2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Keypoint-GraspNet: Keypoint-based 6-DoF Grasp Generation from the Monocular RGB-D input

Yiye Chen[1], Yunzhi Lin[1], Ruinian Xu[1], and Patricio A. Vela[1]

Abstract— **The success of 6-DoF grasp learning with point cloud input is tempered by the computational costs resulting from their unordered nature and pre-processing needs for reducing the point cloud to a manageable size. These properties lead to failure on small objects with low point cloud cardinality. Instead of point clouds, this manuscript explores grasp generation directly from the RGB-D image input. The approach, called *Keypoint-GraspNet (KGN)*, operates in perception space by detecting projected gripper keypoints in the image, then recovering their SE(3) poses with a P*n*P algorithm. Training of the network involves a synthetic dataset derived from primitive shape objects with known continuous grasp families. Trained with only single-object synthetic data, *Keypoint-GraspNet* achieves superior result on our single-object dataset, comparable performance with state-of-art baselines on a multi-object test set, and outperforms the most competitive baseline on small objects. *Keypoint-GraspNet* is more than 3x faster than tested point cloud methods. Robot experiments show high success rate, demonstrating KGN's practical potential.**

I. Introduction

Determining grasp configurations from visual sensor input is a fundamental problem in robot manipulation. While great progress has been made on the detection of top-down grasps using deep learning [1], [2], [3], general purpose object grasping requires 6-DoF grasp poses for task completion purposes or for more flexibility in challenging situations.

A typical 6-DoF grasp system involves a *grasp generation* module for generating grasp candidates and a *grasp evaluation* module for ranking the candidates regarding successful executability based on estimated grasp quality [4], collision considerations [5], [6], or environemental constraints [7], [8]. Some methods include a *grasp pose refinement* module to enhance the grasp success probability [9], which acts an iterative approach to the grasp synthesis. Being the first step, a good *grasp generation* algorithm is a prerequisite for robust grasping. An ideal grasp candidate set should contain poses that are accurate and diverse to ensure functionality and flexibility in constrained environments.

Based on the premise that 6-DoF grasp detection requires 3D geometric reasoning, the majority of approaces process 3D data. Deep point cloud feature extractors, such as PointNet [10] and PointNet++ [11], support grasp detection from point cloud input [12], [13], [14], [15], [16], [17]. The preference for point clouds is due to their more immediate accessibility compared to other 3D data formats (e.g., meshes, voxels). However, geometric information is lost as point clouds are unordered. Recovering lost structure involves methods such as hierarchical grouping and/or geometry-based sampling [11]. Further, point clouds have poor scaling; computational cost soars as input cardinality increases [18]. Point cloud grasp detection approaches also involve pre-processing to limit point cloud cardinality. Methods include segmentation [13], [12] and downsampling [14], which may introduce segmentation error [17] or discard critical information. Point cloud methods may produce biased results due to point distribution imbalance. Observations in §IV-D show that small objects tend to be ignored.

a) Contribution: This paper explores 6-DoF grasp generation directly from monocular RGB-D (2.5D) input. It proposes **a new single-view grasp generation method, named *Keypoint-GraspNet (KGN)*,** inspired by object pose estimation [19], [20] and advances in top-down SE(2) grasping [3]. The method first uses a convolutional neural network (CNN) to predict the 2D perspective projection of a set of predefined 3D keypoints in the gripper frame. A Perspective-n-Point (P*n*P) algorithm then recovers the 3D grasp poses from established 2D-3D correspondences.

To generate sufficiently rich and dense training data, we construct a primitive shape dataset with continuously parameterized and densely sampled grasp families. The tactic leverages the simple, closed-form nature of primitive object geometry to simplify specification of the feasible grasp pose distribution. Once the distribution is defined, sampling involves generating a uniform cover of the feasible grasp set. This overcomes the sampling insufficiency of randomized sampling methods and related sample-based grasp labeling strategies applied to arbitrary objects [21]. Using primitive shapes also provides a principled way to evaluate both accuracy and diversity of the predicted set. While primitive shapes may seem limiting, PS-CNN [22] observed that common household objects are similar to primitive shapes or contain primitive shape parts. PS-CNN achieves top 6-DOF grasping performance by recognizing primitive shape regions. Its drawback is a time-consuming 3D model fitting process for dense grasp generation. Based on these findings, we conjecture that models trained on primitive shape datasets can extrapolate to realistic objects, which is supported by physical experiment outcomes in §IV-G.

Testing shows that KGN, trained only on single primitive shape data with a coarse covering of the grasp families, "fills in" the grasp pose gaps and achieves competitive performance. Specifically, **KGN outperforms baselines on a single-object test set, and achieves comparable results**

*This work was supported in part by NSF Award #2026611

[1] Y. Chen, Y. Lin, R. Xu, and P.A. Vela are with the School of Electrical and Computer Engineering, and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA. {yychen2019, ylin466, rnx94, pvela}@gatech.edu

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161284

on a multi-object test set to a baseline specifically trained for cluttered environments. It is more than 3-times faster than all baselines, and demonstrates a significant advantage on small objects compared to point cloud methods. Physical experiments verify the transferability of the approach to the embodied, real-world setting.

