

# Towards Generalized Manipulation Learning through Grasp Mechanics-based Features and Self-Supervision

Andrew S. Morgan, *Student Member, IEEE*, Walter G. Bircher, *Student Member, IEEE*, and Aaron M. Dollar, *Senior Member, IEEE*

**Abstract**—Learning accurate representations of robot models remains a challenging problem, and is typically approached through large, system-specific feature sets. This method inherently introduces practical shortcomings, as interpretability and transferability of the learned model typically decreases as more features are introduced into the learning framework in order to handle increasing task complexity. In this work, we examine the problem of developing transferable learned models for dexterous manipulation that are able to accurately predict the behavior of physically distinct systems without retraining. We introduce the notion of learning from visually-extracted grasp mechanics-based features, which are formulated by combining geometrically-inspired, analytical representations of the gripper into the feature set to more holistically represent the state of varied systems performing manipulation. We characterize the added utility of using such features through simulation and incorporate them into a classifier to predict specific phenomena, or modes of manipulation, that occur during prehensile within-hand movement. Four modes of manipulation—normal (rolling contact), drop, stuck, and sliding—are defined, collected physically, and trained via a self-supervised learning approach. The classifier is first trained on a single sensorless underactuated hand variant for all four modes. We then investigate the transferability of the learned classifier on 5 different planar gripper variants—analyzing applicability of this approach with both online and offline evaluation.

**Index Terms**—Dexterous Manipulation, Generalized Learning, Compliant Joint/Mechanism, Learning and Adaptive Systems

## I. INTRODUCTION

**D**EVELOPING robots capable of performing tasks in human-made, unstructured environments has remained an overarching research question in robotics for several decades. An important building block to this question is addressing the development of dexterous, within-hand manipulation (WIHM) capabilities for robotic hands. Dexterous manipulation is often characterized as the skillful, coordinated use of an end effector to reposition or reorient an object with respect to the hand frame [1]. An example of this ability includes the task of removing a key from a pocket, reorienting, and inserting into a lock. In this

This work was supported by the United States National Science Foundation under Grant IIS-1734190.

A. S. Morgan, W. G. Bircher, and A. M. Dollar are with the Department of Mechanical Engineering and Materials Science, Yale University, USA ({Andrew.Morgan, Walter.Bircher, Aaron.Dollar}@yale.edu).

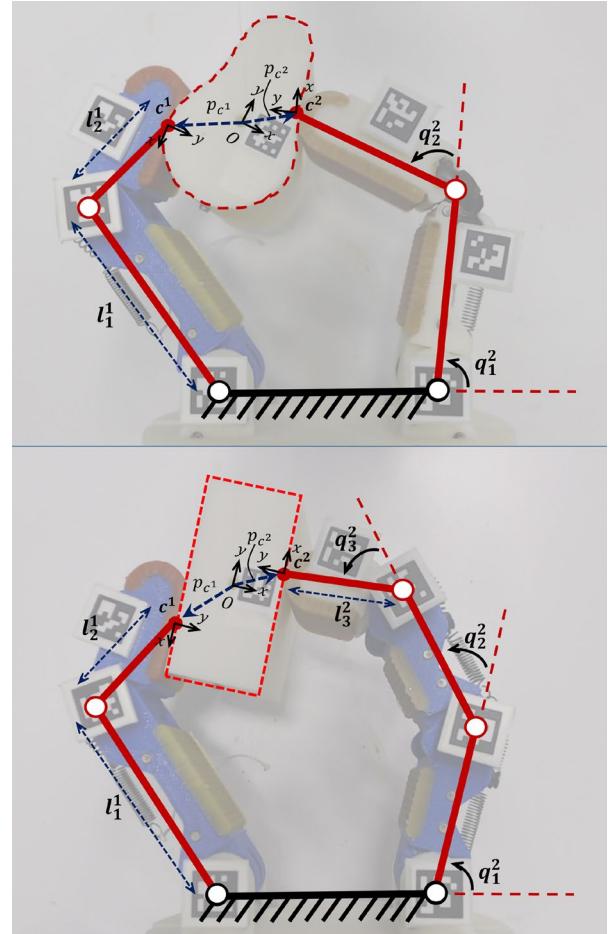


Fig. 1. Geometric features can be extracted visually during manipulation with *a priori* knowledge of the fingertip geometry, object geometry, and the number of finger links. (Top) A pivot-flexure finger manipulates a pear-shaped object with rolling contacts (*Mode: Normal*). (Bottom) A three-link pivot finger manipulates a rectangular object until sliding occurs along the left finger (*Mode: Sliding*).

task, not only does WIHM enable repositioning and reorientation of the key without re-grasping or large whole-arm motions, it also allows the robot to avoid undesired system conditions, such as diverting away from joint singularities, while attempting to reposition the key [2]. WIHM capabilities are especially advantageous for more capable service/home robots—which would be required to perform a variety of daily activities, such as folding clothes or feeding humans [3], [4].

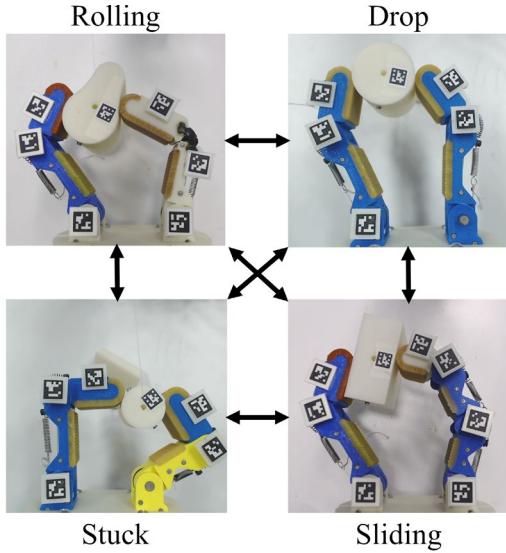


Fig. 2. By predicting modes of manipulation before or at the moment they occur, the user is able to transition between modes (trigger or avoid) for desired manipulation. Modes are typically detected when the hand-object system is in a similar configuration as those shown.

Practical implementation of precision WIHM remains a major challenge as coordinated finger movements with rigid, high degree-of-freedom hands requires accurate hand-object models, accurate parameter estimations of the environment, and advanced control schemas, which may be impossible to derive or estimate. An alternative to WIHM with fully actuated hands has been through the use of soft, compliant, or underactuated grippers that are able to passively adapt to their environment. While this ability enables grippers to more easily handle sensing and perception uncertainty [5], it also introduces difficulties in modeling—the configuration of the hand is dependent on fingertip forces, joint stiffnesses, and contact locations, which may be impossible to accurately measure.

Due to analytical modeling difficulties, machine learning has been introduced into manipulation for both, fully-actuated and underactuated hands. Such approaches are able to intrinsically estimate parameters, e.g. kinetic model parameters or joint stiffness ratios, that can be difficult or impossible to model via human intervention. While these approaches can be fairly successful, they often rely on large, unstructured feature sets for training, e.g. camera or tactile array inputs [6]. In this approach, little intuition is provided by the learned model as to what characteristics of the system are most valuable for the task.

In this work, we address this drawback by utilizing grounded, mechanics-based features that are able to generalize to different system variants. Assuming quasistatic motion of both the hand and the object, purely geometric representations—including finger manipulability measures, grasp quality measures, and hand-object manipulability measures—constitute as elemental, generalized properties of the hand-object system (Fig. 1). We investigate how these features allow trained models to transfer more successfully than traditional joint-based features. We use these features to distinguish between four possible manipulation classes for fingertip-based, prehensile manipulation; namely, normal (rolling contact), drop, stuck,

and sliding. These classes, coined as *modes of manipulation* for this work, can be predicted through a self-supervised learning approach—which would enable the user to either trigger or avoid modes for desired object movement (Fig. 2), as in [7], [8].

The approach of using mechanics-based features is particularly advantageous for generalizing models among a task. Due to reliance on the underlying mechanics of the problem, a single classifier can be self-supervised and trained on one gripper variant and then transferred to another similar but distinct variant without retraining or data adaptation. We theoretically explore this concept and show the bounds by which hand parameters can change before mode distributions of the features become distinct between variants. We also test this experimentally by using estimated Cartesian motion models to randomly manipulate different objects, and self-tagging each of the modes when they occur. A classifier is trained offline using a single gripper variant, and we show the transferability of the learned model for 5 different, asymmetric hands.

This manuscript extends our preliminary conference work [9] in several ways. Specifically, we discuss in detail each of the grasp mechanics-based features and their generalizability to different systems. Moreover, we present the mathematics required for an in-depth simulation of underactuated hands. With this simulation, we show the extended utility of grasp mechanics-based features, providing bounds by which a single hand’s model is able to generalize to other gripper variants, providing justification for our approach. Finally, this work provides a substantial increase in experimentation and analysis—both online and offline—for further validation.

## II. RELATED WORK

In this section, we present traditional methods to modeling within-hand manipulation. Following, we cover recent approaches to learning manipulation and this method’s associated drawbacks, which motivates this work.

### A. Within-hand Manipulation

For several decades, a great deal of research in robot manipulation has focused on explicitly modeling physical interactions that occur between robots and objects in complex, unstructured environments—from fundamentals of interactions such as pushing [10], to object interactions in highly dynamic and unconstrained environments. The study of these interactions is especially entailed in the application of WIHM, that requires coordinated finger movements while maintaining predefined contact scenarios. Since Okada first used inverse kinematics to plan joint trajectories for manipulator motion almost four decades ago [11], nearly every aspect of robot manipulation has been treated with great mathematical rigor in the pursuit of creating more capable robots [12]–[17]. This great volume of work elucidates many powerful relationships between finger joint motion and object motion via classic formulations such as contact curvatures, the Grasp Matrix, the Hand Jacobian, and the Hand-Object Jacobian.

Leveraging these mechanical representations and assuming that specific contact models are warranted by the task, object

motion models can be devised. The point contact with friction model, denoting that forces can be exerted in any direction within the friction cone, is often used and has led to the formulation of the Hand-Object Jacobian [18], which represents the transition map from joint movement into object motion. The work in [17] assumed stationary point contacts without rolling or sliding, virtually fixing the location of the contact frame on the finger to that on the object. Rolling has been taken into account as well [19]–[21], which requires geometric knowledge of the fingerpad and the contact to maintain precision. More advanced contact models, such as those for soft contacts [22], have also been introduced. Generally, all contact models are highly subject to material parameters, such as durometer and texture of the contact, that can change with environmental conditions (humidity or dust). Therefore, parametric estimation often necessitates on-board sensors, which are expensive, inaccurate, and complicate the design and control of the hand.

The addition of compliance to the system through either hardware (soft, underactuated) or software (impedance control [23], soft synergies [24]) can help mitigate uncertainties that would otherwise lead to task failure. Both approaches introduce passive adaptability to the system, which permits a grasp to be maintained under reasonable external disturbances [25]. For example, in [26], this concept is leveraged for in-hand manipulation with simple control over just two degrees of actuation. Though, due to this compliance, precision manipulation remains difficult to accurately model or simulate, since the output space is of higher dimension than the input space [21], [27], [28].

Two promising approaches to planar precision manipulation with underactuated hands have been introduced in our previous works by either using rough gripper models and an MPC visual servoing framework [29], or learning a state transition model of the gripper [30]. Although object precision was increased in both works, manipulation was focused in a specific region of the workspace. Moreover, the models learned were system specific and transfer was not addressed in either of these works providing inspiration for this manuscript—to learn transferrable representations of the gripper to aid in generalizing manipulation.

