

# Computational Design of Passive Grippers

MILIN KODNONGBUA, IAN GOOD, YU LOU, JEFFREY LIPTON, and ADRIANA SCHULZ, University of Washington, USA

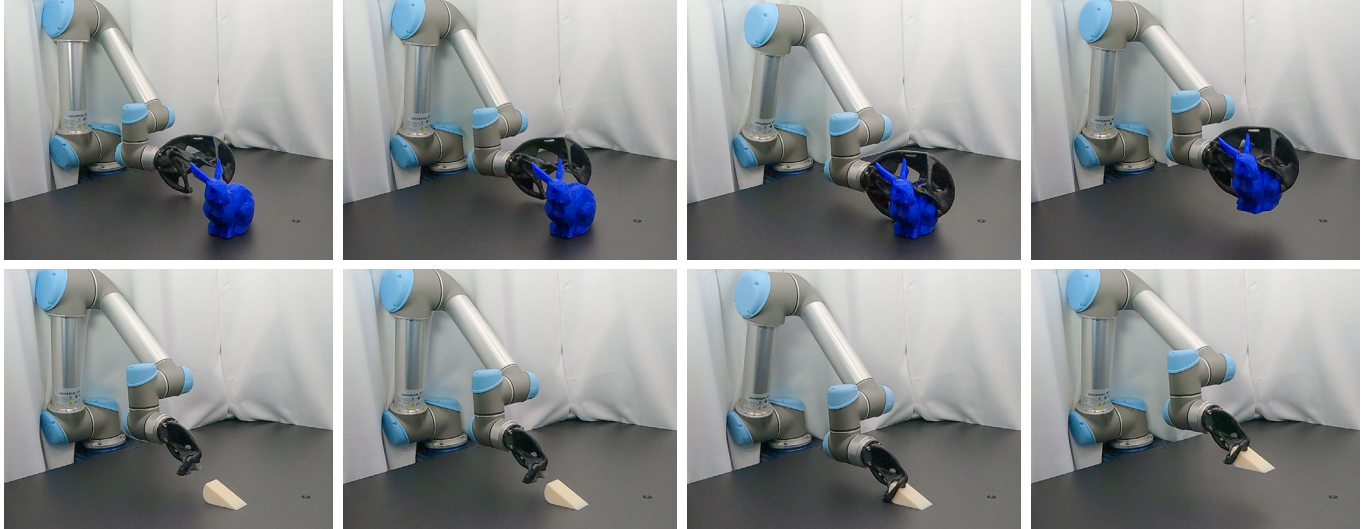


Fig. 1. We present an automated algorithm for designing passive grippers given a target object and its positioning. As our algorithm co-designs both the gripper shape and the insert trajectory, our approach broadens the space of shapes that can be passively grasped, compared to existing methods. The figure shows two of the 21 grippers (out of 23) that can successfully pick up the object in reality.

This work proposes a novel generative design tool for passive grippers—robot end effectors that have no additional actuation and instead leverage the existing degrees of freedom in a robotic arm to perform grasping tasks. Passive grippers are used because they offer interesting trade-offs between cost and capabilities. However, existing designs are limited in the types of shapes that can be grasped. This work proposes to use rapid-manufacturing and design optimization to expand the space of shapes that can be passively grasped. Our novel generative design algorithm takes in an object and its positioning with respect to a robotic arm and generates a 3D printable passive gripper that can stably pick the object up. To achieve this, we address the key challenge of jointly optimizing the shape and the insert trajectory to ensure a passively stable grasp. We evaluate our method on a testing suite of 22 objects (23 experiments), all of which were evaluated with physical experiments to bridge the virtual-to-real gap. Code and data are at <https://homes.cs.washington.edu/~milink/passive-gripper/>

CCS Concepts: • **Computing methodologies** → Shape analysis.

Authors' address: Milin Kodnongbua, [milink@cs.washington.edu](mailto:milink@cs.washington.edu); Ian Good, [iangood@uw.edu](mailto:iangood@uw.edu); Yu Lou, [louyu27@uw.edu](mailto:louyu27@uw.edu); Jeffrey Lipton, [jilipton@uw.edu](mailto:jilipton@uw.edu); Adriana Schulz, [adriana@cs.washington.edu](mailto:adriana@cs.washington.edu), University of Washington, Seattle, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2022 Association for Computing Machinery.

0730-0301/2022/7-ART149 \$15.00

<https://doi.org/10.1145/3528223.3530162>

Additional Key Words and Phrases: passive gripper, generative design, additive manufacturing, fabrication

## ACM Reference Format:

Milin Kodnongbua, Ian Good, Yu Lou, Jeffrey Lipton, and Adriana Schulz. 2022. Computational Design of Passive Grippers. *ACM Trans. Graph.* 41, 4, Article 149 (July 2022), 12 pages. <https://doi.org/10.1145/3528223.3530162>

## 1 INTRODUCTION

Passive grippers are end-effectors with no actuation. They leverage the existing degrees of freedom in a robotic arm to perform grasping tasks. Among the robotics literature, there has been a growing interest in such grasping techniques as they: reduce the manufacturing and deployment cost when compared to active grippers; have lower energy consumption since they do not require continuous power; and enable easier human-robot-interaction as anything passively grasped can be removed by a human without changing the robot state. A fundamental limitation to the widespread use of passive grippers, however, is that they are currently significantly restricted in the types of shapes that can be grasped.

Motivating this work is the insight that rapid-manufacturing and design optimization can be used to expand the space of shapes that can be passively grasped. We propose an algorithm that, given an input shape, can automatically generate a 3D printable custom gripper and an accompanying grasp insertion trajectory for stable pickup. This approach is directly applicable to a plethora of robotics applications that target repetitive tasks. For example, it can be used for task-specific tooling of robotic work cells in assembly lines,

allowing assembly lines instrumented with simple robotic systems (i.e. systems that lack dexterity) to be easily updated to respond to changes in the product. We can simply optimize and print new passive grippers overnight and re-purpose existing hardware.

The challenge with optimizing a passive gripper for a given input is that (1) there is a valid insert trajectory for making contact with the object and (2) once contact is made, the object can be lifted stably. This implies a large-scale search, nominally at the resolution of the 3D printer, where performance evaluation is expensive to compute since it involves physical evaluation and a nested optimization.

We address these challenges with two fundamental insights that allow us to reduce the complexity of this search. First, we observe that stability is a function of the contact points between the gripper and the object. We call this set of contact points the *grasp configuration* (GC) and we expose it as a design variable that should be optimized. We observe that, given the GC, an insert trajectory must ensure that there exists a gripper whose final insert configuration touches the GC and can be inserted without colliding with the rest of the shape. Based on this observation, our second insight is that we can search for such a trajectory by creating an abstraction of the gripper geometry that reflects such minimal requirements and then running a co-optimization over the space of trajectories and abstracted gripper shapes. The abstraction we propose is a parametric skeleton of infinitesimal thickness that connects the points on the GC to the *flange frame's origin* (FFO) of the robot, the center of where we mount the gripper. This reflects the requirement of a rigid object that is attached to the FFO and makes contact with the selected points.

From these two insights, one could argue for a two-step approach that optimizes the GC for stability and then, co-optimizes the trajectory and the gripper abstraction given the GC. The challenge, however, is that the chance of successfully finding a feasible insertion trajectory depends on the choice of the GC. We therefore propose a strategy for computing a ranked list of stable GC candidates that are likely to enable a collision-free insert trajectory. We use this ranked list of candidates for co-optimizing the trajectory and gripper abstraction. Once the trajectory and the GC are found, a straightforward modification of classic topology optimization algorithms can be used for computing the gripper geometry.

We evaluate our method over an experimental dataset of 23 examples, containing both representative samples from standard grasping datasets as well as a set of challenge models that are difficult to pick up. Our method finds solutions for all of these examples (see Figure 1). We evaluate the virtual-to-real gap by running real physical experiments in *all* 23 results. We are able to pick up all but two models whose 3D representation fundamentally differed from the real shape. The vast majority of models have high grasp reliability and a large range of stability, but for one we generated a solution with marginal stability.

