Robotic Pick-and-Place With Uncertain Object Instance Segmentation and Shape Completion

Marcus Gualtieri and Robert Platt

***Abstract*—We consider robotic pick-and-place of partially visible, novel objects, where goal placements are non-trivial, e.g., tightly packed into a bin. One approach is (a) use object instance segmentation and shape completion to model the objects and (b) use a regrasp planner to decide grasps and places displacing the models to their goals. However, it is critical for the planner to account for uncertainty in the perceived models, as object geometries in unobserved areas are just guesses. We account for perceptual uncertainty by incorporating it into the regrasp planner's cost function. We compare seven different costs. One of these, which uses neural networks to estimate probability of grasp and place stability, consistently outperforms uncertainty-unaware costs and evaluates faster than Monte Carlo sampling. On a real robot, the proposed cost results in successfully packing objects tightly into a bin 7.8% more often versus the commonly used minimum-number-of-grasps cost.**

***Index Terms*—Perception for grasping and manipulation, manipulation planning, deep learning in grasping and manipulation.**

I. Introduction

PICK-AND-PLACE is prehensile manipulation where objects are grasped rigidly and placed into desired configurations [1]. This problem has been extensively studied for fully observed objects, resulting in deeper understanding of the problem and efficient planning algorithms [2], [3], [4], [5], [6], [7]. Such a planner could be combined with a separately designed perceptual algorithm for estimating objects' geometry from raw sensor data. However, a system with separately designed perception and planning modules is not always optimal: these methods treat grasping an unobserved part of an object the same as grasping a part that is fully observed, which could lead to avoidable failures.

One approach to this problem is to dispense with the idea of separate perception and planning modules and use reinforcement learning (RL) to train a single module that does both. While some success has been achieved with this idea [8], [9], [10], training is time-consuming, the system is not robust to changes in either task or environment, and performance is often suboptimal, even for simple tasks (cf. placing mugs with an RL approach [8] versus a modular approach [11]).

Another approach is to plan in belief space, i.e., in probability distributions over state [12]. While this handles arbitrary types of uncertainty, there are a couple of important drawbacks. First, these methods often require a detailed description of the observation and state transition models of the system, which can be difficult to obtain [13], [14]. Second, planning takes place in the space of probability distributions over states, which is continuous and, for practical problems, high dimensional. For these reasons, this approach has been confined to problems with few dimensions or other simplifying structure.

We take a new approach to pick-and-place of novel, partially visible objects: (a) use perception to predict the complete geometry of the objects and (b) incorporate instance segmentation and shape completion uncertainty as a planning cost. We compare seven cost functions, four of which explicitly model the probability of successfully executing a regrasp plan, including grasp quality (GQ), Monte-Carlo (MC) sampling, uncertainty at contact points (CU), and success prediction (SP). With only small modifications to existing planners, we efficiently account for perceptual uncertainty.

We test this approach with bin packing and bottle arrangement tasks in both simulation and the real world. Results show perception is indeed a significant source of error and shape completion is critical to regrasp planning. Also, the SP method consistently outperforms three other methods (no cost, step cost, and GQ) which do not account for perceptual uncertainty in terms of avoiding grasp failures. Furthermore, the SP cost is faster than MC sampling.

II. Related Work

Pick-and-place in fully observed environments: Pick-and-place was often studied independently from perception. *Regrasping*, which is to find a sequence of picks and places moving an object to a goal pose, was first explained by Tournassoud et al. [2]. There is a discrete search component, for sequencing grasp-place combinations, and a continuous search component, for connecting grasp-place combinations with a motion plan. Alami et al. generalized regrasping to multiple, movable objects, pointed out the problem is NP-hard, and coined the term *manipulation planning* [3]. Later they considered different cost functions for the discrete search, including path length and number of grasp changes [4]. Nielsen and Kavraki gave a 2-level, probabilistically complete planner for manipulation planning [5]. Wan et al. employed a 3-level planner, where the high-level planner provides a set of goal poses for the objects, the middle-level planner is a regrasp planner, and the low-level planner is a motion planner [7]. For *non-monotonic* rearrangement problems (i.e., objects need moved more than once), a middle-level planner displacing multiple objects was more efficient [6]. Our approach is to start with a well-established regrasp (i.e., middle-level) planner and build an uncertainty capability upon it.

Pick-and-place of known objects: Others have considered pick-and-place of imperfectly perceived objects with known

The authors are with the Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA (email mgualti@ccs.neu.edu). This work was supported in part by NSF (1724257, 1724191, 1763878, 1750649) and NASA (80NSSC19K1474).

shapes. One approach is to match object models to sensor data, as in Tremblay et al. [15]. Morgan et al. used clustering to localize blocks for their box and blocks benchmark task [16]. However, we consider novel objects, i.e., the shapes are not known *a priori*.

Pick-and-place of novel objects: A few projects have considered novel-object pick-and-place, where the complete shapes of objects are not given. The first to address this was Jiang et al. [17] who used random sampling with classification to identify placements that are likely to be stable and satisfy human preference. After this, we approached the problem with deep RL by learning a grasp/place value function [8], [9], [10]. Next, Manuelli et al. proposed a 4-component pipeline: (a) instance segmentation, (b) key point detection, (c) optimization-based planning for task-specific object displacements, and (d) grasp detection [11]. Objects were minimally represented by key points, which are 3D points indicating task-relevant object parts, e.g., the top, bottom, and handle of a mug. Later, Gao and Tedrake augmented this with shape completion, which is useful for avoiding collisions when planning arm motions with the held object [18]. Finally, Mitash et al. addressed the problem by fusing multiple sensor views and allowing a single regrasp as necessary, conservatively assuming the object is as large as its unobserved region [19]. None of these compared different ways of accounting for perceptual uncertainty, as we do here.

Pick-and-place under uncertainty: A general approach to pick-and-place under arbitrary types of uncertainty is to solve a partially observable Markov decision process (POMDP). Kaelbling and Lozano-Pérez focused on symbolic planning in belief space with black-box geometric planners and state estimators [13]. Xiao et al. used POMCP [20] to update their belief about the arrangement of a small set of known objects [14]. However, the POMDP approach requires significant computation and an accurate model of transition and sensor dynamics.

