SuctionNet-1Billion: A Large-Scale Benchmark for Suction Grasping

Hanwen Cao, Hao-Shu Fang, Wenhai Liu and Cewu Lu

Abstract—Suction is an important solution for the long-standing robotic grasping problem. Compared with other kinds of grasping, suction grasping is easier to represent and often more reliable in practice. Though preferred in many scenarios, it is not fully investigated and lacks sufficient training data and evaluation benchmarks. To address that, firstly, we propose a new physical model to analytically evaluate seal formation and wrench resistance of a suction grasping, which are two key aspects of grasp success. Secondly, a two-step methodology is adapted to generate annotations on a large-scale dataset collected in real-world cluttered scenarios. Thirdly, a standard online evaluation system is proposed to evaluate suction poses in continuous operation space, which can benchmark different algorithms fairly without the need of exhaustive labeling. Real-robot experiments are conducted to show that our annotations align well with real world. Meanwhile, we propose a method to predict numerous suction poses from an RGB-D image of a cluttered scene and demonstrate our superiority against several previous methods. Result analyses are further provided to help readers better understand the challenges in this area. Data and source code will be publicly available at www.graspnet.net.

Index Terms—Deep Learning in Grasping and Manipulation, Performance Evaluation and Benchmarking, Perception for Grasping and Manipulation, Computer Vision for Automation

I. INTRODUCTION

Suction grasping is widely used in real-world pick-and-place tasks. Due to its simplicity and reliability, it is often preferred than parallel-jaw or multi-finger grasping in many industrial and daily scenes [1]–[3]. Though effective, it actually draws far less attention from researchers than the other types of grasping [4]–[8].

Recently, data-driven approach has made great progress in robotic grasping [4]. Since learning with real-robot is quite time-consuming, some researchers turned to build simulated environment to obtain large amount of synthetic data [6], [9]. However, learning from synthetic data faces the problem of visual domain gap, as analyzed in Section V-D1. Others tried to collect real-world data and annotate them manually [1]. However, the annotation process requires experience in robotic manipulation and is labor-intensive, especially for dense annotations, causing those datasets to be small in scale. Besides, although previous work [10] trained with with single-object scenes can transfer to cluttered scenes to some extent, later work [2] showed that training on cluttered scenes improved the model (see Section V-C1).

To further boost the development of suction grasping algorithms, a benchmark is needed to i) provide large-scale data with cluttered scenes captured by real-world sensors, ii) have dense annotations which is well aligned with real-robot trials, iii) evaluate suction poses quickly in a unified manner without access to a real robot. This is challenging due to the dilemma mentioned above: domain gap for synthetic data and difficulty to annotate real-world data manually. We address this issue by leveraging the advantage of both sides, which means we collect data using real-world sensors and annotate them by analytic computation with a physical model.

To obtain a large-scale densely annotated suction grasping dataset, we get inspiration from previous literature [4] and propose a two-step annotation pipeline. Specifically, we first generate and annotate suction poses on 88 single objects with their accurate 3D mesh models. Then we utilize objects’ 6D poses to generate dense annotations for totally 190 cluttered scenes by projection and collision detection. Each scene contains 512 RGB-D images captured by two cameras from 256 different views, leading to 97,280 images in total. Contributed
to the effective annotation methodology, we built the first large-scale real-world suction grasping dataset in cluttered scenes with more than one billion suction annotations, which is larger than previous datasets by orders of magnitude. In addition, we provide an online evaluation system to quickly evaluate suction grasping algorithms in a unified manner. We conducted real-world experiments to show that our benchmark aligns well with real-robot manipulations. The pipeline of building our dataset is shown in Fig. 1. We hope the large-scale densely annotated real data as well as the evaluation metric can benefit related research.

With the new large-scale dataset, we propose a new method to predict 6D suction poses directly from single RGB-D image. Firstly, a neural network predicts a pixel-wise seal formation heatmap and a center map for objects from the RGB-D input. Then, we sample high-score suction candidates by grid sampling. Finally, we compute 3D suction points and directions from depth image geometrically with camera intrinsics. The effectiveness of our proposed method is demonstrated by experiments. Previous methods [1], [10] are also evaluated on our benchmark.

