2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Transferring Experience from Simulation to the Real World for Precise Pick-And-Place Tasks in Highly Cluttered Scenes

Kilian Kleeberger[1], Markus Völk[1], Marius Moosmann[1], Erik Thiessenhusen[1], Florian Roth[1], Richard Bormann[1], Marco F. Huber[2,3]

***Abstract*— In this paper, we introduce a novel learning-based approach for grasping known rigid objects in highly cluttered scenes and precisely placing them based on depth images. Our Placement Quality Network (PQ-Net) estimates the object pose and the quality for each automatically generated grasp pose for multiple objects simultaneously at 92 fps in a single forward pass of a neural network. All grasping and placement trials are executed in a physics simulation and the gained experience is transferred to the real world using domain randomization. We demonstrate that our policy successfully transfers to the real world. PQ-Net outperforms other model-free approaches in terms of grasping success rate and automatically scales to new objects of arbitrary symmetry without any human intervention.**

I. Introduction

For robots to work safely and effectively, they must be aware of their environment. One aspect of this is the estimation of the pose of the objects in the scene to be able to avoid collisions and allow robust grasping and manipulation of the components. 6D object pose estimation (OPE) and grasp planning in highly cluttered scenes based on a single depth image is challenging because of sensor noise, incomplete information, and uncertainties about the state of the environment. Furthermore, the robot has to reason on how to manipulate the objects because selecting the wrong object and grasp pose can result in failed grasps.

Works such as [1], [2], [3], [4] focus on robotic grasping and manipulation tasks in scenarios with limited clutter which do not require a defined picking order of the objects. Simply selecting collision-free and kinematically feasible grasps in highly cluttered scenes [5], [6], might lead to a movement of the object relative to the gripper which prevents a precise placement without additional in-hand localization and entanglements with other objects for complex object geometries as visualized in Fig. 2. In this paper, we tackle these challenges by providing a novel learning-based approach for grasp pose evaluation in scenes of many parts in bulk.

Approaches to robotic grasping and manipulation usually rely on datasets consisting of human-labeled grasps [2], [3], [7], which are tedious to get, or on physical grasp outcomes where data collection can take several months [8], [9], [10], [11]. In this work, we execute each predefined grasp pose in a physics simulation and transfer the gained experience to the real world using domain randomization [12] to increase generalization. Our approach directly transfers from simulation to the real world (see Fig. 1).

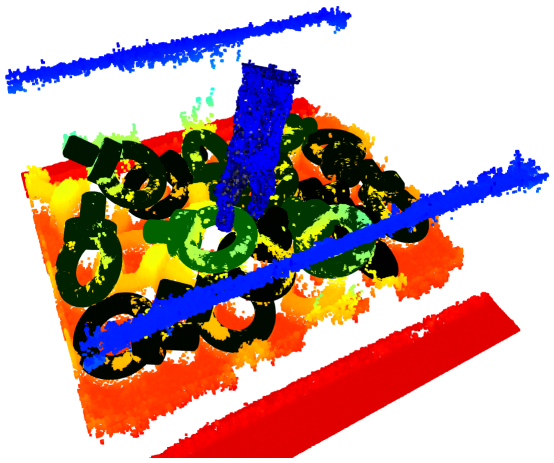


Fig. 1. Estimated object poses of our approach on real-world data after ICP refinement. The greener the object, the more certain the model is that the object can be grasped safely. The gripper (blue) indicates the top ranked grasp based on our policy.

Learning-based approaches to robotic grasping [13], [14], [15], [8], [9], [11] usually rely on top-down grasps and cannot be used for bin-picking due to collisions when attempting to grasp objects close to the border of the bin. Analytical approaches for pick-and-place tasks in cluttered scenes require an object-specific configuration and tuning until a satisfactory system performance is reached, which limits the scalability [5], [16], [17], [18], [19], [20]. PQ-Net configures automatically based on a given object model and does not require any human intervention.

Inspired by the success of single shot approaches [21], [22], [23], [24], we go further and extend OP-Net [25], a single shot approach for OPE outperforming the winner of the "*Object Pose Estimation Challenge for Bin-Picking*" at IROS 2019[1] on the Siléane dataset [26]. To the best of our knowledge, we are the first extending a single shot approach for OPE to grasp success prediction in a joint framework.

Based on a single depth image, PQ-Net predicts the object pose $P \in \mathrm{SE}(3)$ relative to the camera coordinate system and outputs a success estimate for a set of predefined grasps $\mathcal{G}$ defined relative to the object coordinate system.