### B. Learning Manipulation Policies

Learning control policies for dexterous manipulation is a well-studied research area when analytical representations are unavailable. This approach enables the robot to formulate its own representative model without hand-tuned, human intervention. Reinforcement Learning (RL) has shown to be a promising approach to this problem, especially for compliant systems, e.g. [31]. A major drawback to RL is the amount of data required to train the model. As presented in [6], over “a hundred years” of object manipulation was collected in simulation for WIHM of a cube. Though some approaches have addressed this caveat, e.g. by learning from online videos [32] or guiding the manipulation strategies by combining imitation learning of a human expert [33], [34], simulators, which are often not representative of real-world contact scenarios, are normally required to develop these learned models. In addition

TABLE I  
NOMENCLATURE

Symbol	Description
<b>General:</b>	
$q$	Particular hand configuration: $q \in \mathbb{R}^4$ or $\mathbb{R}^5$
$a$	Configuration of the actuators: $a \in \mathbb{R}^2$
$B, F, O$	Pose of the base frame, finger frame, or object frame, respectively: $B, F, O \in SE(2)$
$v$	Velocity = $(v_x, v_y, v_\theta) \in se(2)$ of the object w.r.t. $B$
$J^i$	Jacobian of the $i^{\text{th}}$ finger of the hand: left finger is index 1 and right finger is index 2. $J_h$ is the Hand Jacobian.
$G$	Grasp Matrix: $G \in \mathbb{R}^{3 \times 4}$ in the two-finger, planar case
$H$	Hand-Object Jacobian: $H \in \mathbb{R}^{3 \times 4}$ or $\mathbb{R}^{3 \times 5}$
$P$	Object point cloud ( $P_o \in \mathbb{R}^{2 \times N}$ ) w.r.t. $O$ or fingerpad point cloud ( $P_f \in \mathbb{R}^{2 \times N}$ ) w.r.t. $F$
<b>Grasp Mechanics-based Features:</b>	
$v$	Cartesian velocity reference of the object: $v_x$ in the $x$ -direction and $v_y$ in the $y$ -direction.
$w^i$	Manipulability measure of the $i^{\text{th}}$ finger: $w_p^i$ is the penalized manipulability measure
$g$	Singular values (SV) of $G$ : $g_{\max}$ max SV, $g_{\min}$ min SV
$h$	Singular values of $H$ : $h_{\max}$ max SV, $h_{\min}$ min SV
$c^i$	Curvature of the contact point on the $i^{\text{th}}$ finger: $c_f^i$ is fingerpad curvature and $c_o^i$ is object curvature

to these drawbacks, the input dimensionality used in multilayer perceptrons can be extremely large and will therefore lack interpretability and generalizability for a more enlightened approach to manipulation. For example, in [6], the input vector was a video stream from 3 cameras (thus,  $3 \times 640 \times 480 = 921,600$  pixels/features).

Aside from learning the entire system model for precision manipulation, detecting object phenomena such as sliding has also been reported in the literature. Previous works have learned from tactile “images” to detect the coefficient of friction at the point of incipient slippage [35], [36] and can therefore plan trajectories to avoid slip conditions [37], [38]. Unfortunately, due to the nature of this approach, prior exploration with the object is necessary, which may be infeasible for time-sensitive or mission critical tasks. By leveraging mechanics, slip conditions can also be avoided by ensuring reasonable grasp quality measure values during manipulation [39]. Nevertheless, compliant, soft, and underactuated hands are often not equipped with the sensing modalities required to detect such phenomena, providing inspiration and purpose for this work.

### III. GRASP MECHANICS-BASED FEATURES

In this work, we leverage traditional mechanical models of manipulation to define features generalizable to different hand variants. Specifically, we extract the most common manipulability measures associated with the hand, the object, and the contacts: a Jacobian-based manipulation measure, a penalized Jacobian-based manipulability measure, the singular values of the Grasp Matrix, the singular values of the Hand-Object Jacobian, and the contact curvatures. By learning from these features, which are grounded geometrically to the state of the gripper, we are able to analyze which traditional grasp-mechanics measures are able to best represent the hand-object state. A summary of this manuscript’s nomenclature is

presented in Table I. For the remainder of this manuscript, we will refer to the left finger as index 1 and the right as index 2.

For the following formulations, let's assume the planar manipulator has  $n$  serial-link fingers, each having  $j^i$  joints per finger. The configuration of a single finger,  $q^i \in \mathbb{R}^{j^i}$ , represents its current joint angles. Therefore, the hand configuration,  $q \in \mathbb{R}^{\sum_{i=1}^n j^i}$ , fully describes the state of the hand and denotes the angles associated with each of the joints (Fig. 1). In traditional modeling for grasping and manipulation, the Hand Jacobian, sometimes referred to as the Manipulator Jacobian, is denoted as  $J_h = blkdiag(J^1, \dots, J^k)$ , where  $J^i$  is the Jacobian for a single, serial-link finger. It is important to note that in underactuated or compliant systems, the configuration of the gripper cannot be fully described by the state of the actuators,  $a$ . That is, it is likely the  $\dim(a) < \dim(q)$ , so we must properly denote their differences.

#### A. Finger Manipulability Measures

The current configuration of a serial-link finger determines its manipulability, i.e. how the tip of the finger is able to move given an actuation input about each of the joints, and is represented by the finger Jacobian. This representation is a function from joint input velocity to fingertip velocity:

$$\dot{y}^i = J^i \dot{q}^i \quad (1)$$

where  $\dot{y}^i \in \mathbb{R}^2$  in the planar case ( $\dot{y}^i \in \mathbb{R}^3$  in the spatial case),  $J^i \in \mathbb{R}^{2 \times j^i}$  in the planar case ( $J^i \in \mathbb{R}^{3 \times j^i}$  in the spatial case), and  $\dot{q}^i \in \mathbb{R}^{j^i}$ . In this work, we utilize both, two-link and three-link serial manipulators in the plane. From this, we can formulate the two finger Jacobians:

Two-link finger:

$$J^i(q^i) = \begin{bmatrix} -\mathcal{J}_A - \mathcal{J}_B & -\mathcal{J}_B \\ \mathcal{J}_D + \mathcal{J}_E & \mathcal{J}_E \end{bmatrix} \quad (2)$$

Three-link finger:

$$J^i(q^i) = \begin{bmatrix} -\mathcal{J}_A - \mathcal{J}_B - \mathcal{J}_C & -\mathcal{J}_B - \mathcal{J}_C & -\mathcal{J}_C \\ \mathcal{J}_D + \mathcal{J}_E + \mathcal{J}_F & \mathcal{J}_E + \mathcal{J}_F & \mathcal{J}_F \end{bmatrix} \quad (3)$$

$$\begin{aligned} \mathcal{J}_A &= l_1^i \sin(q_1^i) \\ \mathcal{J}_B &= l_2^i \sin(q_1^i + q_2^i) \\ \mathcal{J}_C &= l_3^i \sin(q_1^i + q_2^i + q_3^i) \\ \mathcal{J}_D &= l_1^i \cos(q_1^i) \\ \mathcal{J}_E &= l_2^i \cos(q_1^i + q_2^i) \\ \mathcal{J}_F &= l_3^i \cos(q_1^i + q_2^i + q_3^i) \end{aligned}$$

where, more specifically  $q^i = [q_1^i, \dots, q_j^i]^T$ , which represents the joint configuration for a single finger,  $i$ . From this Jacobian, we can represent its manipulability measure,  $w^i$ , for each finger in the hand [40]:

$$w^i = \sqrt{\det(J^i * transpose(J^i))} \quad (4)$$

As  $w^i$  approaches zero, this is indicative of the individual mechanism nearing a singularity—which effectively limits the ability to instantaneously move in any direction.

A penalized manipulability measure is also proposed, as it better encapsulates limits of a finger's workspace by incorporating *a priori* knowledge of the hard stops, i.e. a finger link cannot rotate fully around a joint, but typically has a range in which it can operate [41]. Fundamentally, this measure enables the mechanism to determine where mechanical constraints are located and to stay well within the workspace. The penalized manipulability measure,  $w_p^i$ , is the product of a penalty value,  $\pi^i(q^i)$  and the manipulability measure from (4).

$$\pi^i(q^i) = 1 - e^{-\kappa \prod_j \frac{(q_j^i - l_j^{i-})(l_j^{i+} - q_j^i)}{(l_j^{i+} - l_j^{i-})^2}} \quad (5)$$

Here,  $l_j^{i+}$  and  $l_j^{i-}$  represent the upper and lower bounds on joint  $j$ , respectively, and  $\kappa$  is a weighting factor that is tuned to determine how quickly manipulability drops off near the joint limits. This penalty function is calculated for each finger, and is applied to determine  $w_p^i$ ,

$$w_p^i = \pi^i w^i \quad (6)$$

The two finger manipulability measures,  $w^i$  and  $w_p^i$ , are used as mechanics-based features for mode detection in this work. We provide an illustration of these measures in Fig. 3, where it is important to note the similarities of the data distributions as properties of the fingers change.

#### B. Grasp Quality Measures

A grasp quality measure based on the Grasp Matrix is utilized for representing the manipulability of the object, given the current contact configuration. The Grasp Matrix is commonly leveraged as a representation for relating the velocity of the contact to the velocity of the object. Determined by the contact normal directions, in addition to the relative location of the object's fixed frame, the Grasp Matrix is formulated strictly by the geometry of the object and the position of the contacts—force sensing is not required. The model has a desirable quality that, even though the upper bound of its singular values is unbounded, the minimum singular value has a lower bound of zero regardless of the dimensions of the object. This occurs when two or more contact normals are collinear, parallel vectors with respect to the object frame,  $O$ . This quality can be a useful indicator for when the object is likely to drop. A metric based off of singular values of the Grasp Matrix is therefore invariant across systems of different dimensions.

The Grasp Matrix,  $G$ , in the velocity domain represents a map from external contact velocities,  $\dot{z}$ , to object frame velocity,  $v$ . We can represent this as:

$$\dot{z} = G^T v \quad (7)$$

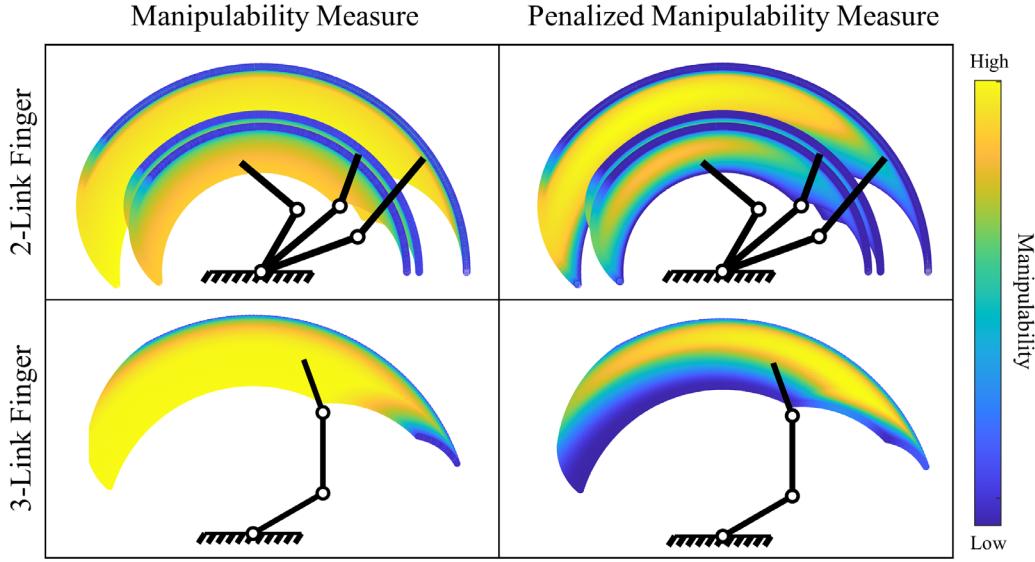


Fig. 3. Free swing manipulability workspaces for both proposed manipulability measures. (Top) Free swing trajectories for three two-link planar fingers used in this work. (Bottom) Free swing trajectory of the three-link planar finger used in this work. Unlike the penalized manipulability measure, the standard manipulability measure does not account for joint hard stops.