## 2 RELATED WORK

*Robotic Co-design.* Robot design involves specifying both geometry and actuation sequences. Traditional methods start with general-purpose geometry and then customize the actuation. This is useful given the cost of manufacturing and the relative ease of re-programming. However, the revolution in digital fabrication and the resulting ease of customization has opened a new era of task-specific robot design. A body of recent work has shown the advantages of jointly optimizing a robot's shape and actuation for a variety of tasks such as ground locomotion [Digumarti et al. 2014; Ha et al. 2017; Luck et al. 2020; Spielberg et al. 2019; Zhao et al. 2020], flying [Du et al. 2016], swimming [Ma et al. 2021], and grasping [Chen et al. 2020; Deimel et al. 2017; Hazard et al. 2018; Pan et al. 2020; Xu et al. 2021]. Our work builds on this new trend, but instead of customizing a whole robot, we propose to enhance a general-purpose robot with customized end-effectors that can be rapidly fabricated, lowering the cost of customization.

*Generative Gripper Design.* Past work on generative gripper design has focused on active grippers. Antipodal grasping was an early target for computational design efforts [Brown and Brost 1999; Velasco and Newman 1998], with researchers developing shaped fingertips for antipodal grasping using direct formulaic approaches [Brown and Brost 1999; Honarpardaz et al. 2017a,b; Schroeffler et al. 2019; Velasco and Newman 1998] and neural network techniques [Ha et al. 2020]. Vacuum-based gripper design is another direction. The user specifies target locations, and then a 3D printable manifold and superstructure are generated for a specific object [Hendrixson 2016; Stevenson 2019]. Our work extends these ideas to passive grippers.

*Passive Grippers.* In literature, many grippers were classified as passive only because they passively conform to an object (as in the case of compliant [Crooks et al. 2017] and soft robotics [Seibel et al. 2020]). Some so-called passive grippers rely on actuators to release a grasp [Nagaoka et al. 2018; Petković et al. 2013; Seibel et al. 2020; Zhang et al. 2020]. Truly passive grippers have zero degrees of freedom and have no reliance on actuators. The most widely deployed passive grippers in industry are forklifts, but they can only be used on objects specifically designed for them, e.g., pallets, shipping containers, and FIBC bags. The passive grippers proposed by Mucchiani et al. [2018]; Mucchiani and Yim [2021] use a rotational motion to engage an antipodal grasp, but they can only pick up extruded cross sections (e.g., a cylinder), the key category not covered by our method. Other work has focused on passive deformations around objects to apply antipodal forces [Brodbeck and Iida 2015; Gupta et al. 2019].

Table 1 summarizes the object shape restrictions posed by different types of grippers. The forklifts can pick up objects around their center of mass (CoM) and through internal pickup (e.g., holes and handles). Active parallel jaw grippers or other passive grippers that generate antipodal grasps work well on a variety of objects because they can create opposing forces anywhere and do not rely on gravity at the CoM for the opposing force. However, they suffer from picking up objects via internal structures and objects that are antipodal resistant. For example, the bottom row of Fig. 1 shows an

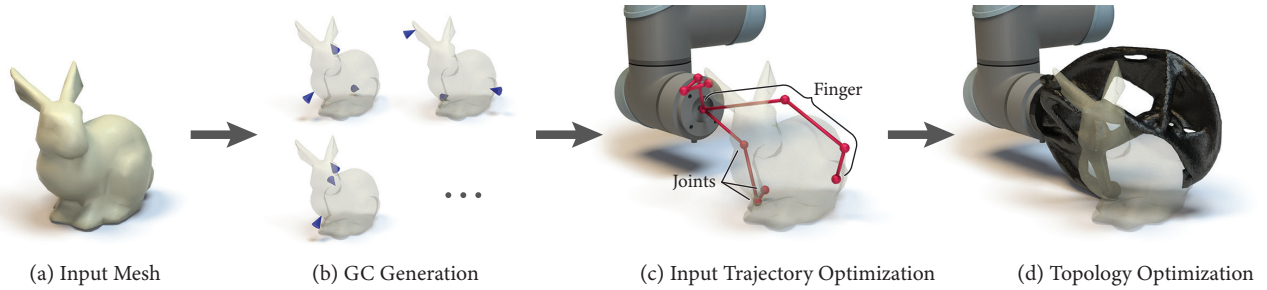


Fig. 2. Steps of our algorithm: (a) Import the object’s shape from a user provided file; (b) Generate multiple promising GC candidates (where the blue cones point to contact points); (c) Optimize gripper shape and trajectory by using a skeleton (shown in red) as the simplified gripper model; (d) Generate the final gripper shape (shown in black) using topology optimization.

Table 1. Comparisons of different grippers on types of objects they are able to pick up.

	Internal Pickup	Antipodal Resistant	Grips Outside CoM	Extruded Cross Section
Ours	✓	✓	✓	
Non-Actuated Fork Lift	✓			
Mucchiani et al. 2018/2021			✓	✓
Antipodal (active or passive)			✓	✓
Antipodal (active) + Custom fingertips		✓	✓	✓
Vacuum-based (active)		✓	✓	✓

object that is graspable outside its CoM and is antipodal resistant. Parallel jaw grippers with custom fingertips may be able to pick up the antipodal resistant part due to their large conformable gripping area. Vacuum-based gripper can pick up a wide variety of objects with large enough flat surface. We stress that Table 1 highlights *characteristics* of objects, not a strict classification. By covering a larger number of characteristics, our work significantly expand the space of objects that can be passively grasped.

*Grasping.* All grippers require a grasp planning step and numerous methods have been applied to solving the problem. A fundamental metric for evaluating a grasp is force closure. A grasp is force closed if it can resist any arbitrary set of forces and torques [Bicchi and Kumar 2000; Bruyninckx et al. 1998]. A grasp can be partially force closed if it resists all but a subset of forces and torques [Kruger and van der Stappen 2011a]. An alternative is to evaluate the caging of an object kinematically, in which an array of obstacles is placed around an object to limit the range of motion of the object [Rodriguez et al. 2012; Seo et al. 2016]. The downside is that one cannot be sure where the object is relative to the robot when grasped, therefore we chose to focus on a grasp being force closed.

The challenge with physics-based methods for grasp point selection is the need to model noise. Deep learning techniques have been shown to generate robust grasp poses both in the constrained 2.5D bin picking setting [Mahler et al. 2017] and even the full cluttered 6D grasp pose setting [Maitin-Shepard et al. 2010; Mousavian et al. 2019; Murali et al. 2020; Qin et al. 2020; Seita et al. 2019; ten Pas et al. 2017]. An alternative is to learn visomotor policies directly from raw visual data [Levine et al. 2016; Nagabandi et al. 2019; Rahmatizadeh et al. 2018]. While learning methods successfully generate grasp candidates, they are trained primarily for antipodal and vacuum grippers and cannot be directly applied to passive grippers.

### 3 OVERVIEW

Our algorithm takes an object’s geometry and positioning, along with a robot’s kinematic structure as input, and generates (1) a passive gripper geometry that can be 3D printed and (2) a collision-free insert trajectory for creating a stable grasp.

Our algorithm involves three steps, as illustrated in Figure 2. First, we generate a ranked list of GC (grasp configuration) candidates. Next, we search for a collision-free trajectory for a given GC. The trajectory search is done by jointly optimizing the trajectory and a gripper abstraction. Finally, given the GC and the insert trajectory, we optimize the gripper shape to minimize compliance and weight using topology optimization.

As discussed, GC selection is uniquely responsible for final stability. However, it also impacts the likelihood for finding a valid insert trajectory in the second step and the possible trade-offs between compliance and weight in the third step. We consider four metrics: (1) a binary metric that validates static stability under gravity; (2) a measurement of robustness—the minimum disturbance force and torque that make the object unstable; (3) a heuristic to identify GCs that are unlikely to have a valid insert trajectory; and (4) an estimate of the final gripper weight. We use (1) and (3) to prune randomly sampled GCs and use (2) and (4) to rank the remaining GCs.