II. Related Work

Here, the focus is on related 3D grasp detection literature. Top-down grasping literature follows a similar taxonomy.

a) Grasp Generation: Model-based methods estimate the object pose first and then generate grasps [23]. Unfortunately, exact object models are not always available in general purpose grasping. Alternatively, GPG [24] employed a heuristic method based on point cloud geometry analysis at sampled points. Candidate grasp hypotheses at the sampled points depend on the local surface normal and principle curvature directions. Though high quality candidates can be generated and selected with state-of-art evaluators [12], [25], the time consumption is large and the diversity of the hypothesis set is limited (§IV-G). Sampling-based methods suffer from sample sparsity in the 6-DoF grasp space.

Deep learning techniques for 3D data and large scale dataset synthesis [26], [27] motivate data-driven approaches. These include methods based on truncated signed distance functions of the scene using voxels [28], which require multi-view scanning of the scene; and point cloud sampling methods such as 6-DOF GraspNet [13] and Contact-GraspNet [17]. While the grasp generator of [13] was trained as a Variational Auto-encoder (VAE), [27] employed a Generative Adversial Network (GAN) formulation. Grasp proposals can be reconstructed from latent space samples.

Recent approaches merge the generative and the discriminative components to result in end-to-end grasping systems. The ability of PointNets [10], [11] to extract local and global feature from point clouds admits candidate-wise quality regression/classification. GPNet [29] adopts the anchor idea from the object detection research [30]. It first generates the initial proposals based on a discrete set of the grid points, then uses a network to verify the antipodal validity [31], regress the approaching direction, and predict the quality. Several methods [14], [16], [32] predict the per-point grasp confidence and configuration, either in the format of translation and rotation directions [14], [16], or the two contact points with the pitching angle [32]. Extending the idea with discrete orientation classification enables multiple (coarse to fine) grasp predictions about the same point [15].

b) Grasp Evaluation and Refinement: The grasp evaluator aims to eliminate false predictions from the candidate set and rank the hypotheses for the selection. GPD [25] projects the local geometry features to the gripper frame planes and trains a CNN to classify the grasp quality. PointNetGPD [11] upgrades the CNN to PointNet [10] and directly processes the point cloud within the gripper closing area. The GPDs show promising results on heuristically sampled grasps [24]. 6DOF-GraspNet [13] follows a similar idea, but represents the candidate as the union of the object and the gripper point sets. Some work also employs a trained evaluator to improve the grasp poses [13], [9], [33], where the idea is to optimize the estimated quality by refining the initial pose locally within the SE(3) space. However, empirical evidence indicates that the performance of a grasp system depends on the quality of the generated grasps hypotheses, even with a capable evaluator [13]. Hence, **this work focuses on grasp generation instead of grasp evaluation.**

c) Grasp Detection From the 2D/2.5D input: Grasp synthesis from point clouds seems reasonable given that solutions follow the traditional sense-reconstruct-analyze pipeline, while leveraging point set deep learning models [10], [11]. However, the increased time costs with point quantity affects runtimes [18]. Speed issue stem from the unordered nature of point cloud data that loses any implicit or explicit geometry. Offsetting the loss requires more computation in the form of Farthest Point Sampling (FPS) and ball-query-based grouping [11]. The need to reduce point cloud cardinality by either segmenting a local point cloud or downsampling the scene point set incurs additional latency.

RGB-D data, on the other hand, encodes spatial structure that CNN's can be trained to extract. Recent CNN-based top-down grasp detection methods achieve high performance with 2D/2.5D input [3] based on network outputs that recover SE(2) grasps. The RGBD-Grasp method [34] employs a similar heatmap-based approach to 6-DoF grasp learning based on discretization of the grasp orientation space, followed by point-wise processing of top candidates. **This paper continues to investigate 6-DoF grasp generation from 2.5D image input.** It explores keypoints as the intermediate grasp representation to bridge the dimensionality gap. Countinuous keypoint coordinates avoid discretization error and lead to efficient keypoint-based 3D pose recovery.