The shape of  $G$  is not absolute as it depends on the contact model used for manipulation. In the planar case with a point contact model and  $c$  number of contacts,  $\dot{z} \in \mathbb{R}^{2c}$ ,  $v \in se(2)$ , thus  $G \in \mathbb{R}^{3 \times 2c}$ . Similarly, for the spatial case not covered in this work,  $\dot{z} \in \mathbb{R}^{3c}$ ,  $v \in se(3)$ , and  $G \in \mathbb{R}^{6 \times 3c}$ . In this work, we will assume a point contact with friction model, forming the basis,  $b_{ci}$ , which states that a force can be applied along the  $x$ - and  $y$ -axes of the contact accordingly so long as it is within the friction cone. Additionally, we must calculate the vector,  $p_{ci}$ , denoting the positional relationship between the contact frame,  $c^i$ , for the  $i^{\text{th}}$  finger and the object frame,  $O$ . The rotational relationship,  $\theta_{\delta i}$ , between the contact frame,  $c^i$ , and  $O$  is also computed. For the two-finger, two-contact case in this work:

$$b_{ci} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (8)$$

$$R_{ci} = \begin{bmatrix} \cos(\theta_{\delta i}) & -\sin(\theta_{\delta i}) \\ \sin(\theta_{\delta i}) & \cos(\theta_{\delta i}) \end{bmatrix} \quad (9)$$

$$p_{ci} = \begin{bmatrix} p_{ci}^x \\ p_{ci}^y \end{bmatrix} \quad (10)$$

where  $\theta_{\delta i} = \theta_{ci} - \theta_O$ , and  $\theta_{ci}$ ,  $\theta_O$  are the angle offsets of the  $i^{\text{th}}$  contact frame and the object frame, respectively. Finally, with these calculated for each contact, we can formulate the Grasp Matrix,  $G$ :

$$Ad_{g_{oc}^{-1}}^T = \begin{bmatrix} R_{ci} & 0 \\ [-p_{ci}^y, p_{ci}^x]R_{ci} & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (11)$$

$$G = \begin{bmatrix} Ad_{g_{oc}^{-1}}^T b_{c^1} & Ad_{g_{oc}^{-1}}^T b_{c^2} \end{bmatrix} \in \mathbb{R}^{3 \times 4} \quad (12)$$

From  $G \in \mathbb{R}^{3 \times 4}$ , there exist three singular values that describe the state of the contacts with respect to the object. In

our feature set, we will denote the maximum singular value as  $\mathcal{G}_{\max}$  and the minimum singular value as  $\mathcal{G}_{\min}$ .

### C. Hand-Object Manipulability Measure

The Hand-Object Jacobian [18] is a map that describes the relationship between actuation input,  $\dot{q}$ , and object velocity,  $v$ . Although this cannot be directly utilized in underactuated hands, due to the inability to control each of the joints individually, i.e.  $\dim(a) < \dim(q)$ , it's geometric representation of the hand-object system can provide insight as to where the object can move given the current hand configuration. This Hand-Object Jacobian,  $H$ , assumes a point contact with friction model and is formulated by combining the Grasp Matrix,  $G$ , and the Hand Jacobian,  $J_h$ . Let's examine  $\dot{y} \in \mathbb{R}^{2k}$ , or the vector of all fingertip velocities of the hand from (1). Let's now also assume, that  $\dot{y} = \dot{z} \in \mathbb{R}^{2k}$ , the object contact velocities from (7). Assuming a point contact with friction model, this further suggests that the location of the contact does not move with respect to the object frame during manipulation, and virtually attaches the finger to the object. With this assumption, we combine (1) and (7),

$$v = (G^T)^+ J_h \dot{q} = H \dot{q} \quad (13)$$

where  $(G^T)^+$  is the pseudo-inverse of the transposed Grasp Matrix. In the planar case with two-links and two contacts,  $H \in \mathbb{R}^{3 \times 4}$ . Here, the singular values of  $H$  represent how close the hand-object system is to a singular configuration, i.e. the ability for the object to move instantaneously in any direction. Similar to those used for  $G$ , we will use the maximum singular value,  $\mathcal{H}_{\max}$ , and the minimum singular value,  $\mathcal{H}_{\min}$ , as features.

### D. Curvature of Contact

As described by Montana [42], the geometric conditions of contact are important as they enable differentiation between contact stability and spatial stability—necessary measures to

track during manipulation. To this end, we propose extracting the local conditions of the contact point for both, the fingerpad and object.

The object point cloud,  $\mathcal{P}_o$ , and fingerpad point cloud,  $\mathcal{P}_f$ , as further described in Sec. V.C, are used for calculating the curvature at the contacts. Let's require that the point clouds are contiguous; that is, neighboring indices indicate neighboring points in the cloud. Given  $\mathcal{P}_o(p_o)$  and  $\mathcal{P}_f(p_f)$ , determined by the KD-Tree to be the two closest points to one another in separate clouds, we calculate the curvature of the contact by evaluating their relationship to neighbors at each contact point. The curvature is therefore equal to the reciprocal of the radius of the circle that fits three neighboring points in the same point cloud. For example, let's calculate the curvature for the object. Given neighboring points on the object,  $p_{-o} = p_o - 1$  and  $p_{+o} = p_o + 1$ , we calculate the Euclidean distance between each of the three sets of points,  $(\mathcal{P}_o(p_o), \mathcal{P}_o(p_{+o}), \mathcal{P}_o(p_{-o}))$  providing distances  $\beta_1$ ,  $\beta_2$ , and  $\beta_3$ . Then,

$$\beta_s = \frac{\beta_1 + \beta_2 + \beta_3}{2} \quad (14)$$

$$\lambda = \sqrt{|\beta_s(\beta_s - \beta_1)(\beta_s - \beta_2)(\beta_s - \beta_3)|} \quad (15)$$

$$c_o^i = \frac{\beta_1 \beta_2 \beta_3}{4\lambda} \quad (16)$$

where  $c_o^i$  from (16) is the curvature of the object at the  $i^{\text{th}}$  contact point. We can similarly calculate the curvature of fingerpad at the  $i^{\text{th}}$  contact point. These curvatures are included as mechanics-based features in this work to aid in determining object stability.

#### IV. BOUNDING FEATURE GENERALIZABILITY

The goal of this section is to investigate the bounds of which grasp mechanics-based features are able to better estimate the state of the hand-object system as physical parameters of the hand change, e.g. link lengths or spring ratios. We compare these bounds to a more traditional feature set used for learning—the joint or motor configuration of the robot. This is accomplished by modeling the mechanics of quasistatic, underactuated manipulation for a two-fingered hand. After modeling, we sequentially vary parameters of the hand-object system beyond that of its original symmetric configuration and run statistical analyses that indicate whether or not the features likely come from similar distributions between different hand variants. In order to maintain brevity and tractability of these results, this section will focus on studying solely finger manipulability measures, and will leave the additional features from Sec. III.B, III.C, and III.D to be discussed in Sec. VII. Physical characteristics of the hand are referenced in Fig. 4.A.

##### A. Mechanics of Underactuated Manipulation

Underactuated hands can be modeled in terms of energy with kinematic, frictional, and actuation constraints. That is, the configuration of the hand after actuation can be determined by solving for the minimum energy configuration objective,

$$U = \frac{1}{2} \sum_i \sum_j k_j^i (q_j^i - q_{j0}^i)^2 \quad (17)$$

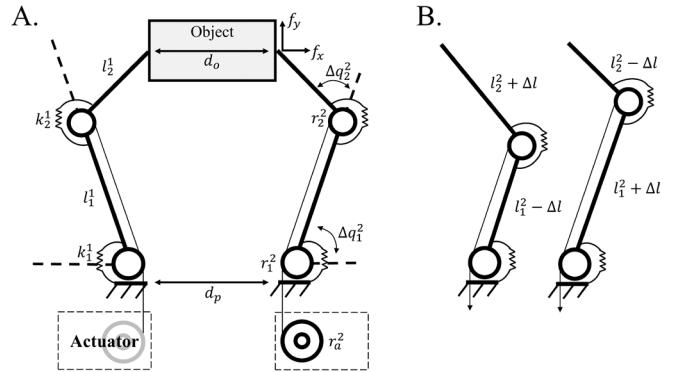


Fig. 4. (A) Annotation of hand parameters required for modeling underactuated manipulation mechanics. (B) Proximal and distal link lengths in simulation are changed by the same value,  $\Delta l$ , as to maintain unit length during simulation.

where  $q_{j0}^i$  is the rest angle of each joint,  $q_j^i$  is the current angle of each joint,  $k_j^i$  is the spring stiffness of each joint, and  $U$  is the total elastic energy of the hand. This is formulated as an optimization problem, guided by both equality and inequality constraints. The quasistatic moment about the finger's proximal joint,  $M_1^i$ , on finger  $i$  is created by both normal,  $f_N^i$ , and tangential,  $f_T^i$ , contact forces,

$$M_1^i = u_{1,\text{eff}}^i \times f_N^i + u_{1,\text{eff}}^i \times f_T^i \quad (18)$$

where  $u_{1,\text{eff}}^i$  is a vector from the proximal joint to the fingertip and  $\times$  is the cross product. In this formulation, we assume that the normal force vector extends along the line joining both contact points to the object. We then define the moment at the finger's distal joint,  $M_2^i$ , created by contact forces,

$$M_2^i = u_{2,\text{eff}}^i \times f_N^i + u_{2,\text{eff}}^i \times f_T^i \quad (19)$$

where  $u_{2,\text{eff}}^i$  is a vector from the distal joint to the fingertip.

Through this analytical modeling of  $M_1^i$  and  $M_2^i$ , we represent the moment balance at the finger's proximal joint,

$$0 = T^i r_1^i - k_1^i \Delta q_1^i + M_{1\tau}^i \quad (20)$$

where  $T^i$  is the force created by the tendon, wrapped about a pulley of radius  $r_1^i$ ,  $k_1^i$  is the proximal spring stiffness,  $\Delta q_1^i = q_1^i - q_{1,0}^i$  is the proximal joint angle w.r.t. its rest orientation, and  $M_{1\tau}^i$  is the out of plane component of the proximal moment vector representing the magnitude of its torque. We similarly model the moment balance at the finger's distal joint,

$$0 = T^i r_2^i - k_2^i \Delta q_2^i + M_{2\tau}^i \quad (21)$$

where  $\Delta q_2^i = q_2^i - q_{2,0}^i$  is the distal joint angle w.r.t. its rest orientation.

In addition to moment balance constraints, the forces applied to the object must be in equilibrium with one another in order to maintain a stable grasp. That is,

$$0 = f_{Nx}^1 + f_{Tx}^1 + f_{Nx}^2 + f_{Tx}^2 \quad (22)$$

$$0 = f_{Ny}^1 + f_{Ty}^1 + f_{Ny}^2 + f_{Ty}^2 \quad (23)$$

where  $f_{Nx}^i, f_{Tx}^i, f_{Ny}^i$ , and  $f_{Ty}^i$  are the  $x$  and  $y$  components of the normal and tangential forces, respectively. During manipulation, we must also satisfy kinematic loop closure,

$$0 = \|u_{B,eff}^1 - u_{B,eff}^2\| - d_o \quad (24)$$

where  $u_{B,eff}^1$  is the position of the left fingertip w.r.t the base frame,  $B$ , and  $u_{B,eff}^2$  is similarly the position of the right fingertip. Here,  $d_o$  is the diameter of the object in contact with the fingertips. The last equality constraint represents the tendon or transmission constraint of underactuated mechanisms, dictating the coupled actuation between both joints,

$$0 = r_1^i \Delta q_1^i + r_2^i \Delta q_2^i - r_a^i \Delta a^i \quad (25)$$

where  $r_a^i$  is the radius of the actuator pulley, and  $\Delta a^i$  is the difference between the resting and set angle of the actuator.

Finally, an inequality constraint on each finger must also be satisfied such that contact normal and tangential forces satisfy Coulomb's Law,

$$0 \geq \|f_T^i\| - \mu_o \|f_N^i\| \quad (26)$$

where  $\mu_o = 1$  and is a conservative coefficient of friction estimate between *rubber fingerpads and a solid object* [43].