Grasping under uncertainty: We extend ideas from grasping under object shape uncertainty to pick-and-place planning. The two most common approaches to grasping under shape uncertainty are (a) evaluate grasp success over an MC sampling of object shapes and (b) evaluate a probabilistic model of grasp success. Kehoe et al. took the MC approach and represented uncertainty as normally distributed polygonal vertices with given means and variances [21]. Hsiao et al. provided a probabilistic model for grasp success given multiple object detections and grasp quality evaluations [22]. Afterward, Gaussian process implicit surfaces (GPISs) were proposed as a representation of object shape uncertainty [23], [24], [25], [26]. GPISs combine multiple observations of an object's signed distance function (SDF) into a Gaussian process – a normal distribution over SDFs [23]. Mahler et al. compared a probabilistic model (based on the variance of the GPIS at contact points) to an MC approach [24]. The MC approach did better but has higher computational cost. Laskey et al. improved the efficiency of MC sampling from the GPIS by employing multi-armed bandit techniques to reduce the number of evaluations for grasps that are unlikely to succeed [25]. Li et al. conducted real-world experiments filtering grasps with different thresholds on variance of the GPIS at contact points [26]. Lundell et al. represented objects as voxels, used a deep network to complete objects, and performed MC sampling using dropout [27].

III. Problem Statement

Consider planning robot motions to place a partially visible object of unknown shape into a goal pose. In particular, we consider this problem in the context of the following system:

Definition 1 (Move-open-close system): A *move-open-close system* consists of one or more objects, a robotic manipulator, and one or more depth sensors, each situated in 3D Euclidean space. Objects are rigid masses $O_1, \ldots, O_{n_{obj}} \subseteq \mathbb{R}^3$, sampled randomly from an unknown probability distribution. The manipulator is equipped with a parallel-jaw gripper with status *empty* or *holding*. The action of the robot is to move the gripper to a target pose $T_e \in SE(3)$, followed by either gripper *open* or *close*. At each step, the robot acquires a point cloud $C \in \mathbb{R}^{n \times 3}$, observes its gripper status, and takes an action.

To simplify planning, we avoid dynamic actions (e.g., pushing). In particular, *close* actions should fix an object rigidly in the gripper, and *open* actions should place an object at rest. Assume an *antipodal grasp* is sufficient to fix an object in the gripper ([28] p. 233), and assume the conditions in [2] are sufficient to stably place an object on a horizontal surface. We now state the problem as follows:

Definition 2 (Regrasping under perceptual uncertainty): Given a move-open-close system, objects represented as point clouds $\{\bar{C}_i \in \mathbb{R}^{\bar{n}_i \times 3}\}_{i=1}^{n_{obj}}$, perceptual uncertainty vectors $\{\mathcal{U}_i \in \mathbb{R}^{d_i}\}_{i=1}^{n_{obj}}$, and a set of goal poses for each object $\{\{T_{ij} \in SE(3)\}_{j=i}^{n_{goal}}\}_{i=1}^{n_{obj}}$, find a sequence of antipodal grasps and stable places maximizing the probability of displacing an object to a goal pose.

There are different ways to represent the uncertainty vectors $\mathcal{U}_i$, including point-wise segmentation/completion uncertainties, Monte-Carlo samples, and grasp/place success prediction networks: these are described in Section V. Intuitively, actions should account for uncertainty in object shape, as grasping and placing on uncertain object parts is likely to result in unpredictable movements of the object.

IV. System Overview

Consider a modular, perception-planning pipeline for displacing partially visible, novel objects, where the regrasp planner addresses the problem of Def. 2. Such a system is summarized in Fig. 1. For each perception-action cycle, the environment produces a point cloud, the geometry of the scene is estimated, a partial plan for displacing an object is found, and the first pick-and-place of the plan is executed. Automatic resensing and replanning accounts for failures, similar to MPC [29]. In this section, each component is briefly described. Regrasping under segmentation and completion uncertainty – the main contribution – is detailed in Section V.

A. Perception

The purpose of the perceptual modules is to reconstruct the geometry of the scene so we can apply geometric planning algorithms. Additionally, they must quantify their own

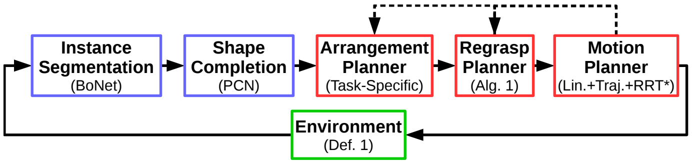


Fig. 1: Diagram of our system architecture. Green represents the environment, blue the perceptual modules, and red the planning modules. Dashed arrows are followed up to a number of times if no plan is found.

uncertainty so plans unlikely to succeed can be avoided. For both instance segmentation and shape completion, we have chosen point clouds as the input/output representation of objects. A point representation consumes less memory than uncompressed voxel grids, enables efficient planning, and, from our previous experience, exhibits good simulation-to-real domain transfer [8], [9], [10].

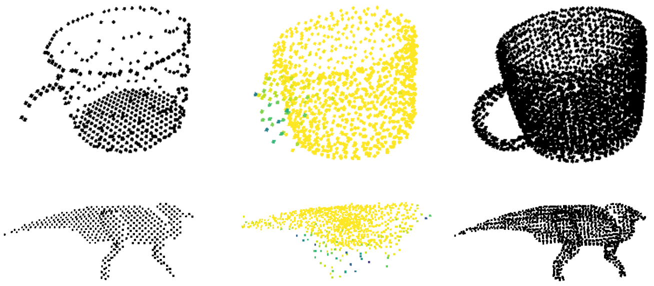
1) Object instance segmentation: The input to the segmentation module is a point cloud $C \in \mathbb{R}^{n \times 3}$, and the output is a point cloud for each object, $\{C_i \in \mathbb{R}^{n_i \times 3}\}_{i=1}^{n_{obj}}$ with $\sum_{i=1}^{n_{obj}} n_i \leq n$, and uncertainties $\{U_i \in \mathbb{R}^{n_i}\}_{i=1}^{n_{obj}}$. Although any object instance segmentation method with this interface can be used in the proposed architecture, our implementation uses BoNet [30]. BoNet produces an $n \times K$ matrix, where K is a predefined maximum number of objects, and each row is a point's distribution over object ID. U_i is the $\max$ of the ith row, which is the estimated probability the ith point is correctly segmented. (And, optionally, points with U_i below a threshold can be omitted.)

