

# Autonomously Untangling Long Cables

Vainavi Viswanath<sup>\*1</sup>, Kaushik Shivakumar<sup>\*1</sup>, Justin Kerr<sup>\*1</sup>, Brijen Thananjeyan<sup>1</sup>, Ellen Novoseller<sup>1</sup>, Jeffrey Ichnowski<sup>1</sup>, Alejandro Escontrela<sup>1</sup>, Michael Laskey<sup>2</sup>, Joseph E. Gonzalez<sup>1</sup>, Ken Goldberg<sup>1</sup>

**Abstract**—Cables are ubiquitous in many settings, but are prone to self-occlusions and knots, making them difficult to perceive and manipulate. The challenge often increases with cable length: long cables require more complex slack management and strategies to facilitate observability and reachability. In this paper, we focus on autonomously untangling cables up to 3 meters in length using a bilateral robot. We develop new motion primitives to efficiently untangle long cables and novel gripper jaws specialized for this task. We present Sliding and Grasping for Tangle Manipulation (SGTM), an algorithm that composes these primitives with RGBD vision to iteratively untangle. SGTM untangles cables with success rates of 67% on isolated overhand and figure eight knots and 50% on more complex configurations. Supplementary material, visualizations, and videos can be found at <https://sites.google.com/view/rss-2022-untangling/home>.

## I. INTRODUCTION

Long cables are commonplace in household and industrial settings, from wires in homes to cables in manufacturing plants to ropes in sailing [13, 15, 21, 27]. Cables—defined here as single-dimensional deformable objects—often tangle and form knots, which can be unsightly, unsafe, and reduce utility. However, autonomously manipulating cables to untangle them is challenging due to their infinite-dimensional configuration spaces and tendency to form self-occlusions and knots [3, 20, 22, 19, 11]. These challenges grow with increasing cable length as longer cables allow more knots to form. Additionally, as robots untangle one segment of a cable, the other cable segments, which we define as the *slack*, could form new knots, occlude visibility, or impede grasping. The stiffness of cables further adds to the slack management challenge, because untangled slack may appear to hold its shape even without a knot present. Robustly manipulating long cables requires policies that perceive cable configurations and effectively manage slack during manipulation.

Prior work in robot cable untangling considers short cables and studies how to untangle dense knots, which have no open space between adjacent cable segments [3, 20, 22], or considers longer cables, estimated to be no longer than one meter in length, whose paths can be clearly traced using analytic methods [11]. This paper studies untangling knots in cables up to 3m long that have sufficient stiffness to prevent the formation of dense knots. Untangling such cables requires policies to actively manage the slack created during manipulation to prevent the formation of new knots and

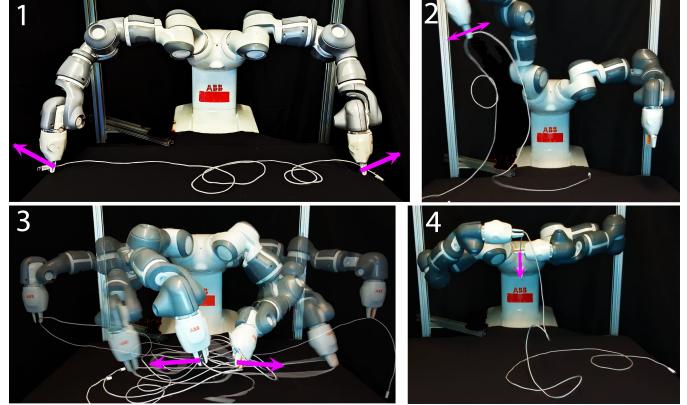


Fig. 1: Overview of Sliding and Grasping for Tangle Manipulation (SGTM): SGTM untangles a long cable by (1) performing endpoint separation to spread out the configuration, 2) shaking to reduce loops and reveal knots, 3) performing a caged pull apart primitive to untangle a knot, and (4) performing physical tracing to verify the only remaining loops are trivial.

crossings. We use overhead RGBD images for perception and a bilateral YuMi robot.