III. Keypoints-Based 6-DoF Grasp Generation

a) Problem Definition: Given a single-view color and depth image pair (i.e. RGB-D), $I \in \mathbb{R}^{H\times W\times 3}$ and $D \in \mathbb{R}^{H\times W}$, the goal is to generate 6-DoF grasp candidates for grasping a visible target object using a parallel-jaw, or equivalent, gripper. Equivalently, it is to define the function:

$$\mathcal{G} = \{(g, w) \,|\, g \in SE(3), w \in \mathbb{R}^{+}\} = f(I, D),$$

with grasp pose g and gripper open width w outputs.

b) Keypoint Grasp Representation: The operation will include an intermediate grasp representation given by 4 keypoint coordinats with pre-defined gripper frame coordinates $p_g = (p_g^1, p_g^2, p_g^3, p_g^4)$, where $p_g^j \in \mathbb{R}^3$, $j = 1, 2, 3, 4$. As a first step, a CNN named Keypoint-GraspNet will generate image space (projected) keypoint coordinates and open widths:

$$\hat{P}_{2d} = \{\hat{p}_I^i = (\hat{p}_I^{i1}, \hat{p}_I^{i2}, \hat{p}_I^{i3}, \hat{p}_I^{i4})\}_{i=1}^{N}, \quad \{\hat{w}_i\}_{i=1}^{N},$$

where N is the number of the grasp proposals. The second step applies a P*n*P algorithm to recover the gripper frame with respect to the camera frame,

$$\hat{\mathcal{G}} = \{(\hat{g}_i, \hat{w}_i) \,|\, \hat{g}_i = \text{P}n\text{P}(p_g, \hat{p}_I^i, K)\}_{i=1}^{N},$$

given the camera intrinsic matrix K. Four keypoints guarantee a solution from the chosen PnP algorithm [35].

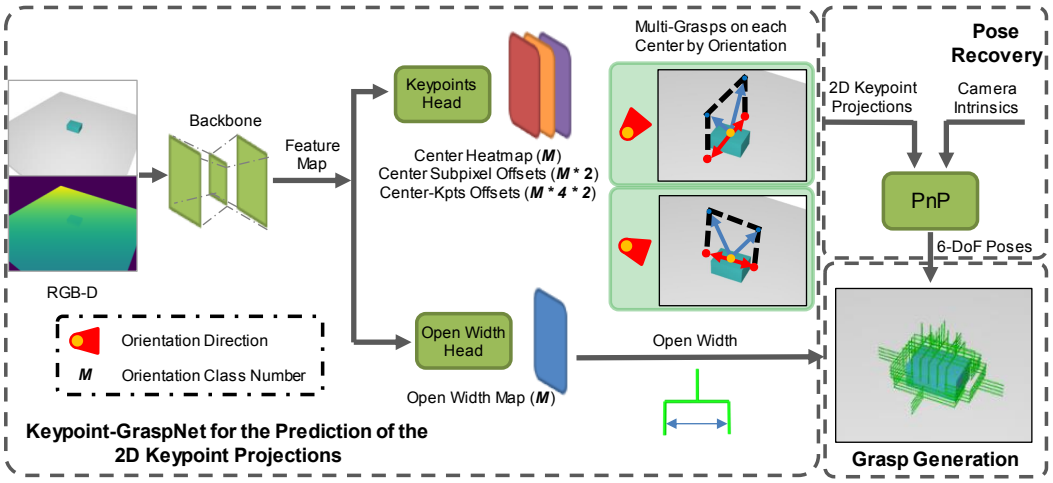


Fig. 1: The method pipeline. The bold values in parentheses are channel dimensions. Keypoint-GraspNet predicts the probability of each pixel being a grasp center. Center prediction is done for each orientation class, to detect multiple grasps with a shared center. For each grasp center, the network regresses to subpixel-level the offsets of four image-space keypoints and the grasp open widths. The outputs recover four 2D keypoint projection coordinates, called *assistant points*, used to recover 6-DoF grasp poses with a P*n*P algorithm.

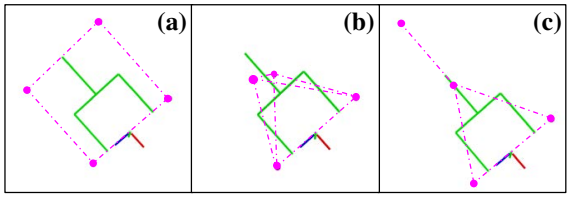


Fig. 2: The gripper frame and keypoint options. The gripper frame is defined at the finger tips midpoint. *Red*, *green*, *blue* axes represent x, y, z directions, respectively. Keypoints are color-coded as magenta. The following keypoint options are considered: **(a)** box-type; **(b)** tetrahedron-type; **(c)** tail-type. Option **(a)** achieves superior performance in (§IV-D).

A. Grasp Representation

Let the 4 keypoints, called *assistant points*, be the corners of a virtual planar square representing the grasp as depicted in Fig. 2(a). As noted earlier, this point quantity suffices to provide a unique solution from P*n*P algorithms [35], [36], [37]. The gripper frame is depicted in Fig. 2, such that the square is in the gripper x-z plane with one edge aligning with the z-axis. Keypoint distances are set to a canonical value l. Alternative keypoint options include a non-planar tetrahedron-type (Fig.2(b)) and a planar tail-type (Fig.2(c)). Experiments in §IV-D indicate that the box-type exhibits the best resilience to the noise for pose estimation.

