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

Marcus Gualtieri and Robert Platt

***Abstract*— In this paper we consider joint perception and control of a pick-place system. It is important to consider perception and control jointly as some actions are more likely to succeed than others given non-uniform, perceptual uncertainty. Our approach is to combine 3D object instance segmentation and shape completion with classical regrasp planning. We use the perceptual modules to estimate their own uncertainty and then incorporate this uncertainty as a regrasp planning cost. We compare 7 different regrasp planning cost functions, 4 of which explicitly model probability of plan execution success. Results show uncertainty-aware costs improve performance for complex tasks, e.g., for a bin packing task, object placement success is 6.2% higher in simulation and 4.1% higher in the real world with an uncertainty-aware cost versus the commonly used minimum-number-of-grasps cost.**

## I. Introduction

*Pick-place* is prehensile manipulation where grasped objects are fixed in the hand as the arm moves and rest stably after being placed [1]. Owing to the simplifying, static nature of this problem, an interesting computational aspect has emerged: search can be separated into a discrete phase over grasp-place combinations and a continuous phase over arm motions, naturally enabling hierarchical planning [2], [3]. It has been more or less assumed that, a planner exploiting this property can be combined with a perception module to create a working system. However, perception of the object's geometry is a difficult and error-prone process in itself. Furthermore, separately designing perception and planning modules in this way is not necessarily optimal, e.g., these methods treat grasping a completely unobserved part of an object the same as grasping a part that is fully observed, which could clearly 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 learn a single module that does both. While some success has been achieved with this idea (e.g., [4], [5], [6]), 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 in [4] vs. a pipelined approach [7]).

Another approach is to plan in belief space, i.e., in probability distributions over the current state [8]. While this is the most principled approach, there are still a couple of important drawbacks. First, these methods almost always require a detailed description of the observation and state transition models of the system, which can be very difficult to obtain (e.g., [9], [10]). 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.

In this work, we take a different approach, which is to (a) use perception to predict the complete geometry of the objects and (b) incorporate instance segmentation and shape completion uncertainty as a planning cost at the level of discrete search. With only small modifications to existing components, we efficiently account for perceptual uncertainty. Our results for a bin packing task show that perception is indeed a significant source of error and that some of this error can be compensated for by penalizing uncertain grasps/places. We compare four different ways of modeling probability of pick-place execution success, including grasp quality (GQ), Monte-Carlo (MC) sampling, uncertainty at contact points (CU), and success prediction (SP) to two baselines. We find either SP or a combination of GQ and MC performs best, depending on the scenario. We test the applicability of the approach with real robot experiments on three benchmark tasks: block arrangement, bottle arrangement, and bin packing.

## II. Related Work

### A. Pick-place in deterministic, fully observed environments

Pick-place was often studied independently from perception. The structure of the problem for one movable object (called *regrasping*) 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 [11]. Nielsen and Kavraki gave a 2-level, probabilistically complete planner for manipulation planning [12]. Manipulation planning is related to the more general concept of multi-modal planning, which deals with discontinuities in the configuration space [13]. Wan et al. employ 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 [14]. For *non-monotonic* rearrangement problems (i.e., objects need moved more than once), a middle-level planner that displaced multiple objects

The authors are with the Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA (email mgualti@ccs.neu.edu).

was more efficient [15]. Our approach is to start with a well-established regrasp (i.e., middle-level) planner and build an uncertainty capability upon it.

### B. Pick-place of novel objects

A few projects have considered novel-object pick-place, where the complete shapes of the objects are no longer given. The first to address this was Jiang et al. [16] 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 [4], [5], [6]. 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 [7]. 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 was useful for avoiding collisions when planning arm motions with the held object [17]. 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 [18]. None of these considered multiple regrasps or compared different ways of explicitly accounting for segmentation and completion uncertainty, as we do here.

### C. Pick-place under uncertainty

A general approach to pick-place under arbitrary types of uncertainty is to solve a partially observable Markov decision process (POMDP). Kaelbling and Lozano-Pérez focus on symbolic planning in belief space with black-box geometric planners and state estimators [9]. Xiao et al. use POMCP [19] to update their belief about the arrangement of a small set of known objects [10]. Although the POMDP approach is very general, it requires significant computation and an accurate model of transition and sensor dynamics.