This paper makes 3 contributions: (1) Novel gripper jaw hardware for untangling that enables two modes of grasping: caging and pinching. Caging grasps enable the cable to slide along the gripper jaws without falling out, which is useful for physically tracing along the cable. Pinching grasps firmly hold the cable, preventing it from moving. (2) Three novel bilateral manipulation primitives for cable untangling: shaking, tracing, and caged pull-apart. Shaking moves can simplify the cable state before attempting other primitives. Physical tracing can slide along the cable to actively uncover knots and loosen dense structures. Caged pull-apart actions can loosen and undo knots using learning-based perception to identify isolated knots in the scene. The caged pull apart action slides apart from a knot center, avoiding the need to pinch and regrasp the cable repeatedly to undo knots as in prior work [20, 3, 22]. Building on our previous work, this paper contributes a learning-based perception pipeline which leverages RGBD sensing to identify cable endpoints and knots [22, 20, 3], which are used to select and parameterize the novel manipulation primitives. (3) A novel untangling algorithm, Sliding and Grasping for Tangle Manipulation (SGTM) (Fig. 1), to untangle cables of up to 3 meters in length from starting configurations that contain up to 2 overhand and figure 8 knots (Fig. 2), as well as self-crossings that do not form knots. To the best of our knowledge, this is also the first work on robot cable untangling that considers cables up to 3 meters with multiple knots.

<sup>1</sup>AUTOLAB at the University of California, Berkeley

<sup>2</sup>Toyota Research Institute

<sup>\*</sup>equal contribution

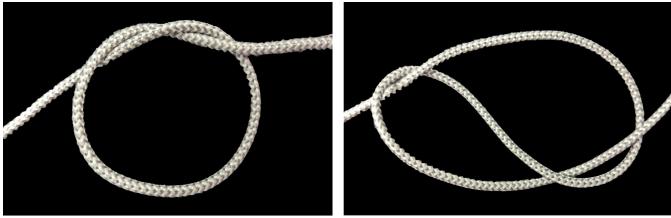


Fig. 2: **Knots that SGTM Untangles:** Left: an overhand knot. Right: a figure 8 knot.

## II. RELATED WORK

### A. Deformable Object Manipulation

Autonomously manipulating deformable objects such as cables (1D), fabric (2D), and bags (3D) is a challenging problem, as such materials have near-infinite configuration spaces, form self-occlusions, and are subject to complex dynamics. This work focuses on cables, which can present particularly complex configurations due to their tendency to form knots. While recent years have seen significant progress in robot deformable manipulation, for instance perception-driven algorithms for untangling cables [3, 20, 22, 11], smoothing and folding fabric [16, 24, 2, 5, 6], and placing objects into bags [17], to the best of our knowledge, the task of autonomously untangling long household cables remains unstudied.

Developing effective manipulation algorithms for deformable objects often involves choosing an abstract object representation on which an algorithm can operate. Existing work on deformable manipulation has leveraged a number of approaches for representing deformable objects. For instance, dense descriptors [1], which map object images to a pixel-wise embedding, have been successfully applied to both cable knot tying [19] and fabric folding and smoothing [2]. Yet, this approach is not robust to the severe deformation and occlusion common with lengthier cables. Other works attempt to address this challenge by learning dynamics models of deformable objects, which can then be used for planning [5, 23, 28, 9]. These methods have positioned non-knotted cables [28, 23] and fabric [5, 28, 9] into target configurations. Yet, for heavily self-occluded knotted cables, robust state estimation remains challenging, while dynamics are governed by complex knot structure. Instead of estimating the full state of the cable, our work leverages perception-based keypoint prediction methods and knowledge about the geometric problem structure.

Several works meanwhile propose end-to-end model-free visuomotor learning approaches for deformable manipulation. These include model-free reinforcement learning, which has been applied successfully to fabric smoothing and folding [12, 25, 7] and straightening unknotted rope [25], as well as leveraging optical flow to predict fabric manipulation actions given a target image [24]. Imitation learning approaches, which require demonstrations, have also seen success in manipulating fabric and non-knotted ropes [16, 17, 14]. However, these model-free approaches are highly sample inefficient for cable untangling, as they do not leverage geometric knowledge specific to cable structure or untangling.