2) Shape completion: The input to the shape completion module is a point cloud $C \in \mathbb{R}^{n \times 3}$, and the output is a point cloud $\bar{C} \in \mathbb{R}^{\bar{n} \times 3}$ that is a dense sampling of points on all object faces, including faces not visible to the sensor. We also require an uncertainty estimate for each completed point, $\bar{U} \in \mathbb{R}^{\bar{n}}$. Although any shape completion method with this interface can be used in the proposed architecture, our implementation uses a modified version of PCN [31]. PCN consists of an encoder (two PointNet layers [32]) and a decoder (three fully connected, inner product layers). We augmented the original version of PCN with a second decoder for uncertainty estimates. In particular, the uncertainty decoder is trained using a binary cross-entropy loss to predict the probability each point is within Euclidean distance $\beta \in \mathbb{R}_{++}$ of the nearest ground truth point. So the uncertainty values should be interpreted as the estimated probability each completed point is accurate. Example completions are shown in Fig. 2b.

B. Planning

We use a 3-level planner, similar to Wan et al. [7].

1) Arrangement planner: The input to the arrangement planner is a list of completed clouds, $\bar{C}_1, \dots, \bar{C}_{n_{obj}}$, and the output is a set of triples $\{(T, c, i)_j\}_{j=1}^{n_{goal}}$, where T is a goal pose for object i and c is an associated goal cost. The reason the arrangement planner produces multiple goals for multiple objects is to increase the chances one of them is feasible. Besides, not all goals are equal: some may be more preferable to the task. For example, in bin packing, some placements will result in tighter packings than others. This is captured by the goal cost, c. We implement a different arrangement planner for each task.



(a) Observed cloud. (b) Completed cloud. (c) Ground truth.

Fig. 2: Shape completions with PCN. Yellow represents high $\bar{U}$ values (near 1), and blue represents low $\bar{U}$ values (near 0.5).

2) Regrasp planner: The regrasp planner takes in the triples from the arrangement planner and produces a sequence of picks and places, i.e., effector poses, that displaces one object. If a regrasp plan is not found, more goals can be requested from the arrangement planner (as indicated by dashed lines in Fig. 1).

3) Motion planner: The motion planner finds a continuous motion between picks and places. Any off-the-shelf motion planner will do: we use a 3-level planner that first attempts a linear motion, then Trajopt [33], and then RRT* with timeout [34]. If no motion plan is found, the regrasp planner can be resumed from where it left off, but marking the infeasible section so the same solution is not found again.

V. Regrasp Planning Under Uncertainty

Regrasps are needed due to kinematic constraints: the grasps at the object's current pose may all be in collision or out of reach at the object's goal poses. In this case, a number of temporary places (i.e., non-goal places) are needed. Our regrasp planner (Alg. 1) extends Tournassoud et al.'s [2] to handle multiple goals for multiple objects, arbitrary additive costs, and discrete grasp/place sampling. Related planners (e.g, [3], [4], [5], [6]) could also have been adapted to the purpose: the main point is to incorporate segmentation and shape completion uncertainty into the cost.

Algorithm 1: Regrasp planner: run for each object.

```
Input: Number of sampling iterations $N$, completed
       cloud $\bar{C}$, uncertainty vector $\mathcal{U}$, goal poses and
       costs $\{(T, c)_j\}_{j=1}^{n_{goal}}$, and $costLowerBound$.
$RG \leftarrow []$
for $i \leftarrow 1, \dots, N$ do
    $G, gc \leftarrow$ SampleGrasps($\bar{C}, \mathcal{U}$)
    $P, pc \leftarrow$ SampleTemporaryPlaces($\bar{C}, \mathcal{U}$)
    $RG \leftarrow$ UpdateRegraspGraph($RG$,
      $\{(T, c)_j\}_{j=1}^{n_{goal}}, G, gc, P, pc$)
    $plan, cost \leftarrow$ A*($RG$)
    if $cost \leq costLowerBound$ then break
return $plan$
```

A key part of Alg. 1 is the regrasp graph, *RG*. The *regrasp graph* is a matrix where rows refer to grasps and columns refer to places. When the object has been grasped, column changes are allowed to switch the object's placement, and when the object has been placed, row changes are allowed to switch grasps [2]. To Tournassoud et al.'s regrasp graph we add costs: matrix values are the sum of the corresponding grasp and place costs if the grasp-place combination is feasible (i.e., there is a collision-free IK solution) and infinity otherwise.

Alg. 1 is run in parallel for each object that has at least one goal pose. For N steps, additional grasps (G with costs gc) and temporary places (P with costs pc) are randomly sampled. Given the shape completion $\bar{C}$, grasp samples are constrained to satisfy the geometric antipodal conditions ([28] p. 233), and place samples are constrained to satisfy the stability conditions ([2]). The function `UpdateRegraspGraph` adds a row for each sampled grasp and a column for each sampled place to *RG* and then checks IK and collisions for the new grasp-place combinations. Finally, A* with a consistent heuristic finds an optimal pick-and-place sequence, given the current samples [35]. Next we define the cost function used by A* and give different ways of calculating grasp and place costs.