[1]Department Robot and Assistive Systems, Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstraße 12, 70569 Stuttgart, Germany kilian.kleeberger@ipa.fraunhofer.de

[2]Center for Cyber Cognitive Intelligence (CCI), Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstraße 12, 70569 Stuttgart, Germany marco.huber@ipa.fraunhofer.de

[3]Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, Allmandring 35, 70569 Stuttgart, Germany marco.huber@ieee.org

[1]http://www.bin-picking.ai/en/competition.html

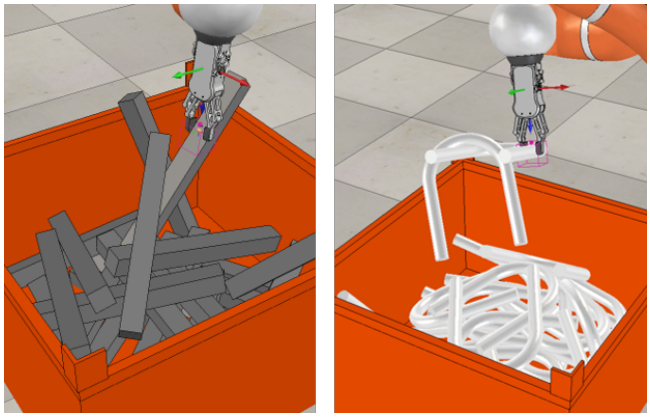


Fig. 2. Failure cases for picking tasks in cluttered scenes: (left) The robot picks an object (IPABar) which moves relative to the gripper during lifting due to overlaps with other objects. The object cannot be placed precisely anymore. The objects in the bin have to be picked in a defined order. (right) The robot picks an object (IPAUBolt) which is entangled with another object. Therefore, a goal is to select objects which do do not entangle with other objects.

Furthermore, we introduce graspability metrics which allow a gentle component removal and avoid entanglements.

In summary, the main contributions of this work are:

- Extension of a single shot approach for object pose estimation to grasp pose success prediction suitable for precise object placement
- Novel metrics for the assessment of the graspability of objects in highly cluttered scenes
- Scalable system, which enables a robot to learn how to place objects precisely on the basis of an object model of arbitrary symmetry (automatic configuration)
- Extension of the Fraunhofer IPA [27] and Siléane [26] datasets with grasp annotations and provide data for two new objects. All datasets are publicly available at http://www.bin-picking.ai/en/dataset.html.

The paper is structured as follows. The next section reviews related work. In Section III the proposed approach is described. Experimental evaluations are provided in Section IV. Pros and cons of our approach are discussed in Section V. The paper closes with a conclusion.

II. Related Work

Methods for robotic grasping can roughly be categorized in analytical and data-driven methods [28], [29], [17].

A. Analytical Approaches

Analytical approaches (model-based approaches) use an object model with predefined grasps. First off, they localize the object in the scene [30]. Based on this, they try to find a collision-free and kinematically feasible path for grasping and placing [5], [6]. Especially for highly cluttered scenes, they require significant effort for manually tuning suitable grasp poses and grasp priorities to reach a satisfactory system performance, limiting the scalability to new objects [5], [16], [17], [18], [19], [20]. Usually, the grasp poses are prioritized independent of the object pose. Furthermore, zones on the object can be specified where no measurement point of the 3D point cloud should be contained in order to pick the candidate next. This can be used to specify a picking order of the localized objects in the scene.

B. Learning-based Approaches

Approaches to Robotic Grasp Detection estimate oriented rectangles in the input image which represent a grasp configuration for parallel jaw grippers [31]. Public datasets are the Cornell Grasping Dataset [2] providing 1,035 manually annotated samples of 280 objects and the Jacquard Dataset [4] with over 50,000 synthetic samples on a large diversity of objects (11,000), each with multiple labeled grasps.

MultiGrasp [3] uses the Cornell dataset to train a neural network to predict oriented rectangles (bounding boxes) in an image together with a confidence and makes local predictions based on global information by discretizing the output in $S \times S$ grid cells. This work led to the YOLO [21], [22] approach for object detection. With a two-stage system that first samples grasp candidates and ranks them using neural networks, Lenz et al. [2] demonstrated that this parameterization (oriented rectangles) can be used for real-world robotic grasping tasks.

GG-CNN [7], [32] predicts a quality and configuration of grasps at every pixel of the input image using a lightweight convolutional neural network trained on the Cornell and Jacquard dataset [4]. The generated antipodal grasps that are executed closed-loop and allow grasping in cluttered scenes and non-static environments.

Dex-Net makes use of large scale synthetic data collection for learning grasping policies for parallel jaw [13], [14] and suction grippers [15] using analytic metrics. The sampled grasps are ranked using a neural network which gets a cropped depth image and grasp candidate as input. Dex-Net observes a local image patch, and is not designed to execute grasps in a defined order or avoid entangled objects due to missing global scene information.

Levine et al. [8], [9] parallelizes the real-word data collection to up to 14 robot and collect 800,000 samples in two months for robust grasping. QT-Opt [11] makes use of reinforcement learning to train robotic grasping and manipulation policies based on self-supervision on real-world systems. Because of the time-consuming and hardware demanding data collection procedure, works such as GraspGAN [33] or RCAN [34] focus on reducing the need of real-world data collection.

While all these aforementioned model-free approaches to robotic grasping show promising results, they do not provide a solution for a precise placement of the objects and only consider pick-and-drop tasks using top-down grasps (4D). Using grasps in this grasp representation has limitations, e.g., for bin-picking due to collisions with the bin when attempting to grasp objects at the border. Therefore, works such as [8], [9], [13], [14], [11], [35], [36] use bins with slanted or no high bin walls to ensure that the top-down

grasps work. Furthermore, it is not possible to blindly move into the bin for data collection due to damaging the gripper because of unknown fill levels of the bins. Picking multiple objects is often also considered as a successful grasp [8], [9], [11].