### B. Cable Manipulation and Untangling

Existing works on robot cable untangling do not consider long cables, such as those commonly found in households, which pose additional challenges compared to short cables. This paper extends several ideas from prior methods, which use a combination of vision-based perception and domain structure to extract the state features necessary for untangling. Firstly, Lui and Saxena [11] start from initial configurations in which the cable's self-crossings are distinctly visible and use RGBD sensing to extract the cable's full state via classical feature extraction. The features are used to construct a graphical representation of the cables' self-crossings, which are used for untangling. The cables in this work are of unspecified length but do not appear to be longer than one meter in length. Meanwhile, in [3, 22, 20], the robot does not perform full state estimation and instead predicts keypoints to parameterize open-loop untangling actions such as crossing removal and pulling the endpoints apart. The robot untangles very densely knotted single and multi-cable knots in short cables without significant slack. Similar to these prior works, SGTM uses RGBD sensing and predicts parameterized actions that do not explicitly require full state estimation. However, while prior works focus on short cables, SGTM is designed for long cables, in which the cable's underlying knot structure can be significantly obfuscated due to the additional slack, which can create a number of self-crossings and self-occlusions that are not part of a knot.

This work defines novel motion primitives for cable untangling and employs perception-based methods to detect keypoints for executing these primitives. Several of these novel manipulation primitives slide along the length of the cable, which enable the robot to undo knots faster and physically trace the cable to check for knots. Sliding motions in cable manipulation were studied in [18] using a tactile gripper with learning-based control to keep the cable centered in the grippers when sliding along its length. Instead of using tactile sensing to fix the cable in the grippers, we propose a physical gripper augmentation that enables two grasping modes without sophisticated control techniques: caging, which enables sliding, and pinching, which constrains all movement in the gripper (Figure 4). This approach allows for a lower-profile gripper which can fit in tighter knot configurations and can speed up manipulation time due to the absence of active control. In addition, caging grasps enable gripper motion along the cable without applying force to the cable, a mode in which a tactile sensor could not collect feedback.

## III. PROBLEM STATEMENT

### A. Workspace Definition and Assumptions

We define an  $(x, y, z)$  Cartesian coordinate frame containing a bilateral robot and a flat manipulation surface that is parallel to the  $xy$ -plane. We assume that the workspace contains an overhead RGB-D sensor facing the manipulation surface from above and further assume that the rigid transformations between the sensor, robot, and workspace coordinate frames are known.

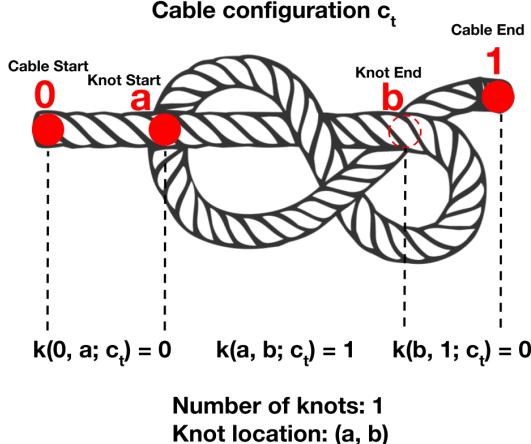


Fig. 3: **Knot Definition:** for a cable configuration at time  $t$ ,  $c_t$ , a knot exists between two points  $(a, b)$  if pulling apart at those points will not result in a straight cable. In this example, no knots exist in  $(0, a)$  and  $(b, 1)$ ; thus,  $k(0, a; c_t) = 0$  and  $k(b, 1; c_t) = 0$ . A figure 8 knot exists between indices  $a$  and  $b$ , and so  $k(a, b; c_t) = 1$ . Observe that  $(a, b)$  is the smallest interval that contains the knot, such that  $\mathcal{K}(c_t) = \{(a, b)\}$ .

At the start of a trial (time  $t = 0$ ), the manipulation workspace entirely contains a cable  $\mathcal{C}$  with cross-section radius  $r$  and length  $l$ , which traces a continuous volume  $c_t(s) : [0, 1] \rightarrow (x, y, z)$  in the workspace. The parameter  $s \in [0, 1]$  is used to index into the path, with  $c_t(s)$  corresponding to the center of the circular cable cross-section with distance  $sl$  from the endpoint defined as the first endpoint. The circular slice lies in the plane orthogonal to the cable's direction,  $\nabla_s c_t(s)$ . We assume that the cable can be segmented from the manipulation surface using color thresholding and that in the starting configuration  $c_0$ , the entire cable is contained in the workspace. The initial cable configuration can form self-crossings, which could include knots or loops. A cable segment contains at least one *knot* if pulling it taut by pulling cable segments on either side of it in opposite directions does not result in a straight cable, i.e. one with no crossings. In contrast, for simple unknotted loop structures, performing such an action would straighten the cable. We assume that any knots in the initial configuration are either overhand or figure 8 knots (Fig. 2), and that the space within each knot is large enough to fit both robot jaws inside. We assume that individual knots are distinct, such that no knots are embedded within other knots. We classify the initial cable configurations we consider into tiers of difficulty in Section V.

