcal{V}$.

Reachability predictor consists of an input layer, one hidden layer of size 16, one hidden layer of size 8 and a final output layer. The input to the neural network is a 6-dimensional reachability determinant $\mathbf{a} = (x, y, z, a_x, a_y, a_z)$, and the output is a binary classification result, where 1 denotes the grasp pose is reachable.

To integrate these two networks, we embed the grasping reachability into the grasping success probability by multiplying the two results. The final output $p_g$ indicates the grasping success probability of $\mathbf{X}$.

### D. Data Collection and Training

Training 3D CNN with multiple objects is ineffective because their shape varies for every grasp pose, preventing 3D CNN from generalizing object geometries. We used 8 primitive shapes from [1] as our training objects. The self-supervised robot interacts with a single object and collects 60,000 labeled voxel grids to train the 3D CNN and 10,000 data to train the RP. A testing dataset of 1000 data is reserved to evaluate the prediction accuracy of 3D CNN. Fig. 4 compiles this result with respect to the training data size from 1,000 to 60,000. It is clear that the prediction accuracy benefits from large-scale training data. We also show two prediction results of the RP in Fig. 5, as expected, an approaching direction toward the robot itself results in narrower reachable space. Our approach learns to select candidates with more vertical approaching direction as they

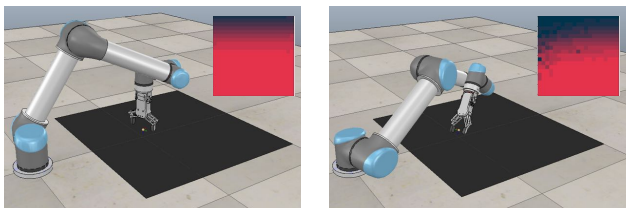
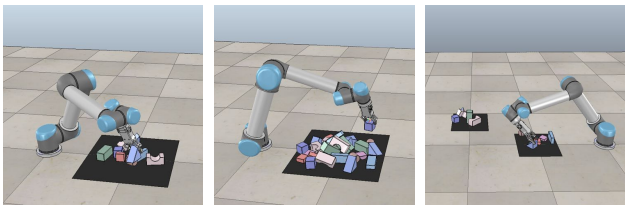


Fig. 5: **Visualization of Reachability Predictor**. We visualize the predictions for two poses over an arbitrary workspace, colored in black. Dark blue indicates poor reachability.



(a) Standard (b) Challenging (c) Ablation

Fig. 6: **Simulation experiments.** Our approach is evaluated in (a) the standard scenario with randomly arranged 10 objects, (b) the challenging scenario with 30 randomly dropped objects and (c) ablation study.

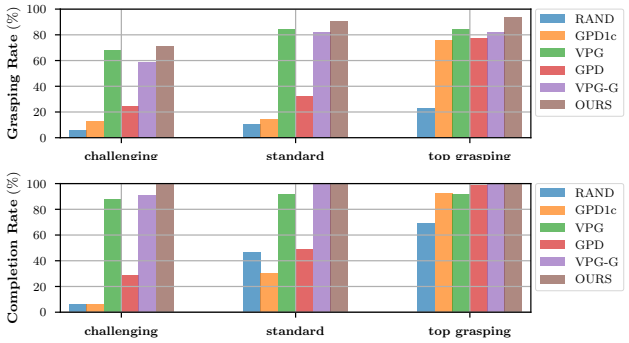



Fig. 7: **Performance in simulation.** The grasping success rate (top) and completion rate (bottom) of each approach in three different settings. The plots clearly show the effectiveness of our approach, which achieves 71.5% grasping success rate in challenging scenario and 100% completion rate in all experiments.

TABLE I: Ablation Study of Reachability Predictor

| | RAND | 3DCNN | OURS |
|---|---|---|---|
| Grasping Success Rate | 13.33 | 32.67 | **82.67** |
| Planning Success Rate | 26.67 | 37.33 | **96.00** |

are generally more stable and reachable in our training and testing environments.

## V. EXPERIMENTS

Our experiments aim to answer the following four questions: i) How much does our approach outperform other approaches? ii) Can our approach generalize to cluttered objects and novel objects? iii) Is the reachability predictor effectively targeting reachable objects? iv) Do our networks trained in simulation work well in real robot?

To answer these questions, we conducted experiments in both simulated and real settings. We compare our approach with 3 other approaches: 1) **RAND**, a baseline that randomly generates a grasp pose without any learning, 2) **GPD**, Grasp Pose Detection proposes 6-DoF grasp candidates based-on geometric reasoning and evaluates them with RGB-D images trained CNN [2] and 3) **VPG**, Visual Pushing for Grasping [1] learns the synergy between pushing and grasping with a reinforcement learning to clean cluttered objects.