III. Placement Quality Network

In this section, we describe the routine for automatically generating grasp poses for object models, the process of data generation for training our neural network using a physics simulation, the proposed definitions for the graspability of objects, the parameterization of the network's output, the loss function, the network architecture, and the training procedure together with the technique for a robust transfer of the model from simulation to the real world. Fig. 4 shows an overview of our approach.

A. Automatic Grasp Pose Generation

To avoid the need for manually defining grasp poses $G \in \mathrm{SE}(3)$ on the object model, we provide a method that automatically generates a set of grasp poses $\mathcal{G}$ for common gripper types such as parallel jaw, suction, and magnetic grippers based on a given 3D object model. Each grasp $G \in \mathcal{G}$ is represented by $(R; t) \in \mathrm{SE}(3)$ where $R \in \mathrm{SO}(3)$ and $t \in \mathbb{R}^3$ are the rotation and translation of grasp G.

As a first step of our technique, points are sampled on the surface of the object. For parallel jaw grippers, we check the distance between all pairs of points to verify whether it is smaller than the opening distance of the gripper, filter the candidates using the normal information of the 3D points, discretize the rotation around the straight line between any two points in 20° steps, and finally filter the candidates with a collision check using the gripper model. For suction and magnetic grippers, we sample grasp poses by evaluating the flatness of the object locally. Depending on the shape of the gripper, we define cylinder or cuboid elements, which should and should not contain points of the object model while also taking surface normals into account.

The proposed procedure results in a high number of grasp poses. We make use of unsupervised learning to reduce the amount of data while keeping a high diversity in terms of position and orientation. We apply partitioning around medoids (PAM) [37] clustering to reduce the number of grasps to approximately 500. Fig. 3 exemplary shows automatically generated grasp poses using our technique.

B. Physics Simulation for Data Generation

1) Scene Generation: We use the physics simulation V-REP / CoppeliaSim [38] to create scenes with a high amount of clutter. These scenarios are challenging because the robot has to avoid collisions with other objects in the scene and carefully select which object to pick next. Analogous to the Siléane [26] and Fraunhofer IPA [27] datasets, we drop objects in a random position and orientation above a bin to generate chaotic scenes typical for bin-picking. We save the RGB image, depth image, and segmentation masks together with the visibility $v \in [0, 1]$ and pose $P \in \mathrm{SE}(3)$

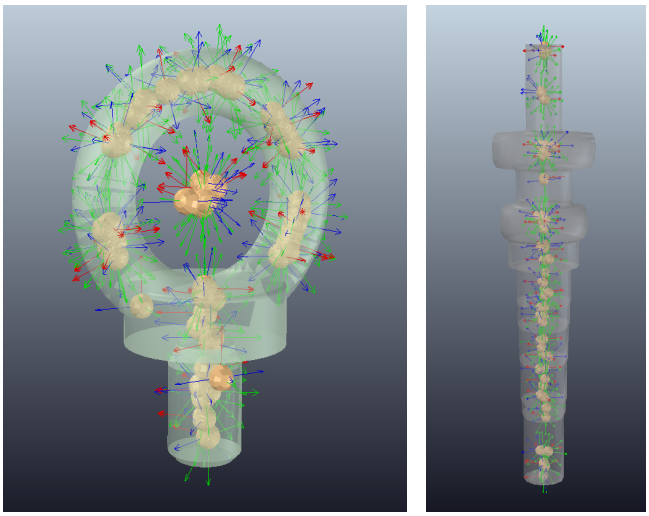


Fig. 3. Automatically generated grasp poses exemplary visualized for the IPARingScrew (left) and IPAGearShaft (right) for a parallel jaw gripper. Objects are taken from the Fraunhofer IPA dataset [27].

for each object in the scene. The number of objects that are dropped is increased iteratively until a predefined drop limit is reached, resulting in a uniform distribution over different fill levels of the bin (see also [27]).

2) Grasping: Using the filled bins, we loop over each (automatically generated) grasp pose for all objects in the scene. First off, we check the collision of the gripper at every grasp pose with the environment (other objects and the bin). In case no collision occurs, we try to find a kinematically feasible robot configuration and plan a collision-free path to the grasp pose using the OMPL [39] module integrated in V-REP / CoppeliaSim [38].

In case a suitable path was found, we execute the grasp and place the object at the defined target pose. We log whether an object is in the gripper after lifting and after placement of the object. Furthermore, we log the pose difference after grasping and lifting the object (chosen grasp pose relative to the gripper TCP) and placement (current object pose relative to defined placement pose). We consider an object as successfully lifted / grasped or placed precisely enough if the distance between the pose representatives based on [40] is less than 0.1 times the diameter of the smallest bounding sphere of the object. This is analogous to the metric used for object pose estimation in computer vision proposed by Brégier et al. [40], [26] and allows to properly consider all possible kinds of object symmetry. Furthermore, we log for each grasp pose whether an entanglement with other objects in the scene occurred.