A. Maximize probability of regrasp plan execution success

The aim is to choose a regrasp plan that maximizes the joint probability each grasp is antipodal and each temporary place is stable, i.e., maximize Eq. 1, where G_i is the event the ith grasp is antipodal, P_i is the event the ith place is stable, and m is total number of picks and places. Assuming each grasp/place is independent of previous steps in the plan, we arrive at Eq. 2.[1] Taking the log and abbreviating $\Pr(G_i)$ as g_i and $\Pr(P_i)$ as p_i yields Eq. 3.

$$\begin{aligned}&\Pr(G_1, P_1, \dots, G_{m/2}) \\ &= \Pr(G_{\frac{m}{2}} | G_1, P_1, \dots, P_{\frac{m}{2}-1}) \cdots \Pr(P_1|G_1)\Pr(G_1) \qquad (1)\\ &\approx \Pr(G_{m/2}) \cdots \Pr(P_1)\Pr(G_1) \qquad (2)\end{aligned}$$

$$\log\left[\Pr(G_1, \dots, G_{m/2})\right] \approx \sum_{i=1}^{m/2} \log(g_i) + \sum_{i=1}^{m/2-1} \log(p_i) \quad (3)$$

Negating Eq. 3 results in a non-negative, additive cost: the form required by A*. We account for plan length and task cost by adding these as objectives to a multi-criterion optimization problem ([36] pp. 181-184). Scalarization results in Eq. 4, where $w_1, \dots, w_4 \in \mathbb{R}_{++}$ are trade-off parameters and $c \in \mathbb{R}$ is the task cost associated with the goal placement (from the arrangement planner). This is the cost used by our regrasp planner. To complete the description, we next look at different ways of estimating g_i and p_i.

$$w_1 m - w_2 \sum_{i=1}^{m/2} \log(g_i) - w_3 \sum_{i=1}^{m/2-1} \log(p_i) + w_4 c \quad (4)$$

B. Probability grasps are antipodal and places are stable

1) Grasp quality (GQ): One way to estimate g_i is via a measure of "robustness" of the grasp to small perturbations in the nominal shape completion. For antipodal grasps, Murray et al. suggest choosing grasps where the line between contacts is inside and maximally distant from the edges of both friction cones ([28] p. 233). This way, a grasp will satisfy the geometric antipodal conditions under small perturbations to the object's shape.

We place this idea into our probabilistic framework. For both grasp contacts, $j = 1, 2$, let $\theta_j \in [0, \pi]$ be the angle between the surface normal n_j and the normalized, outward-pointing vector b_j connecting both contacts. Assume θ_j is distributed according to a truncated normal distribution with mode μ_j and scale σ, where the angle μ_j (Eq. 5) is derived from the nominal object shape and σ is given. The probability b_j lies in the friction cone is then $\Pr(\theta_j \leq \theta_{max}) = F(\theta_{max}; \mu_j, \sigma, 0, \pi)$, where F is the cumulative density function of the truncated normal distribution and θ_{max} is half the angle of the friction cone. We make the simplifying assumption that this probability is independent between contacts, giving Eq. 6.

$$\mu_j = \arccos(b_j \cdot n_j) \quad (5)$$

$$g_i = \prod_{j=1}^{2} F(\theta_{max}; \mu_j, \sigma, 0, \pi) \quad (6)$$

The effect of the GQ estimator is to choose grasps that are as centered as possible in both friction cones, given the estimated object shape. The scale parameter σ makes the trade-off between regrasp plan length and centering of grasps: small σ prefers centered grasps over short plans and large σ prefers short plans over centered grasps.

2) Monte Carlo (MC): Another approach is to estimate g_i and p_i via segmentation and completion samples, as was done for grasping under shape uncertainty [21], [24], [25], [27]. The idea is to randomly generate multiple segmentations then completions and average grasp/place antipodal/stability.

Let $\Pr(\bar{\mathcal{C}}_i|C)$, for $i = 1, \dots, n_{obj}$, be a distribution over object shapes, where C is the input point cloud, from which we collect samples. This could be implemented with segmentation/completion networks with randomized components, e.g., using dropout [27]. However, to compare to the CU method (described next), we use the point-wise uncertainty outputs of the networks (U_i and $\bar{U}_i$ in Section IV-A) as follows.

For segmentation, the object ID for each point is independently sampled from the distributions given by the segmentation matrix. (To reduce noise, we only sample points whose U-value is below a threshold.) For shape completion, assume the ith point's offset from the nominal point is i.i.d. $\sim \mathcal{N}(0, \sigma_i^2)$. Since $\bar{U}_i$ is the estimated probability the point is offset no more than β, the standard deviation of the point's offset is derived from the Gaussian CDF as in Eq. 7. To sample a shape: (a) sample a segmentation point-wise using the segmentation mask and (b) compute the shape completion given this segmentation. Then, for each point in the completion, (b.1) sample a direction uniformly at random and (b.2) sample an offset along this direction from a normal distribution with 0 mean and standard deviation given by Eq. 7.

[1] Assuming knowledge that a previous grasp/place was successful does not decrease the joint probability of success, Eq. 2 is a lower bound.

$$\sigma_i = \frac{\beta}{\sqrt{2}\,\mathrm{erf}^{-1}(\bar{U}_i)} \tag{7}$$

Regardless of implementation, g_i is estimated as #antipodal$/M$ and p_i is estimated as #stable$/M$ where M is the number of shape samples and #antipodal is the number of shapes for which the ith grasp is antipodal and #stable is the number of shapes for which the ith place is stable.

3) Contact uncertainty (CU): Computing g_i and p_i with MC is computationally expensive if M is large. This motivates considering uncertainty only at contact points. For instance, placing an object on its unseen, predicted geometry could likely be unstable, so we penalize grasps/places on uncertain object parts. The same idea is behind penalizing high-variance grasp contacts [24], [26].

Formally, suppose for the ith point, the segmentation network estimates $\Pr(V_i)$, where V_i is the event the ith point is segmented correctly. Suppose the shape completion network estimates $\Pr(\bar{V}_i|V_i)$ where $\bar{V}_i$ is the event the ith point is within Euclidean distance β of a ground truth point. Assuming whether a grasp (place) is antipodal (stable) depends only on whether each contact point is correctly segmented and is within Euclidean distance β of the nearest ground truth point, and assuming independence between contacts, g_i and p_i are estimated via Eq. 8 and 9, where contacts are explained in Fig. 3.

$$g_i \approx \Pr(\bar{V}_l|V_l)\Pr(V_l)\Pr(\bar{V}_r|V_r)\Pr(V_r) \tag{8}$$

$$p_i \approx \prod_{i=1}^{3} \Pr(\bar{V}_{t^i}|V_{t^i})\Pr(V_{t^i}) \tag{9}$$

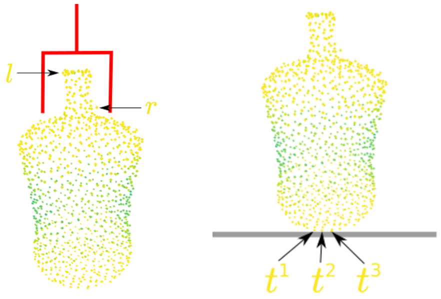


Fig. 3: **Left**. For an antipodal grasp (shown in red), there are at least 2 contact points, l and r. **Right**. For a stable placement on a flat surface, there are at least 3 contact points, t^1, t^2, and t^3. Colors represent estimates of $\Pr(\bar{V}_i|V_i)\Pr(V_i)$, where yellow represent higher probabilities.