The next step is to optimize the insert trajectory to avoid collisions with the object. We consider one GC from the ranked list at a time and proceed to the next best GC if we fail to find the trajectory. Collision avoidance depends on both the trajectory and the gripper’s overall shape, making it the most challenging objective. We first simplify the problem by ignoring the structural cost. We observe that the geometric aspect that affects collision is how the contact points are connected to the FFO and how the connecting structure moves through time. We propose to represent the gripper geometry using a *skeleton* (See Fig. 2c.), which is parameterized curves that connect the FFO to each grasp point. We then jointly optimize over the space of skeletons and trajectories by minimizing a cost function associated with collision. We also add a trajectory complexity regularization term to the cost function to reward a simpler trajectory which reduces the chance of collision, leaves more collision-free space to optimize gripper shape, and lowers the robot’s energy consumption.

The final step is to compute the gripper geometry. While the skeleton could serve as a workable gripper, it is too fragile. An

alternative is to compute the volumetric region that does not intersect with the target object during insert motion, and use this full *collision-free volume* as the gripper. However, this gripper is too heavy. We therefore propose to use topology optimization to generate the final gripper geometry, simultaneously minimizing compliance and weight, while constraining the gripper to lie in the collision-free volume (See Fig. 2d).

#### 4 GC CANDIDATES GENERATION

Recall that a GC (grasp configuration) is a set of three contact points on the target object. We choose three because it is the minimum number of points that constrains a rigid body. More contact points will kinematically over-constrain the solution, make it harder to find a collision-free trajectory, and make the gripper more sensitive to the approach direction [Blanding 1999]. Our proposed strategy for generating this list of candidates involves three steps. First, we propose a method for sampling GCs. Our sampling scheme identifies points on the surface that do not make contact with the floor and that can be directly connected to the FFO without colliding with the object; sets of three identified points are selected uniformly at random. We then drop GCs that are not statically stable under gravity (Section 4.1) and the ones that we identify as *unreachable*—i.e., candidates that are unlikely to have a collision-free gripper shape and grasp trajectory (Section 4.2). Finally, we rank the remaining GCs to maximize the robustness to external disturbances once the object is grasped, while trading-off compliance and weight of the gripper shape by using another heuristic metric (Section 4.3).

##### 4.1 Static Stability Under Gravity

Each contact point makes a contribution to the object’s stability by exerting forces onto the object at different angles and location. We assume Coulomb’s model of friction that limits the angle with respect to the surface at which the forces can be exerted. For contact point  $i$ ,  $|\mathbf{f}_i^T| \leq \mu_i |\mathbf{f}_i^N|$  where  $\mathbf{f}_i^T$  and  $\mathbf{f}_i^N$  are the tangential and normal forces exerted, respectively; and  $\mu_i$  is the coefficient of friction at that point. This constraint can be viewed as a cone and can be approximated using a polyhedral cone with  $q$  sides where the allowed force is the non-negative linear combination of the edges of the polyhedral cone, called basis. Let  $\mathbf{w}_{ij}$  be the wrench (force and torque) generated by  $j$ -th basis of the cone at point  $i$ . The contribution from the  $i$ -th contact point can be written as  $\sum_{j=1}^q k_{ij} \mathbf{w}_{ij}$  where  $k_{ij} \geq 0$ .

We determine if the GC is stable using *partial force closure*, which specifies whether a grasp can withstand a particular external disturbance (in our case, gravity) [Kruger and van der Stappen 2011b]. Formally, the partial force checks if there exists  $k_{ij} \geq 0$  such that  $\sum_i^3 \sum_j^q k_{ij} \mathbf{w}_{ij} = [\mathbf{g} \ 0]^T$  where  $\mathbf{g}$  is the unit vector of gravity. We say a GC is *stable* if it meets the partial force closure condition.

There is a slight problem with using this formulation of force contribution for passive grasping because the force of gravity does not directly cause top-facing contact points to generate forces, instead, the gripper only generates a contact force if there is a torque around the center of mass. To account for this, we set the frictional force generated by the top-facing contact points to zero and set those generated by the bottom-facing points depending on the angle with

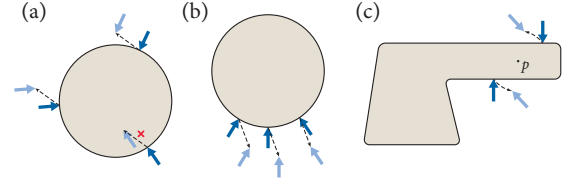


Fig. 3. Illustrations of the reachability heuristic. The dark blue arrows point to the contact points. The dashed black arrows show a candidate rigid motion to break all contact points simultaneously. (a) contact points cannot instantaneously break contact with the circle; (b) contact points can break the contact by translation; (c) contact points can leave the object by rotating counterclockwise around the point  $p$ .

respect to the ground. Formally, we set  $\mu_i = \max(0, \mathbf{n}_i \cdot \mathbf{g}) \mu$  where  $\mu$  is the base coefficient of friction.

There is a stronger condition called *force closure* which specifies whether a grasp can withstand any external forces and torques [Bicchi and Kumar 2000; Bruyninckx et al. 1998]. However, this type of grasp which typically involves contact points on different sides of the object makes breaking contact more difficult. We observe that, by utilizing gravity, the object can still be held stable without a force closure grasp. Hence, the partial force closure condition is sufficient for our purpose.

##### 4.2 Identifying Unreachable GC

Given a GC, it is expensive to determine if a collision-free trajectory exists—it involves solving the co-optimization problem in Section 5. Thus, we need a fast heuristic that identifies unreachable GCs. Our insight is to check if there exists an instantaneous motion where all contact points simultaneously break contact with the object.

We will motivate our heuristic in 2D for ease of illustration; these ideas extend naturally to 3D. Imagine grabbing a circle at three equally spaced locations in a 2D space (see Fig. 3a.). It is trivial to see that there is no rigid motion to break all three contacts simultaneously without colliding with the circle, so we can drop it without running the expensive optimization. However, if the contact points are roughly on the same side (see Fig. 3b.), then we can break the contacts just by moving the contact points away; hence, we keep this GC. One might be tempted to just check if the normals of each contact point are pointing in the same hemisphere. This is too strict. Consider grabbing the object in Fig. 3c. The two contacts’ normals are pointing in the opposite direction. However, this GC is reachable since we can rotate counterclockwise around the midpoint between the two contact points.

Our heuristic searches for an instantaneous rigid motion (translation and rotation) so that all contact points simultaneously sufficiently break contact with the object. We define the instantaneous motion of the gripper as  $(\mathbf{v}, \boldsymbol{\omega}, \mathbf{c})$ , where  $\mathbf{v}$  is the instantaneous translational velocity and  $\boldsymbol{\omega}$  is the instantaneous angular velocity around the rotation center  $\mathbf{c}$ . We can derive the instantaneous velocity of a single contact point as  $\mathbf{v}_i = \mathbf{v} + \boldsymbol{\omega} \times (\mathbf{c}_i - \mathbf{c})$  where  $\mathbf{c}_i$  is position of the contact point. The contact point breaks contact if  $\mathbf{v}_i \cdot \mathbf{n}_i \geq 0$  where  $\mathbf{n}_i$  is the normal direction at the contact point. We also limit the angle between  $\mathbf{v}_i$  and  $\mathbf{n}_i$  to  $\theta_{\max}$  to avoid nearly parallel motion to



the surface. Formally, if there does not exist a motion  $(v, \omega, c)$  such that  $v_i \cdot n_i \geq \|v_i\| \cos(\theta_{\max})$  for all  $i$ , then the GC is *unreachable*.

Inspired by the barrier method [Fiacco and McCormick 1964], we construct a loss function that penalizes motions that violate the constraints. We then solve the following minimization problem for a valid motion using a standard gradient-based optimization:

$$\min_{(v, \omega, c)} \sum_i [[\cos(\theta_{\max}) - v_i \cdot n_i]_+ + [\|v_i\| - 1]_+]$$

where  $[\cdot]_+$  denotes  $\max(0, \cdot)$ .

Note that we do not normalize  $v_i$  to avoid numerical instability when the value approaches zero. The angle between two vectors will not exceed  $\theta_{\max}$  because we also constrain  $\|v_i\| \leq 1$ . A GC is said to be *unreachable* if the loss function does not converge to zero.