The rest of our paper is organized as follows: we firstly introduce previous methods and datasets in Sec. II. Next, in Sec. III, we detail the construction of our benchmark, including our analytical models, annotation procedure and evaluation metrics. Our network is then introduced in Sec. IV. Finally, abundant analyses to our benchmarks and several baseline methods are carried out in Sec. V.

II. RELATED WORK

In this section, we will review some previous suction-based grasping algorithms, previous suction-related datasets and suction analytic models.

A. Suction Datasets

Zeng et al. [1] built a real-world dataset by seeking experts who had experience with suction grasp to annotate actionable and non-actionable areas on raw RGB-D images. However, the labor-intensive procedure and requirement of expertise makes it hard to build large-scale dataset. In the contrast, Mahler et al. [6], [10] turned to simulation and built a synthetic dataset by using 3D models and annotated suction by analytical computing. The dataset contained depth images rendered from those 3D models. They built a much larger dataset but suffered from the problem of domain gap of visual perception and did not have RGB images. [2] also pointed out that methods trained on single object scenes does not transfer well to cluttered scenes. Other datasets [11], [12] are either small in scale or publicly unavailable. In this work, we aim to provide a dataset that is both large in scale and aligns with real visual perception. An online evaluation system is further provided to enable algorithm evaluation without the need to access real robots.

Meanwhile, our dataset contains 6D poses and thus can be used by researchers in 6D pose estimation community [13]–[15]. And these datasets can be used for suction purpose with our annotation pipeline.

B. Suction Analytic Models

In the literature of graphics, Provot et al. [16] modeled deformable objects like cloth and rubber using a spring-mass system with several types springs. Mahler et al. [10] followed this work and modeled the suction cup as a quasi-static spring system. They evaluated whether the suction can form a seal against object surface by estimating the deformation energy. Readers can refer to a review by Ali et al. [17] for more constitutive models of rubber-like materials.

Assuming a seal is formed, several models have been designed to check force equilibrium. Most early works regarded the suction cup as rigid and considered forces along surface normals, tangential forces due to friction and pulling forces due to motion. Bahr et al. [18] and Morrison et al. [12] augmented the model by further considering moment resistance and torsional friction. Mahler et al. [10] combined both torsional friction and moment resistance. We follow the suction model in [10] due to its effectiveness and make some modifications to achieve fast and accurate annotations.

C. Suction-based grasping

Eppner [19] firstly detected objects and then pushed objects from top or side until suction was achieved. Hernandez et al. [11] firstly recognized objects and estimated their 6D poses [20], [21]. They then generated suction candidates on the 3D model of the item based on primitive shapes. Those candidates were finally scored based on geometric and dynamic constraints. Zeng et al. [1] used a neural network trained on a human-labeled dataset to predict grasping affordance map from the RGB-D input and chose the suction point with max affordance value. Morrison et al. [12] firstly obtained semantic segmentation of the scenes and scored the suctions using heuristics like distance from boundary and object curvature. Mahler et al. [10] designed a network to score a suction candidate with the depth image centered at the suction point and suction pose information as input. Correa et al. [22] further proposed a toppling strategy to expose access to robust suction points and increase the suction reliability. Shao et al. [9] proposed an online self-supervised learning method by predicting suction and getting results from a simulated environment.

III. SUCTIONNET-1BILLION

In this section, we provide details of our benchmark and illustrate how we build it.

A. Overview

Previous suction datasets are either small in scale [1] due to labor intensity and expertise required for annotators, or only contain synthetic data [2], [10] which can cause performance decrease in real world due to the domain gap. Also, DexNet3.0 [10] only focuses on single objects, which does not transfer well to cluttered scenes pointed out by [2]. To address those problems, we build a large-scale real-world dataset with cluttered scenes and dense suction pose annotations named SuctionNet-1Billion.
B. Suction Pose Annotation

We define a suction pose as a 3D suction point and a direction vector starting from the suction point and pointing outside of the object surface. Inspired by DexNet 3.0 [10], given a suction pose, we evaluate seal formation and wrench resistance separately. However, instead of labeling suction poses with binary scores, we assign them two continuous scores, namely seal score and wrench score, both in the range of [0, 1]. We define the overall score as the product of these two scores $S = S_{\text{seal}} \times S_{\text{wrench}}$. In the following, we will describe how we compute the scores and present the overall annotation procedure.