The keypoint distance l need not equal the grasp open width w. The motivation of this design choice is that when w is small (e.g. the grasps for a thin stick), then P*n*P algorithms applied to a concentrated point set is noise-sensitive. It disentangles pose learning from open width learning in the canonical keypoint distance design.

B. Keypoint-GraspNet

Fig. 1 depicts the processing flow, its input/output structure, and the network Keypoint-GraspNet architecture. The trained network outputs a set of 2D keypoint projection groups P_{2d} with corresponding open widths $\{w\}^N$. RGB-D inputs have better performance over solely RGB or depth, since complementary visual and scale information is available. One challenge is that P_{2d} can be dense and involve overlapping points due to the quantity of grasp candidates per object. We employ a center point [38], [39], defined to be the gripper frame origin in Fig. 2, as the grouping clue.

The network detects keypoint groups relative to the center such that the raw network output includes: (1) The center heatmap $\hat{Y}$ as the per-pixel possibility of being a grasp center; (2) The center subpixel offset map $\hat{O}$; (3) The center-to-keypoint offset maps $\hat{J}$ for the displacement between the center and 4 keypoints; and (4) The open width map $\hat{S}$.

At the inference stage, first select top-k scores from $\hat{Y}$ and remove low values with a fixed threshold ϵ_c, to obtain center coordinates $\hat{c}^i = (\hat{c}^i_x, \hat{c}^i_y)$ of the candidate grasp set. Obtain keypoint coordinates by adding the center-keypoints offsets to the center coordinates: $p^{ij}_I = \hat{c}^i + \hat{J}_{\hat{c}^i j}$, $j = 1, 2, 3, 4$, which leads to the 6-DoF poses using a P*n*P algorithm. Finally, recover the open width prediction by indexing $\hat{S}$ at the center location $\hat{w}_i = \hat{S}_{\hat{c}^i}$. The poses and the corresponding open widths constitute the generated grasp set $\hat{\mathcal{G}}$.

Detecting one grasp per center will omit grasps when multiple occupy the same. To address this problem, we detect M sets of outputs distinguished by the orientation direction of the gripper z axis in the image space. Due to the symmetry of the grasp pose, the range of the orientation is assumed to be $[-\pi/2, \pi/2]$ discretized into M intervals. At the training phase, the ground truth outputs are labeled at the corresponding channel index. The network is trained to produce a set of outputs for each orientation bin, for up to M grasp predictions per center pixel. During inference, candidates with inconsistent orientation angle and class are discarded. Though orientation is quantized, keypoint coordi-

nates are continuous in $\mathbb{R}^2$, which avoids quantization error.

C. Training Loss

The proposed network is end-to-end trainable with the ground truth outputs, which can be synthetically generated (see §IV-A). The heatmap branch is trained using the penalty-reduced binary focal loss [40]:

$$L_Y = -\frac{1}{N}\sum_{xym} \begin{cases} (1-\hat{Y}_{xym})^{\alpha}\log(\hat{Y}_{xym}) & \text{if } Y_{xym}=1 \\ (1-Y_{xym})^{\beta}(\hat{Y}_{xym})^{\alpha}\log(1-\hat{Y}_{xym}) & \text{otherwise} \end{cases}$$

with the choices $\alpha = 2$ and $\beta = 4$ following [38]. The center subpixel offset loss L_O, the center-keypoint offset loss L_J, and the open width loss L_S are formulated as L_1 losses on the ground truth centers. The total loss is the weight sum of each branch loss:

$$L = \gamma_Y L_Y + \gamma_O L_O + \gamma_J L_J + \gamma_S L_S.$$

The weights used were: $\gamma_Y = 1$, $\gamma_O = 1$, $\gamma_J = 1$, $\gamma_S = 10$.

IV. Experiments

A. Dataset Generation

The ground truth training annotations are generated from known grasp poses and camera poses. To train and evaluate KGN, we generate a synthetic dataset based on the primitive shape and grasp family idea [22]. A grasp family is a pre-defined parametric grasp pose space for a shape primitive instance. With the continuous, closed-form grasp pose descriptions, uniformly sample the grasps and create a set covering based on spatial and angular distance thresholds.

Six shape categories are used [22]: *Cylinder*, *Ring*, *Stick*, *Sphere*, *Semi-sphere*, *Cuboid*. For single-object scenes, randomly choose a shape class, and generate an object of random size and color. Place the object on a table with a random stable pose. For grasp annotations, sample object-frame grasp poses from uniform set coverings of the pre-defined grasp families. Grasps causing collision between a virtual gripper model and the table or other objects are removed. For multi-object data, the procedure is repeated to place one object per shape in the scene. Finally, 5 virtual camera poses are sampled for RGB-D image rendering. We generate 1000 single-object scenes following the above procedure, and divide them into an $80\%/20\%$ training/testing split. We also create, 200 multi-object test scenes to examine KGN's generalization ability to cluttered test cases. In total, 4000 single-object training , 1000 single-object testing , and 1000 multi-object testing data instances are generated.