### 4.3 Ranking

The final step is to produce a ranked list of candidates for the next stage. We consider two metrics.

The first metric evaluates the minimum disturbance force that causes the object to become unstable. This metric measures how stable the grasp will be after the object is picked up and experiencing the transfer motion. Inspired by the well-known minimum wrench metric, we introduce a new metric called *partial minimum wrench*, which measures the minimum additional external wrench to violate the partial force closure condition. Intuitively, this metric corresponds to the maximum tilt angle and/or acceleration that an object held by this GC can experience before falling off. A higher partial minimum wrench is better.

The second metric is the estimated finger length. If the contact points are far from the FFO, the gripper will be cumbersome and the path to reach the point will be long. We therefore estimate the finger length by finding the shortest non-colliding path from the FFO to each contact point and taking the maximum path length. A shorter length is better.

Since the two metrics can be conflicting, our ranking scheme is inspired by the non-dominated sorting criteria in multi-objective optimization [Deb et al. 2000]. We segment the candidates into multiple Pareto frontiers, and sort them by the estimated finger length within each frontier.

## 5 TRAJECTORY OPTIMIZATION

In this stage, for each GC candidate, we want to search for an insert trajectory that avoids collision. Since collisions depend both on the gripper's shape and the trajectory, we need to design them together. The fundamental challenge in the co-design of a gripper and trajectory is the complexity of the search space. The nominal search space of the gripper geometry corresponds to the resolution of the voxel grid that a 3D printer can afford, and the search space of a trajectory can be described by the degrees of freedom of a robotic arm over time. We propose a novel reduced representation of the search space to make this search tractable.

### 5.1 Gripper Abstraction

We represent a gripper design as a *skeleton* which is comprised of three *fingers*, each with multiple *joints* connecting the FFO to the three contact points (Fig. 2c). This representation stems from the

weakest constraint that the grasping points need to be connected to the FFO. We use a skeleton of infinitesimal thickness to evaluate collisions over 2D curves as opposed to volumes; we account for manufacturability by expanding the mesh by an offset that corresponds to half of the printer resolution.

The insert trajectory is represented as a linear interpolation between a list of robot states in joint space. We call these robot states the *keyframes* of the trajectory. The first keyframe defines the robot when the gripper is outside of the object's proximity, and the last keyframe defines the robot when making contacts with the target object.

We define the gripper skeleton as  $\mathcal{G} \in \mathbb{R}^{3 \times m \times 3}$  where  $m$  is the number of joints in each finger and define the trajectory as  $\mathcal{T} \in \mathbb{R}^{n \times d}$  where  $n$  is the number of keyframes and  $d$  is the robot's degrees of freedom. Our goal is to optimize the tuple of the gripper skeleton and the trajectory:  $(\mathcal{G}(\mathbf{x}), \mathcal{T}(\mathbf{x}))$  where  $\mathbf{x} \in \mathbb{R}^N$  are the adjustable parameters, namely the intermediate joint positions for every finger, and intermediate trajectory keyframes, for a total of  $N := 3 \cdot (m-2) \cdot 3 + d \cdot (n-2)$  parameters. In our implementation, we set  $m = 4$ ,  $n = 4$  and  $d = 6$ , resulting in 30 degrees of freedom in total. We denote  $(\mathcal{G}(\mathbf{0}), \mathcal{T}(\mathbf{0}))$  as the initial guess from the initialization method described in Section 5.2. We specify appropriate ranges  $\sigma \in \mathbb{R}^N$  to each of the adjustable parameters (i.e.  $-\sigma_i \leq x_i \leq \sigma_i$ ) to ensure the search space is connected and the solution remains valid (e.g. the robot does not self intersect and the skeleton maintains its overall shape). In the following discussion, we omit  $\mathbf{x}$  and hence write  $\mathcal{G}$  and  $\mathcal{T}$  to mean  $\mathcal{G}(\mathbf{x})$  and  $\mathcal{T}(\mathbf{x})$ .

### 5.2 Co-Optimization of Trajectory and Gripper Skeleton

To co-optimize the trajectory and the gripper skeleton, we propose an objective function with four energy terms: gripper collision energy  $E_g$ , trajectory collision energy  $E_t$ , robot collision energy  $E_r$ , and the trajectory regularizer  $L$ . The first three terms focus on collision, while the regularizer penalizes complex trajectories.

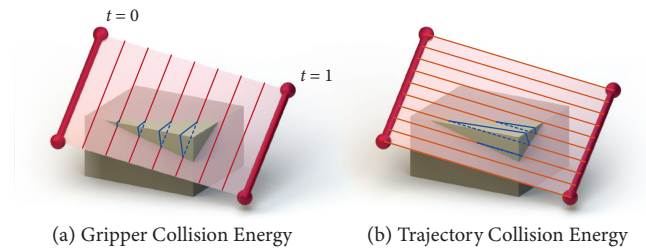


Fig. 4. Our collision energy measures the collision of the surface swept by the skeleton over trajectory shown in pink. We compute collision using path intersections along two directions: In (a), the path along the skeleton is tested at different time steps in the trajectory (shown by vertical red lines); In (b), the path along the trajectory is tested at different points on the skeleton (shown by horizontal orange lines). Collisions by the paths are shown in blue. Maximum collision of the paths in (a) defines the gripper collision energy, and likewise maximum of those in (b) defines the trajectory collision energy. Note: a simple box and a linear path is used for simplicity of the visualization.

The gripper collision energy is defined as the maximum collision at any point of the trajectory measured over the whole gripper skeleton. Similarly, the trajectory collision energy is defined as the maximum collision of a single point in the gripper measured over the whole trajectory. These two metrics are illustrated in Figure 4. We note that, in theory, either one of these metrics can individually represent the collision error we target. In practice, however, we cannot directly compute them, but must instead evaluate them over a discretization of the trajectory and skeleton. Our insight for combining these metrics is that they describe the two basis directions for the surface, making the evaluation over a discretization robust to small features.

Both metrics depend on a measurement of collision over a path: the collision of the skeleton at any point can be expressed as the sum of the collision of its fingers (piece-wise linear paths) and the collision of a point throughout the whole trajectory is measured from the path this point traces. We will describe our measurement of collision over a path later in this section.

Since the robot end-effector cannot go lower than a certain clearance height  $h$ , we introduce a robot collision energy term  $E_r$  which is the maximum penetration into the ground by FFO's position. We penalize the complexity of the trajectory with the L2 norm of the trajectory variations from the initial guess (i.e. the trajectory components in  $\mathbf{x}$ ).

Our optimizer minimizes the following energy term:

$$\min_{\mathbf{x}} E_g(\mathcal{G}, \mathcal{T}) + E_t(\mathcal{G}, \mathcal{T}) + \lambda_1 E_r(\mathcal{T}) + \lambda_2 L(\mathcal{T})$$

where  $\lambda_1$  and  $\lambda_2$  are the significance of the robot collision energy and the trajectory regularizer. Since this objective is non-differentiable, we use controlled random search (CRS) with local mutation for minimizing this energy [Kaelo and Ali 2006].

*Collision Along a Path.* There are several approaches to compute the collision of a given path. One trivial approach is to use the length of the path that lies inside the object. We call this length the *inside distance*. However, this approach does not capture the degree of collision, as illustrated in Figure 5. We propose to include a measurement of the shortest additional distance to make the path collision-free. We call it the *wrap-around distance* (See Fig. 5). To measure the wrap-around distance, we compute the geodesic distance between a pair of points when the path enters and exits the object. We take the sum of the geodesics if the path enters and exits multiple times, and we do not include the geodesic if one or both endpoints are in the object. The collision of a path is defined as the sum of the inside and wrap-around distance (See Fig. 6).

*Discretization.* We compute  $E_g$  and  $E_t$  by sampling the trajectory and skeleton, respectively. We adaptively subdivide the trajectory so that the skeleton moves by at most some distance  $d_{\text{sub}}$  and we adaptively subdivide the fingers by that same distance. This sampling is repeated for every iteration as lengths of individual segments can significantly change when adjustable parameters change.