### A. Simulation Experiments

The simulation environment is built in V-REP [28] with Bullet [29] Physics engine 2.83. It includes a UR5 robot arm equipped with an RG2 gripper. We noticed **GPD** under this one camera setup (**GPD1c**) can not perform well as their geometric-based grasp sampling suffers from partial point cloud. Another camera was added for **GPD** to reproduce their complete setup. The experiments are illustrated in Fig. 6. For all the experiments, a grasp is successful only if the object is lifted by 15 cm. If the robot failed 10 times (including planning failures) consecutively or all the objects have been removed, this run is completed.

We adopt the same testing setup as reported in [1] for a fair comparison. In a standard multiple objects grasping scenario. The goal is to grasp 10 objects that are randomly dropped to the center of the ground. We notice that the RP effectively penalizes approaching directions toward the robot when grasping point exceeds its learned thresholds, as shown in Fig. 5. To fairly compare our approach and **GPD** with **VPG**, we also report the top grasping success rate for each method. An interesting observation is that the performance of **GPD** improves significantly as top-down grasping poses are mostly stable and reachable in this setting. This indirectly corroborates the importance of reachability awareness when proposing 6-DoF poses. The challenging scenario compares our approach to the others in a densely cluttered setting where 30 objects are randomly dropped to the center. This triples the workspace density thus demonstrates our ability to generalize to more cluttered scenarios.

Fig. 7 presents the average grasping success rates and completion rates over 30 runs for each method, where grasping success rate $= \frac{\text{number of successful grasps}}{\text{number of proposed grasps}}$ and completion rate $= \frac{\text{number of objects grasped}}{\text{total number of objects}}$. With only one camera and no help of pushing to declutter the challenging scenario, our approach achieved the highest grasping success rate and completion rate. Our approach suggests feasible grasp poses as long as the object is within the view, while **VPG** may accidentally force objects out of its workspace.

We report an ablation study of the reachability predictor by grasping from two clusters of objects, one of which is partially unreachable. We tested our approach with (**OURS**) and without (**3DCNN**) the RP to grasp the objects. Given only 5 chances for each run, the robot has to choose the most stable and reachable pose to succeed. We report the average grasping success rate and planning rate over 30 runs for **RAND**, **3DCNN** and **OURS** in Table I, for a total of 150 grasps. Planning efficiency $= \frac{\text{number of successful planning}}{\text{total number of grasping planned}}$. Our


Fig. 8: **Real-world novel objects.** We use five novel objects to show the generalization capability of our approach. The shape, texture and size of the testing objects are different from the training objects in simulation as well.

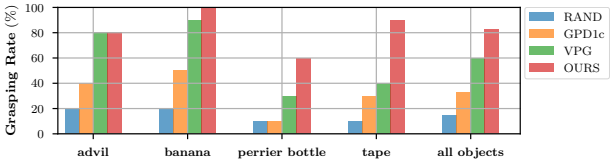



Fig. 9: **Grasping generalizability.** The grasping success rate of four approaches on the 5 test objects. The average grasping success rate of our approach is 82.5%, surpassing all others on novel objects.

reachability aware 3D CNN is able to achieve 96% planning rate, an improvement of 58.67% compared to 3D CNN only.

### *B. Real Robot Experiments*

We directly evaluate our approach on a Franka Emika Panda robot arm without any fine-tuning. Fig. 8 and Fig. 10 show the real robot experiments, which include four scenarios: 1) challenging scenario, 2) ablation study, 3) random household objects and 4) obstacle rich environment.

We compare our approach with **`RAND`**, **`GPD1c`** (**`GPD`** with one camera, due to hardware limitation), and **`VPG`** in the challenging scenario, reproduced from simulation by 30 toy blocks. Table II compiles the performance of each method from averaging the results of 10 runs. Our approach is able to perform consistently in real-world and achieves the highest grasping success rate and completion rate. Despite the robustness of 3D CNN to sensor noise [4], our grasp sampling algorithm may propose a vacant grasping point that contributes to a misaligned pose. **`VPG`** occasionally forces objects out of its workspace and ignores them. **`GPD1c`** suffers from predicting unreachable poses and partial point cloud.

Ablation scenario is reconstructed with toy blocks in the same fashion as in simulation, our approach demonstrates an efficiency improvement as seen previously, but the gap between the simulated UR5 and the real Panda arm deteriorates the performance. Quantitative results are shown in Table III.