### D. Grasping under uncertainty

Our approach is to extend ideas from grasping under object shape uncertainty to pick-place planning. The two most common approaches to grasping under shape uncertainty are (a) evaluate force closure 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 and center of mass with given means and variances [20]. Hsiao et al. provide a probabilistic model for grasp success given multiple object detections and grasp quality evaluations [21]. Soon afterward, Gaussian process implicit surfaces (GPISs) were proposed as a representation of object shape uncertainty for grasping [22], [23], [24], [25]. GPISs combine multiple observations of the object's signed distance function (SDF) into a Gaussian process – a normal distribution over SDFs [22]. Mahler et al. compare a probabilistic model (based on the variance of the GPIS at contact points) versus an MC approach [23]. The MC approach does 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 [24]. Li et al. conducted real-world experiments filtering grasps with different thresholds on variance of the GPIS at contact points [25]. Finally, Lundell et al. represented objects as voxels, used a deep network to complete objects, and performed MC sampling using dropout [26].

## III. Problem Statement

*Definition 1:* Move-binary-effect system. A *move-binary-effect system* (cf. move-effect system [6]) is a discrete-time system consisting of a robotic manipulator, one or more depth sensors, and one or more objects, each situated 3D Euclidean space. The manipulator has configuration $q \in \mathbb{R}^d$ and is equipped with an effector with status *empty* or *holding*. The action of the robot is to move along a trajectory, $f : [0, 1] \rightarrow q$, followed by an effector operation, either *open* or *close*. At each step, the depth sensors acquire a point cloud $C \in \mathbb{R}^{n\times 3}$, sampling points on the object surfaces. The objects are rigid polyhedrons and can be either fixed or movable. At each step the robot observes $q$, its effector status, and a point cloud and takes an action.

*Definition 2:* Rearrangement of unknown objects. Within a move-binary-effect system, given a set of goal arrangements (where an *arrangement* is a pose for each movable object w.r.t. a fixed frame, $\{T_i \in \mathrm{SE}(3)\}_{i=1}^{n_{\mathrm{obj}}}$), find a minimum number of actions that is guaranteed to achieve a goal arrangement.

Since objects are unknown, the set of goal arrangements cannot be specified explicitly as a list of poses. Instead, it is specified with a boolean property, e.g., using first-order logic (e.g., all bottles are upright on coasters). Reasonable variations of this problem are also possible, such find a minimum number of actions that, with given probability, achieves a goal arrangement [9].

As this problem is PSPACE-hard [27], approximate solutions are needed. Our approach is to break the problem into two subproblems: (a) find (possibly multiple) sequences of explicit, single-object displacements (independent of how it is moved by the robot) that are likely to be executable and to achieve a goal arrangement and (b) find a regrasp plan that is most likely to achieve a single-object, goal displacement. For this paper, we focus on subproblem (b); subproblem (a) is implemented task-by-task.

## IV. System Overview

The proposed system for rearranging unknown objects 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 achieving a goal arrangement is found, and the first pick-place of the plan is executed. Automatic resensing and replanning accounts for failures, similar to MPC [28]. In this section, each component is briefly described; regrasping under segmentation and

completion uncertainty – the main contribution – is described next (Section V).

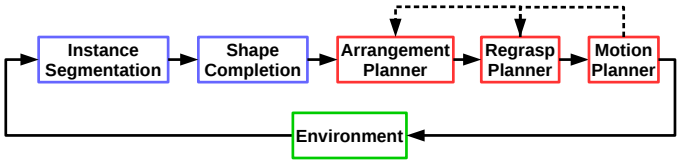


Fig. 1. Diagram of the proposed system architecture. Green represents the environment, blue the perceptual modules, and red the planning modules. Dashed arrows are followed up to a number of times if no plan is found.