Rationale behind primitive shapes: We first argue for the benefit of primitive shapes for evaluation. Evaluating both the accuracy and comprehensiveness of a predicted set requires annotations with sufficient coverage of grasp space [2], [13], [17]. Recent literature [41], [26], [27] adopts a sample-then-verify process done in simulation. However, sampling leads to biased or incorrect results [21], [3], which negatively impacts training and evaluation.

Sampling problems are caused by the fact that the grasp distribution for an arbitrary (non-primitive) object is complex and has no explicit formula. To tackle the challenge, we restrict the objects to primitive shapes. Due to the simplicity of the object geometry, closed-form expressions of the object's grasp families sufficiently cover the feasible grasp modes. The closed-form nature of the grasp families permits deterministic annotation sampling with a desired density.

Lastly, it is observed that primitive shapes comprise a wide range of common household objects. State-of-art performance has been achieved by explicitly targeting basic geometrical regions [22].

Extrapolation from sparse to dense: Another challenge is the gap between the continuous grasp space and the discrete labels for training. To investigate the effect of training with sparse annotations, we vary the sample density between the training and testing split. The training set uniformly samples 5 and 11 elements from the *translational freedom* and *rotational freedom*, respectively, whereas the testing set samples 10 and 30 elements. The increased density (and stricter error threshold) at the evaluation stage examines the network's ability to learn the underlying distribution.

B. Implementation Details

The backbone network is DLA-34 [42] with deformable convolution layers [43], designed by [38], which produces feature maps one-fourth the input dimensions. Each task head is a shallow network of a 3×3 convolution layer with 256 channels followed by a 1×1 convolution layer to the desired channel dimension. There are $M = 9$ orientation classes.

The network training data is the training split dataset described in §IV-A. The labeled grasps are conditionally flipped (rotated by 180° around the x axis) to consistently point the gripper y axis to the left in the image space. Only one grasp per orientation class per center pixel is kept. The input image data is augmented with random cropping, flipping, and color jittering.

Network training uses the Adam optimizer for 400 epochs. The initial learning rate is $1.25e^{-4}$, reducing by 10x at epochs 350 and 370. For all experiments, the center heatmap threshold at the inference phase is fixed to $\epsilon_c = 0.3$, and the top-k number is set to $k = 100$.

C. Synthetic Dataset Experiment Setup

Evaluation with the test splits of the synthetic dataset uses several metrics. Three point cloud methods are compared with as baselines.

Metrics: The metrics quantify when two grasp poses are similar. Two $SE(3)$ elements g and g^* are similar up to the threshold ϵ_T, ϵ_R if their translational and rotational components satisfy:

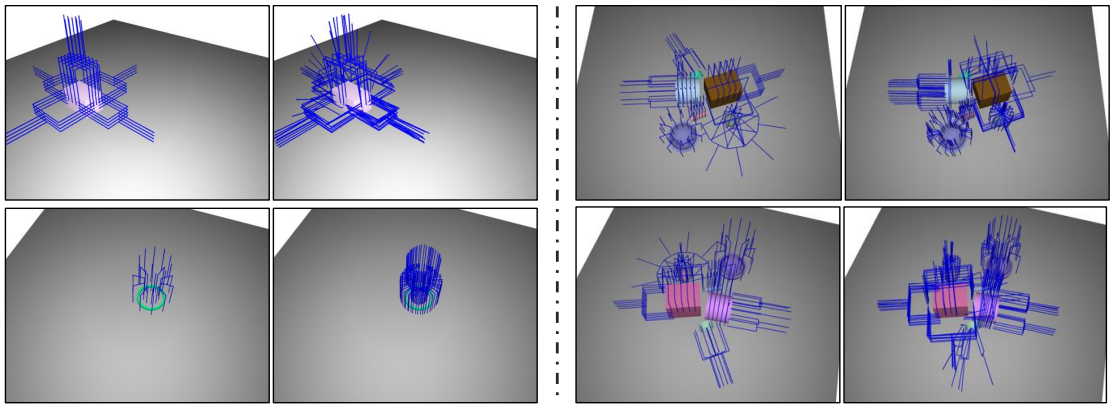
$$d_T(T, T^*) = \|T - T^*\|_2 \leq \epsilon_T$$
$$d_R(R, R^*) = \arccos(\frac{1}{2}\operatorname{Tr}(RR^*) - \frac{1}{2})) \leq \epsilon_R$$

where T and R represent the translation vector and rotation matrix from an $SE(3)$ element. The rotation distance is the minimum angle required to align two rotations [44], [45].