To find where the path enters and exits the object, we use a ray-tracing algorithm to find intersections. Since the path for trajectory is not linear, we adaptively subdivide the path with some linearity tolerance  $d_{\text{lin}}$  into connected line segments.

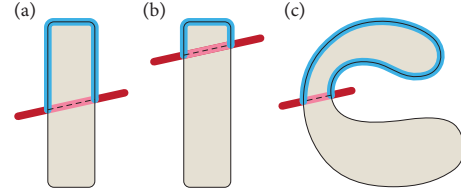


Fig. 5. Illustrations of the *inside distance* (pink) and *wrap-around distance* (cyan) of the skeleton (red) through some objects. The path in (b) is closer to a collision-free state than that in (a), but the inside distance is the same. The path in (c) has lower inside distance than (a) and (b), but it needs more work to reach a collision-free state.

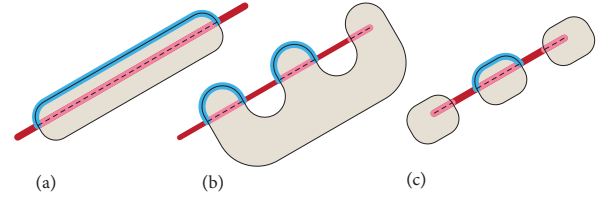


Fig. 6. The *wrap-around distance* is the sum of the geodesics (cyan) between every pair of entering and exiting intersections. The path in (a) has one such pair and in (b) has two. Some endpoints in (b) and (c) do not have a pair and therefore the geodesics are not defined.

To efficiently compute the geodesic distance between any given two points, we need to allow some accuracy trade-offs. We approximate the geodesic distance by running a standard shortest path algorithm on the vertices and edges of the triangle mesh. We use the isotropic remeshing algorithm to generate a new mesh whose edge lengths are as close to one another as possible. This makes the approximation more accurate. We precompute the shortest distance between every pair of vertices. The approximate geodesic distance for any given two points on the surface is computed as the distance of those two points to their closest vertices plus the precomputed geodesics between the two vertices.

*Initialization.* We initialize the skeleton by computing the shortest, non-colliding curves that connect from the FFO to each contact point and then simplifying these curves to the right number of joints by discretizing the curves, expanding them by the surface normal, and removing vertices that does not contribute to collision avoidance. We initialize the trajectory to be a straight line towards the object.

## 6 TOPOLOGY OPTIMIZATION AND REFINEMENT

At our last step, we generate the gripper design by performing discrete topology optimization over the *collision-free* volume [Bendsoe and Sigmund 2003]. Since the trajectory is known, this volume can be computed as the complementary space of the swept volume of an object moving away from the gripper (See Fig. 7). For the boundary conditions, we set the external forces to be the forces exerted by the contact points of the known GC along the normal directions and fix parts of the gripper around the FFO. In post-processing, we apply a smoothing kernel and run marching cubes to retrieve a smooth mesh. We further refine the gripper geometry by adding a small sphere at each contact point and subtracting the result with the

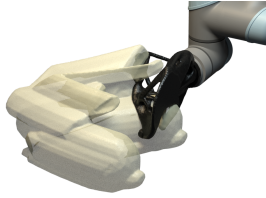


Fig. 7. In the robot's reference frame, the swept volume of the bunny (shown in white) is the space occupied by the bunny throughout the trajectory. The complementary space is the *collision-free* space for topology optimization. The topology optimized gripper is shown in black.

swept volume to ensure an accurate geometry around the contact points. This increases the area of contact and improves the robustness of the grasp. Lastly, we add robot-specific mounting structures such as holes and a mounting plate for fast installation.

## 7 RESULTS

We evaluate our algorithm on its success rate and demonstrate different designs fabricated in real. We refer the reader to the supplementary video for demonstrations of the grippers in motion.

### 7.1 Evaluation Set

We created a testing set of 22 objects (See Fig. 8).

Five models are engineered to demonstrate the unique capabilities of the generated passive grippers. A1-A3 are designed to represent internal structures. A1 contains an L-shaped hole at the front. The pyramid (A2) contains a curved hole that requires a twist motion



Fig. 8. Evaluation set. A: engineered models; B: bunny; C: challenge models; D: samples from YCB dataset.

to insert. The *Top Key* (A3) contains a rectangular slot at the top and requires a quarter turn to insert. The pallet (A4) is designed to test if our algorithm can generate a fork-lift. The wedge (A5) is engineered to resist an antipodal grasp because of its tapered shape at all orientations. To the best of our knowledge, objects (A2) and (A5) cannot be passively picked up by any prior work.

In addition to our engineered models, we include a representative set from prior work. First, to uphold the SIGGRAPH tradition, we add the Stanford bunny (B) to our evaluation set. We choose some challenging objects (C1-C8) from the Fit2Form dataset [Ha et al. 2020] representing adversarial objects from Dex-Net 2.0 (C1,C2, C4,C6-8) [Mahler et al. 2017] and objects from ShapeNet (C3, C5) [Chang et al. 2015]. We pick a subset of YCB objects [Calli et al. 2015a,b] that are feature-rich and contain no moving parts (D1-D8).

The input to our algorithm is the shape and its positioning. We run two experiments with (A1) using orthogonal positions: inserting from the front and from the side, resulting in a total of 23 experiments. We refer to them as *Front Key* (A1-f) and *Side Key* (A1-s).

### 7.2 Implementation and Experimental Setup

Our algorithm is implemented in C++, using libigl [Jacobson et al. 2018] and CGAL [The CGAL Project 2021] for most mesh processing tasks. We use UR5 as our robot, which has six degrees of freedom. We use the code from Sellán et al. [2021] to calculate swept volume, and we use ToPy [Hunter et al. 2017] to run topology optimization with the voxel size of 2 millimeters. We ran the algorithm on all 23 experiments on a cluster with 40 cores.

In our GC generation stage, we randomly sample 1,000 points on the mesh and generate 3,000 GCs. We assume the base coefficient of friction  $\mu$  is 0.5 in modeling the contacts. For the reachability heuristic, we set  $\theta_{\max}$  to  $80^\circ$ . In computing collision of a path, we set both the subdivision distance threshold ( $d_{\text{sub}}$ ) and the linearity threshold of the trajectory ( $d_{\text{lin}}$ ) to 1 millimeter. In computing the total objective function, we set the robot floor clearance ( $h$ ), the robot energy significance ( $\lambda_1$ ), and the regularizer significance ( $\lambda_2$ ) to 0.05 meters, 1000, and  $1e-6$ , respectively. In the optimization stage, we use NLOpt's implementation of CRS for trajectory optimization [Johnson 2014] with the population size of 10,000 and relative tolerance of  $1e-6$ . We allow the intermediate finger joints to vary within 1 centimeter, and we allow the six joints to deviate within  $5^\circ$ ,  $5^\circ$ ,  $5^\circ$ ,  $45^\circ$ ,  $25^\circ$ , and  $90^\circ$ , respectively.

We set the robot grasping position (the last keyframe) so that the end effector is pointing forward and we place the object right in front of the end effector. This is true for all the models except for the Top Key (A3) which we point the end effector downwards. We ran the optimization stage until we were able to find three candidates that succeeded or until we ran out of grasp candidates, whichever happened first. We picked one candidate to fabricate and validate in the real world.