From these constraints, which are guided by the mechanics of manipulation, we solve for the equilibrated joint configuration,  $q^*$ , and contact forces on each finger,  $f_N^i$  and  $f_T^i$ , by solving the optimization problem,

$$(q^*, f_N^i, f_T^i) = \underset{q}{\operatorname{argmin}} U(q) \text{ s.t. (18-26)} \quad (27)$$

### B. Mode Characterization in Simulation

Following these formulations, we create a simulation modeling the motion of an object given an actuation input. Although our simulation can represent any underactuated two-fingered hand-object variant, we decide to limit the parameter variation to just three characteristics in order to maintain tractability of the results. Specifically, we begin with a symmetric two-fingered hand (base variant) and sequentially change link lengths of the right finger, joint stiffnesses of the right finger, and object diameters, while keeping object contact locations constant. To avoid highly asymmetric cases, we choose to incorporate a variational term,  $\Delta l$ , where, if this term is added to one link, it is conversely subtracted from the other in order to maintain unit length of the finger (Fig. 4.B).

We collect observations of the hand-object system when actuated and save two feature sets of its state. Particularly, these feature vectors consist of both types of finger manipulability measures in addition to the joint configurations of the hand. Concretely, we represent these as feature sets,  $\mathcal{S}_m = (w^1, w^2, w_p^1, w_p^2)$  and  $\mathcal{S}_q = (q_1^1, q_1^2, q_2^1, q_2^2)$ , which are then both tagged with a mode of manipulation, as determined by the results of the optimization process:

- 1.) *Drop* - The hand is unable to provide force closure (i.e. when frictional fingertip contacts can equilibrate an external wrench perturbation) on an object of 20 grams, with gravity pointing into the manipulation plane.
- 2.) *Stuck* - The object is no longer able to move in the direction dictated by actuation forces, creating an excessively large internal object force. This normally occurs at joint limits.
- 3.) *Sliding* - The object exhibits sliding contacts when normal forces lie outside of the friction cone, as determined by  $\mu_o$  and fingertip forces from (27).
- 4.) *Normal* - The object is manipulable within the gripper's workspace and modes 1-3 are not satisfied.

We complete the simulation with a total of 867 hand-object variants. Each variant is actuated with a total of 900 distinct actuation pairs, and from each pair, the two feature vectors and mode of manipulation is recorded. Table II provides a summary of the simulated hand parameters.

TABLE II  
SIMULATION HAND PARAMETERS

Symbol	Value	Symbol	Value
$l_1^1$	6cm	$\Delta l$	[-2.4 – 2.4]cm (17 total)
$l_2^1$	4cm	$d_o$	[2.0 – 6.0]cm (17 total)
$k_2^1/k_1^1$	2.0	$k_2^2/k_1^2$	[2, 2.5, 3] (3 total)
$d_p$	6cm	$r_1^i/r_2^i$	1.2

### C. Bounding Feature Distributions by Statistical Testing

The goal of this simulation is to provide general bounds by which grasp mechanics-based features are able to better generalize to different hands compared to joint-based features. More specifically, we approach this study by analyzing the data distributions of each of those feature sets with respect to the modes realized within those distributions, while varying hand-object parameters. It further follows that if mode distributions do not greatly change between hand variants, we are likely able to better transfer learned models to other hands without retuning or retraining with new data.

We perform this analysis by sequentially conducting a One-Way Multivariate Analysis of Variance (MANOVA) test while increasing the difference between tested hand variants. Due to this sequential testing, we adjust the *p-value* required to reject the null hypothesis according to the Bonferroni correction method, starting with a value of 0.05 at the first variation of testing. This analysis is conducted as follows: given an object diameter and a right finger stiffness ratio, we select three hand variants—two with  $\pm \Delta l$ , and one where  $\Delta l$  is equal to zero (the base variant). We run MANOVA and according to the *p-value*, decide whether to reject the null hypothesis. For this type of statistical analysis, the null hypothesis tests whether the mode data from the three hand variants come from the same distributions. If the *p-value* is less than the Bonferroni adjusted threshold, we can reject the null hypothesis, meaning that there is sufficient evidence that the three hand variants do not come from the same data distributions. Performing these tests for all simulated hand-object variants, we compare the *p-values* of both the mechanics-based feature set,  $\mathcal{S}_m$ , and the joint-based feature set,  $\mathcal{S}_q$ . The results are presented in Fig. 5.

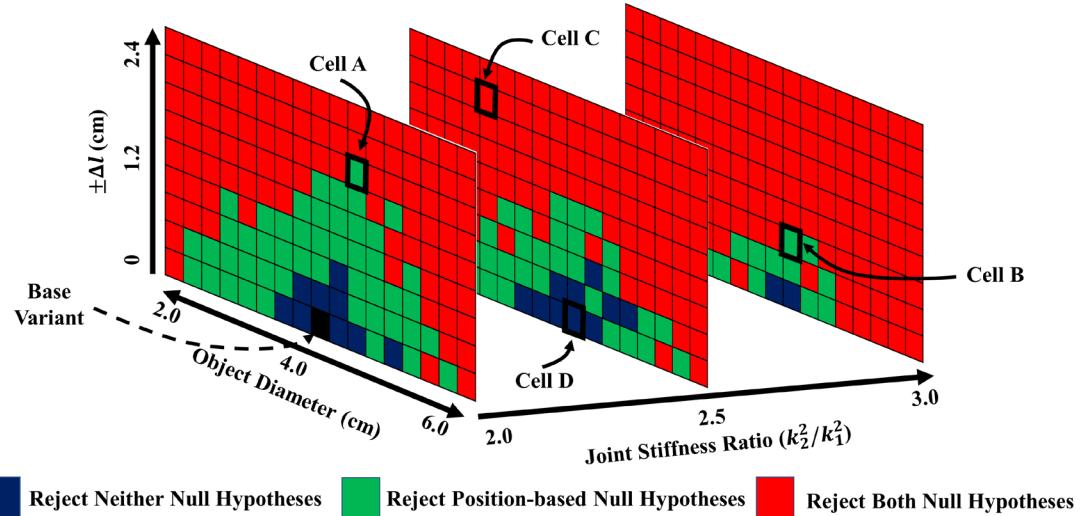


Fig. 5. MANOVA mode distribution testing of grasp mechanics-based features and position-based features. Green cells generally indicate the extended bounds by which mechanics-based features are able to transfer beyond that of their position-based counterpart (blue cells). Red cells indicate that neither of the feature sets likely share data distributions with the base variant (solid black cell).

These results indicate the general bounds by which hand properties can change without varying the feature data distributions for each of the four modes of manipulation. More intuitively, Fig. 5 shows that, when starting at the base variant (solid black cell), the green cells are able to extend beyond that of the blue cells, where the green cells denote mechanics-based features and the blue cells denote both, position-based and mechanics-based features. Alternatively, red cells indicate hand variants where both null hypotheses were rejected, i.e. neither feature sets can sufficiently represent the base variant's data.

In fact, while keeping the object diameter around that of the base variant, and while holding the joint stiffness ratio static,  $\Delta l$  can change by  $\pm 1.8\text{cm}$  without being statistically significant from the other data distributions (Cell A in Fig. 5). Notably, this variation cannot extend as drastically when the joint stiffness of the finger also changes. For instance, while still rejecting the position-based null hypothesis, we are only able to change  $\Delta l$  by  $\pm 0.6\text{cm}$  comparatively, but this is when we increase the joint-stiffness ratio from 2 to 3 (Cell B). These two cells, interestingly, have similar  $p$ -values of 0.07 for  $s_m$ .

This depiction serves to broadly represent the extent of the generalization bounds by analyzing various cells, like Cell C where we can reject both null hypotheses, and Cell D where we cannot reject either of the null hypotheses according to their  $p$ -values. While this analysis is not definitive in that it does not necessarily directly transfer to more advanced non-linear learned regression models, it provides a general basis for understanding the added utility of the mechanics-based features from a statistical, data-distribution perspective, and we use this concept to motivate the continuation of this work.

## V. SELF-SUPERVISED TAGGING AND OBJECT RESET

Beyond that of statistical evaluation, we seek to test the reliability of mechanics-based features empirically on a physical hand-object system to further analyze their applicability in real-world environments. We employ such experimentation on an underactuated Yale OpenHand Model T42, that is not equipped with joint encoders or tactile sensors

at the fingertips. Due to this limitation, we must reformulate the definitions of the four modes of manipulation:

- 1.) *Drop* - The hand-object configuration is in a state where the object is just about to drop and will drop shortly thereafter the commanded next actuation.
- 2.) *Stuck* - The object is no longer able to move in the commanded actuation direction due to the hand-object configuration of the gripper, or the joint has reached a physical hard stop.
- 3.) *Sliding* - The object exhibits a sliding contact with respect to the gripper's distal link, i.e. mechanical rolling contact conditions are not satisfied.
- 4.) *Normal* - The object is manipulable within the gripper's workspace while maintaining a rolling contact, and modes 1-3 are not satisfied.

### A. Manipulation Primitives

The Model T42 is underactuated and thus mechanically compliant, which enables passive reconfiguration post-contact and mitigates potential overconstraint as in a fully actuated hand. This compliance is advantageous for manipulation, as it enables the hand to reconfigure with noisy or imprecise control input. Though due to the nature of this mechanism, we cannot control all degrees of freedom of the object simultaneously, but a 2D submanifold of the object's 3D configuration space. We employ manipulation primitives on the hand by generating an approximated Jacobian for an arbitrary object that relates to the velocity,  $v = [v_x, v_y, v_\theta]^T$ , of the object frame,  $O \in SE(2)$ , to an actuation velocity,  $\dot{a} = [\dot{a}_1, \dot{a}_2]^T$ , all with respect to the base frame,  $B \in SE(2)$  [44]. These primitive actuation sequences are estimates of the true Jacobian, and are selected according to the commanded Cartesian velocity reference in the  $x$ -direction,  $v_x$ , and in the  $y$ -direction,  $v_y$  (Fig. 6).

### B. Geometric Hand-Object Representation

The geometric representation of a hand-object system can generally be extracted through various sensing modalities, e.g. cameras, tactile sensors, and joint encoders or IMUs. Albeit, not

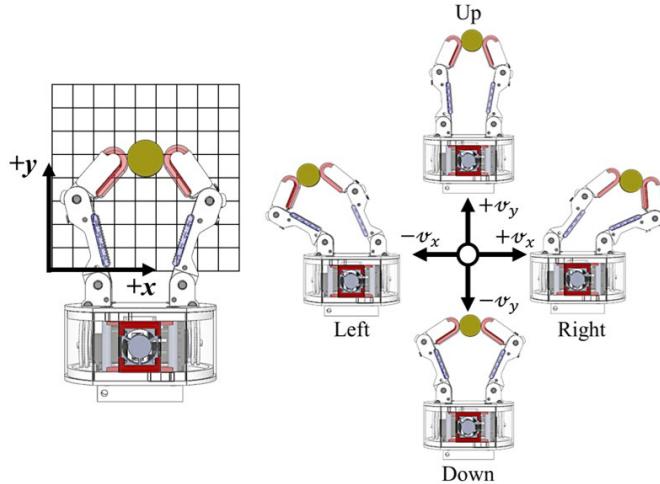


Fig. 6. Simple manipulation primitives enable planar motion within the workspace of a Yale OpenHand Model T42 gripper. These primitives enable the object to move up, down, left, or right depending on the Cartesian velocity reference,  $\mathbf{v}$ .

all hands are equipped with such capabilities, as these types of sensors are generally not required for grasping with compliant hands. Thus, in this work, we focus on a vision-based approach with a fixed overhead camera (30Hz). During manipulation, the gripper configuration is tracked via ArUco markers attached to rigid links of the hand. *A priori* knowledge of the hand includes the number of finger links, the object geometry, and the geometry of the fingerpads. Principally, the camera pose w.r.t. the task can vary, as long as the markers are visible by the camera sensor such that the 6D pose of the attached markers can be tracked, e.g. offset from a robot wrist or on another robot.