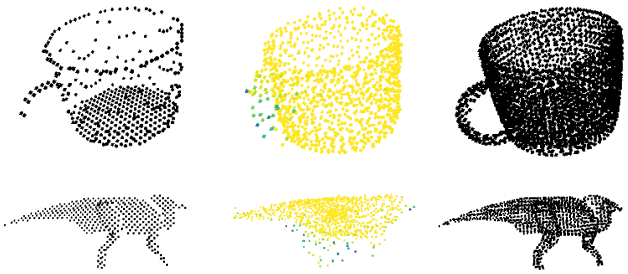

Fig. 2. Shape completions with PCN. Columns are (**left**) observed cloud, (**center**) completed cloud, and (**right**) ground truth. Yellow represents high $U$ values (near 1) and blue represents low $U$ values (near 0.5).

### 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 uncertainty so that 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 [4], [5], [6], exhibits good simulation-to-real domain transfer.

*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, $\{C_i \in \mathbb{R}^{n_i\times 3}\}_{i=1}^{n_{\text{obj}}}$ with $\sum_{i=1}^{n_{\text{obj}}} n_i \leq n$, and uncertainties $\{U_i \in \mathbb{R}^{n_i}\}_{i=1}^{n_{\text{obj}}}$. Although any object instance segmentation method with this interface can be used in the proposed architecture, our implementation uses BoNet [29]. BoNet produces an $n \times K$ matrix, where $K$ is a predefined maximum number of objects, and each row is a distribution for each point over object ID. The column-wise maximum value of this matrix is used for $U_i$, which is interpreted as the estimated probability each 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. For robust regrasp planning, we also require an uncertainty estimate for each completed point, $U \in \mathbb{R}^{\bar{n}}$. Although any shape completion method with this interface can be used in the proposed architecture, our implementation uses a modified version of PCN [30]. PCN consists of an encoder (two PointNet layers [31]) and a decoder (three fully connected, inner product layers[1]). 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. 2.

[1]The "detailed output" layers were omitted in our implementation, and the CD loss was used for the shape completion branch. (See [30].)

### B. Planning

We use a 3-level planner, similar to Wan et al. [14].

*1) Arrangement planner:* The input to the arrangement planner is a list of completed clouds, $\bar{C}_1, \ldots, \bar{C}_{n_{\text{obj}}}$, and the output is a set of triples $\{(T, c, i)_j\}_{j=1}^{n_{\text{goal}}}$, where $T$ is a target 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, $c$. For this paper, we implement a different arrangement planner for each task.

*2) Regrasp planner:* The regrasp planner takes in the triples from the arrangement planner and produces a sequence of pick-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 [32], and then RRT* with timeout [33]. 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 that of [2] to handle multiple goals for multiple objects, arbitrary additive costs, and discrete grasp/place sampling. Related planners (e.g, [3], [11], [12], [13], [15]) could also have been adapted to the purpose: the main point is to incorporate segmentation and shape completion uncertainty into the cost.

Alg. 1 is run in parallel for each object that has at least one goal pose. For $N$ steps (or until a timeout is reached), additional grasps (with costs $gc$) and temporary placements (with costs $pc$) are sampled, the regrasp graph is updated with the new samples, and A* with a consistent heuristic finds

**Input:** Completed cloud $\bar{C}$, per-point uncertainties $U$, goal poses and costs $\{(T,c)_j\}_{j=1}^{n_{\text{goal}}}$, and $costLowerBound$.

$RG \leftarrow []$
**for** $i \leftarrow 1,\ldots,N$ **do**
  $G, gc \leftarrow \text{SampleGrasps}(\bar{C}, U)$
  $P, pc \leftarrow \text{SampleTemporaryPlaces}(\bar{C}, U)$
  $RG \leftarrow \text{UpdateRegraspGraph}(RG, \{(T,c)_j\}_{j=1}^{n_{\text{goal}}}, G, gc, P, pc)$
  $plan, cost \leftarrow \text{A*}(RG)$
  **if** $cost \leq costLowerBound$ **then break**
**end**
**return** $plan$

**Algorithm 1:** Regrasp planner: run for each object. If using MC, replace $U$ with samples of the object's shape.

an optimal pick-place sequence, given the current samples. Grasps that are force-closure and temporary places that are stable, w.r.t. the completed cloud, are randomly sampled. The *regrasp graph* is represented by 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. 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.