Since the grasp poses are defined relative to the object coordinate system, pose ambiguities due to object symmetries result in convergence issues during neural notwork training. To avoid this, we introduce a unique object pose definition. For discrete symmetries, we ensure to pick the pose where the z-component of a non-symmetry axis (x- or y-axis) is maximal (assuming the axis of symmetry is the z-axis). For

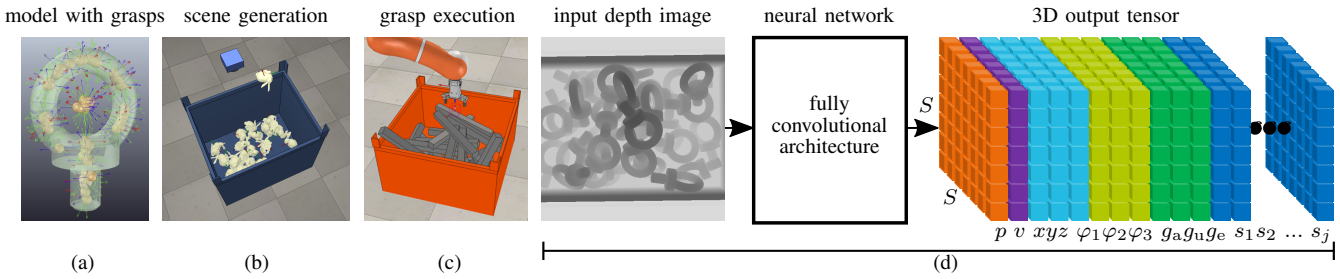


Fig. 4. Overview of our approach: (a) 3D object model with automatically generated grasp poses (b) Physics simulation for scene generation (c) Physics simulation for grasp execution with a robot (d) We train a deep neural network for 6D object pose estimation and grasp pose success estimation to transfer the knowledge gained in simulation to the real world using domain randomization [12]. The output of our network is a 3D tensor comprising estimates of the probability $\hat{p}$, visibility $\hat{v}$, positions $\hat{x}$, $\hat{y}$, $\hat{z}$, Euler angles $\hat{\varphi}_1$, $\hat{\varphi}_2$, $\hat{\varphi}_3$, graspabilies $\hat{g}_\mathrm{a}$, $\hat{g}_\mathrm{u}$, $\hat{g}_\mathrm{e}$, and success $\hat{s}_j$ for each grasp pose $G_j \in \mathcal{G}$.

continuous symmetries, we maximize the z-component of a non-symmetry axis by rotating around the axis of symmetry.

C. Graspability Metrics

Using the results from all executed grasps, we define instance based metrics to assess the graspability of each object in the scene. The graspability of an object based on the accessibility of the grasp poses $g_\mathrm{a} \in [0,1]$ is defined as the ratio between the number of collision-free grasps and the number of total grasps J. Fig. 5 (a) shows the ground truth labels g_a for the IPARingScrew. Some fully visible objects (which are easy to localize) cannot be grasped because other objects are in the way, demonstrating that visibility and graspability of objects are not fully correlated.

For a removal that is gentle on the components and to avoid movements of the grasped object relative to the gripper due to overlapping objects, entanglements, or jams preventing a precise object placement, we log the movement in terms of position $\mathbf{x} \in \mathbb{R}^3$ of all other objects in the scene before grasping (t_0) and after lifting (t_1). This information is used to define the graspability of object k based on the unrest caused in the bin during grasping ("mikado metric")

$$g_{\mathrm{u},k} = 1 - \min\left(\sum_{n=1,n\neq k}^{N} \|\mathbf{x}_{n,t_0} - \mathbf{x}_{n,t_1}\|, 1\right) \tag{1}$$

with $\|.\|$ being the L^2 norm and N being the number of objects in the scene without the picked object k [41]. Fig. 5 (b) shows exemplary ground truth $g_\mathrm{u} \in [0,1]$ labels for the IPABar object. It can be seen that the objects at the top of the bin have a high graspability value regarding the unrest.

The graspability of object k based on the entanglement with other objects is

$$g_{\mathrm{e},k} = \begin{cases} 0, & \text{if an entanglement occurred for any grasp pose} \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

Fig. 5 (c) shows the ground truth g_e labels for the IPAUBolt. A goal for the robot is to avoid picking objects which can potentially entangle.

D. Parameterization of the Output

Similar to [25], we introduce a spatial discretization of the 3D scene into $S \times S$ volume elements (see white grid in Fig. 5) and solve a regression problem locally, i.e., individually for each volume element. Each volume element comprises an $(11 + J)$-dimensional vector containing the probability p, visibility v, positions x, y, z, Euler angles φ_1, φ_2, φ_3, graspabilities g_a, g_u, g_e of the object, and a J-dimensional vector with a success label $s \in \{0,1\}$ for each grasp pose G for the considered task (grasping, precise placement). For the ground truth generation, the objects are assigned to the volume element which contains the origin of the object coordinate system. In case multiple objects fall into the same volume element, we assign the object with the highest visibility v as ground truth. All volume elements not containing an object are filled with a zero vector. The output of the network is a $S \times S \times (11 + J)$ tensor as depicted in Fig. 4 (d).