The state of the contacts is tracked, which subsequently allows the system to detect sliding during manipulation (Sec. V.C), by superimposing a 2D point cloud on both, the fingertips and the object, with respect to the marker frames attached to each. As the hand-object configuration changes during actuation, fundamentally changing characteristics about the grasp such as the effective link length, the superimposed 2D point clouds are tracked and analyzed. We solve for the contact location between the fingerpad and the object by querying a KD-Tree constructed with the object's point cloud.

### C. Self-Supervised Mode Detection

All four modes described in this work can be detected solely by an overhead camera that monitors the hand-object state during manipulation. This observation forms the basis of our self-supervised learning approach, where we can monitor features of the hand and of the object to determine and autonomously tag the current mode of manipulation.

1.) *Detecting Drops*: Drop detection is achieved by recording the state history of the object during manipulation. Simply, if the object marker is no longer within the manipulation plane, or the marker is currently absent from visual detection, the object is declared to be dropped. To reduce the potential for drop detection error, the history over the past 10 frames (0.3 seconds or two hand actions) is used to determine such occurrences, whereas this threshold is tuned heuristically during the experimentation setup. If this condition is satisfied, the system

accesses the recorded state of the gripper 10 frames prior (directly before the object was dropped) and self-tags a drop observation. The object is then reset via an object reset system (further described in Sec. V.D) and manipulation continues.

2.) *Detecting Stuck*: The object is considered stuck if it is no longer manipulable in the direction desired, which is determined by the current Cartesian velocity reference. Typically, this mode occurs when both fingers reach hard stops, limiting additional manipulation towards the palm (Fig. 2). Alternatively, stuck cases are also detected when the current configuration of the hand-object system is not able to reconfigure, limiting the movement of the object in the reference direction. When stuck is detected, the system self-tags an observation and the object is reset via an object reset system (Sec. V.D) for manipulation to continue.

3.) *Detecting Sliding*: Sliding is the most difficult of the four modes to detect and is done so when kinematic rolling conditions cannot be satisfied. In order for one surface to be considered rolling on top of another, we choose to track two of the three sliding constraints—the position of the point of contact and the velocity at the point of contact must be the same between the two bodies [45].

Consider the scenario depicted in Fig. 7. Here, for the planar case,  $O \in SE(2)$  is the object frame and  $F \in SE(2)$  is the finger frame. To maintain generality,  $F$  can be either finger frame, where the left and right finger frames are  $F^1$  and  $F^2$ , respectively. By parameterizing the object and the fingerpad surface locally in  $O$  and  $F$ , respectively, we effectively develop a point cloud for the object,  $\mathcal{P}_o \in \mathbb{R}^{N \times 2}$ , and for the fingerpad,  $\mathcal{P}_f \in \mathbb{R}^{N \times 2}$ , where the interacting array index from the object point cloud is  $\mathcal{P}_o$  and the interacting array index for the fingerpad point cloud is  $\mathcal{P}_f$ , as determined by the KD-Tree. The value for  $N$  can be arbitrarily assigned such that points sufficiently cover the surface of the fingerpad and the object. For clarification, in the object point cloud, the location  $\mathcal{P}_o(\mathcal{P}_o) \in \mathbb{R}^2$  is in contact with point  $\mathcal{P}_f(\mathcal{P}_f) \in \mathbb{R}^2$  from the fingerpad point cloud. Let's consider that the location of  $O$  is  $x_o = (O_x, O_y) \in \mathbb{R}^2$ , and the location of the  $F$  is  $x_f = (F_x, F_y) \in \mathbb{R}^2$ , both with respect to  $B$ . We denote the 2D rotation matrices  $R_o$  and  $R_f$  for these respective frames. It follows the elementary consideration that the location of the contact point on the object ( $x_o^c$ ), that is with respect to the base frame, can be calculated as,

$$x_o^c = x_o + R_o \mathcal{P}_o(\mathcal{P}_o) \quad (27)$$

and is similarly calculated for the fingerpad, denoted  $x_f^c$ . To satisfy the positional constraint of a rolling contact, within some user-defined threshold,  $\epsilon_p$ , the following must be valid:

$$x_o^c - \epsilon_p \leq x_f^c \leq x_o^c + \epsilon_p \quad (28)$$

The velocity constraint can be similarly constructed, where we can differentiate the two positions,  $x_o$  and  $x_f$ , with respect to time to form  $\dot{x}_o$  and  $\dot{x}_f$ . Since the body rotations are also functions of time, we must also differentiate body rotations of the object and fingertips to form velocity dependent rotation matrices,  $\dot{R}_o$  and  $\dot{R}_f$  (see [45]). We can then solve for the velocity of the object contact about the base frame,  $\dot{x}_o^c$ , by,

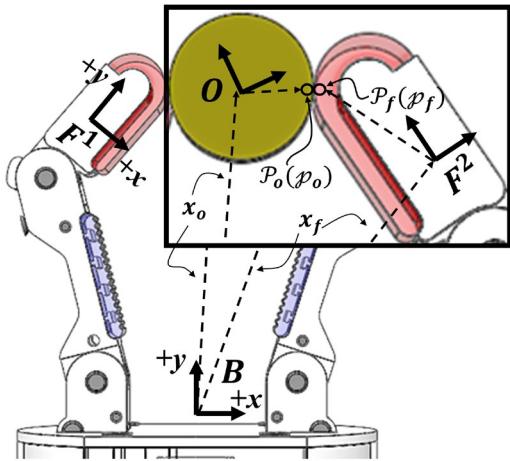


Fig. 7. Sliding contacts are detected by verifying rolling contact constraints cannot be satisfied.

$$\dot{x}_o^c = \dot{x}_o + \dot{R}_o \mathcal{P}_o(\mathcal{P}_o) \quad (29)$$

We similarly calculate  $\dot{x}_f^c$ . Given the velocity threshold,  $\epsilon_v$ , we develop our final constraint:

$$\dot{x}_o^c - \epsilon_v \leq \dot{x}_f^c \leq \dot{x}_o^c + \epsilon_v \quad (30)$$

Thresholds  $\epsilon_p$  and  $\epsilon_v$  are tuned heuristically according to the frequency of the camera and the accuracy of contact point estimation. If constraints (28) or (30) do not hold, it further implies that sliding occurred at the contact. Upon detection, the state of the system is self-tagged and manipulation continues without object reset.

#### D. Standardizing Object Reset

Collecting training data for dexterous manipulation is a labor-intensive process, as constant monitoring and manual intervention is frequently required to reset the system due to object drops during manipulation, or from other undesired system scenarios, e.g. actuators at torque limits. Moreover, during reset, it is unlikely that a human can completely standardize the initial grasp of the object, as a human placing the object within the grasp may often cause undesired deviations in the initial pose of the object before manipulation. In order to collect data in a self-supervised manner, we fabricated a system to autonomously and precisely reset the object as to standardize the initial grasp before manipulation.

The automated reset system (Fig. 8) is comprised of an object crane and a stabilization beam with an affixed magnet on the end. For each of the objects tested, two magnets were affixed to opposite sides of the body and a lightweight fishing line was strung through the center. During the case of object drop or stuck, the crane raised and the stabilization arm was lowered to the reset position as to adhere to the object magnets. Once the hand reacquired the grasp, which is standardized due to the positioning of the magnets, the stabilization beam lifts out of the way and the crane lowers. This provides slack to the connection between the crane and the object, and allows the hand to freely manipulate the object once again.

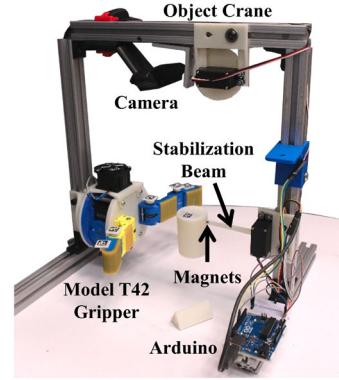


Fig. 8. An object crane and stabilization beam with affixed magnets accurately resets the object into the same configuration for each trial.

## VI. DATA COLLECTION

We design gripper variants that are generally within the bounds identified in Sec. IV and test the applicability of mechanics-based features empirically, as to evaluate their robustness in physical environments. In a self-supervised manner, we autonomously collect and tag data on 6 gripper variants—one variant for training and five for testing. Once a grasp is acquired after reset, the object was manipulated with randomly selected Cartesian velocity references that operated for a period between 0.5 and 2.5 seconds. A “normal” observation was collected once no other mode was detected for more than 5 seconds. The self-supervised training data was first collected online, randomly selected as to adhere to the leveling of the data distributions, and was then trained and tested offline.

A total of six 3D Printed ABS objects of negligible weight (~20g) and differing geometries in the manipulation plane were created for experimentation (Fig. 9, 10). The center of each object contained a hole where the object crane was attached. For each object, magnets were affixed to opposite ends as to enable attachment to the stabilization beam for object reset. In the training data, only four objects were used. The other two objects, the oval and the pear, were used as novel test objects.

Training data was collected with a single, symmetric Model T42 gripper variant (PL-PL) with Dynamixel RX-28 actuators. This naming convention signifies a “pivot-long proximal link, and a pivot-long distal link” configuration. From four objects, the two rectangles and the two circles, a total of 3500 modes were collected for training, with an equal mode distribution over each of the objects: 1000 normal, 1000 drop, 1000 stuck, and 500 sliding. Data was tagged and collected until the minimum for each mode was fulfilled. Afterward, overflow mode observations were selected randomly and excluded from the observation set. It is important to note sliding only occurs on objects with flat surfaces, i.e. the rectangular objects (Fig. 2, 9). Therefore, the number of sliding points recorded for each variant was determined by which type of objects were used during collection. The training data workspace is presented in Fig. 11. We note that, generally, the mode regions are symmetric about the central axis of the gripper.

Testing data was collected by equipping the hand with 5

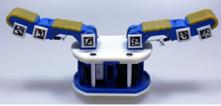
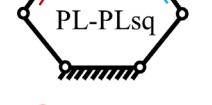
	Variant	Data	Properties	Objects
Training	 	1000 Rolling 1000 Drop 1000 Stuck 500 Sliding	<ul style="list-style-type: none"> <li>‘Pivot Long – Pivot Long’</li> <li>Dynamixel RX-28</li> <li>Hard Stops: 90°</li> </ul>	
Testing	 	100 Rolling 100 Drop 100 Stuck 50 Sliding	<ul style="list-style-type: none"> <li>‘Pivot Long – Pivot Short’</li> <li>Dynamixel RX-28</li> <li>Hard Stops: 90°</li> </ul>	
	 	100 Rolling 100 Drop 100 Stuck 50 Sliding	<ul style="list-style-type: none"> <li>‘Pivot Short – Pivot Long’</li> <li>Dynamixel RX-28</li> <li>Hard Stops: 90°</li> </ul>	
	 	100 Rolling 100 Drop 100 Stuck 0 Sliding*	<ul style="list-style-type: none"> <li>‘Pivot Long – Pivot Long Squared’</li> <li>Dynamixel XM-430</li> <li>Hard Stops: 90°</li> </ul>	
	 	100 Rolling 100 Drop 100 Stuck 50 Sliding	<ul style="list-style-type: none"> <li>‘Pivot Short – Flexure Long’</li> <li>Dynamixel XM-430</li> <li>Hard Stops: 70°</li> </ul>	
	 	100 Rolling 100 Drop 100 Stuck 50 Sliding	<ul style="list-style-type: none"> <li>‘Pivot Short – Pivot Short – Pivot Short’</li> <li>Dynamixel XM-430</li> <li>Hard Stops: 60°</li> </ul>	

Fig. 9. Manipulation was performed on 6 different gripper variants. The base variant used in training, the symmetric PL-PL gripper, was evaluated with four different objects (small circle, large circle, small rectangle, and large rectangle). A total of 3500 points for training were collected for the four identified modes. The five test variants (PL-PS, PS-PL, PL-PLsq, PS-FL, and PS-PS-PS) then performed manipulation with two of the six test objects. Two novel objects were added in testing (medium oval and medium pear). During manipulation, 50 occurrences of each mode were collected for each gripper-object combination. A quarter is placed next to the objects for size reference. \*Sliding only occurs with rectangular objects, therefore limiting the number of sliding cases.