It is desirable to choose grasps that fix the actual object in the effector as the arm moves and temporary places that, when the effector releases the object, the object rests stably. This way, the pick-place plan will execute predictably. For the case of a parallel-jaw gripper, an *antipodal grasp*[2] will fix the object (i.e., it is force-closure [34] p. 233). For the case of a fixed, horizontal support surface, conditions for a *stable placement* are given in [2]. While it is possible a goal placement can be achieved without antipodal grasps and stable placements, the outcome is more difficult to model and predict, so we conservatively try to avoid this situation.

### 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 $i$th grasp is antipodal, $P_i$ is the event the $i$th 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[3]. Taking the log and abbreviating $\Pr(G_i)$ as $g_i$ and $\Pr(P_i)$ as $p_i$ yields Eq. 3.

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

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

*1) Multi-criterion cost:* Negating Eq. 3 results in a non-negative cost. We account for additional factors in the regrasp planning problem, such as plan length and task cost, by adding these as objectives to a multi-criterion optimization problem ([35] pp. 181-184). Scalarization results in the cost (4), where $w_1,\ldots,w_4 \in \mathbb{R}_{++}$ 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 "grasp quality" evaluated on the nominal shape completion. For example, consider a distance measuring how far the line connecting the contact points is from the centers of both friction cones (cf. [34] pp. 233). Intuitively, if the line between contacts is near the edge of either friction cone, small errors in shape completion are likely to result in an incorrect antipodal assessment (see footnote 2).

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

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

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

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

[2]A parallel-jaw gripper forms an *antipodal grasp* on an object iff the line connecting the contact points lies inside both friction cones ([34] p. 233).

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

*2) Monte Carlo (MC):* Another approach is to estimate $g_i$ and $p_i$ via segmentation and completion samples. Suppose we are given a mechanism for sampling from the distribution $\Pr(\bar{C}_i|C)$, for $i = 1, \ldots, n_{\text{obj}}$, where $\bar{C}_i$ is a random variable over object shapes and $C$ is the input point cloud. Such a mechanism could be implemented as an ensemble of segmentation and completion networks, e.g., multiple networks trained with different weight initializations or datasets. Or, this could be implemented as a pair of networks with randomized components, e.g., dropout (as in [26] for grasping under uncertainty) or VAEs. Or, the option used here for sake of comparison to the CU method, one could use the point-wise uncertainty outputs of the networks ($U$ 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, assuming the $i$th point's offset from the nominal point is i.i.d. $\sim \mathcal{N}(0, \sigma_i^2)$, $\sigma_i$ is given by Eq. 7, since $U_i$ is the (estimated) probability the point is offset no more than $\beta$. To summarize, to sample a shape: (a) sample a segmentation point-wise using the segmentation mask and (b) compute the shape completion given this segmentation, and then, for each point, (c) sample a direction uniformly at random and (d) sample an offset along this direction from a normal distribution with 0 mean and standard deviation given by Eq. 7.

$$\sigma_i = \frac{\beta}{\sqrt{2}\text{erf}^{-1}(U_i)} \tag{7}$$

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

*3) Contact Uncertainty (CU):* Computing $g_i$ and $p_i$ using an MC method is expensive if $N$ is large. This motivates using the network uncertainties directly. The basic idea is to penalize grasp/place contact points with low $U$-values. Formally, suppose for the $i$th point, the segmentation network estimates $\Pr(S_i)$, where $S_i$ is the event the $i$th point is segmented correctly. Suppose the shape completion network estimates $\Pr(\mathcal{U}_i|S_i)$ where $\mathcal{U}_i$ is the event the $i$th point is within Euclidean distance $\beta$ 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 $\beta$ 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.

$$g_i \approx \Pr(\mathcal{U}_l|S_l)\Pr(S_l)\Pr(\mathcal{U}_r|S_r)\Pr(S_r) \tag{8}$$

$$p_i \approx \prod_{i=1}^{3} \Pr(\mathcal{U}_{t^i}|S_{t^i})\Pr(S_{t^i}) \tag{9}$$