E. Loss Function

To train the network, the multi-task loss function

$$\mathcal{L} = \sum_{i=1}^{S^2} \left(\lambda_1 \mathcal{L}_\mathrm{p} + \left[\lambda_2 \mathcal{L}_\mathrm{v} + \lambda_3 \mathcal{L}_\mathrm{pose} + \lambda_5 \mathcal{L}_\mathrm{g} + \lambda_6 \mathcal{L}_{\mathrm{gp_s}}\right] p_i\right) \tag{3}$$

is optimized. The λ-factors are manually tuned weights for the different loss terms. While $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_4 = 1$, $\lambda_5 = 1$, and $\lambda_6 = 1/J$ are constant, $\lambda_3 = (g_\mathrm{a} + g_\mathrm{u} + g_\mathrm{e})^3$ is a function of the ground truth graspabilities $g_\mathrm{a}, g_\mathrm{u}, g_\mathrm{e}$ to make the network focus on the relevant objects for grasping.

For the loss of the pose

$$\mathcal{L}_\mathrm{pose} = \mathcal{L}_\mathrm{pos} + \lambda_4 \mathcal{L}_\mathrm{ori} \tag{4}$$

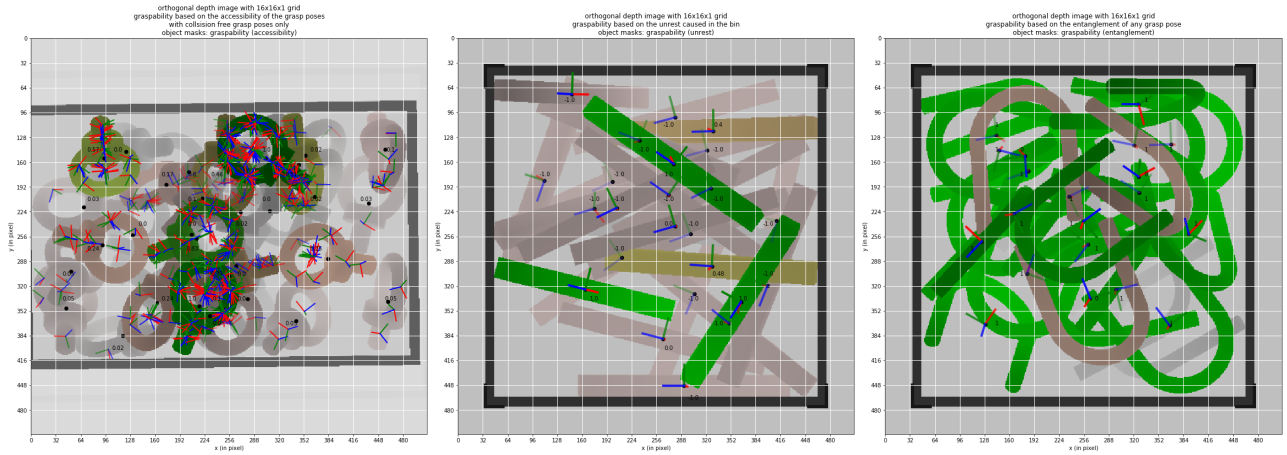
we use

$$\mathcal{L}_\mathrm{pos} = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \tag{5}$$

with $\mathbf{x} = [x, y, z]^\top$ and

$$\mathcal{L}_\mathrm{ori} = \|\boldsymbol{\varphi} - \hat{\boldsymbol{\varphi}}\|^2 \tag{6}$$

with $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \varphi_3]^\top$ and $\varphi_1, \varphi_2 \in [0, 2\pi)$ and $\varphi_3 \in [0, 2\pi/k)$, where $k \in \mathbb{N}$ represents the order of the cyclic symmetry.



(a) (b) (c)

Fig. 5. Exemplary ground truth samples: (a) Graspability $g_\mathrm{a} \in [0, 1]$ of the IPARingScrew based on the accessibility of the grasp poses (each collision-free grasp pose is indicated by a small coordinate system). (b) Graspability $g_\mathrm{u} \in [0, 1]$ of the IPABar based on the unrest caused in the bin during grasping. (c) Graspability $g_\mathrm{e} \in \{0, 1\}$ of the IPAUBolt based on entanglements with other objects after grasping. Objects shown in green have a high rating for tangibility. The more difficult an object is to grasp, the more the colour changes from green to yellow to red with increasing transparency.

To stabilize the training, the position $\mathbf{x} \in \mathbb{R}^3$ of the object is estimated relative to the volume element, i.e., $x, y, z \in (0, 1)$, while z is the position between the near and far clipping plane of the 3D sensor, and the angles are bounded, i.e., $\varphi_1, \varphi_2, \varphi_3$ are mapped to $[0, 1)$. For objects with a revolution symmetry, we omit the respective output feature-map.