### B. Knot Definition

Let  $a, b \in [0, 1]$  be two indices on the cable's path, where  $a < b$  without loss of generality. We say that a knot exists at time  $t$  between indices  $a$  and  $b$  if grasping and pulling apart the points  $c_t(a)$  and  $c_t(b)$  with sufficient force does not straighten the cable segment (i.e., by removing all crossings) between  $a$  and  $b$ . We let  $k(a, b; c_t) = 1$  if a knot exists between  $a$  and  $b$  on cable  $c_t$ . For a particular cable configuration  $c_t$ , we can represent its knot structure by the minimal set of intervals

$$\mathcal{K}(c_t) = \{(a_1, b_1), \dots, (a_n, b_n)\} \text{ containing a knot:}$$

$$\begin{aligned} \mathcal{K}(c_t) = \operatorname{argmin}_{\mathcal{K}(c_t) \in \bigcup_{k=0}^{\infty} ([0, 1] \times [0, 1])^k} \sum_{(a_i, b_i) \in \mathcal{K}(c_t)} |b_i - a_i| \\ \text{s.t. } k(a_i, b_i; c_t) = 1, \forall i \in \{1, \dots, |\mathcal{K}(c_t)|\}, \\ a_i < b_i \forall i; b_i \leq a_{i+1} \forall i, \\ k(\tilde{a}, \tilde{b}; c_t) = 0 \forall \tilde{a}, \tilde{b} \text{ s.t. } b_i \leq \tilde{a} \leq \tilde{b} \leq a_{i+1}, b_0 := 0. \end{aligned}$$

This definition finds the smallest interval surrounding each distinct knot in the cable. For a given cable configuration  $c_t$ , the number of distinct knots is the number of intervals in  $\mathcal{K}(c_t)$ . This notation is illustrated in Figure 3 for a single figure 8 knot on a cable.

### C. Task Objective

The goal of the cable untangling problem is to manipulate the cable into a configuration  $c_t$  such that  $\mathcal{K}(c_t) = \emptyset$ , indicating that no knots exist in the cable. We call such a configuration *untangled*. This is equivalent to the condition that  $k(0, 1; c_t) = 0$ , which implies that the cable can theoretically be grasped at the endpoints and pulled apart to remove all crossings. The cable may still have crossings in its ending configuration; however, these crossings cannot form any knots.

*1) Algorithm Inputs and Outputs:* The algorithm is provided with overhead RGBD images as input, which are received between manipulation actions. At each step in the untangling process, the algorithm outputs either an open-loop trajectory to execute with the bilateral robot or a termination signal. To output a termination signal on completion, the robot must detect that it has successfully untangled the cable. Because the RGBD sensor uses structured light, the workspace must be static during image capture.

*2) Termination Condition:* The untangling algorithm terminates successfully if it outputs a termination signal and has fully untangled the cable, such that  $\mathcal{K}(c_t) = \emptyset$ . The algorithm terminates in failure if it outputs a termination signal  $\tau_{\text{done}}$  when  $\mathcal{K}(c_t) \neq \emptyset$  or reaches a fixed timeout  $T_{\text{timeout}}$  without successfully completing the task.

*3) Performance Metrics:* We record the algorithm's success rate across three difficulty tiers of problem instances, as described in Section V. We also report the time taken for the algorithm to terminate in each trial, as well as the time taken to first reach an untangled state, which may differ from the total time if the algorithm takes additional time to identify that it has fully untangled the cable.