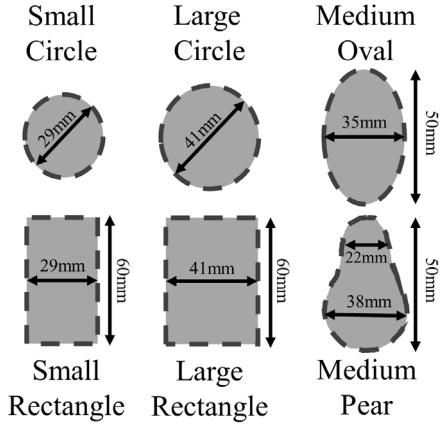


Fig. 10. Six objects were used for testing and training. In the manipulation plane, object geometries are classified either as a circle, rectangle, oval, or pear.

different finger configurations (Fig. 9). Variations incorporated changes to the link lengths, fingerpad curvatures, joint types, actuator models, and the number of links compared to the original PL-PL setup. The five variants included PL-PS and PS-PL fingers (Dynamixel RX-28), and PS-FL, PL-PLsq, and PS-

PS-PS fingers (Dynamixel XM-430). Different joint stiffness ratios were observed for the PS-FL and PS-PS-PS setups compared to the other four variants. Additionally, the distal hard stop was 70° for the PS-FL variant, and the two hard stops were 60° for the PS-PS-PS variant, compared to all other variants with a distal hard stop of 90°. In each of the five variants used for testing, a total of 50 observations were collected for each mode-object pair. Since in most cases two objects were tested and only one object recorded any sliding, we recorded 100 normal, 100 drop, 100 stuck, and 50 sliding for each variant.

More formally, during data collection we form a feature set,  $\mathcal{S}$ , comprised of features from Sec. III, and a class set,  $\mathcal{R}$ , while manipulating the objects. Denoted by,

$$\mathcal{S}_n = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{w}^1, \mathbf{w}^2, \mathbf{w}_p^1, \mathbf{w}_p^2, \dots, \mathbf{g}_{min}, \mathbf{g}_{max}, \mathbf{h}_{min}, \mathbf{h}_{max}, c_f^1, c_o^1, c_f^2, c_o^2) \in \mathbb{R}^{14}$$

an input feature, and

$$\mathcal{r}_n = (m) \in \{normal, drop, stuck, slide\}$$

an output feature. The dataset is defined as,

$$\mathcal{S} = \{\mathcal{S}_n\}_{n=1:M} \quad \mathcal{R} = \{\mathcal{r}_n\}_{n=1:M}$$

where its size,  $M$ , has the same number of normal, drop, stuck, and sliding cases for each gripper-object combination.

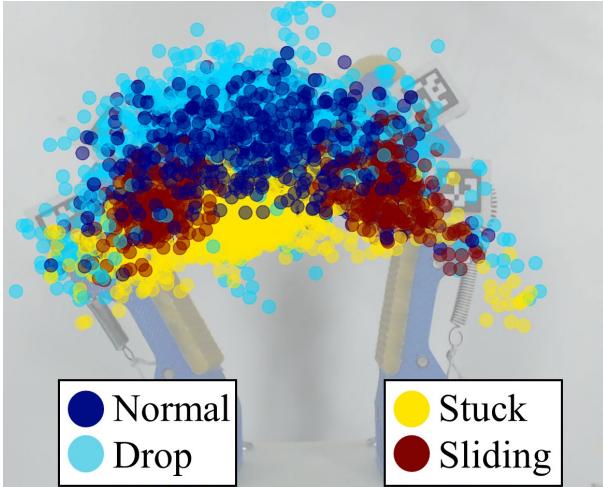


Fig. 11. Regions in the workspace where modes occur. Markers indicate the centroid of the object when a mode was detected.

## VII. RESULTS

### A. Classifier Identification and Observation Reduction

We were first interested in obtaining the best cross validation score given the training feature set, ( $\mathcal{S}_{train} \in \mathbb{R}^{3500 \times 14}$ ,  $\mathcal{R}_{train} \in \mathbb{R}^{3500}$ ), of the symmetric PL-PL setup. In the self-supervised learning approach taken in this work, we evaluated three different predictive models: Random Forests (RF) [46], Support Vector Machines–linear kernel (SVM-l), and Support Vector Machines–radial kernel (SVM-r). We chose these three classifiers for their extended use in the robotics literature, and due to the fact that other classifiers, e.g. Neural Networks, likely need more data than what was collected in this work. To determine the best classifier for this data, we performed a five-fold cross validation on the training dataset using each classifier. As presented in Table II, the RF classifier performed the best, with an accuracy of 92.3% for all four modes, followed by 88.6% (SVM-r) and 85.4% (SVM-l). For the RF classifier, we calculate a classification accuracy of 85%, 94%, 95%, and 86% for the normal, drop, stuck, and sliding cases, respectively. We note that drop and stuck cases are often classified with higher accuracy than sliding and normal cases. This quality is advantageous as it allows the system to more correctly avoid potentially hazardous modes to stay well within the workspace. For the remainder of this work, we evaluate classification with the RF classifier by building 50 weak learners (shallow trees of depth 10) split according to a Gini impurity measure and averaging each tree’s prediction to determine mode classification.

We were interested in how much data was required to maintain high classification accuracies via self-testing. Using all 14 features from  $\mathcal{S}_n$  and the RF classifier, we split the data into two sections: one with 2800 observations (training) and the other with 700 observations (testing), all while keeping the number of modes in each balanced. We continually reduce the number of data points in the training set by 100, removing observations randomly, and test on all 700 test observations. After training the classifier once observations were sequentially removed, we note that the classifier performs similarly with

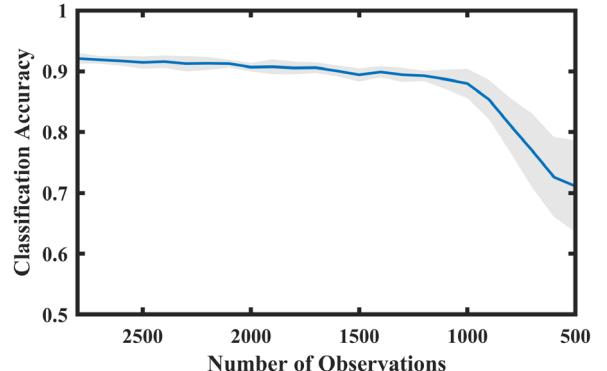


Fig. 12. Validation of training data size by reducing observations.

1200 observations as it does with 3500 observations (accuracy reduces by 3.4%). This 1.7:1 data ratio underscores that sufficient data was collected via self-supervision (Fig. 12).

TABLE II  
FIVE-FOLD CROSS VALIDATION SCORES ON TRAINING SET

Classifier	Random Forests	SVM - Linear	SVM - Radial
Score	$92.3 \pm 0.4\%$	$85.4 \pm 0\%$	$88.6 \pm 0\%$

### B. Classification Accuracy

For the remainder of our analysis, the RF classifier was trained with all 3500 data points using the symmetric PL-PL gripper variant. The particular test set, ( $\mathcal{S}_{test}, \mathcal{R}_{test}$ ), was changed according to which of the five gripper variants was being tested.

Using all 14 features from  $\mathcal{S}_n$ , each of the test grippers were evaluated individually (Fig. 13). The classification accuracy of the PL-PS and PL-PLsq variants were highest, with a classification accuracy of 90.6%. The second highest classification accuracy was realized in the PS-PL variant with an accuracy of 85.1%. As provided in the decision matrices in Fig. 13 (leftmost column), the PL-PS variant was able to classify normal, drop, and stuck with 84%, 97%, and 99% accuracy, respectively. Classification for sliding dropped to 72%, where it had difficulty distinguishing from the normal mode. The PL-PLsq variant did not have sliding modes, since data was not collected with rectangular objects. Therefore, the lowest classification accuracy was observed with the drop mode (80% accuracy).

This high misclassification of drop is interesting, as it is significantly lower than other variants (97%, 93%, 96%, 88%) with the same feature set. This can be largely attributed to the shifted workspace of the PL-PLsq gripper. As provided by the workspace plots in the rightmost column of Fig. 13, compared to the other variants, the modes detected for this variant are shifted to the left of the workspace. Additionally, many drop cases seem to occur in the middle of the workspace, where normal classification would typically be predicted. This artefact is due to the differing geometry of the fingerpad, as it was difficult for the finger to manipulate on the right side of the workspace since the “sharp” edge of the finger prevented a rolling contact to the tip of the finger.

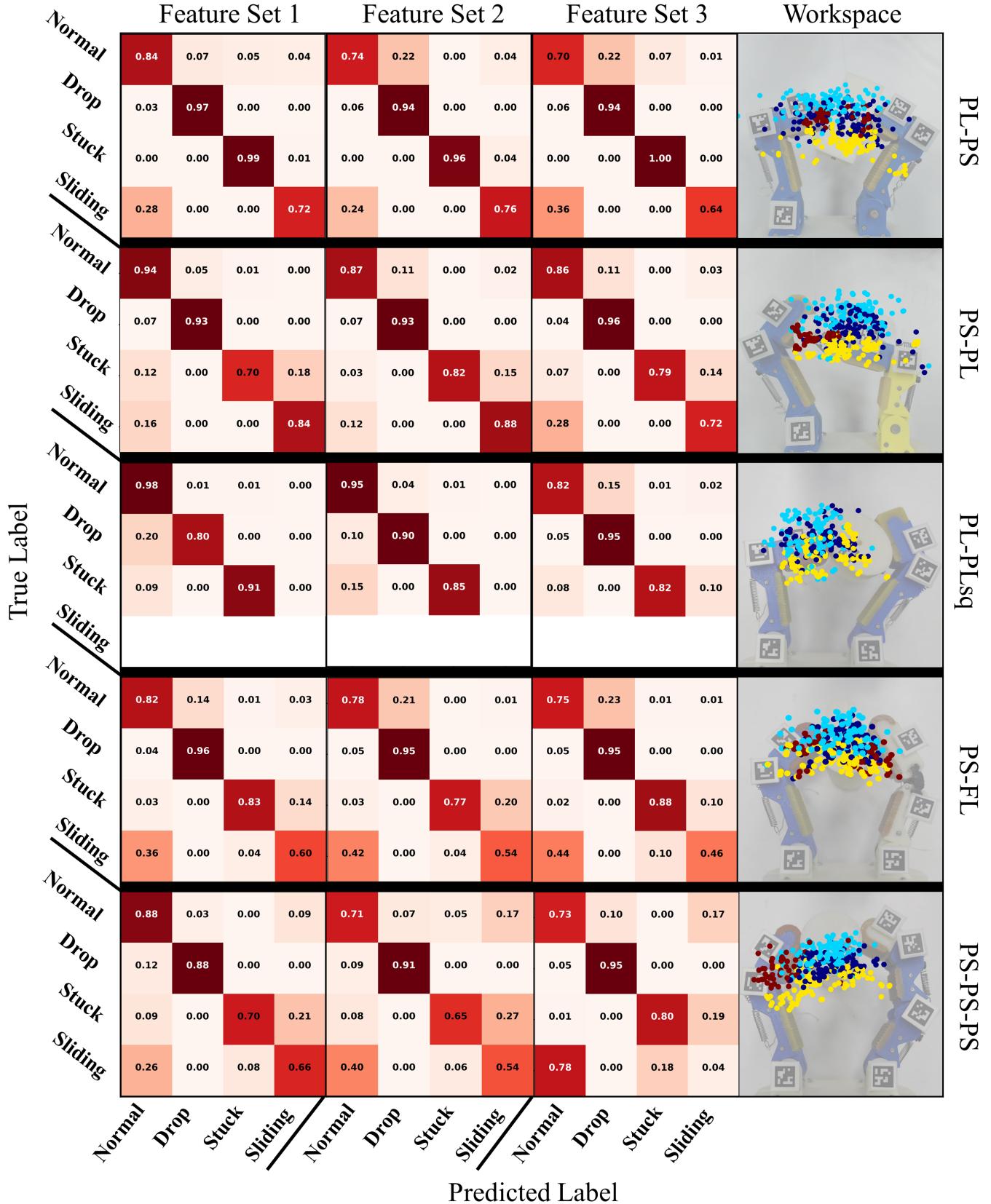


Fig. 13. (Left three columns) Confusion matrices for each gripper variant given differing feature sets (described in Sec. VII.C). (Right column) Object centroid position for modes detected within the workspace of each gripper variant. (Light Blue-Drop, Dark Blue-Normal, Yellow-Stuck, Red-Sliding)