The uncertainty values from PCN ($\bar{U}_i$ in Section IV-A) are used to estimate $Pr(\bar{V}_i|V_i)$. Estimating $\Pr(V_i)$ from the uncertainty values from BoNet (U_i in Section IV-A) is less straightforward since, for each completed point we must associate a corresponding segmentation uncertainty. A heuristic we found that works well for this is, for each point in the completed cloud, take the nearest neighbor in the segmented cloud.

4) Success prediction (SP): g_i and p_i can also be directly estimated with a neural network. The encoding of grasp/place as input to the neural network is an important design choice that affects performance [37]. Here, we encode grasps as the points from the shape completion, $\bar{C}$, inside the gripper's closing region w.r.t. the gripper's reference frame (cf. [38]). For places, the completed cloud, $\bar{C}$, is transformed to the place pose and translated with the bottom-center of the cloud at the origin. For network architecture, we use PCN with a single output with sigmoid activation, trained with the binary cross-entropy loss. Training data is generated in simulation, so labeling ground truth antipodal/stable is straight-forward.

VI. Experiments

We ran experiments in simulation and the real world to compare the different ways in Section V-B for accounting for object shape uncertainty in regrasp planning. We also compare to two baselines which do not account for object shape uncertainty: *no cost*, which takes the first regrasp plan found, and *step cost*, which includes the step cost term only ($w_1 = 1$ in Eq. 4). The step cost appears almost exclusively in the regrasping literature, e.g., [2], [3], [4], [7].

A. Setup

The experimental environment is illustrated in Fig. 4, left. We evaluate the proposed system on the following tasks:

1) **Bottle arrangement.** Place 2 bottles upright onto 2 coasters (from our prior work [9], [10]).
2) **Bin packing.** Place 6 objects into a box minimizing packing height. This is known as the 3D irregular-shaped open dimension problem [39]. This is easier to evaluate than smallest bin size [40] in the real world.
3) **Canonical placement.** Place any 1 of 5 objects into a goal pose. The arrangement planner is an oracle which consistently gives the same goal pose for an object. The purpose is to analyze regrasp performance independent from arrangement planner errors.

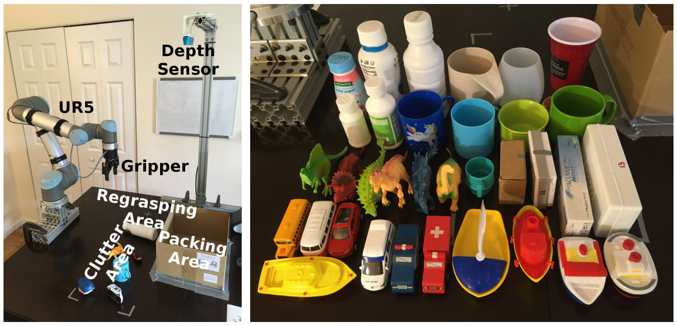


Fig. 4: **Left**. Environment includes a UR5 arm, a Robotiq 85 gripper, and a Structure depth sensor. **Right**. 34 same-category novel objects used for real-world packing experiments.

B. Simulation experiments

The environment is simulated by OpenRAVE [41] using 3DNet objects [42]. Objects are partitioned into 3 sets: *Train* for training all deep networks, *Test-1* for same-category novel objects (boat, bottle, box, car, dinosaur, mug, and wine glass), and *Test-2* for novel-category objects (airplane, bowl, and stapler). A depth sensor, situated above the objects, captures a point cloud which is then passed into BoNet for segmentation. Grasps succeed if (a) exactly 1 object intersects the hand closing region, (b) the antipodal condition with 24° friction cone is met, and (c) the robot is collision-free. Places are stable if the conditions in Tournassoud et al. [2] are met.

We evaluate *place execution success rate* – the proportion of regrasp plans with no grasp failures – and *temporary place stability rate* – the proportion of temporary places that are stable. These metrics are fast to compute and indicate how well the regrasp plans succeed in placing objects. For the bin packing task we also have "Packing height of 5", which refers to the end-of-episode packing height when 5/6 objects are placed. This is a measure of how well the arrangement planner chooses goals. We use a 1-sided, same-variance, unpaired t-test to decide if one method significantly outperforms another. (If $p \leq 0.05$, we accept the hypothesis that the treatment outperforms the baseline.)

1) Perception ablation study: We quantify the potential benefit of accounting for uncertainty for the bin packing task. We evaluate performance with ground truth perception (GT Seg. & Comp.), imperfect completion (GT Seg.), imperfect segmentation and completion (Percep.), and without shape completion (GT Seg. & No Comp.) "Imperfect" means the objects' segmentation/completion is estimated from the observed point cloud. Step and task costs are used, i.e., $w_1 = w_4 = 1$ and $w_2 = w_3 = 0$ in Eq. 4, where the task cost, c, is the estimated final packing height in centimeters.

Results (shown in Table I) are as expected. A clear drop in performance is observed as perception becomes imperfect (down 18% for imperfect completion and another 4% for imperfect segmentation). Thus, a large source of error is due to perception, so there is space for improvement by accounting for perceptual uncertainty. Without shape completion, regrasp planning is crippled (regrasp plan found rate drops from 94.1% for Percep. Test-1 to just 10.0%). This is because insufficient grasp and place samples are found to displace objects.

2) Regrasp cost comparison: We test the hypothesis that a method estimating perceptual uncertainty (either MC, CU, or SP from Section V-B) selects regrasp plans that execute successfully more often on average than other methods (e.g., no cost, step cost, and GQ). Additional methods can be obtained by combinations, e.g., "MC+GQ" refers to the case where MC and GQ costs are summed.