## IV. METHODS

SGTM (Figure 7) proceeds in iterations consisting of two phases. In the first phase, SGTM actively manipulates the cable to identify knots by pulling apart the endpoints, shaking the cable, and tracing along the cable physically. In the second phase, it identifies exposed knots and undoes them using a novel bilateral manipulation primitive that cages the knot and slides it apart. In this section we will describe 1) the novel gripper jaw design, which allows multiple grasping modes with a compact form factor and no additional moving parts; 2)



Fig. 4: **Pinch-Cage (PC) Jaws.** Left: a rendering of the design of the pinch-cage gripper jaws, which attach to the YuMi’s standard white gripper. Center: the cage grasp, in which the perpendicular “foot” segment of the attachment enables the cable to slide freely. Right: the pinch grasp, in which the cable is held tightly, preventing slack from slipping through.

the manipulation primitives used as sub-components in SGTM; 3) the perception systems used, and 4) how SGTM uses these components to untangle long cables.

#### A. Pinch-Cage (PC) Grippers

Long cables can limit the efficacy of pinch grasps, in which both grippers firmly grasp the cable, because such grasps are unable to efficiently manage slack between grippers. Prior work demonstrates the promise of sliding grippers along cables [18], and to this end we present a passive mechanical design that facilitates pinch-pinch, cage-cage, and cage-pinch grasps. The design, shown in Fig. 4, attaches to the YuMi robot’s standard parallel jaw grippers. Each jaw includes a perpendicular “foot” segment that facilitates both **caging grasps**, where the jaws open partially and the feet prevent the cable from slipping out, and **pinching grasps**, where the jaws firmly grasp the cable. Caging grasps enable the cable to freely slide inside the gripper, while pinch grasps impart more force and prevent slack from slipping through. These two modes of cable grasping enable a variety of manipulation primitives that are particularly suitable for long cables, without the need for sensing or closed-loop control.

#### B. Manipulation Primitives

SGTM employs 5 cable manipulation primitives which are facilitated by the PC jaws and use an analytic grasp planner which selects collision-free gripper poses such that the grippers avoid each other as well as neighboring cable segments.

1) *Reidemeister move:* To uncover the cable’s underlying knot structure  $\mathcal{K}(c_t)$ , the robot spreads it apart by pinching both of the endpoints,  $c_t(0)$  and  $c_t(1)$ , and pulling them 1.2 meters apart towards opposite ends of the workspace. This is an example of a Reidemeister move in knot theory, and in this case it serves to spread out slack to reveal knots.

2) *Cable Shaking:* The Reidemeister move requires both cable endpoints to be visible and graspable. When this condition does not hold, we leverage dynamic shaking actions, a popular manipulation primitive in deformable manipulation [4, 29]. In this paper, the robot performs shaking actions to uncover occluded endpoints and loosen knot structure. If no endpoints are identifiable, the robot computes a random point on the cable’s mask in the overhead RGBD image. Using the

pointcloud from the RGBD image, the robot identifies and pinches the 3D point corresponding to this point,  $c_{\text{shake}}$ . After pinching, the robot raises its arm 0.7 meters off the table and executes a shaking motion by rotating the wrist joint 3 times by 2 radians, at a frequency of 1.5Hz, with a radius of 0.1 meters. These parameters were determined empirically to maximize the disruption to the cable’s visual state. If exactly one endpoint is visible and graspable, and the previous move was not a shaking action, the robot pinches this endpoint,  $c_{\text{shake}} = c_t(0)$  or  $c_{\text{shake}} = c_t(1)$ .

We also make use of the shaking action as a recovery move in case of any failures that may occur in other primitives.

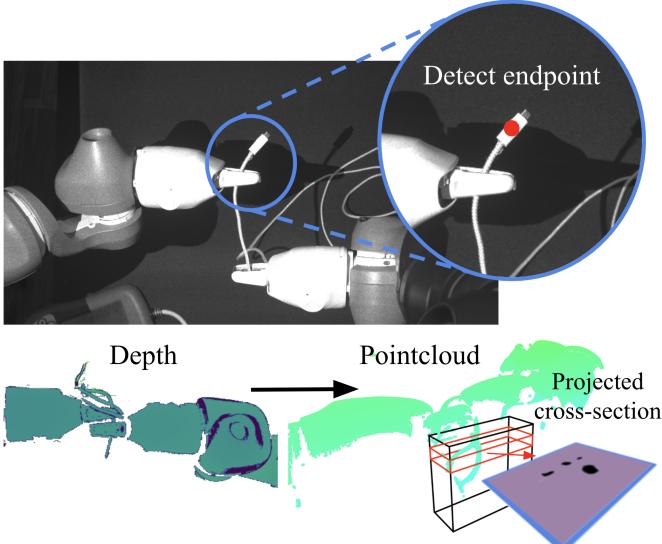
3) *Bimanual physical tracing:* During bimanual physical tracing, the robot slides along the cable until it visually identifies either an endpoint or a knot. In this motion, the robot first pinches two nearby points on the cable  $c_t(s_{\text{trace},l})$  and  $c_t(s_{\text{trace},r})$  with the left and right arms respectively. Without loss of generality, let the left arm be the *pulling arm* and the right arm be the *sliding arm*. The robot alternates between pinching and caging between the two arms to slide the cable through one gripper with the other. The robot first cages the cable with the sliding arm while the pulling arm pinches and pulls the cable through by 0.1 meter chunks as shown in Fig 5. After each chunk, the sliding arm pinches the cable and the pulling arm cages the cable, allowing it to move forward to meet the sliding arm without dropping or pushing the cable. The robot iterates this procedure until it either detects an endpoint or a knot approaching the sliding arm from overhead images. The detection algorithms are described in Section IV-C.