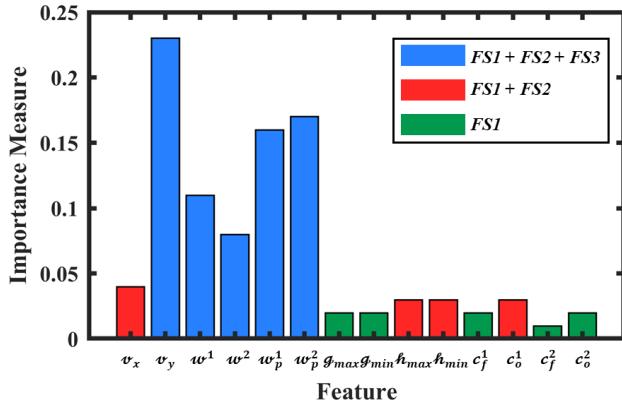


Fig. 14. Feature importance measures provided by the Random Forest Classifier via Gini impurity. Features in blue are included in Feature Sets 1,2,3, features in red are included in Feature Sets 1,2, and features in green are included in Feature Set 1. See Table III.

Of the other gripper variants, the 3-link PS-PS-PS performed the worst with a total classification accuracy of 79.8%. In general, for all variants, predicting sliding was difficult as without tactile sensors and with differing friction within the joints of the fingers, the classifier struggles to determine forces applied at the contact point.

### C. Feature Reduction

A benefit of the Random Forests classifier is its ability to inherently provide “feature importance measures”, or values that signify how much each feature contributed to the classification decision—providing intuition as to which features were most important during manipulation. In this work, we use a Gini impurity measure to calculate this importance metric, which is a standard often used in ensemble tree classifiers. It works as follows: once a random set of features are selected to determine a split, the Gini impurity represents the likelihood that an accurate classification is predicted given a random class from the distribution of labels. As these splits are calculated for each tree in the forest, the importance measure averages the Gini measures for each split and further signifies the feature’s importance, or more generally, how much “purity” they contributed to the forest. Feature importance measures are reported in Fig. 14, where we note that the largest contribution to classification success is attributed to the y-axis Cartesian velocity reference ( $v_y$ ), finger manipulability measures ( $w^1, w^2$ ), and the penalized finger manipulability measures ( $w_p^1, w_p^2$ ).

Using the feature importance measures, we define 3 feature sets ( $FS1, FS2, FS3$ ) consisting of 14, 9, and 5 features, respectively, to further test classification (Table III). By testing accuracy with each feature set, we can provide greater intuition as to what features were most important. The results from this analysis are reported in Table IV and Fig. 13. Note that the results from  $FS1$  were previously described in Sec. VII.B.

After feature reduction, some variants such as PS-PL ( $FS1: 85.1\%, FS2: 85.0\%, FS3: 84.4\%$ ) and the PL-PLsq ( $FS1: 90.6\%, FS2: 89.2\%, FS3: 87.6\%$ ) provided consistent classification scores even with the reduction of features. Interestingly, the PS-

PS-PS variant obtained nearly the same accuracy between  $FS1$  (79.8%) and  $FS2$  (79.0%), even with the reduction of 5 total features (14 features to 9 features). The PS-FL sees a sharp decrease in classification from  $FS1$  to  $FS2$ , but then maintains a similar classification accuracy for  $FS3$ . What is also interesting to note, while the overall accuracy of the PL-PLsq decreases, the drop accuracy increases with the reduction of features ( $FS1: 80\%, FS2: 90\%, FS3: 95\%$ ).

TABLE III  
FEATURE SETS DETERMINED BY FEATURE REDUCTION

	Feature Vector
Feature Set 1 ( $FS1$ )	$s_n = (v_x, v_y, w^1, w^2, w_p^1, w_p^2, \dots, g_{min}, g_{max}, h_{min}, h_{max}, c_f^1, c_o^1, c_f^2, c_o^2)$
Feature Set 2 ( $FS2$ )	$s_n = (v_x, v_y, w^1, w^2, w_p^1, w_p^2, h_{min}, h_{max}, c_o^1)$
Feature Set 3 ( $FS3$ )	$s_n = (v_y, w^1, w^2, w_p^1, w_p^2)$

TABLE IV  
CLASSIFICATION ACCURACY WITH DIFFERING FEATURE SETS

Variant	Feature Set 1	Feature Set 2	Feature Set 3
PL-PS	$90.6 \pm 1.2\%$	$87.9 \pm 0.9\%$	$84.1 \pm 1.6\%$
PS-PL	$85.1 \pm 0.9\%$	$85.0 \pm 1.1\%$	$84.4 \pm 2.1\%$
PL-PLsq	$90.6 \pm 2.2\%$	$89.2 \pm 1.2\%$	$87.6 \pm 1.4\%$
PS-FL	$84.8 \pm 1.8\%$	$78.3 \pm 2.4\%$	$77.6 \pm 1.6\%$
PS-PS-PS	$79.8 \pm 0.7\%$	$79.0 \pm 1.3\%$	$70.0 \pm 0.8\%$

Of the five gripper variants tested, the PL-PLsq maintained the best classification score. This success is likely attributed to two things. First, this variant, dimensionally, is the closest variant to the original PL-PL used in training, as the only difference is the squared fingertip on the right finger. Second, this variant was tested with two rounded objects (circle and oval), and therefore no sliding occurred during manipulation, which is normally the most difficult to classify. While evaluated variants that were tested with sliding cases, the PS-PL performed the best with a total classification accuracy of 84.4% ( $FS3$ ). This variant also had the highest sliding accuracy among any of the five variants throughout all features sets, which can likely be attributed to the fact that this variant has the same distal link as the training PL-PL variant. The PS-PS-PS gripper variant performs the worst of the five variants—this variant has a more limited workspace due to the hard stops at  $60^\circ$  at each of the links. Additionally, sliding only occurs on the left side of the workspace, since the flat surface of the right most-distal link rarely comes in contact with the object. As depicted by the confusion matrices in Fig. 13, in general, as the number of features is reduced, the ability for the system to accurately predict sliding greatly reduces. For example, in the PS-PS-PS variant for  $FS1$ , the classification for sliding is 66%, but in  $FS3$  the accuracy is just 4%. This is a fairly specific case, as the classification accuracy for sliding only differs from a maximum of 12% for the three other gripper variants (from  $FS1$  to  $FS3$ ).

As previously discussed, the rightmost column of Fig. 13 provides workspace plots for the 5 test gripper variants. Plotted

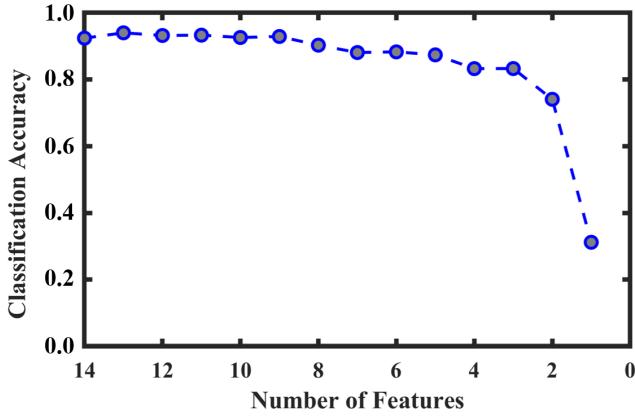


Fig. 15. Five-fold cross validation accuracy of the PL-PL training variant. Features were reduced one at a time subject to their classification accuracy contribution (see Fig. 14).

points depict the centroid of the object when a mode was detected. While tested objects were of various geometries, this plot generally presents where modes were likely to occur within the workspace. It is interesting to note how the “regions” for different modes change according to the gripper variant, especially how varied they are compared to the training hand, PL-PL, in Fig. 11. For example, sliding only occurred on the left side of the workspace for the PS-PL and PS-PS-PS variants, and a large number of drop cases occurred in the middle of the workspace for the PS-FL variant. These workspace plots underscore how, where joint configuration and object center location inside of the workspace is important, the properties of the hand-object system must be accounted for in order to accurately predict modes of manipulation.

#### D. Single-Component Feature Reduction

As stated in Sec. VII.B, some variants were not as susceptible to higher classification errors given feature reduction techniques, while others were more affected. It was our interest to perform feature reduction techniques by removing one feature at a time, instead of in sets, as to validate our approach. We begin by removing features from least important to most important according to the measures presented in Fig. 13.

The results to this feature reduction are presented in Fig. 15. While performing this task on the PL-PL variant with a total of 3500 observations, we note that the cross-validation accuracy remains around 93% while having 9 or more features. Thereafter, when only 8 features remain, the accuracy drops to 87% and continues until 5 features remain. Once only 4 features are used for classification, the accuracy starts to decline, as it is difficult to determine the decision boundary. This feature reduction test validates the decision for 14, 9, and 5 features for *FS1*, *FS2*, and *FS3*, respectively (Sec. VII.C), as these are volatile intervals when accuracy will likely drop.

#### E. Online Classification

Detection of modes, and their associated regions, are somewhat fluid (as presented by the workspace plots) and in general, we are interested to see if modes can be successfully predicted online to promote safe manipulation. We

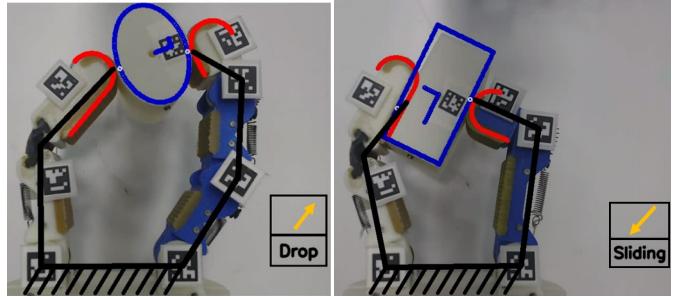


Fig. 16. Online classification of two novel gripper variants. The arrow signifies the Cartesian velocity reference and the text (Drop or Sliding) signifies the predicted mode. (Left) A PS-FL left finger and a PS-PS-PS right finger perform manipulation and the online classifier predicts a drop will occur given the Cartesian velocity reference. (Right) A PS-FL left finger and a PL-PS right finger predicts sliding will occur during manipulation.

implemented this RF prediction model in an online framework to evaluate classification accuracy with two novel gripper variants not evaluated in the previous sections (Fig. 16). The first variant was tested with the medium oval and consisted of a PS-FL left finger and a PS-PS-PS right finger. The second variant was tested with the small rectangle and was comprised of a PS-FL left finger and a PL-PS right finger. As before, the mechanics-based features used for testing were extracted online using markers attached to rigid links of the hand. The gripper was commanded through random Cartesian velocity references for a period between 0.5-4.0 seconds to attempt to cover the entire workspace. Once a mode other than normal was detected for a period between 0.1-1.0 seconds, the Cartesian velocity reference changed randomly to either stop manipulation or guide the object back towards the middle of the workspace.