Results for bin packing are shown in Table II. For Test-1, MC+GQ has the best grasp performance while SP has the best temporary place stability rate. MC+GQ significantly outperforms GQ ($p = 0.0092$ for place execution success and $p = 0.0073$ for grasp antipodal), suggesting the network's uncertainty estimates are useful for planning. For Test-2, SP has the best grasp performance (vs. step cost, $p = 0.023$ for place execution success and $p = 0.0023$ for grasp antipodal), and MC+GQ has the best temporary place stability rate. CU does not significantly outperform the baselines for either dataset: it is not sufficient to account for uncertainty only at the contact points. For bin packing, we do not see a significant improvement for place stability over the step cost, but this is because regrasps are rare with the step cost, obscuring the significance of the results.

We see a bigger difference with the canonical task on Test-1 (Table III). In this case, MC, CU, and SP methods have significantly higher temporary place stability rates than no cost (which happened to do better than step cost) ($p = 0.010$, 0.005, and 2.4×10^{-10}, respectively). There is no doubt SP outperforms GQ for place execution success rate ($p = 9.7 \times 10^{-9}$) and for grasp antipodal rate ($p = 3.8 \times 10^{-9}$).

For both packing and canonical tasks, the SP method does significantly better than the baselines or GQ in terms of place execution success (packing Test-1 $p \leq 0.032$, packing Test-2 $p \leq 0.013$, canonical Test-1 $p \leq 9.7 \times 10^{-9}$, and canonical Test-2 $p \leq 2.9 \times 10^{-4}$), which supports the hypothesis.

C. *Real world experiments*

We seek to (a) see if the perceptual components, trained with simulated data, work well with real sensor data and (b) verify the importance of uncertainty seen in simulation results. For these experiments, same-category novel objects are used (Fig. 4, right).

To answer part (a), no domain transfer was needed for bin packing. For bottles, BoNet (but not PCN) overfit to simulation data. This problem was mitigated by adding simulated sensor noise. To answer part (b), a regrasp cost comparison for bin packing is shown in Table IV. Both MC and SP methods significantly outperform the step cost (which outperforms GQ) ($p = 0.019$ and $p = 0.012$, respectively). Example packing and regrasp sequences are shown in Fig. 5.

We also compare bottle arrangement performance to our previous method, which uses RL to learn a pick-and-place policy [10]. Many of the same bottles as before are included, but 4/15 of them are more challenging. Two of the bottles are difficult to distinguish orientation (size of tops near size of bottoms), and two are near the 8.5 cm gripper width. Results are shown in Table V. With the proposed method, all places are correct. Only the grasp success rate is lower than before, but all 3 grasp failures are with the wider bottles. Overall, we conclude the modular approach performs better (80% vs. 67% task success rate).[2]

VII. Conclusion

Object instance segmentation and shape completion enable use of existing planning algorithms for pick-and-place of sensed objects. However, perceptual errors are still a major source of failure. To compensate for this, we compare different planning costs modeling probability of successfully executing a regrasp plan. Results show the SP cost, which uses separate networks to predict grasp/place success, consistently performs nearly as well as or outperforms all other costs. We attribute this to: (a) unlike baseline and GQ costs, SP can detect when perception is uncertain based on the distribution of perceived points; (b) unlike the CU cost, which considers uncertainty only at contact points, SP considers uncertainty at many points;

[2] Source code and additional results are available at https://github.com/mgualti/GeomPickPlace.

	GT Seg. & Comp.	GT Seg. (Train)	GT Seg. (Test-1)	Percep. (Train)	Percep. (Test-1)	GT Seg. & No Comp.
Place Execution Success	0.929 ± 0.008	0.767 ± 0.013	0.747 ± 0.013	0.718 ± 0.014	0.710 ± 0.014	0.508 ± 0.046
Regrasp Plan Found	0.957 ± 0.006	0.882 ± 0.009	0.939 ± 0.007	0.879 ± 0.009	0.941 ± 0.007	0.100 ± 0.009
Grasp Antipodal	0.931 ± 0.007	0.779 ± 0.013	0.761 ± 0.013	0.755 ± 0.013	0.736 ± 0.013	0.563 ± 0.047
Temporary Place Stable	1.000 ± 0.000	0.769 ± 0.122	1.000 ± 0.000	0.828 ± 0.071	0.826 ± 0.081	0.500 ± 0.500
Packing height of 5 (cm)	12.27 ± 0.315	12.36 ± 0.331	12.18 ± 0.306	12.37 ± 0.447	12.44 ± 0.307	–
Regrasp planning time (s)	35.62 ± 1.103	38.46 ± 1.115	38.68 ± 1.141	35.76 ± 1.059	35.05 ± 1.077	15.86 ± 1.482

TABLE I: Perception ablation study for bin packing. Showing average ± standard error over 200 episodes.

	No Cost	Step Cost	GQ	MC	MC + GQ	CU	SP
Place Execution Success	0.651 ± 0.013	0.725 ± 0.012	0.748 ± 0.012	0.756 ± 0.012	**0.787** ± 0.011	0.712 ± 0.013	0.779 ± 0.012
Grasp Antipodal	0.737 ± 0.011	0.751 ± 0.012	0.794 ± 0.011	0.811 ± 0.011	**0.830** ± 0.010	0.743 ± 0.012	0.823 ± 0.010
Temporary Place Stable	0.784 ± 0.024	0.857 ± 0.097	0.845 ± 0.030	0.904 ± 0.028	0.883 ± 0.031	0.848 ± 0.054	**0.959** ± 0.018
Plan Length	2.665 ± 0.031	**2.038** ± 0.008	2.293 ± 0.021	2.222 ± 0.019	2.201 ± 0.018	2.105 ± 0.013	2.233 ± 0.019
Regrasp planning time (s)	**4.904** ± 0.230	7.201 ± 0.393	84.56 ± 0.827	90.10 ± 0.892	126.5 ± 1.029	72.00 ± 0.835	86.61 ± 1.040
Place Execution Success	0.412 ± 0.017	0.417 ± 0.017	0.395 ± 0.017	0.458 ± 0.017	0.422 ± 0.017	0.429 ± 0.017	**0.465** ± 0.017
Grasp Antipodal	0.484 ± 0.017	0.449 ± 0.017	0.450 ± 0.017	0.504 ± 0.017	0.472 ± 0.017	0.457 ± 0.017	**0.518** ± 0.017
Temporary Place Stable	0.704 ± 0.051	0.714 ± 0.125	0.533 ± 0.075	0.750 ± 0.083	**0.800** ± 0.082	0.778 ± 0.101	0.686 ± 0.080
Plan Length	2.514 ± 0.036	**2.094** ± 0.015	2.247 ± 0.024	2.167 ± 0.020	2.150 ± 0.019	2.118 ± 0.017	2.193 ± 0.022
Regrasp planning time (s)	**6.030** ± 0.237	8.484 ± 0.408	51.61 ± 1.113	58.56 ± 1.064	71.38 ± 1.333	50.92 ± 1.177	53.35 ± 1.159