1) Seal Score Evaluation: DexNet 3.0 [10] models a suction cup as a quasi-static spring system with topology shown in Fig. 2 left. Specifically, the model has three kinds of springs: 1) Perimeter Springs – those linking vertex $v_i$ to vertex $v_{i+1}$; 2) Cone Springs – those linking vertex $v_i$ to vertex $a$; 3) Flexion Springs – those linking vertex $v_i$ to vertex $v_{i+2}$.

They project the cup along the suction direction to the suction point on the object surface. The length of each spring will change due to the irregularity of the surface. If each spring’s length change is within 10%, seal formation is labeled as positive, otherwise negative.

We modify the annotation process as follows. Firstly, we remove Cone Springs and Flexion Springs and only keep Perimeter Springs since the three types of springs are coupled with each other. Secondly, we model the score as a continuous value instead of a binary one. Define $l_i$ as the length of the perimeter spring linking vertex $v_i$ and vertex $v_{i+1}$. Given the original length $l_i$ and the length after projection $l_i'$, we get the change ratio as:

$$r_i = \min(1, \frac{l_i' - l_i}{l_i}).$$

The spring deformation score is then defined as:

$$S_{\text{deform}} = 1 - \max\{r_1, r_2, ..., r_n\}. \quad (2)$$

Thirdly, to compensate for accuracy loss induced by representing a circle as polygon, we fit a plane for object surface points around the contact ring using least-squares solution and calculate the average of squared errors from the surface points to the plane, denoted as $E_{\text{square}}$. The plane fit score is defined as:

$$S_{\text{fit}} = e^{-cE_{\text{square}}}. \quad (3)$$

Where $c > 0$ is some fixed coefficient. Finally, the seal score is defined as the product of $S_{\text{deform}}$ and $S_{\text{fit}}$:

$$S_{\text{seal}} = S_{\text{deform}} \times S_{\text{fit}}. \quad (4)$$

We conduct real-robot experiments in Sec. V-A to demonstrate that our annotations align well with real world suckings.
TABLE I

| Range | Train | Test-Seen | Test-Similar | Test-Novel |
|-------|-------|-----------|--------------|-----------|
|       | $S_{\text{seal}}$ | $S_{\text{wrench}}$ | $S$ | $S_{\text{seal}}$ | $S_{\text{wrench}}$ | $S$ | $S_{\text{seal}}$ | $S_{\text{wrench}}$ | $S$ |
| (0.0, 0.2) | 35.5% | 24.8% | 55.5% | 35.3% | 23.4% | 53.0% | 27.4% | 18.2% | 41.8% | 72.4% | 24.5% | 82.8% |
| (0.2, 0.4) | 5.5% | 11.7% | 12.0% | 5.5% | 11.5% | 12.2% | 3.7% | 13.6% | 13.9% | 4.5% | 13.2% | 7.0% |
| (0.4, 0.6) | 5.6% | 22.6% | 17.0% | 5.5% | 22.6% | 17.2% | 4.1% | 26.5% | 23.2% | 4.2% | 23.6% | 6.1% |
| (0.6, 0.8) | 9.4% | 28.9% | 13.7% | 9.4% | 28.9% | 13.7% | 9.6% | 28.5% | 16.0% | 4.7% | 26.6% | 3.3% |
| (0.8, 1.0) | 43.9% | 13.9% | 3.6% | 44.4% | 13.5% | 3.8% | 55.3% | 13.3% | 5.2% | 14.2% | 12.1% | 0.8% |

To remove the Friction constraint is that it requires the suction cup
to stick on the object surface while keeping the seal, which is
usually satisfied in suction manipulations. As for the Suction
constraint, it does not contribute to the suction policy because if $V$
is less than the object gravity, then all suctions will fail
no matter where the suction point is. Therefore, we assume
this constraint is always satisfied.