Online classification using the first novel variant properly classified modes normal, drop, and stuck within its workspace for the oval object. In addition to these three modes, the second variant also included the “sliding” mode. To test the efficacy of this online detection, the classifier was run on each hand 5 times for 5 minutes. Cartesian velocity references were selected randomly, with a goal of remaining within the manipulable region of the gripper. For the first variant, 3 out of the 5 executions were successfully run for a total of five minutes. The object was manipulated safely within the workspace and was diverted towards the center of the workspace when a mode other than normal was detected. For the other two executions, a dropped object was detected at 3 minutes 21 seconds and 4 minutes 5 seconds. For the second variant, 4 out of the five executions successfully completed 5 minutes of manipulation. The final failed execution successfully manipulated the object for 2 minutes and 34 seconds. This failure was due to the amount of sliding the object had undergone without detection 12 seconds before task failure.

## VIII. DISCUSSION AND FUTURE WORK

In this work, we showed that by learning from mechanics-based features, which represent high-level properties of the hand-object system, we were able to successfully transfer mode prediction accuracies between different gripper variants. We first provided bounds by which mechanics-based features were

likely to better transfer than their joint-based counterparts. We then tested this notion physically with different hand variants. Specifically, the 92.3% five-fold cross validation accuracy of the training variant was marginally greater than the 90.6% classification accuracy of the PL-PLsq and PL-PS variants. However, we did note that these features presented shortcomings in accurately predicting sliding when stiffness ratios changed between variants. Additional sensing modalities such as tactile sensing at the fingertips would likely be beneficial for classification.

The features included in *FS3*, or the set containing finger manipulability measures, contribute the most to the success of the classifier (75.2% of the Gini impurity measure). The  $v_y$  component benefits classification in that, according to the workspace plots, it likely discriminates between the drop, normal, and stuck regions as the  $y$ -position component passes through all three. When coupled with the finger manipulability measures, these values together determine where the object is within the workspace and where it is headed, and in general, the hand-object configuration. It is our belief that the other features defined, such as singular values of the Hand-Object Jacobian and the contact curvatures, are important for stable manipulation capabilities when fingerpad curvatures change more drastically, or different gripper types (underactuated vs. fully actuated) are observed.

This work elucidates the beginning of what we consider a promising approach for learning models in dexterous manipulation. While we recognize the drawbacks and inaccuracies in predicting sliding as the gripper becomes more asymmetric, this approach has proven to be successful for the other three modes, and was completed without the use of tactile sensing. Although conceptually backed by simulation, the majority of our analysis consisted of data that was collected physically, which allows us to capture uncertainties of the real world. In future work, we plan to investigate this approach further by extending this sort of classification to the spatial manipulation case, investigating how time series data aids in prediction accuracy (e.g. HMMs), further modeling this approach for deformable contacts and objects, and testing such methods on more commercial, readily-available robot grippers. Furthermore, we are interested in how adding single unit tactile sensors at the fingertips may be beneficial in detecting sliding cases when using different gripper variants.

While this approach of using mechanics-based features for learning dexterous manipulation can be applied to any hand design, it is particularly useful for soft, compliant, or underactuated hands that typically do not have tactile sensors or joint encoders. Fundamentally, dexterous manipulation extends the workspace of the manipulator and is a valuable tool for the future of robotics in society. We hope that the robustness demonstrated by testing different gripper variants encourages researchers to search for features that represent higher-level properties of the system for a more enlightened discussion on learning system models.

## REFERENCES

- [1] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 652–662, 2000.
- [2] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*, 2011, pp. 1–7.
- [3] J. van den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-Based Robotic Cloth Folding," Springer, Berlin, Heidelberg, 2010, pp. 409–424.
- [4] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, "Towards Robotic Feeding: Role of Haptics in Fork-Based Food Manipulation," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1485–1492, Apr. 2019.
- [5] R. Balasubramanian, J. T. Belter, and A. M. Dollar, "External Disturbances and Coupling Mechanisms in Underactuated Hands," in *Volume 2: 34th Annual Mechanisms and Robotics Conference, Parts A and B*, 2010, pp. 175–184.
- [6] OpenAI *et al.*, "Learning Dexterous In-Hand Manipulation," *eprint arXiv:1808.00177*, Aug. 2018.
- [7] B. Calli, K. Srinivasan, A. Morgan, and A. M. Dollar, "Learning Modes of Within-Hand Manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3145–3151.
- [8] B. Calli, A. Kimmel, K. Hang, K. Bekris, and A. Dollar, "Path Planning for Within-Hand Manipulation over Learned Representations of Safe States," *Int. Symp. Exp. Robot.*, 2018.
- [9] A. S. Morgan, W. G. Bircher, B. Calli, and A. M. Dollar, "Learning from Transferable Mechanics Models: Generalizable Online Mode Detection in Underactuated Dexterous Manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5823–5829.
- [10] K. M. Lynch, "Estimating the friction parameters of pushed objects," *Proc. 1993 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS '93)*, vol. 1, no. 1, pp. 186–193, 1993.
- [11] T. Okada, "Computer Control of Multijointed Finger System for Precise Object-Handling," *IEEE Trans. Syst. Man. Cybern.*, vol. 12, no. 3, pp. 289–299, 1982.
- [12] J. Kerr and B. Roth, "Analysis of Multifingered Hands," *Int. J. Rob. Res.*, vol. 4, no. 4, pp. 3–17, Jan. 1986.
- [13] D. L. Brock, "Enhancing the dexterity of a robot hand using controlled slip," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, IEEE, 1988.
- [14] A. M. Dollar and R. D. Howe, "The Highly Adaptive SDM Hand: Design and Performance Evaluation," *Int. J. Rob. Res.*, vol. 29, no. 5, pp. 585–597, Apr. 2010.
- [15] J. C. Trinkle, "A quasi-static analysis of dexterous manipulation with sliding and rolling contacts," *Int. Conf. Robot. Autom.*, pp. 788–793, 1989.
- [16] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, vol. 29. 1994.
- [17] M. T. Mason and J. K. Salisbury, *Robot hands and the mechanics of manipulation*. MIT Press, 1985.
- [18] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 255–262.
- [19] K. Hertkorn, M. A. Roa, and C. Borst, "Planning in-hand object manipulation with multifingered hands considering task constraints," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 617–624.
- [20] R. Michalec and A. Micaelli, "Stiffness modeling for multi-fingered grasping with rolling contacts," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 601–608.
- [21] L. U. Odhner and A. M. Dollar, "Dexterous manipulation with underactuated elastic hands," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 5254–5260.
- [22] Yanmei Li and Imin Kao, "A review of modeling of soft-contact fingers and stiffness control for dexterous manipulation in robotics," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, pp. 3055–3060.
- [23] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *J. Dyn. Syst. Meas. Control*, vol. 107, no. 1, pp. 1–7, Mar. 1985.
- [24] D. Prattichizzo, M. Malvezzi, M. Gabiccini, and A. Bicchi, "On

Motion and Force Controllability of Precision Grasps with Hands Actuated by Soft Synergies,” *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1440–1456, Dec. 2013.

[25] R. Balasubramanian, J. T. Belter, and A. M. Dollar, “Disturbance Response of Two-Link Underactuated Serial-Link Chains,” *J. Mech. Robot.*, vol. 4, no. 2, p. 021013, May 2012.

[26] C. Della Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, “Toward Dexterous Manipulation with Augmented Adaptive Synergies: The Pisa/IIT SoftHand 2,” *IEEE Trans. Robot.*, vol. 34, no. 5, pp. 1141–1156, Oct. 2018.

[27] A. Rocchi, B. Ames, Zhi Li, and K. Hauser, “Stable simulation of underactuated compliant hands,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4938–4944.

[28] T. Laliberté and C. M. Gosselin, “Simulation and design of underactuated mechanical hands,” *Mech. Mach. Theory*, vol. 33, no. 1–2, pp. 39–57, Jan. 1998.

[29] B. Calli and A. M. Dollar, “Robust Precision Manipulation With Simple Process Models Using Visual Servoing Techniques With Disturbance Rejection,” *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 406–419, Jan. 2019.

[30] A. Sintov, A. S. Morgan, A. Kimmel, A. M. Dollar, K. E. Bekris, and A. Boularias, “Learning a State Transition Model of an Underactuated Adaptive Hand,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1287–1294, Apr. 2019.

[31] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4639–4644.

[32] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, “Robot Learning Manipulation Action Plans by ‘Watching’ Unconstrained Videos from the World Wide Web,” *Twenty-Ninth AAAI Conf. Artif. Intell.*, pp. 3686–3692, 2015.

[33] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, “Learning dexterous manipulation for a soft robotic hand from human demonstrations,” in *IEEE International Conference on Intelligent Robots and Systems*, 2016, vol. 2016-Novem, pp. 3786–3793.

[34] N. Abdo, H. Kretzschmar, L. Spinello, and C. Stachniss, “Learning manipulation actions from a few demonstrations,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 1268–1275.

Z. Su *et al.*, “Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor,” in *IEEE-RAS International Conference on Humanoid Robots*, 2015, vol. 2015-Decem, pp. 297–303.

[35] M. R. Tremblay and M. R. Cutkosky, “Estimating friction using incipient slip sensing during a manipulation task,” in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 429–434.

[36] H. Dang and P. K. Allen, “Learning grasp stability,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2392–2397.

[37] K. Hang *et al.*, “Hierarchical Fingertip Space: A Unified Framework for Grasp Planning and In-Hand Grasp Adaptation,” *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 960–972, Aug. 2016.

[38] M. A. Roa and R. Suárez, “Grasp quality measures: review and performance,” *Auton. Robots*, vol. 38, no. 1, pp. 65–88, Jan. 2014.

[39] T. Yoshikawa, “MANIPULABILITY OF ROBOTIC MECHANISMS.,” *Int. J. Rob. Res.*, vol. 4, no. 2, pp. 3–9, Jun. 1985.

[40] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, “Manipulability analysis,” *IEEE-RAS Int. Conf. Humanoid Robot.*, no. May, pp. 568–573, 2012.

[41] D. J. Montana, “Contact Stability for Two-Fingered Grasps,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 4, pp. 421–430, 1992.

[42] W. G. Bircher, A. M. Dollar, and N. Rojas, “A two-fingered robot gripper with large object reorientation range,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 3453–3460.

[43] B. Calli and A. M. Dollar, “Vision-based model predictive control for within-hand precision manipulation with underactuated grippers,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 2839–2845.

[44] A. B. A. Cole, J. E. Hauser, and S. S. Sastry, “Kinematics and control of multifingered hands with rolling contact,” *IEEE Trans. Automat. Contr.*, vol. 34, no. 4, pp. 398–404, Apr. 1989.

[45] J. L. Bentley and J. Louis, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.



**Andrew S. Morgan** (S’15) received the B.E. degree in Computer Engineering and the B.S. degree in Computer Science in 2017, from Youngstown State University, Youngstown, OH, USA, and the M.S. degree in Mechanical Engineering in 2019, from Yale University, New Haven, CT, USA. He is currently working towards the Ph.D. in Engineering in the GRAB Lab under Prof. Aaron Dollar. He is a recipient of the National Science Foundation Graduate Research Fellowship Program, awarded in 2019.

His research focuses on advancing dexterous manipulation capabilities of robots by coupling adaptive, compliant mechanism capabilities with machine learning.



**Aaron M. Dollar** (S’02-M’07-SM’13) received the B.S. degree in Mechanical Engineering from University of Massachusetts at Amherst, Amherst, MA, USA, in 2000, and the S.M. and Ph.D. degrees in Engineering Sciences from Harvard University, Cambridge, MA, in 2002 and 2007, respectively. He is a Professor of Mechanical Engineering and Materials Science and Computer Science at Yale University, New Haven, CT, USA.

His research interests include robotic grasping and manipulation, prosthetics and rehabilitation robotics, human and animal grasping and manipulation, and robot locomotion.



**Walter G. Bircher** (S’16) received the B.S. degree in Mechanical Engineering in 2014, from the University of Nebraska-Lincoln, Lincoln, NE, USA, and the M.S. degree in Mechanical Engineering in 2018, from Yale University, New Haven, CT, USA, where he is currently working toward the Ph.D. degree in engineering.

His research focuses on improving the dexterous capabilities of robotic grippers through design.