We use the binary cross-entropy loss for the probability channel, the visibility channel, the three graspability channels, and the grasp pose result channels to compute $\mathcal{L}_\mathrm{p}$, $\mathcal{L}_\mathrm{v}$, $\mathcal{L}_\mathrm{g}$, and $\mathcal{L}_{\mathrm{gp_s}}$, respectively, while only backpropagating the loss for the elements that contain ground truth by multiplying each channel element-wise with the ground truth probability channel.

F. Network Architecture

The input of our model is a single normalized depth image which is processed with a fully convolutional architecture and mapped to a 3D output tensor as shown in Fig. 4 (d). In the experiments, we use an input resolution of 128×128 pixel, a DenseNet-BC [42] with 40 layers and a growth rate of 50, which represents the number of feature-maps being added per layer, and $S = 16$. We choose a DenseNet-BC because it promotes gradient propagation by introducing direct connections between any two layers with the same feature-map size and has a high parameter efficiency. The network architecture consists of four dense blocks and downsampling is performed three times via 2×2 average pooling to reduce the size of the feature-maps from 128×128 to 16×16 and preserve the spatial information. ReLU activation functions are employed in the dense blocks and sigmoid functions for the 3D output tensor. With this architecture, forward passes are performed with a frame rate of 92 fps on a Nvidia Tesla V100.

G. Training

During training, the error of the entire probability channel and the error of the remaining elements from the 3D output tensor that contain ground truth are backpropagated. Since annotating the grasps is time-consuming, we do not annotate the whole training dataset and only backpropagate $\mathcal{L}_\mathrm{g}$ and $\mathcal{L}_{\mathrm{gp_s}}$ for the samples, where ground truth annotations are available. In our experiments, we only annotate 100 out of 750 cycles from the Fraunhofer IPA [27] and our newly provided dataset, resulting in 100 uniformly distributed samples over different fill levels of the bin.

We augment the training data by rotating around the z-axis of the camera coordinate system and mirroring the images if the object is symmetric with respect to a plane while adjusting the ground truth pose annotations accordingly. To not lose the information of the robot placement relative to the bin, we only backpropagate $\mathcal{L}_{\mathrm{gp_s}}$ for the non-augmented samples.

For a robust Sim-to-Real Transfer, we use domain randomization [12]. To allow the model generalizing on real-world data, we apply different augmentations with varying intensity to the rendered training images, e.g., adding noise, blurring, elastic transformations, dropout, etc. This allows PQ-Net generalizing to different 3D sensor technologies.

We use the Adam optimizer with an initial learning rate of 0.01, monitor the validation loss, reduce the learning rate by a factor of 10 if the loss did not improve for three epochs, and train the network for about 50 epochs on the synthetic data.

H. Policy

Based on a single depth image I with global scene information, the neural network f with weights θ outputs a 3D tensor $\hat{T}$. Our policy π uses the network output to

select the highest quality grasp weighted with $\hat{p}$, $\hat{v}$, $\hat{g_{\mathrm{a}}}$, $\hat{g_{\mathrm{u}}}$, and $\hat{g_{\mathrm{e}}}$ from all S^2 volume elements for execution

$$\pi(f_\theta(I)) = \underset{i,j}{\operatorname{argmax}}(\hat{s}_{j,i} \cdot \hat{p}_i \cdot \hat{v}_i \cdot \hat{g_{\mathrm{a}}}_i \cdot \hat{g_{\mathrm{u}}}_i \cdot \hat{g_{\mathrm{e}}}_i) \quad (7)$$

with $i = 1, ..., S^2$, $j = 1, ..., J$ with J being the number of predefined grasp poses, and $\hat{s}_{j,i}$ being the success estimate of grasp pose G_j at volume element i for the considered task (grasping, precise placement).

IV. Experimental Evaluation

In this paper, we focus on parallel jaw grippers using a RG2 gripper [43]. Given a proper physics simulation, our approach can easily be transferred to other gripper types.

A. Sim-to-Real Transfer

For demonstrating a robust transfer of PQ-Net to the real world, we extend the Siléane [26] and Fraunhofer IPA [27] datasets with annotations for the collision-free reachability of approximately 500 densely sampled grasp poses (examples see Fig. 3). In Table I we report the success rate of our grasping policy and the precision and recall over all grasp poses in the scene. Applying randomizations on the synthetic images during training (see Section III-G) allows PQ-Net providing robust pose estimates and very high success rates of the policy on real-world data recorded with different 3D sensors.

Table I gives the average precision (AP) results for the object pose estimation based on the metric provided by Brégier et al. [40], [26] from OP-Net [25] with exactly the same depth image resolution, network architecture, and output discretization as used for PQ-Net. Even with a success estimate for approximately 500 grasp poses together with the graspability g_{a}, PQ-Net only loses very few points in terms of AP. In addition to the datasets, videos of real-world experiments are available at `http://www.bin-picking.ai/en/dataset.html`.

B. Benchmarking in Simulation

For evaluation, we compare the performance of three approaches in simulation on two very challenging objects. Each approach gets to observe the same 250 scenes for both the IPABar (see Fig. 2 left) and IPAUBolt (see Fig. 2 right), respectively, and executes one grasp per scene to ensure a comparison under the same conditions (the approaches face the exactly same scenarios). Table II reports the success rates for each method.