The wrench score is also modified to be continuous in $[0, 1]$. Specifically, we define a torque threshold as $\tau_{\text{thr}} = \pi rk$ and
overall tangential torque as $|\tau_e| = \sqrt{\tau_x^2 + \tau_y^2}$. The wrench
score is defined as:

$$S_{\text{wrench}} = 1 - \min(1, |\tau_e|/\tau_{\text{thr}}).$$

Note that $|\tau_e|$ is actually related to the weight of object but we simply assign the same weight to each object. The weight actually does not contribute to the policy because given
any weight $m$, the gravity wrench is $\tau = mgl$ where $g$ is
the gravity acceleration constant. The policy should try to minimize the force arm $l$ which is dependent on the suction
point and direction (i.e. suction pose) regardless of the weight.
In other words, the optimal suction pose is actually the same for different weights.

TABLE II

| Data | Labels/scene | Obj./scene | Total labels | Total img. | Modal. | Data | CAD |
|------|--------------|------------|--------------|------------|--------|------|-----|
| [10] | 1 | 1 | 6 | 28M | depth | sim. | Yes |
| [4]  | 1 | 1 | 1.2M | 28M | depth | sim. | Yes |
| [1]  | ∼100K | - | 191M | 1837 | rgbd | 1 cam. | No |
| [14] | None | ∼5 | None | 134K | rgbd | 1 cam. | Yes |
| ours | ∼5.1M | ∼10 | 1.1B | 97K | rgbd | 2 cam. | Yes |

3) Overall Procedure:

   a) 6D Pose Annotation: We adopt the 6D pose annotations from GraspNet-1Billion [4]. The annotations are obtained
by propagating the annotated 6D poses of the first frame of each scene using the recorded camera poses:

$$P_i^t = \text{cam}_i^{-1} \text{cam}_0 P_0^i,$$

where $P_i^t$ is the 6D pose of object $j$ at frame $i$ and $\text{cam}_i$ is the camera pose of frame $i$. We refer readers to [4] for more
details about object collection and 6d pose annotation.

   b) Suction Pose Annotation: Following GraspNet-1Billion [4], we adopt the two-step pipeline to annotate suction
poses: annotate single objects and then project to scenes.

Firstly, we sample and annotate suction poses on each single
object. The 3D mesh models are first voxelized (voxel side
length is 5 mm) and suction points with the corresponding
directions (surface normals) are sampled uniformly in voxel space.

Secondly, for each scene, these suction poses are projected using the annotated 6D object poses:

$$P_i^t = \text{cam}_i^{-1} \text{cam}_0 P_0^i,$$

where $\text{P}_i$ is the 6D pose of the $i$-th object in the world frame, $S_i^j$ is a set of suction poses in the object frame and $S_i^j$ contains the corresponding poses in the world frame.

The seal scores can be directly obtained from each single objects since they only depend on local surface geometry while the wrench scores should be annotated per-scene since the wrenches depend on object poses. We also check collision to avoid invalid suctions. Specifically, we model the suction end-effector as a cylinder and check if there are object points in it. After the above three steps, we get dense suction annotations for each scene. The key parts of annotation is illustrated in Fig. 3.

The ratios of each score interval for both Seal Score and Wrench Score are listed in Table I. Seal scores are mostly distributed in the ranges of $[0.0, 0.2]$ and $[0.8, 1.0]$ since it is usually quite clear whether it can form a seal or not while wrench scores are distributed more evenly since suction points are sampled uniformly from the 3D space.

C. Evaluation Metrics

   a) Dataset Split: We use the same split as [4]. Among all 190 scenes, there are 100 for training and 90 for testing.
   The testing set can be divided into 3 categories: 30 scenes with seen objects, 30 with unseen but similar objects and 30 for novel and adversarial objects. As we can see in Table I, the Test-Seen and Test-Similar set share similar distribution with the Train set. The Test Novel set is very hard, where only a small number of suctions get high scores.

   b) Metrics: Given a suction pose, we first associate it with the target object by checking which object is closest
to the suction point. Then the evaluation process is similar
with annotation. After computing the seal score $S_{\text{seal}}$ and
wrench score $S_{\text{wrench}}$ separately, the final score is obtained
by production $S = S_{\text{seal}} \times S_{\text{wrench}}$.