TABLE I: Vision Dataset Evaluation

Methods	Modality	Single-Object Evaluation (GSR% / GCR% / OSR%)			Multi-Object Evaluation (GSR% / GCR% / OSR%)			FPS
		1cm + 20°	2cm + 30°	3cm + 45°	1cm + 20°	2cm + 30°	3cm + 45°	
PointNetGPD	PC	0.43 / 0.13 / 1.50	1.52 / 0.90 / 3.57	20.5 / 4.80 / 16.0	0.00 / 0.00 / 0.00	8.33 / 0.02 / 0.80	41.7 / 0.33 / 3.20	0.008
6DoF-GraspNet	PC	3.78 / 6.78 / 35.4	16.5 / 39.6 / 79.1	35.9 / **73.9** / 97.7	0.20 / 0.10 / 0.70	2.00 / 0.50 / 5.27	8.66 / 2.68 / 16.7	0.240
Contact-Graspnet[1]	PC	29.9 / 24.9 / 77.0	60.1 / 32.0 / 81.7	81.6 / 36.5 / 84.2	**22.1** / **15.5** / **44.1**	**54.2** / **28.5** / 51.4	**78.4** / **34.5** / 54.4	2.109
RGB-Matters	RGB-D	0.74 / 0.12 / 1.9	4.68 / 1.30 / 8.5	21.92 / 6.13 / 20.6	1.36 / 0.22 / 1.65	6.33 / 1.67 / 5.88	26.40 / 6.46 / 16.73	0.409
KGN	RGB-D	**55.5** / **42.9** / **97.0**	**78.5** / **63.3** / **99.6**	**86.9** / 73.2 / **99.9**	10.8 / 5.48 / 28.7	30.6 / 18.7 / **51.8**	49.6 / 33.8 / **62.4**	**9.290**

[1] Contact-GraspNet is trained on cluttered scenes as opposed to other methods. In addition, ground truth object segmentation mask is provided to cluster input point clouds. Hence, it serves as a upper bound for the multi-object evaluation.



Training Labels **KGN** **Training Labels** **KGN**

Fig. 3: Example results on synthetic single-object and multi-object datasets. Trained only on single-object data with sparse annotations, Keypoint-GraspNet predicts dense grasp poses (left) and extrapolates to multi-object scenarios (right).

Given a set of predicted grasps $\mathcal{G}$ and ground truth grasps $\mathcal{G}^*$, three metrics evaluate the predictions: *grasp success rate (GSR)* [13], *grasp coverage rate (GCR)* [13], and *object success rate (OSR)* [2], [3]. The *GSR* is the percentage of the successful predictions to ground truth grasps (i.e., the predicted grasp is similar to a ground truth grasp); The *GCR* is the percentage of the ground truth grasps similar to at least one of predicted grasp; The *OSR* is the percentage of objects with one or more successful predictions. The three metrics measure the accuracy, the diversity, and the practicality of the predicted set.

The visual experiments record the metrics under three distance threshold levels from strict to loose: $(\epsilon_T, \epsilon_R) = (1\text{cm}, 20°)$, $(2\text{cm}, 30°)$, and $(3\text{cm}, 45°)$.

Baselines: Comparison is with three point cloud grasp generation baselines, namely *PointNetGPD* [12], *6DoF-GraspNet* [13], and *Contact-GraspNet* [17]; and one RGB-D method *RGB-Matters* [34], The first three choices cover distinct pipelines: heuristic grasp sampling followed by ranking with learnt evaluator [12]; generator-based grasp synthesis selected and refined by a trained evaluator [13]; and the per-point grasp scoring and parameter regression [17]. For these baselines, we remove the tabletop points to reduce the input point cloud cardinality, which is not required by KGN. The generator of 6DoF-GraspNet is trained from scratch on our training set. Pretrained weights are used for the other modules from the baselines due to their hardware requirements. The workstation used has an NVIDIA GTX 1080 GPU and an Intel i7-7700 CPU @ 3.60GHz.

D. Synthetic Dataset Experiment Results

Single-Object Evaluation: Results from KGN and the baselines on the single-object test set are shown in Table I. KGN outperforms the baselines under all distance thresholds and metrics except for the GCR at the largest error tolerance. The grasp success rate gap is consistantly over 45%, and the grasp coverage rate gap is over 30% at the strictest error tolerance level. Although trained with a coarse set covering for each grasp family, the high GCR outcomes with a denser set covering indicates that KGN captures the underlying grasp distribution shown in Fig. 3.