A robust picking of the IPABar objects from a cluttered bin is challenging because they require a defined picking order. If an occluded object is chosen for grasping, the grasp trial might fail completely or the object might move relative to the gripper which hinders a precise placement of the object. Therefore, the right object and grasp pose has to be chosen from the highly cluttered scene. This is especially important for friction grasps (force closure) because for a form closure it is unlikely that object moves relative to the gripper.

The IPAUBolt is challenging because it can potentially entangle with other objects in the bin. In case a wrong object and grasp pose for picking is chosen, the robot might lift multiple objects resulting in failed grasp because no collision-free placement of the object is possible.

GG-CNN [7], [32] and other model-free approaches [13], [14], [15], [3], [2], [8], [9], [35], [11] focus on generalization performance, use top-down grasps for pick-and-drop of the objects, and cannot solve precise pick-and-place tasks. To compare our approach with these methods, we use the logged grasping success labels for training. Table II reports the success rates for each approach. Our approach outperforms GG-CNN because of operating in 6D and being specifically configured to the object. Furthermore, GG-CNN collides with the bin when attempting to grasp object close to the border due to the limited flexibility in the gripper orientation.

The analytical approach considers the collision-free reachability and kinematical feasibility of the grasp pose and the path only and does not give an estimate on the actual physical outcome of the grasp, e.g., whether the object might move relative to the gripper in the given scenario, jamming, or entanglements with other objects in the bin.

With our simulation-driven and learning-based approach, we let our system autonomously learn how to localize and grasp the objects and transfer the automatically gained experience from the simulation to the real world without any time-consuming object-specific manual tuning.

V. Discussion

In the following, we summarize strengths and discuss limitations of our approach.

A. Strengths

PQ-Net gets to observe the whole depth image and selects highly robust grasps on a global level because of not looking at local patches of the image only. Our approach can operate in a closed-loop fashion with 92 fps for the forward pass (for OPE and grasp planning) and is therefore suitable for grasping in non-static environments. Our graspability metrics allow a gentle removal of the components and avoiding to grasp entangling objects. Furthermore, our approach can be extended to prioritizing grasp poses which allow a precise placement without re-grasping (e.g., important for objects without symmetries) which allows to reduce cycle times. PQ-Net provides robust estimates on real-world data independent of the actual 3D technology being used and does not require any human labeled data or grasping trials on the real-world system, facilitating scalability. Furthermore, it automatically configures for new object geometries using simulation and machine learning for precise pick-and-place tasks in highly cluttered scenes by providing an object model only. Our approach properly considers all possible kinds of object symmetries during data generation in the physics simulation and in the loss function for the regression of the angles. Furthermore, our provided simulated scenes can be used to benchmark further approaches.

B. Limitations

PQ-Net is a model-based approach and, therefore, does not generalize to unseen objects. Instead, it configures for

TABLE I

PREDICTION OF COLLISION-FREE REACHABILITY OF GRASP POSES OF OUR APPROACH ON REAL-WORLD / NOISY DATA FOR DIFFERENT OBJECTS WITH DIFFERENT KINDS OF OBJECT SYMMETRY FORM THE SILÉANE [26] AND FRAUNHOFER IPA [27] DATASETS.

object	SileaneBunny	SileaneCandlestick	SileanePepper	SileaneGear	SileaneTLess20	IPARingScrew	IPAGearShaft
object symmetry based on [40], [26]	no proper symmetry	revolution	revolution	revolution	cyclic (order 2)	cyclic (order 2)	revolution
PQ-Net success rate of policy	0.98	0.97	0.98	0.99	0.98	0.98	0.99
PQ-Net precision (all grasp poses)	0.57	0.70	0.71	0.73	0.77	0.75	0.83
PQ-Net recall (all grasp poses)	0.52	0.67	0.66	0.64	0.65	0.45	0.65
PQ-Net success rate OPE for chosen object	0.89	0.92	0.95	0.98	0.98	0.98	0.99
PQ-Net success rate OPE for chosen object with ICP	0.91	0.93	0.95	0.99	0.99	0.99	0.99
PQ-Net AP (OPE) whole scene	0.86	0.88	0.92	0.74	0.82	0.86	0.98
OP-Net [25] AP (OPE) whole scene	0.92	0.95	0.98	0.82	0.85	0.88	0.99

TABLE II

COMPARISON OF THE PERFORMANCE OF PQ-NET WITH OTHER APPROACHES AND PERFORMANCE EVALUATION.

object	IPABar	IPAUBolt
object symmetry based on [40], [26]	finite non trivial	cyclic (order 2)
GG-CNN [7], [32] success rate for grasping	0.78	0.67
GG-CNN [7], [32] success rate for grasping without bin	0.81	0.72
PQ-Net (ours) success rate for grasping	0.99	0.87
PQ-Net (ours) precision (all grasp poses)	0.63	0.59
PQ-Net (ours) recall (all grasp poses)	0.65	0.57
analytical approach [5] success rate for precise placement	0.85	0.80
PQ-Net (ours) success rate for precise placement	0.89	0.81
PQ-Net (ours) success rate OPE	0.96	0.85
PQ-Net (ours) precision (all grasp poses)	0.57	0.66
PQ-Net (ours) recall (all grasp poses)	0.52	0.60

novel objects without any human intervention. The approach requires a large dataset to train on, where the process of data generation (grasp execution) can take long, especially, when increasing the number of grasp poses. Still the process for data generation can run much faster than real time and can easily be parallelized across multiple machines. Furthermore, our method cannot do pre-grasp manipulations on the objects to change their poses in order to more robustly grasp them. While PQ-Net can avoid entangled object situations, we do not propose a solution to unhook very complex object geometries, for which no general solution has been proposed so far [44], [45].