Given a cluttered scene, suction pose prediction algorithms should predict multiple grasps. Following [4], we use Precision@k as our metrics, which measures the precision of top-$k$ ranked suctions. Since the evaluation score is continuous instead of binary, we define several score thresholds to decide
whether a suction is positive, that is, if the score of a suction is larger than a threshold \( s \), then it is regarded as positive under threshold \( s \). AP\(_s\) denotes the average of Precision@\( k \) under the threshold \( s \), where \( k \) ranges from 1 to 50. Similar to previous works [4], AP\(_s\) under different threshold \( s \) is reported. We also report the overall AP as the average of AP\(_s\) with \( s \) ranging from 0.2 to 0.8, with an interval \( \delta s = 0.2 \).

To prevent hacking on test set by predicting many similar suction poses and encourage diverse suction poses, we run Non-Maximal Suppression (NMS) [4] before evaluation and restrict the maximal number of suction on a single object to 10.

Since we usually perform the best suction candidate given any scenes, we also report AP and AP\(_s\) of the top-1 suction pose (which only calculates Precision@1), denoted as AP-top1 and AP\(_s\)-top1.

D. Summary

As mentioned before, real-world data and dense annotations contradict to each other to some extent but thanks to our annotation system and data collected by [4], we build a real-world dataset with much larger scale and diversity. Comparison with other publicly available suction dataset is shown in table II.

As for evaluation, continuity in suction space leads to infinite number of possible suction candidates. Therefore, no matter by human annotation [1] or simulation, it is unlikely to cover all possible suction predictions. To this end, following [4], we do not provide pre-computed labels for test set, but directly evaluate them by computing scores with our evaluation system. The API will be made publicly available.

A robot system can basically be separated into two parts: perception part and control part. Our benchmark falls into the first category, focusing on the problem of suction pose prediction from visual input and leaving out the control part. We make our annotations irrelevant to real robotic design choices as much as we can. Although the data comes from the previous work [4], due to the different topology and working mechanism of the end-effectors, the annotation algorithm is totally different. As mentioned in Sec. I, suction has several advantages compared with parallel-jaw grasping but lacks enough investigation, we hope our benchmark can help researchers design their suction prediction algorithms and compare with others fairly.

IV. Method

In this section, we will describe our end-to-end suction pose prediction framework. We will first introduce the problem formulation and then describe our framework pipeline.

### A. Problem Formulation

To let a robot perform a suction grasping, we need to tell it where to place the suction cup and how it should approach the object. Therefore, we define a suction grasping pose as:

\[
S = [p, u],
\]

where \( p \) is the suction point, a.k.a., the center of contact ring between suction cup and object and \( u \) is the direction vector pointing from the suction point outside the object surface.

We define the prediction task as, given a RGB-D image of a cluttered scene from a specific viewpoint, the algorithm should predict a set of reasonable suction poses \( \mathbb{S} \). Based on that, our framework is formulated as follows.

#### B. Framework

We assume the camera intrinsics are known so we can project a 2d pixel coordinate to its 3D point. Our pipeline is illustrated in Figure 4, which is composed of 3 steps: 1) 2D heatmap prediction, which predicts pixel-wise suction scores; 2) sampling a fixed number of suction points; 3) suction direction estimation.

![Fig. 4. Our proposed pipeline. Given an RGB-D image, our network predicts a seal score heatmap and center score heatmap separately. The two heatmaps are then multiplied to get a final score map. We perform grid sampling on the final score and project the sampled pixels to 3D space. Suction directions are simply obtained by estimating the normals of corresponding points.](image-url)
heatmap around each projected pixel. As for wrench heatmap label, inspired by CenterNet [25], we draw a center heatmap to approximate the wrench heatmap for efficiency. The network is trained with MSE (Mean Squared Error) loss function.

b) Downsampling Suctions: The goal of our framework is to predict a fixed number of suction poses for each scene. However, the predicted heatmap is too dense to directly recover suction poses from pixels. Therefore, we propose a simple but effective sampling strategy. Specifically, we first divide the image to several regular grids and sample the suction with the highest predicted score within each grid. We then rank these sampled suction proposals and output the pixel coordinates of the top 1024 suctions, which can be further converted to 3D points with the camera intrinsics.

c) Suction Direction Estimation: We simply use the surface normals of the suction points as direction. Specifically, we first create point cloud from the depth image and use Open3D [26] to directly estimate normals.