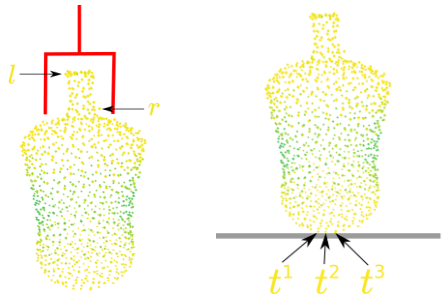


Fig. 3. **Left**. For an antipodal grasp (shown in red), there are at least 2 contact points, $l$ and $r$. **Right**. For a stable placement on a flat surface, there are at least 3 contact points, $t^1$, $t^2$, and $t^3$. Colors represent estimates of $\Pr(\mathcal{U}_i|S_i)\Pr(S_i)$, where brighter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

The uncertainty values from PCN (Section IV-A) are used to estimate $\Pr(\mathcal{U}_i|S_i)$. Estimating $\Pr(S_i)$ from the uncertainty values from BoNet (Section IV-A) is less straight-forward 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 estimated with a neural network. The encoding of grasp/place as input to the neural network is an important design choice that affects performance [36]. 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. [37]). 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

### A. Setup

We evaluated the proposed system in the environment shown in Fig. 4, left, consisting of a UR5 arm, Robotiq 85 gripper, and Structure depth sensor on the following tasks:

1) **Block arrangement.** Arrange 5 rectangular blocks from tallest to shortest according to the longest edge (reminiscent of "blocks world" [38]).
2) **Bottle arrangement.** Place 2 bottles upright onto 2 coasters (from our prior work [5], [6]).
3) **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 easer to evaluate than smallest bin size [40] in the real world.

In each case, same-category novel objects were tested. Novel-category objects were also tested for bin packing. Arrangement planners were designed separately for each task, without considering uncertainty. For brevity, we primarily discuss results for the most difficult task – bin packing. Similar trends are observed in the other tasks, but perception is more accurate, reducing the urgency to account for uncertainty.


Fig. 4. **Left**. Experimental setup. **Right**. 34 same-category novel objects for real-world packing experiments.

### B. Simulation results

The environment was simulated by OpenRAVE [41] using 3DNet objects [42]. Train/Test-1 categories included boat, bottle, box, car, dinosaur, mug, and wine glass. Test-2 (novel) categories included airplane, bowl, and stapler. Scenes were initialized without objects touching, as segmentation otherwise performs poorly. Grasps succeeded if (a) exactly 1 object intersected the hand closing region, (b) the antipodal condition with $24^\circ$ friction cone was met, and (b) the robot was collision-free. These conditions are conservative relative to reality, e.g., as the hand closes or the arm moves, an object in a non-antipodal grasp may rotate but still remain in the gripper and arrive near the goal. Unstable temporary places were recored, but otherwise ignored, because, in reality they never resulted in arrangement failures but required replanning. In Tables I and II, "Place Execution Success" refers to the proportion of successfully executed regrasp plans and "Packing Height of 5" refers to the end-of-episode packing height when 5/6 objects were placed.

*1) Perception ablation study:* The purpose of this study is to evaluate the perceptual modules in terms of pick-place performance and to quantify the potential benefit of accounting for uncertainty. 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.) Step and task costs were used, i.e., $w_1 = w_4 = 1$ and $w_2 = w_3 = 0$ in (4), where the task cost, $c$, was the estimated final packing height in centimeters.

The results (shown in Table I) are mostly as expected. A clear drop in performance is observed as perception becomes imperfect (down 18% for imperfect completion and another 4% for imperfect segmentation). A slight drop is noticeable from train to test objects. Without shape completion, planning is crippled (not shown in table, regrasp plan found rate drops from 94.1% for Percep. Test-1 to just 10.0%). A similar but more extreme trend is seen with Test-2.

*2) Regrasp cost comparison:* We compare four different ways of evaluating probability grasps (places) are antipodal (stable) (Section V-B) to two baselines – "No Cost", which takes the first regrasp plan found, and "Step Cost", which includes the step cost term only ($w_1 = 1$ in (4)). The step cost appears almost exclusively in the regrasping literature (e.g., [2], [3], [11], [14]). For simplicity, task cost is not included in this evaluation, but the 1st step of 16 packing solutions (top-2 solutions on each of 8 threads after 1 minute) are used as goal poses. "GQ + MC" refers to the case where GQ and MC costs are summed together.