Multi-Object Evaluation: Next KGN *trained only with single-object data* is applied to the multi-object test set. We observe that the predicted *scale* (i.e. translation magnitude along camera projection) deteriorates in this setting, as the keypoints proximity in the image space is unstable in the presence of visual disturbance. To compensate, we added a depth-based scale refinement: $T = T\|T\|_2/D_c$, where T is predicted camera-to-grasp translation, and D_c is the depth image value at predicted grasp center. Scale prediction is an known deficiency in monovular pose estimation from 2D image points. Exploration is needed on how the depth channel can best help to resolve scale. Meanwhile, Table I collects the test results. KGN continues to outperform PointNetGPD and 6DoF-GraspNet. It is second best to Contact-GraspNet, which is *explicitly trained for cluttered environments*. Factors influencing the KGN outcomes are the fixed confident threshold and upper limit on number of grasp

TABLE II: Performance on Small Objects

Methods	Stick	Ring
	*Avg GCR% / OSR%	*Avg GCR% / OSR%
Contact-GraspNet	13.8 / 7.10	16.1 / 15.8
KGN	**37.9 / 22.2**	**45.7 / 36.3**

* Numbers averaged over three error tolerance levels.

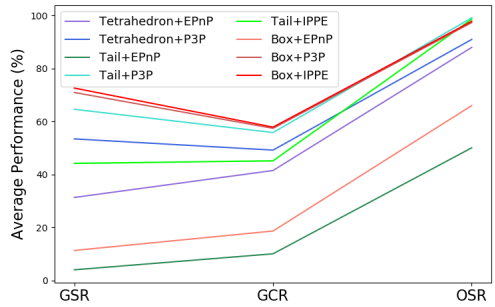


Fig. 4: Ablation study on keypoint and P*n*P options with single-object evaluation. Numbers are averaged over all error levels. The Box+IPPE combination yields the best results.

predictions for more objects.

Advantage for Small Objects: Although Contact-GraspNet outperforms KGN in the multi-object test, KGN is more effective on small objects. Table II reports the GCR and OSR for the two smallest objects, namely the *stick* and the *ring*, which have the the smallest number of occupied observable pixels on average. KGN performs better than Contact-GraspNet. A low OSR indicates that the point cloud method tends to ignore small objects due to relative paucity of points versus larger objects.

E. Abalation Study on Keypoint Options

Testing across keypoint grouping geometry and P*n*P algorithm options on the single-object test set in Sec.IV-D leads to the outcomes in Fig. 4. Per Fig. 2, the keypoint types are: box, tetrahedron, and tail. The P*n*P algorithms tested are: P3P [36], EP*n*P[37], and IPPE [35]. The results indicate that the combination of planar box-type keypoints with the IPPE algorithm is the best option.

F. Advantage Highlights

We summarize the benefits of our RGB-D approach over point cloud methods observed from the experiments:

Preprocessing free: KGN directly processes the input data. In contrast, point cloud methods require preprocessing such as background removal to reduce input point number.

Speed: Although all baselines have an advantage from excluding segmentation timing, KGN still achieves the lowest times and is over 3x faster than point cloud methods using pre-processed inputs.

Superior performance on small objects: Point cloud approaches tend to have poor performance on small objects due to point cloud paucity. With the aid of RGB visual cues, KGN achieves higher performance.

G. Physical Experiments

Physical experiments are conducted to validate the grasp generation approach under the covariate shift and real-world sensor measurements. They involve single object pick-and-place experiments, where the goal is to pick up a randomly placed target and drop it at the designated location. A trial is successful if the object is moved to the spot. The *success rate*–defined as the ratio of the successful trials to total trials–is the evaluation criteria. We adopt 8 common household objects that covering the primitive shape classes as shown in Fig. 5. For each object, 5 trials are performed.

TABLE III: Physical Experiment Results

Object	Succ	Object	Succ
Ball	4/5	Cable	5/5
Tape	5/5	Box	5/5
Bowl	4/5	Mug	4/5
ToothBrush	5/5	Clamp	3/5
Mean	87.5%		

Fig. 5: Physical object set

A simple grasp candidate ranking and refinement process is employed. For the ranking, we score the candidates combining the center confidence in the keypoint detection stage and the reprojection error (RE) induced by the pose recovery stage: $s(g) = Y_{xym} + RE$. The RE, defined as the image distance between the input and estimated 2D keypoints, quantifies the uncertainty of the P*n*P algorithm. We refine the translation scale based on the observed depth from the depth image as per Sec. IV-D.

The physical experiment results are shown in Table III. Trained on a simple synthetic dataset, KGN achieves satisfying success rate on real objects, indicating its potential. For comparison, 6-DOF GraspNet had an 88% success rate for simple objects (box, cylinder, bowl, and mug) similar to KGN, but involved more extensive and computationally involved physics-based simulation. The primitive shapes with continuously parametrized models achieve similar performance with a fraction of the effort and with a faster runtime.