![Fig. 5. The qualitative results of our method. Scenes are constructed from RGB-D image using the camera intrinsics. Suctions are represented by blue cylinders.](image)

V. EXPERIMENTS

In this section, we will first show that our ground-truth annotations align well with real-world suction through robotic experiments and then compare several representative methods with our method using the unified metrics introduced in Sec. III-C.

A. Annotation Evaluation

We conduct real-robot experiments to show that our Seal Score annotations align well with the real world. The Wrench Score annotations are omitted because we make it a relative score to remove the dependence on specific object weights. We paste ArUco code on the objects to get their 6D poses in our experiment.

For the hardware setting, we use a common UR5 robotic arm with an Intel RealSense D435 RGB-D camera. 12 objects are picked from our dataset (shown in Fig. 6). To verify the consistency of our annotation and the real-world suction success rate, we choose the annotated suction poses from three representative score ranges: [0,0.2], [0.4,0.6], [0.8,1.0] and conduct real-world suction. For each score range, 25 suction poses are picked randomly. The suction success rates and the average annotation scores of executed suction poses are shown in Table III. As we can see, the success rates align well with the scores.

B. Baseline Introduction

We evaluate three baseline methods listed below:

a) Normal STD: This method estimates the corresponding normal for each pixel and compute the standard deviation (STD) $\sigma$ within a local patch. The $\sigma$ is normalized to $[0,1]$ by dividing the maximal standard deviation value in the image. The final score for each pixel is computed as $S = 1 - \sigma$. A bounding box indicating the location of the objects is used as an additional input otherwise most predicted suction points will lie on the flat table.

b) DexNet 3.0: Mahler et al. [10] trained a network named GQCNN to predict a score given the suction pose and the depth image centered at the suction point. To get suction poses, they first uniformly sample initial suction poses, predict their scores and finally post process the suctions with Cross Entropy Method. We directly evaluate their pretrained model on our benchmark.

c) Fully Convolutional Network: Zeng et al. [1] propose to use a fully convolutional network to predict heatmap given a RGB-D image. Specifically, they use two branches of ResNet-101 [24] to learn from RGB image and depth image separately. The features from two branches are then fused to predict suction score. Finally, the predicted scores are upsampled bilinearly to original resolution. We use the same training data, loss function, downsampling and normal estimation steps as ours, mentioned in Sec. IV-B, to modify the network to our setting.

![Fig. 6. objects we use to do real-robot evaluation.](image)
C. Results and Analysis

The results are listed in Table IV. We provide some analysis below.

1) Performances of Different Algorithms: As we can see, our method performs the best among all the methods even without additional object bounding box as input. DexNet 3.0 [10] performs worse than [1] and our method. There can be two reasons. The first one is due to domain gap as shown in Section V-D1. The second is that DexNet 3.0 was trained with single objects so it had difficulty transferring to multi-object scenes. The evidence of the difficulty can be found in Figure 3C of DexNet 4.0 [2], the network trained with 100 objects and 2.5k heaps has a higher reliability compared with the one trained with 1.6k objects and 100 heaps, especially in level 2.

2) Performances on Different Splits: As we can see, performances of different algorithms on the Test-Similar set is generally the highest. The reason is that this set contains objects easier to perform suction on, just as indicated in Table I. Though simpler, the set is still suitable to evaluate algorithms’ ability of generalization, which is indicated by the relative performance difference between the Normal STD method and our neural network. Since the Normal STD method does not have the problem of generalization, less advantage of our method indicates the need of generalization. The performances on the Test Novel set are all quite low. The reason is that the set contains new and adversarial objects with few proper suction points, as shown in Table I.