TABLE II: Cost comparison for bin packing for (**top**) Test-1 (230 episodes) and (**bottom**) Test-2 (200 episodes).

	No Cost	Step Cost	GQ	MC	MC + GQ	CU	SP
Place Execution Success	0.727 ± 0.010	0.777 ± 0.009	0.856 ± 0.008	0.852 ± 0.008	0.861 ± 0.008	0.830 ± 0.008	**0.913** ± 0.006
Grasp Antipodal	0.833 ± 0.007	0.824 ± 0.009	0.906 ± 0.006	0.902 ± 0.006	0.908 ± 0.006	0.857 ± 0.008	**0.951** ± 0.005
Temporary Place Stable	0.785 ± 0.015	0.623 ± 0.067	0.700 ± 0.031	0.852 ± 0.022	0.784 ± 0.030	0.885 ± 0.029	**0.967** ± 0.012
Plan Length	3.061 ± 0.029	**2.079** ± 0.009	2.273 ± 0.016	2.286 ± 0.016	2.220 ± 0.014	2.157 ± 0.013	2.239 ± 0.015
Regrasp planning time (s)	**2.462** ± 0.061	6.413 ± 0.353	62.19 ± 0.326	117.6 ± 0.724	121.1 ± 0.577	54.88 ± 0.366	61.54 ± 0.900
Place Execution Success	0.446 ± 0.011	0.535 ± 0.012	0.520 ± 0.012	0.543 ± 0.012	0.566 ± 0.012	0.533 ± 0.012	**0.591** ± 0.011
Grasp Antipodal	0.585 ± 0.010	0.592 ± 0.011	0.612 ± 0.011	0.630 ± 0.011	0.650 ± 0.011	0.590 ± 0.011	**0.674** ± 0.010
Temporary Place Stable	0.690 ± 0.021	0.555 ± 0.046	0.608 ± 0.030	0.717 ± 0.032	0.621 ± 0.034	0.671 ± 0.036	**0.742** ± 0.027
Plan Length	3.265 ± 0.035	**2.323** ± 0.018	2.686 ± 0.025	2.501 ± 0.022	2.474 ± 0.021	2.419 ± 0.020	2.518 ± 0.023
Regrasp planning time (s)	**4.278** ± 0.156	14.84 ± 0.539	68.87 ± 0.657	99.36 ± 0.818	99.02 ± 0.819	60.05 ± 0.633	74.08 ± 0.732

TABLE III: Cost comparison for canonical task for (**top**) Test-1 and (**bottom**) Test-2 over $2,000$ episodes.

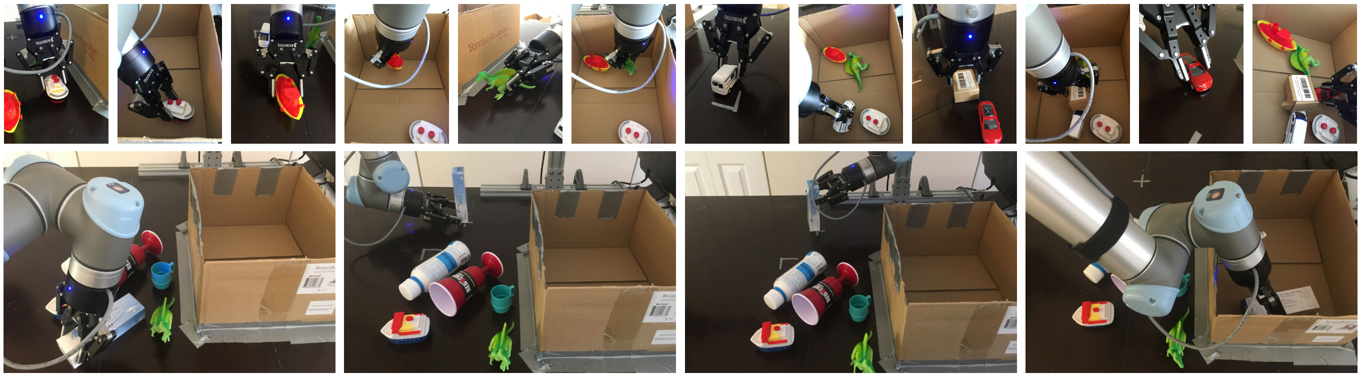


Fig. 5: **Top**. Example packing sequence. **Bottom**. Example situation requiring a regrasp.

	Step Cost	GQ	MC	SP
Place Success Rate	0.839 ± 0.027	0.833 ± 0.028	0.911 ± 0.021	**0.917** ± 0.021
Grasp Success Rate	0.883 ± 0.023	0.866 ± 0.024	**0.947** ± 0.016	0.933 ± 0.017
Grasp Attempts	196	201	207	210
Number of Regrasps	**17**	21	27	30
Packing height of 5 (cm)	7.333 ± 0.858	**7.050** ± 0.650	7.588 ± 1.132	7.711 ± 0.880

TABLE IV: Packing performance on the real robot. Showing average ± standard error over 30 episodes, each with 6 objects.

and (c) unlike the MC cost, which requires sampling and evaluating multiple shapes, SP is computationally cheaper. On the other hand, when shape completion is accurate, e.g., when trained with one category like bottles, the step cost is a reasonable choice as planning and execution is faster than SP.