### 7.3 Algorithm Performance

The GC candidate generation step took less than a minute for each model. The trajectory generation took between 7 and 26 minutes, and 13 minutes on average per GC candidate per model. This step



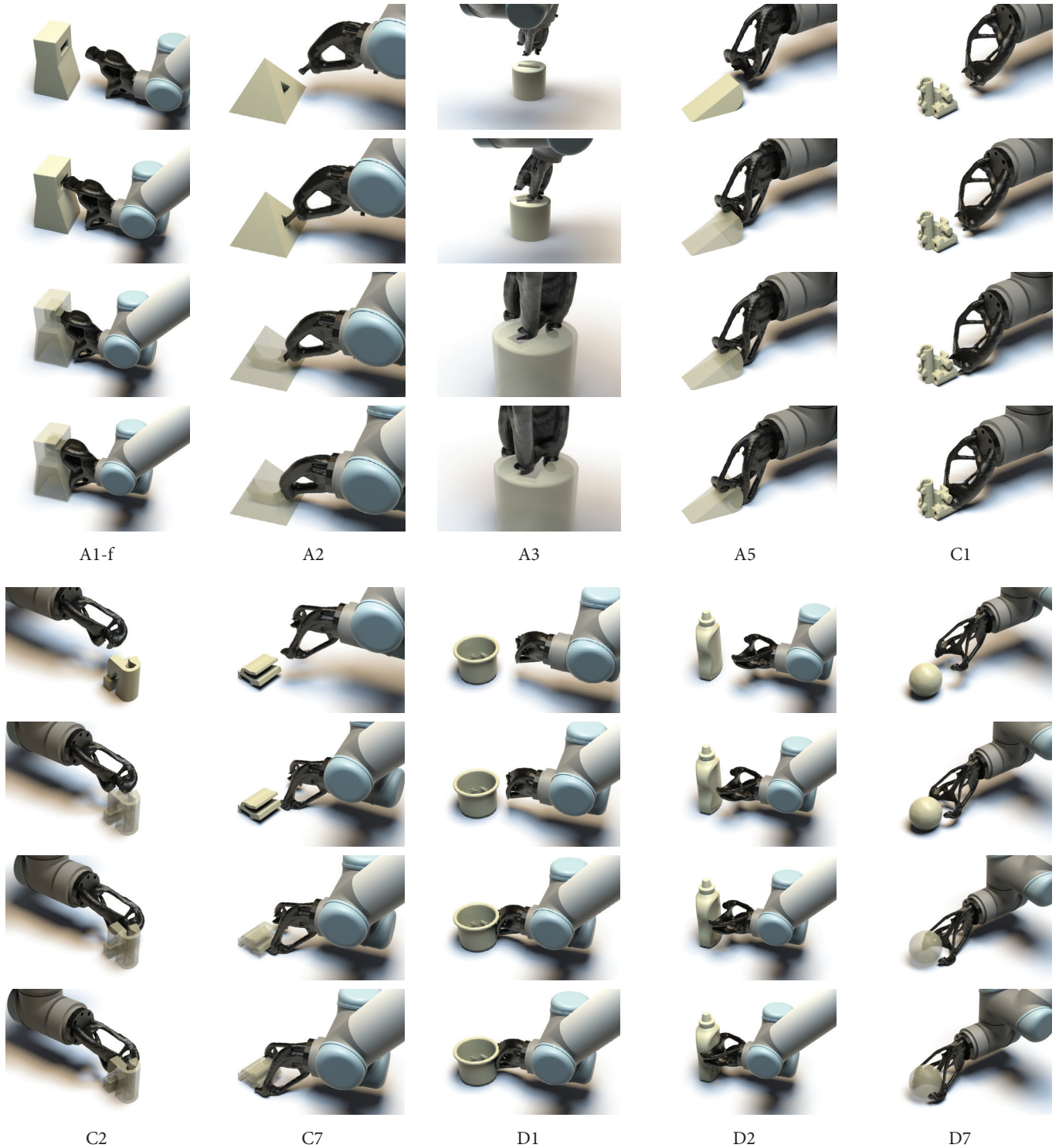


Fig. 9. Example grippers and their trajectories from objects in our dataset. Generated grippers and trajectories vary significantly across different models. These gripper designs mimic everyday mechanical tools such as wrenches (A5 and D1) and tongs (C7, D2, and D7), but are highly tailored to fit the objects. The accompanying trajectories can be classified into insert motions (A5, C2, and C7) and twist motions (A2, A3) and are nontrivial to design by hand.



terminated after running on 5 GC candidates on average and succeeded on at least one candidate on every model. The topology optimization took on average 1 hour and 26 minutes, but varied largely with the dimensions of the bounding box.

The whole pipeline takes on average 2 hours and 38 minutes per model, and on average 54% of the time is spent running topology optimization. While several computation steps could be sped up with straightforward implementation updates (for example, faster topology optimization [Zhu et al. 2017]), our computation speed is on par with SOTA generative design systems and suitable for practical use, particularly when considering manufacturing time.

#### 7.4 Qualitative Analysis

As illustrated in Figure 9, the gripper shapes and trajectories are highly customized for each particular object. While many trajectories may seem obvious given a gripper, our algorithm can generate non-intuitive solutions. For example, it took one of the authors (who had seen all visualizations) almost two minutes to figure out how one of the grippers fits into its object. This highlights the tightly coupled nature of the problem and the need for co-optimization.

While all solutions are unique, we note that it is possible to classify gripper shapes into three categories. *Inserts* take most of the load in a single point, typically inside a cavity on the object, as is shown for examples A1-f, A2, A3, C1, and C2. *Tongs* rely on a wide separation of support from underneath, as shown in examples C7, D2, and D7. Finally *wrenches* have a narrow grasp from opposite sides that distributes the load on all three grasp points, as shown in A5 and D1. While customized to particular shapes, the designs replicate standard types of simple tools.

The trajectories most grippers follow also fall into a triadic grouping. Most objects fall into the *front insert* category where the gripper moves directly in and grabs the object from the bottom. Some examples include the front key (A1), the pallet (A4), the cup (D1), the orange (D7), and the challenge object C1. Although the trajectories may seem simple, they are not trivial to be designed by a user because they can be sensitive to approach angles and sequencing. For example, to get the orange, we need to go slightly down then insert to avoid the convexity of the sphere, and to get the front key (A1), we need the right amount of in and up motion.

A number of designs fall into the *side insert* category where the required motion is to slide horizontally into the desired position. Some examples include our wedge (A5), our Side Key (A1-s), objects C2 and C7. Our algorithm is able to find the solution even if the required motion is not perfectly straight. In the bunny example, the robot needs to slightly turn to get around the convex part.

Our pyramid (A2) and *Top Key* (A3) fall in to the *twist insert* category. These objects are designed to only be grabbed by rotation motions which our algorithm is able to find.

Together, these results suggest that our algorithm is successful in understanding the affordances of each object. For example, object A1 is picked up with a front or side motion depending on the orientation of the cavity. Objects with bilateral symmetry are grasped on opposing sides, and objects with recesses have smooth insertion trajectories.

#### 7.5 Physical Experimental Setup

We 3D print the grippers using two materials. We use a material with high coefficient of friction for the contact region (TangoBlackPlus) and a material with high stiffness and low cost for the rest of the gripper (ABS). While these could be printed together in a multi-material printer, to reduce cost, we use the Stratasys J750 Digital Anatomy for the contact region and Stratasys FDM 3D Printers (Fortus 250mc, F120, F170) for the rest and assemble the two parts.

All objects and printed grippers were tested on the UR5 arm in moveJ mode with the trajectories specified by our algorithm. We tested object pick up as well as resistance to dropping once the object was correctly seated in the gripper. This involved rotating clockwise, counterclockwise, and a forward roll until the object fully fell out of the gripper and can be seen in Fig.10. For the *Top Key* we roll in the opposite direction due to joint limitations in the UR5 at this position. We repeat every test 10 times per object.

#### 7.6 Real World Validation

Of the 23 experiments, 21 lead to successful pickups and the results are shown in Table 2. 17 of these experiments had 100% success rate and only two had success rates below 80%. These results, and the following analysis of the failure cases, show that our method is able to successfully bridge the virtual-to-reality gap and displays high grasp reliability in real experiments. Visualization of these results are shown in Figure 1 and our supplemental video.

The two failure cases (D4 and C8, not shown on the table) had significant discrepancies in their virtual representations. Since our gripper and trajectory were computed over the wrong input, the results were not well suited in practice. The 3D mesh we used to represent the bowl (D4) is thinner than the actual bowl, causing the gripper to collide during insertion. The challenge model (C8) was not oriented in its rest position causing inaccuracies during stability analysis. We note that the two models that had 80-90% success rates had similar discrepancies, but those affected our gripper to a lesser extent because of symmetries in the models and the locations where the discrepancies happened.