3) Top-1 performance: We can see from IV that our method also outperforms previous methods under the Top-1 metric. An exception is that our network fails to outperform the traditional Normal STD method on the Similar split with Kinect. The reason is that Kinect camera has a wide view so the objects look small in the images, with an additional bounding box indicating where the objects are, the traditional method does not face the problem. In practice, we can eliminate such problem by cropping the images.

D. Ablation Study

1) Domain Gap Analysis: We analyze the domain gap of both RGB images and depth images. To show the domain gap of RGB images, since our network is trained with real RGB-D images, we further test it on the synthetic RGB with real depth images and see how much the performance will decrease, which can give us the same insight with training on synthetic data and testing on real data. To show the domain gap of depth images, we compare the performance difference of DexNet 3.0 [10] since it only takes depth images as input.

Table V shows the results on the similar split. There do exist domain gaps with regard to the two different inputs, especially for RGB images. The results of depth images are relatively small but the gap will increase if we only compare the top 1 suction poses.

Examples of the depth images and RGB images can be found in Figure 7.

2) Different data scale: To show the necessity of the large data scale, we test the results of training with different data scale: 1/10, 1/3 and full dataset. The results can be found in Table VI. The performance increases significantly with the increase of data scale.

E. Real-robot Experiments

To show the effectiveness of our overall system, we deploy our network and DexNet 3.0 [10] separately on a UR5 robot arm and compare their performances. To get a comprehensive comparison, we adopt three different metrics. The first is the ratio of the number of successful grasps to the number total grasps, denoted as $R_{\text{grasp}}$; the second is the ratio of the number of successfully cleared objects to the number of total objects, denoted as $R_{\text{object}}$; the last one is the ratio of the number of successfully cleared objects to the number of total
grasps on those objects (the number of total grasps minus the grasps on the failed objects), denoted as $R_{\text{mixture}}$. We judge if the experiment terminates by observing if the robot fails to grasp any object in consecutive 3 times. As shown in Table VIII, our network outperforms DexNet3.0 [10] in terms of all the three metrics.

### Table VIII

| Metrics | DexNet 3.0 [10] | Ours |
|---------|----------------|------|
| $R_{\text{grasp}}$ | 16/25 = 64.00% | 25/31 = 80.65% |
| $R_{\text{object}}$ | 16/20 = 80% | 25/25 = 100% |
| $R_{\text{mixture}}$ | 16/21 = 76.19% | 25/31 = 80.65% |

### VI. Conclusion

In this paper, we provide a large-scale dataset annotated on real-world data of cluttered scenes. To get grid of the need of intensive human labor and expertise, we design a physical model to analytically annotate large numbers of suction poses and adapt a two-step methodology to generate scene annotations from single object annotations. The accuracy of analytical annotations is proved by real-robot experiments. An unified evaluation system is further provided for fair comparison between algorithms. Meanwhile, we propose a pipeline to predict suction poses from an RGB-D image and provide detailed analysis. We hope our work can benefit related research.

### References

[1] A. Zeng, S. Song, K.-T. Yu, E. Doulon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo et al., “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching,” in 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 1–8.

[2] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, “Learning ambidextrous robot grasping policies,” Science Robotics, vol. 4, no. 26, p. eaau4984, 2019.

[3] M. Schwarz, C. Lenz, G. M. Garcia, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, “Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 3347–3354.

[4] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “Grasnet:1-billion: 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.

[5] M. Gou, H. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, “Rgb matters: Learning 7-dof grasp poses on monocular rgb images,” International Conference on Robotics and Automation (ICRA), pp. 1–8, 2021.

[6] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojeda, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” Robotics: Science and Systems (RSS), pp. 1–8, 2017.

[7] A. ten Pas, M. Guazzetti, 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.

[8] 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.

[9] Q. Shao, J. Hu, W. Wang, Y. Fang, W. Liu, J. Qi, and J. Ma, “Suction grasp region prediction using self-supervised learning for object picking in dense clutter,” in 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR). IEEE, 2019, pp. 7–12.