Results for bin packing are shown in Table II. For Test-1, GQ+MC has the best grasp performance while SP has the best temporary place stability rate. GQ+MC significantly outperforms GQ (for 1-sided, same-variance, unpaired $t$-test, $p = 0.0092$ for 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 execution success and $p = 0.0023$ for grasp antipodal), and GQ+MC has the best temporary place stability rate. It was disappointing that CU did not significantly outperform the baselines for either dataset: it is apparently not sufficient to account for uncertainty only at the contact points. Also, while GQ has a significantly higher antipodal rate than the baselines for Test-1, the same is not true for Test-2, suggesting the GQ method can tolerate only small errors in shape completion.

For bin packing, we did not see a significant improvement for place stability over the step cost, but this is because regrasps were rare with the step cost, obscuring the significance of the results. To better test place stability, we designed a scenario with the same Test-1 objects used in bin packing, but where there is no bin and, for each episode, 1 of 5 objects, each with exactly 1 goal pose, has to be placed. The result is shown in 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 $2.9 \times 10^{-15}$, respectively). Interestingly, unlike with packing, CU has a significantly higher place success rate compared to step cost ($p = 1.3 \times 10^{-5}$), so all methods significantly outperformed the baselines in terms of place success rate for the canonical task (for SP vs. step cost, $p = 2.4 \times 10^{-33}$).

### C. Real robot results

The purpose of the real-world experiments is 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. To answer part (a), no domain transfer was needed for bin packing. For blocks and bottles, BoNet (but not PCN) severely overfit to simulation data. This problem was mitigated by using the network trained for bin packing for blocks and adding simulated sensor noise for bottles. For part (b), results for bin packing with step and MC cost are shown in Table IV. Although the MC method appears to be doing better, the gap is relatively small (4.1%). This may be because many non-antipodal grasps still succeed in placing the object into the bin, as we see the grasp success rates are higher in reality. An example packing sequence and a regrasp with blocks are shown in Fig. 5.

We also compare bottle arrangement performance to our previous method, which uses RL to learn a pick-place policy

| | 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 |
| 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 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 AND (**BOTTOM**) TEST-2. SHOWING AVERAGE ± STANDARD ERROR OVER 230 EPISODES FOR TEST-1 AND 200 EPISODES FOR TEST-2.

| | No Cost | Step Cost | GQ | MC | GQ + MC | CU | SP |
|---|---|---|---|---|---|---|---|
| Place Success | 0.725 ± 0.010 | 0.775 ± 0.009 | 0.854 ± 0.008 | 0.849 ± 0.008 | 0.860 ± 0.008 | 0.828 ± 0.008 | **0.911** ± 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 |

TABLE III

COST COMPARISON FOR CANONICAL TASK FOR TEST-1. SHOWING AVERAGE ± STANDARD ERROR OVER 2,000 EPISODES.

| | Step | MC |
|---|---|---|
| Place Success Rate | 0.867 ± 0.031 | 0.908 ± 0.026 |
| Grasp Success Rate | 0.908 ± 0.025 | 0.940 ± 0.021 |
| Number of Grasp Attempts | 131 | 134 |
| Number of Regrasps | 11 | 14 |
| Packing height of 5 (cm) | 7.4 ± 1.0 | 7.9 ± 1.2 |

TABLE IV

PACKING PERFORMANCE ON THE REAL ROBOT. SHOWING AVERAGE ± STANDARD ERROR OVER 20 EPISODES, EACH WITH 6 OBJECTS.

| | Shape Completion | HSA [6] |
|---|---|---|
| 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 [6]. SHOWING AVERAGE ± STANDARD ERROR OVER 30 EPISODES.

[6]. Many of the same bottles as before were included, but 4/15 of them were more challenging. Two of the bottles were difficult to distinguish orientation (size of tops near size of bottoms), and two were near the 8.5 cm gripper width. Results are shown in Table V. With the proposed method, all places were correct. Only the grasp success rate is lower than before, but all 3 grasp failures were with the wider bottles. Overall, we conclude the pipelined method performs much better (80% vs. 67% task success rate).