To test our real-world generalizability, we further experiment with novel objects, shown in Fig. 8. We ran 10 tests on each object, the result is summarized in Fig. 9. Our approach is able to extract 3D geometric features from novel shapes and select grasp pose candidates accordingly. We noticed that

TABLE II: Challenging Scenario in Real-world

| | RAND | GPD1c | VPG | OURS |
|---|---|---|---|---|
| Grasping Success Rate | 13.10 | 31.23 | 64.76 | **75.20** |
| Completion Rate | 14.58 | 12.75 | 94.33 | **100.0** |

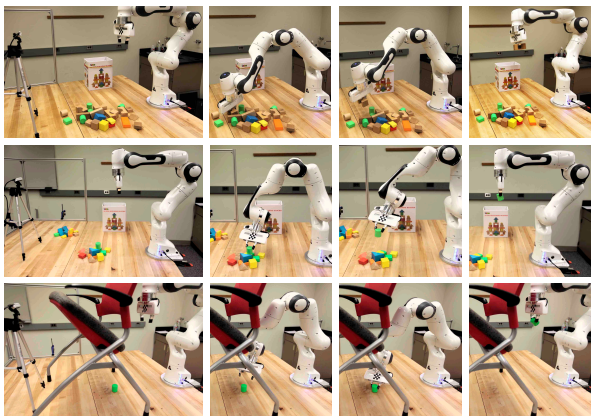


Fig. 10: **Real-world experiment pictures.** The voxel-based 3D CNN approach minimizes the gap between simulation and real-world. Our simulation trained network is able to clean dense clutters (top), differentiate reachable grasps (middle) and grasp an object in a constrained environment which is not feasible by the top grasping approaches, such as VPG (bottom).

TABLE III: Reachability Predictor in Real-world

| | RAND | 3DCNN | OURS |
|---|---|---|---|
| Grasping Success Rate | 3.33 | 23.33 | **66.67** |
| Planning Efficiency | 26.67 | 34.67 | **88.67** |

when grasping considerably larger objects such as the water bottle, our approach prefers smaller shapes such as the bottle neck and cap. This is due to the limited physical voxel grid size that can only fit in shapes that are similar in size to our training objects. We also give an example of our system's grasping flexibility by positioning an object under a chair to reflect real-world challenges. As shown in Fig. 10, the robot is able to complete the task by selecting a feasible grasp which is not possible with the top grasping approaches.

## VI. CONCLUSIONS

In this work, we presented a deep learning approach to generate 6-DoF grasp poses with reachability awareness. A 3D CNN model that estimates grasping stability was trained with a large-scale dataset obtained from simulated self-supervision. A reachability predictor that improves reachability awareness of 3D CNN was trained similarly. Our approach outperformed several comparable deep learning approaches in both simulation and real-world. Furthermore, our method achieved 82.5% grasping success rate on unknown objects. Ablation study showed the RP significantly increased the planning efficiency of 3D CNN by 54% in real-world experiments.

The limitations of our work suggest two directions for future work. First, our point cloud-based sampling algorithm is susceptible to sensor noise. It is of our interest to investigate whether a voxel-based method is able to enhance its robustness. Second, we only applied RP to grasping and would like to explore if RP can help other motion primitives.

## REFERENCES


[1] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[2] 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.

[3] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[4] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.

[5] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

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

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

[8] C. Ferrari and J. F. Canny, "Planning optimal grasps." in *ICRA*, vol. 3, 1992, pp. 2290–2295.

[9] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 557–562.

[10] 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.

[11] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4304–4311, 2015.

[12] 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.

[13] 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.

[14] H. Liang, X. Lou, and C. Choi, "Knowledge induced deep q-network for a slide-to-wall object grasping," *arXiv preprint arXiv:1910.03781*, 2019.

[15] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *IEEE Robotics and Automation Letters*, 2020.

[16] 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.

[17] 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.

[18] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *arXiv preprint arXiv:1806.09266*, 2018.

[19] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," *arXiv preprint arXiv:1810.05687*, 2018.

[20] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[21] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[22] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746–1754.

[23] F. Park and J. W. Kim, "Manipulability of closed kinematic chains," *Journal of mechanical design*, vol. 120, no. 4, pp. 542–548, 1998.

[24] S. M. LaValle, J. J. Kuffner, and Jr., "Rapidly-exploring random trees: Progress and prospects," 2000.

[25] J. J. Kuffner Jr and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *ICRA*, vol. 2, 2000.

[26] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First Iberian Robotics Conference*, M. A. Armada, A. Sanfeliu, and M. Ferre, Eds. Cham: Springer International Publishing, 2014, pp. 703–718.

[27] I. Akinola, J. Varley, B. Chen, and P. K. Allen, "Workspace aware online grasp planning," 2018.

[28] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.

[29] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, ser. SIGGRAPH '15. New York, NY, USA: ACM, 2015. [Online]. Available: http://doi.acm.org/10.1145/2776880.2792704