V. Conclusion

This paper investigated a keypoint-based solution to the problem of 6-DoF grasp generation from RGB-D input. Named *Keypoint-GraspNet (KGN)* the proposed solution detects image space gripper keypoint projections from which it recovers the 6-DoF grasp pose using a P*n*P algorithm. On a synthesized primitive shape dataset, KGN is shown to generate diverse and accurate grasp candidates. It improves grasp performance for small objects and has lower runtime costs compared to the selected baselines. Physical experiments show that the trained model applies to real-world sensors and manipulators without further finetuning.

Improvements can be made to this proof-of-concept work. The first is to explore pose scale estimation more robust to novel visual contents and object spacing. Furthermore, the PS dataset labels grasp poses in a gripper-agnostic manner to examine KGN's abilty to approximate distribution in SE(3) space. Future work should explore modifying the training dataset with labels verified by simulation for specific gripper geometry. Lastly, applying KGN from multiple views might improve the grasp pose accuracy around occluded areas.

REFERENCES

[1] 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] 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.

[3] R. Xu, F.-J. Chu, and P. A. Vela, "GKNet: Grasp keypoint network for grasp candidates detection," *The International Journal of Robotics Research*, vol. 41, no. 4, pp. 361–389, 2022.

[4] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-Dof grasping interaction via deep geometry-aware 3D representations," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 3766–3773.

[5] X. Lou, Y. Yang, and C. Choi, "Collision-aware target-driven object grasping in constrained environments," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 6364–6370.

[6] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object rearrangement using learned implicit collision functions," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 6010–6017.

[7] W. Yang, C. Paxton, A. Mousavian, Y.-W. Chao, M. Cakmak, and D. Fox, "Reactive human-to-robot handovers of arbitrary objects," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 3118–3124.

[8] C. C. B. Viturino, D. M. de Oliveira, A. G. S. Conceição, and U. Junior, "6D robotic grasping system using convolutional neural networks and adaptive artificial potential fields with orientation control," in *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*. IEEE, 2021, pp. 144–149.

[9] Y. Zhou and K. Hauser, "6Dof grasp planning by optimizing a deep learning scoring function," in *Robotics: Science and Systems Workshop on Revisiting Contact-turning a Problem into a Solution*, vol. 2, 2017, p. 6.

[10] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: Detecting grasp configurations from point sets," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3629–3635.

[13] A. Mousavian, C. Eppner, and D. Fox, "6-Dof graspNet: Variational grasp generation for object manipulation," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[14] 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.

[15] K.-Y. Jeng, Y.-C. Liu, Z. Y. Liu, J.-W. Wang, Y.-L. Chang, H.-T. Su, and W. Hsu, "GDN: A coarse-to-fine (c2f) representation for end-to-end 6-Dof grasp detection," in *Conference on Robot Learning*. PMLR, 2020, pp. 220–231.

[16] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "PointNet++ grasping: learning an end-to-end spatial grasp generation algorithm from sparse point clouds," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 3619–3625.

[17] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-Dof grasp generation in cluttered scenes," *IEEE International Conference on Robotics and Automation*, 2021.

[18] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-Net: Efficient semantic segmentation of large-scale point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

[19] S. Huang, Y. Chen, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu, "PerspectiveNet: 3D object detection from a single RGB image via perspective points," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-stage keypoint-based category-level object pose estimation from an RGB image," in *IEEE International Conference on Robotics and Automation*, 2022.

[21] 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," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 890–905.

[22] Y. Lin, C. Tang, F.-J. Chu, R. Xu, and P. A. Vela, "Primitive shape recognition for object grasping," *arXiv preprint arXiv:2201.00956*, 2022.

[23] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1284–1306, 2011.

[24] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3D point clouds," in *Robotics Research*. Springer, 2018, pp. 307–324.

[25] 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.

[26] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 444–11 453.

[27] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 6222–6227.

[28] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 Dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[29] C. Wu, J. Chen, Q. Cao, J. Zhang, Y. Tai, L. Sun, and K. Jia, "Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 174–13 184, 2020.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[31] I.-M. Chen and J. W. Burdick, "Finding antipodal point grasps on irregularly shaped objects," *IEEE transactions on Robotics and Automation*, vol. 9, no. 4, pp. 507–512, 1993.

[32] A. Alliegro, M. Rudorfer, F. Frattin, A. Leonardis, and T. Tommasi, "End-to-end learning to grasp from object point clouds," *arXiv preprint arXiv:2203.05585*, 2022.

[33] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *Robotics Research*. Springer, 2020, pp. 455–472.

[34] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "RGB matters: Learning 7-Dof grasp poses on monocular RGBD images," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 13 459–13 466.

[35] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 252–286, 2014.

[36] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[37] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate O*(n)* solution to the P*n*P problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[38] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019.

[39] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *IEEE/CVF international conference on computer vision*, 2019, pp. 6569–6578.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

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

[42] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2403–2412.

[43] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 764–773.

[44] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6Dof outdoor visual localization in changing conditions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[45] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013.