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed an novel learning-based approach for grasping in highly cluttered scenes and precise object placement based on depth images. Our approach outputs the 6D object poses together with a graspability and quality estimate for each automatically generated grasp pose for multiple objects simultaneously in a single forward pass in a joint framework (running at 92 fps). All densely discretized and automatically generated grasp poses are executed in a physics simulation and the gained experience is transferred from simulation to the real world. Our approach outperforms model-free approaches in terms of grasping success rate and does contrary to analytical approaches not require any human involvement (automatic configuration). We demonstrate that our approach can be used for precise real-world robotic pick-and-place tasks, although being entirely trained on simulated data.

In future work, we plan to extend the approach to mixed bins and study how to reduce the time for data generation and training to allow a faster deployment of our solution. Furthermore, we want to study whether the generated data (grasping trials) can be used for model-free robotic grasping approaches in 6D.

ACKNOWLEDGMENT

This work was partially supported by the Federal Ministry of Education and Research (Deep Picking – Grant No. 01IS20005C) and the Ministry of Economic Affairs of the state Baden-Württemberg (Center for Cognitive Robotics — Grant No. 017-180004 and Center for Cyber Cognitive Intelligence (CCI) – Grant No. 017-192996). We would like to thank our colleagues for helpful discussions and D. Unruh for the support with experiments.

REFERENCES

[1] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, 2018.

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

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

[4] 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 (IROS)*, 2018.

[5] F. Spenrath, A. Spiller, and A. Verl, "Gripping point determination and collision prevention in a bin-picking application," in *German Conference on Robotics (ROBOTIK)*, 2012.

[6] F. Spenrath and A. Pott, "Gripping point determination for bin picking using heuristic search," in *CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME)*, 2016.

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

[8] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," in *International Symposium on Experimental Robotics (ISER)*, 2016.

[9] 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," in *The International Journal of Robotics Research (IJRR)*, 2018.

[10] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[11] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[12] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

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

[14] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems (RSS)*, 2017.

[15] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[16] R. D. Schraft and T. Ledermann, "Intelligent picking of chaotically stored objects," *Assembly Automation*, vol. 23, no. 1, pp. 38–42, 2003.

[17] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, 2020.

[18] M. El-Shamouty, K. Kleeberger, A. Lämmle, and M. Huber, "Simulation-driven machine learning for robotics and automation," *tm – Technisches Messen*, vol. 86, no. 11, pp. 673–684, 2019.

[19] T. Ledermann, "Partikel-Schwarm-Optimierung zur Objektlageerkennung in Tiefendaten," Dissertation, University of Stuttgart, 2012.

[20] M. Palzkill, "Heuristisches Suchverfahren zur Objektlageerkennung aus Punktewolken für industrielle Zuführsysteme," Dissertation, University of Stuttgart, 2014.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016.

[24] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] K. Kleeberger and M. F. Huber, "Single shot 6d object pose estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[26] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6d object pose estimation dataset for industrial bin-picking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[28] 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.

[29] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[30] Y. Konishi, K. Hattori, and M. Hashimoto, "Real-time 6d object pose estimation on cpu," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[31] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[32] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," in *The International Journal of Robotics Research (IJRR)*, 2019.

[33] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[34] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[36] L. Berscheid, T. Rühr, and T. Kröger, "Improving data efficiency of self-supervised learning for robotic grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[37] E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: Improving the pam, clara, and clarans algorithms," in *International Conference on Similarity Search and Applications (SISAP)*, 2019.

[38] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[39] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[40] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Defining the pose of any 3d rigid object and an associated distance," *International Journal of Computer Vision (IJCV)*, vol. 126, no. 6, pp. 571–596, 2018.

[41] M. Moosmann, "Increasing the grasp reliability for bin picking by using machine learning," master's thesis, University of Stuttgart, 2019.

[42] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] OnRobot, "RG2 – flexible 2 finger robot gripper with wide stroke." [Online]. Available: https://onrobot.com/en/products/rg2-gripper

[44] R. Matsumura, Y. Domae, W. Wan, and K. Harada, "Learning based robotic bin-picking for potentially tangled objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[45] M. Moosmann, F. Spenrath, K. Kleeberger, M. U. Khalid, M. Mönnig, J. Rosport, and R. Bormann, "Increasing the robustness of random bin picking by avoiding grasps of entangled workpieces," in *CIRP Conference on Manufacturing Systems (CIRP CMS)*, 2020.