	Shape Completion	HSA [10]
Number of Objects Placed	1.800 ± 0.074	1.667 ± 0.088
Task Success Rate	0.800 ± 0.074	0.667 ± 0.088
Grasp Success Rate	0.948 ± 0.029	0.983 ± 0.017
Place Success Rate	1.000 ± 0.000	0.900 ± 0.040

TABLE V: Bottles performance for the proposed method versus [10]. Showing average ± standard error over 30 episodes.

We note some limitations with our approach. First, the regrasp planner is much slower with a more sophisticated cost function than the step cost. This is because the step cost can exit the sampling loop when a two step plan is found, which occurs often in our experiments, while the other costs have no easy stopping criterion. Second, segmentation and completion accuracy is much lower with novel object categories. Third, integrating additional views to decrease uncertainty is an important aspect not considered.

Acknowledgements

We thank Yuchen Xiao, Andreas ten Pas, Mary Gualtieri, Lawson Wong, Chris Amato, and the anonymous reviewers for their insightful feedback on early drafts.

References

[1] M. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 1–28, 2018.

[2] P. Tournassoud, T. Lozano-Pérez, and E. Mazer, "Regrasping," in *IEEE Int'l Conf. on Robotics and Automation*, vol. 4, 1987, pp. 1924–1928.

[3] R. Alami, T. Siméon, and J.-P. Laumond, "A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps," in *Int'l Symp. on Robotics Research*, 1991, pp. 453–463.

[4] R. Alami, J.-P. Laumond, and T. Siméon, "Two manipulation planning algorithms," in *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, 1995, pp. 109–125.

[5] C. Nielsen and L. Kavraki, "A two level fuzzy prm for manipulation planning," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, vol. 3, 2000, pp. 1716–1721.

[6] A. Krontiris and K. Bekris, "Dealing with difficult instances of object rearrangement," in *Proceedings of Robotics: Science and Systems*, July 2015.

[7] W. Wan, H. Igawa, K. Harada, H. Onda, K. Nagata, and N. Yamanobe, "A regrasp planning component for object reorientation," *Autonomous Robots*, vol. 43, no. 5, pp. 1101–1115, 2019.

[8] M. Gualtieri, A. ten Pas, and R. Platt, "Pick and place without geometric object models," in *IEEE Int'l Conf. on Robotics and Automation*, 2018.

[9] M. Gualtieri and R. Platt, "Learning 6-DoF grasping and pick-place using attention focus," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87, Oct 2018, pp. 477–486.

[10] ——, "Learning manipulation skills via hierarchical spatial attention," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1067–1078, 2020.

[11] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kPAM: Keypoint affordances for category-level robotic manipulation," in *Int'l Symp. on Robotics Research*, 2019.

[12] L. P. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[13] L. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *The Int'l Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[14] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *IEEE Int'l Conf. on Robotics and Automation*, 2019, pp. 8241–8247.

[15] 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 *Proceedings of The 2nd Conference on Robot Learning*, vol. 87, Oct 2018, pp. 306–316.

[16] A. Morgan, K. Hang, W. Bircher, F. Alladkani, A. Gandhi, B. Calli, and A. Dollar, "Benchmarking cluttered robot pick-and-place manipulation with the box and blocks test," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 454–461, 2019.

[17] Y. Jiang, C. Zheng, M. Lim, and A. Saxena, "Learning to place new objects," in *Int'l Conf. on Robotics and Automation*, 2012, pp. 3088–3095.

[18] W. Gao and R. Tedrake, "kPAM-SC: Generalizable manipulation planning using keypoint affordance and shape completion," *arXiv preprint arXiv:1909.06980*, 2019.

[19] C. Mitash, R. Shome, B. Wen, A. Boularias, and K. Bekris, "Task-driven perception and manipulation for constrained placement of unknown objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5605–5612, 2020.

[20] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems*, 2010, pp. 2164–2172.

[21] B. Kehoe, D. Berenson, and K. Goldberg, "Toward cloud-based grasping with uncertainty in shape: estimating lower bounds on achieving force closure with zero-slip push grasps," in *IEEE Int'l Conf. on Robotics and Automation*, 2012, pp. 576–583.

[22] K. Hsiao, M. Ciocarlie, and P. Brook, "Bayesian grasp planning," in *ICRA Workshop on Mobile Manipulation*, 2011.

[23] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *IEEE Int'l Conf. on Robotics and Automation*, 2011, pp. 2845–2850.

[24] J. Mahler, S. Patil, B. Kehoe, J. van den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "GP-GPIS-OPT: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *IEEE Int'l Conf. on Robotics and Automation*, 2015, pp. 4919–4926.

[25] M. Laskey, J. Mahler, Z. McCarthy, F. Pokorny, S. Patil, J. van den Berg, D. Kragic, P. Abbeel, and K. Goldberg, "Multi-armed bandit models for 2D grasp planning with uncertainty," in *IEEE Int'l Conf. on Automation Science and Engineering*, 2015, pp. 572–579.

[26] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

[27] J. Lundell, F. Verdoja, and V. Kyrki, "Robust grasp planning over uncertain shape completions," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2019, pp. 1526–1532.

[28] R. Murray, Z. Li, and S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[29] M. Morari and J. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[30] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3D instance segmentation on point clouds," in *Advances in Neural Information Processing Systems*, 2019, pp. 6737–6746.

[31] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Int'l Conf. on 3D Vision*, 2018, pp. 728–737.

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

[33] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: Science and Systems*, vol. 9, no. 1, 2013, pp. 1–10.

[34] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The Int'l Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[35] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[36] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[37] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2016.

[38] 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 Int'l Conf. on Robotics and Automation*, 2019, pp. 3629–3635.

[39] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.

[40] F. Wang and K. Hauser, "Stable bin packing of non-convex 3D objects with a robot manipulator," in *Int'l Conf. on Robotics and Automation*, 2019, pp. 8698–8704.

[41] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, 2010.

[42] W. Wohlkinger, A. Aldoma, R. Rusu, and M. Vincze, "3DNet: Large-scale object class recognition from CAD models," in *IEEE Int'l Conf. on Robotics and Automation*, 2012, pp. 5384–5391.