The two models with marginal stability (C5 and C6) were examples from the adversarial dataset, which are notoriously challenging to grasp. These failed because the grasp locations we chose did not leave much room for misalignment. The grasp location must be robust to errors in the gripper fabrication (order of 0.5mm), the motion of the robot (order of 0.1mm) and human positioning the object (order of 0.5mm in the best case scenario). This uncertainty



Fig. 10. Post-grasp stability test for the bunny. We measure the maximum roll angle both CW and CCW (left) and forward tilt angle (right) before the bunny falls.

Table 2. The number of successful pickups and maximum roll and pitch angles before the object falls out of the gripper with ten tests per category per object. For roll and pitch, the object was correctly loaded into the gripper to represent stability once an object was successfully picked up. Some objects (A3, C1, C3, C4, C6, D7) demonstrated multiple falling modes where an object could find additional regions of stability, resulting in large standard deviations.

Object	Pickups	CW (deg)		CCW (deg)		Fwd Tilt (deg)	
		mean	sd	mean	sd	mean	sd
A1-f	10	170	3	-162	6	68	9
A1-s	10	34	6	-23	7	30	4
A2	10	360	0	-360	0	53	23
A3	10	360	0	-359	3	-292	144
A4	10	237	3	-28	3	25	2
A5	10	52	6	-59	12	34	4
B	10	60	3	-194	1	145	3
C1	10	341	34	-246	41	118	15
C2	10	38	11	-238	5	276	10
C3	10	242	65	-189	5	83	3
C4	10	360	0	-332	59	316	72
C5	2	149	12	-14	4	9	2
C6	6	308	113	-358	4	28	69
C7	10	255	22	-66	5	34	9
D1	10	62	16	-107	44	68	16
D2	10	92	2	-101	2	26	5
D3	9	27	2	-21	4	19	5
D5	10	127	15	-59	3	14	2
D6	10	101	8	-89	4	69	33
D7	8	97	6	-178	102	40	7
D8	10	50	9	-104	10	30	2

was not accounted for by our algorithm that evaluates the stability of a grasp configuration (Section 4.1). We highlight, however, that our system was robust to these errors in all but these two adversarial examples, which shows the effectiveness of our approach.

### 7.7 Limitations and Future Work

The most evident direction for improvement is stability evaluation for selecting the grasp configurations. This would make our method more robust to errors in object placement and gripper trajectory encountered in real experiments. As discussed in Section 2, the SOTA algorithms for evaluating the quality of a grasp configuration in active grippers use machine learning approaches, but these results cannot be directly applied for passive gripper design. The results in this work invites new avenues of research in this direction. Future work should also consider modeling external forces that may occur during pickup, for example when picking up an object from a bin.

On the algorithmic side, while the random sampling of GC candidates shows numerous successes, they may be sub-optimal in terms of robustness. Future work should employ a better sampling algorithm for maximal robustness and diversity. Further analysis on the collision avoidance objective should also be done to find the optimal trade-off between computational time and convergence as we vary the significance of each cost component and reduce the sampling resolution. In particular, we observed that dropping the trajectory collision energy and the wrap-around distance achieves better convergence when the trajectory samples are sufficiently

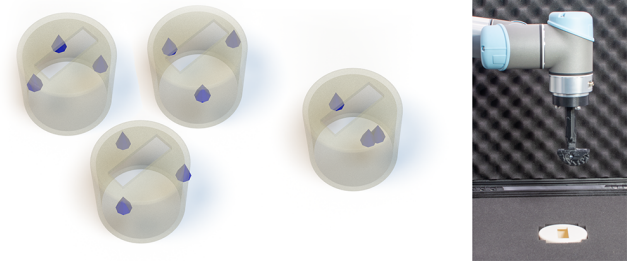


Fig. 11. While our algorithm is able to grasp A3 with wrench and twist motion (see Fig. 9), it cannot find a solution with insert and twist motion. In this experiment, we restricted the contact point to lie inside the object, and the only GCs found (see examples on the left) have no feasible solution since the algorithm prioritizes stability without knowing the shape of the slot. Given a manually specified GC (middle), our algorithm is able to find the solution (see example that has been validated in reality with 100% reliability on the right). This example illustrates avenues of future work on GC and trajectory co-design.

high. However, adding these components to the cost function is important for convergence with sparse samples.

It would be interesting to investigate alternatives to jointly optimize the contact point selection and trajectory optimization which cannot be completely decoupled (see Fig. 11). One other challenge is that the trajectory optimization can get stuck in a local minimum and not find a collision-free trajectory for a GC even if one exists. When this happens, the algorithm will choose a lower ranked GC. Additional freedom in object placement could also be achieved by optimizing robot's state in the grasping position.

Future work should also investigate different post-grasp trajectories. By assuming a vertical motion, we are unable to grasp objects with no bottom support such as a cylinder or a cone. Such objects can be handled by incorporating complex grasping strategies, such as rotating the object after being grasped [Mucchiani et al. 2018; Mucchiani and Yim 2021].

Finally, while we argue that customization for a single input has important practical applications, it would be interesting to relax some of the assumptions on our input. For example, we improve grasp reliability by considering random deviations of the object's position and geometry. Another interest direction for future work is to build on our optimization techniques to handle two or more input shapes or classes of shapes.

## 8 CONCLUSION

This work introduces a novel application for a generative design that has a high potential for impact in industrial applications of robotics. Robotic systems in industrial settings are highly inflexible—any change requires expensive re-design of autonomous components. The true cost of this rigidity has been particularly apparent during recent changes in demand during the COVID-19 crisis. Our algorithm addresses these challenges by establishing a new framework for creating robotic systems that can easily adapt to different scenarios without requiring additional dexterity, programming, or system complexity. Our key insight is that we can achieve this flexibility through re-design of passive grippers. We address the challenges of

exploring the complex design space of passive grippers with novel insights on representation abstractions and co-design. We validate our findings with extensive physical experiments and discuss limitations, paving the way to exciting avenues of future work in this domain.

## ACKNOWLEDGMENTS

This work was funded by NSF grants 2035717, 1954028, 3922035717, and 2017927; the Office of Naval Research grant DB2240; and the Murdock Foundation Materials Foundry grant 201913596. Special thanks to Tyler Freitas for his assistance in data collection.