## VII. CONCLUSION

These results demonstrate that object instance segmentation and shape completion are accurate enough to enable difficult pick-place tasks such as bin packing. However, perceptual errors are still a major cause of failures. Some of these failures can be avoided by simply not grasping or placing on object parts where uncertainty is high. We formalize this idea with four different regrasp costs which account for perceptual uncertainty, GQ, MC, CU, and SP. We find SP or a combination of GQ and MC performs best and is more robust than the classical step cost.

To guide future work, we note some important limitations with the current system. First, the regrasp planner is much slower when using a more sophisticated cost function than the step cost. This is mainly due to having to sample plenty of grasps and places, to improve the likelihood the plan is executed successfully, without having a good stopping criterion. Another issue is to identify under what conditions the overall system (Fig. 1) is guaranteed to converge to a goal arrangement if a feasible path to one exists. Finally,

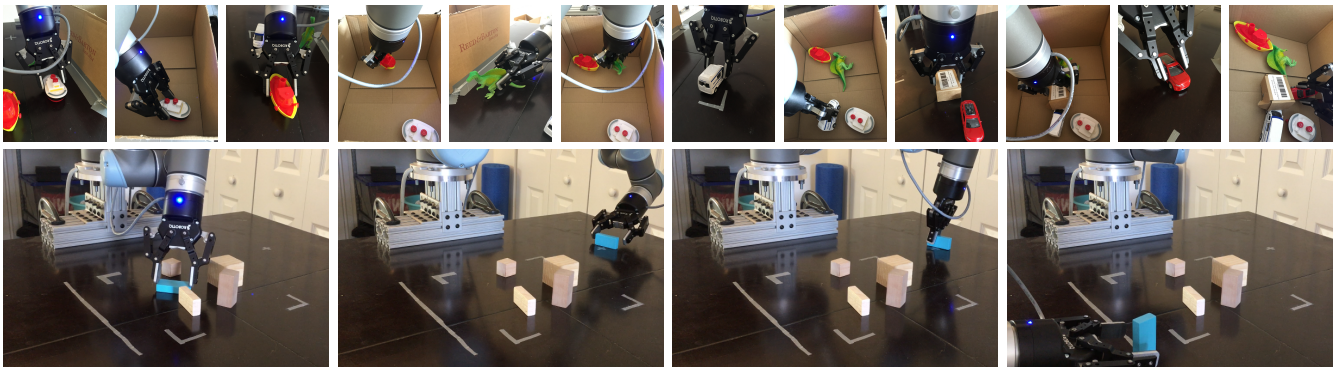

Fig. 5. **Top**. Example packing sequence. **Bottom**. Example block arrangement sequence requiring regrasping.

integrating additional views to decrease uncertainty is an important aspect that we do not examine here.

## ACKNOWLEDGEMENTS

We thank Yuchen Xiao and Andreas ten Pas for reviewing an early draft of this paper and Lawson Wong and Chris Amato for discussions during early stages of this project.

## 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*. Cambridge, MA, USA: MIT Press, 1991, pp. 453–463.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] L. P. 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.

[10] 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.

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

[12] 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.

[13] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The Int'l Journal of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[14] 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.

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

[16] 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.

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

[18] 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.

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

[20] 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 *Int'l Conf. on Robotics and Automation*. IEEE, 2012, pp. 576–583.

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

[22] 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*. IEEE, 2011, pp. 2845–2850.

[23] 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*. IEEE, 2015, pp. 4919–4926.

[24] 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*. IEEE, 2015, pp. 572–579.

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

[26] 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.

[27] G. Wilfong, "Motion planning in the presence of movable obstacles," *Annals of Mathematics and Artificial Intelligence*, vol. 3, no. 1, pp. 131–150, 1991.

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

[29] 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.

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

[31] 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*. IEEE, 2017, pp. 652–660.

[32] 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.

[33] 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.

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

[35] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[36] 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.

[37] 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*. IEEE, 2019, pp. 3629–3635.

[38] D. Chapman, "Penguins can make cake," *AI magazine*, vol. 10, no. 4, pp. 45–45, 1989.

[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*. IEEE, 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.