## REFERENCES

- Martin Philip Bendsoe and Ole Sigmund. 2003. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.
- Antonio Bicchi and Vijay Kumar. 2000. Robotic grasping and contact: A review. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, 348–353.
- D. Blanding. 1999. *Exact Constraint: Machine Design Using Kinematic Principles*. ASME Press. <https://doi.org/10.1115/1.800857>
- Luzius Brodbeck and Fumiya Iida. 2015. An extendible reconfigurable robot based on hot melt adhesives. *Autonomous Robots* 39, 1 (2015), 87–100.
- Russell G Brown and Randy C Brost. 1999. A 3-D modular gripper design tool. *IEEE Transactions on Robotics and Automation* 15, 1 (1999), 174–186.
- Herman Bruyninckx, Sabine Demey, and Vijay Kumar. 1998. Generalized stability of compliant grasps. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, Vol. 3. IEEE, 2396–2402.
- Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. 2015a. The YCB object and Model set: Towards common benchmarks for manipulation research. *2015 International Conference on Advanced Robotics (ICAR)* (Jul 2015). <https://doi.org/10.1109/icar.2015.7251504>
- Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. 2015b. Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set. *IEEE Robotics & Automation Magazine* 22, 3 (Sep 2015), 36–52. <https://doi.org/10.1109/mra.2015.2448951>
- Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.
- Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. 2020. Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning. *arXiv preprint arXiv:2008.04460* (2020).
- Whitney Crooks, Shane Rozen-Levy, Barry Trimmer, Chris Rogers, and William Messner. 2017. Passive gripper inspired by Manduca sexta and the Fin Ray® Effect. *International Journal of Advanced Robotic Systems* 14, 4 (2017), 1729881417721155.
- Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature*. Springer, 849–858.
- Raphael Deimel, Patrick Irmisch, Vincent Wall, and Oliver Brock. 2017. Automated co-design of soft hand morphology and control strategy for grasping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1213–1218.
- Krishnamanaswi M Digumarti, Christian Gehring, Stelian Coros, J Hwangbo, and Roland Siegwart. 2014. Concurrent optimization of mechanical design and locomotion control of a legged robot. In *Mobile Service Robotics*. World Scientific, 315–323.
- Tao Du, Adriana Schulz, Bo Zhu, Bernd Bickel, and Wojciech Matusik. 2016. Computational multicopter design. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–10.
- Anthony V. Fiacco and Garth P. McCormick. 1964. The Sequential Unconstrained Minimization Technique for Nonlinear Programming, a Primal-Dual Method. *Manage. Sci.* 10, 2 (jan 1964), 360–366. <https://doi.org/10.1287/mnsc.10.2.360>
- Varan Gupta, Rithul Perathara, Aditya K Chaurasiya, and Jitendra P Khatait. 2019. Design and analysis of a flexure based passive gripper. *Precision Engineering* 56 (2019), 537–548.
- Huy Ha, Shubham Agrawal, and Shuran Song. 2020. Fit2Form: 3D Generative Model for Robot Gripper Form Design. *arXiv preprint arXiv:2011.06498* (2020).
- Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. 2017. Joint Optimization of Robot Design and Motion Parameters using the Implicit Function Theorem.. In *Robotics: Science and systems*, Vol. 8.
- Christopher Hazard, Nancy Pollard, and Stelian Coros. 2018. Automated design of manipulators for in-hand tasks. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 1–8.
- Stephanie Hendrixson. 2016. Laser-Sintered Grippers Solve Robotic Packing Challenges. <https://www.additivemanufacturing.media/articles/laser-sintered-grippers-solve-robotic-packing-challenges>
- Mohammadali Honarpardaz, Martin Meier, and Robert Haschke. 2017a. Fast grasp tool design: From force to form closure. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 782–788.
- Mohammadali Honarpardaz, Mehdi Tarkian, Johan Ölvander, and Xiaolong Feng. 2017b. Finger design automation for industrial robot grippers: A review. *Robotics and Autonomous Systems* 87 (2017), 104–119.
- William Hunter et al. 2017. ToPy - Topology optimization with Python. <https://github.com/williamhunter/topy>.
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Steven G. Johnson. 2014. The NLOpt nonlinear-optimization package. <http://github.com/stevengj/nlopt>.
- P Kaelo and MM Ali. 2006. Some variants of the controlled random search algorithm for global optimization. *Journal of optimization theory and applications* 130, 2 (2006), 253–264.
- Heinrich Kruger and A Frank van der Stappen. 2011a. Partial closure grasps: Metrics and computation. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 5024–5030.
- Heinrich Kruger and A. Frank van der Stappen. 2011b. Partial closure grasps: Metrics and computation. In *2011 IEEE International Conference on Robotics and Automation*. 5024–5030. <https://doi.org/10.1109/ICRA.2011.5980171>
- S. Levine, C. Finn, T. Darrell, and P. Abbeel. 2016. End-to-End Training of Deep Visuomotor Policies. *Journal of Machine Learning Research (JMLR)* (2016).
- Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. 2020. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning. In *Conference on Robot Learning*. PMLR, 854–869.
- Pingchuan Ma, Tao Du, John Z. Zhang, Kui Wu, Andrew Spielberg, Robert K. Katzschmann, and Wojciech Matusik. 2021. DiffAqua: A Differentiable Computational Design Pipeline for Soft Underwater Swimmers with Shape Interpolation. *ACM Trans. Graph.* 40, 4, Article 132 (jul 2021), 14 pages. <https://doi.org/10.1145/3450626.3459832>
- J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Ojea, and K. Goldberg. 2017. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In *Proc. of Robotics: Science and Systems (RSS)*.
- Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. 2010. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2308–2315.
- Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 2019. 6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In *International Conference on Computer Vision*. arXiv:1905.10520 <http://arxiv.org/abs/1905.10520>
- Cao Mucchiani, Monroe Kennedy, Mark Yim, and Jungwon Seo. 2018. Object picking through in-hand manipulation using passive end-effectors with zero mobility. *IEEE Robotics and Automation Letters* 3, 2 (2018), 1096–1103.
- Cao Mucchiani and Mark Yim. 2021. Dynamic Grasping for Object Picking Using Passive Zero-DOF End-Effectors. *IEEE Robotics and Automation Letters* 6, 2 (2021), 3089–3096.
- A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox. 2020. 6-DOF Grasping for Target-driven Object Manipulation in Clutter. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.
- A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. 2019. Deep Dynamics Models for Learning Dexterous Manipulation. In *Proc. of the Conference on Robot Learning (CoRL)*.
- Kenji Nagaoka, Hayato Minote, Kyohei Maruya, Yuki Shirai, Kazuya Yoshida, Takeshi Hakamada, Hirotaka Sawada, and Takashi Kubota. 2018. Passive spine gripper for free-climbing robot in extreme terrain. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1765–1770.
- Xinlei Pan, Animesh Garg, Animashree Anandkumar, and Yuke Zhu. 2020. Emergent Hand Morphology and Control from Optimizing Robust Grasps of Diverse Objects. *arXiv preprint arXiv:2012.12209* (2020).
- Dalibor Petković, Nenad D Pavlović, Shahabuddin Shamshirband, and Nor Badrul Anuar. 2013. Development of a new type of passively adaptive compliant gripper. *Industrial Robot: An International Journal* (2013).
- Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. 2020. KETO: Learning Keypoint Representations for Tool Manipulation. (2020).
- R. Rahmatizadeh, P. Abolghasemi, L. Boloni, and S. Levine. 2018. Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-to-End Learning from Demonstration. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.



- Alberto Rodriguez, Matthew T Mason, and Steve Ferry. 2012. From caging to grasping. *The International Journal of Robotics Research* 31, 7 (2012), 886–900.
- Andreas Schroeffler, Christoph Rehekampff, and Tim C Lueth. 2019. An Automated Design Approach for Task-Specific two Finger Grippers for Industrial Applications. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 184–189.
- Arthur Seibel, Mert Yildiz, and Berkan Zorlubaş. 2020. A Gecko-Inspired Soft Passive Gripper. *Biomimetics* 5, 2 (2020), 12.
- Daniel Seita, Nawid Jamali, Michael Laskey, Ajay Kumar Tanwani, Ron Berenstein, Prakash Baskaran, Soshi Iba, John Canny, and Ken Goldberg. 2019. Deep transfer learning of pick points on fabric for robot bed-making. (2019).
- Silvia Sellán, Noam Aigerman, and Alec Jacobson. 2021. Swept Volumes via Spacetime Numerical Continuation. *ACM Transactions on Graphics* (2021).
- Jungwon Seo, Mark Yim, and Vijay Kumar. 2016. A theory on grasping objects using effectors with curved contact surfaces and its application to whole-arm grasping. *The International Journal of Robotics Research* 35, 9 (2016), 1080–1102.
- Andrew Spielberg, Allan Zhao, Yuanming Hu, Tao Du, Wojciech Matusik, and Daniela Rus. 2019. Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations. *Advances in Neural Information Processing Systems* 32 (2019), 8284–8294.
- Kerry Stevenson. 2019. What Every 3D Printing Company Should Be Doing. Online. <https://www.fabbaloo.com/blog/2019/11/19/what-every-3d-printing-company-should-be-doing>
- Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. 2017. Grasp pose detection in point clouds. *The International Journal of Robotics Research* 36, 13-14 (2017), 1455–1473.
- The CGAL Project. 2021. *CGAL User and Reference Manual* (5.3.1 ed.). CGAL Editorial Board. <https://doc.cgal.org/5.3.1/Manual/packages.html>
- VB Velasco and Wyatt S Newman. 1998. Computer-assisted gripper and fixture customization using rapid-prototyping technology. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, Vol. 4. IEEE, 3658–3664.
- Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit Agrawal. 2021. An End-to-End Differentiable Framework for Contact-Aware Robot Design. In *Robotics: Science and Systems*.
- Haijie Zhang, Elisha Lerner, Bo Cheng, and Jianguo Zhao. 2020. Compliant Bistable Grippers Enable Passive Perching for Micro Aerial Vehicles. *IEEE/ASME Transactions on Mechatronics* (2020).
- Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2020. RoboGrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–16.
- Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. arXiv:1706.03189 [cs.CE]