While sliding along the cable, the robot dangles the cable 0.45 meters above the work surface, causing loose loops to fall away while knots remain. SGTM uses this behavior to closely inspect the cable to actively locate knots in the presence of excess slack.

This motion begins at an endpoint, so that during execution the portion of cable that has been physically traced contains no knots, an invariant useful for managing slack in subsequent steps and indicating untangling success. If the robot slides uninterrupted from one endpoint to the other without detecting knots, it can verify that the entire cable contains no more knots. This is used as a termination condition by SGTM.

4) *Knot isolation:* Immediately after physical cable tracing terminates at a knot, we execute a slack management motion which separates the knot in-hand from other slack accumulated on the table; this primitive maximizes the likelihood of successful untangling in future steps. During this action, the arm holding the free endpoint places it toward one side of the workspace, then moves directly under the arm holding the knot. It rotates outwards and sideways, cradling the knot and connected slack above it and gently pushes it to the side. The arm holding the knot then deposits it onto the workspace.

5) *Cage-cage knot untying:* To untie individual knots, the robot attempts to cage two points inside the knot and then slowly pull its arms apart while wiggling the wrist joint 15 times by 0.2 radians. This wiggling helps reduce friction



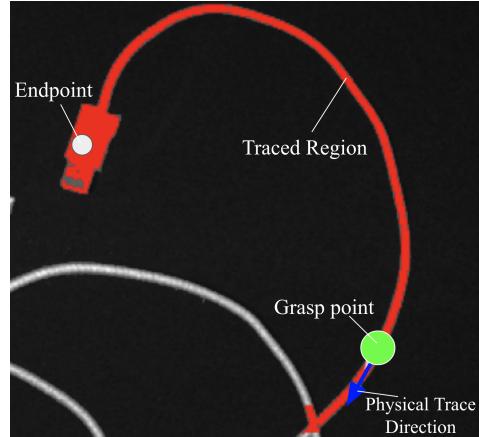
**Fig. 5: Physical Tracing Stop Condition:** *Top:* Physical cable tracing uses an endpoint detector to locate endpoints and terminate when any reach close to the gripper. *Bottom:* Physical cable tracing detects knots by horizontally slicing the pointcloud in front of the gripper (right), then analyzing the cross-sections for the number of connected components. In this case, the figure 8 knot contributes 4 connected components (black) to a cross section and reaches the threshold for number of points, so a knot is detected. If neither an endpoint nor a knot is detected, the robot continues sliding by beginning at the position shown in the bottom image, caging the left gripper and pinching the right gripper, and pulling the right gripper back as in the top image, tracing along the cable.

between segments of cable sliding past each other. Precisely, given a knot  $[a, b]$  in cable  $c_t$ , the robot grasps two graspable points  $c_t(s_{\text{knot},l})$  and  $c_t(s_{\text{knot},r})$  with the left and right arms respectively, such that  $s_{\text{knot},l}, s_{\text{knot},r} \in (a, b)$ . The double cage grasps allow the cable to freely slip through the fingers as the knot loosens, allowing the robot to untie knots in one action where the arms move as far apart as kinematically feasible, approximately 1 m apart. If the robot encounters sufficient resistance due to an endpoint or knot at either gripper, it will automatically stop and reset.

### C. Perception Systems

In contrast to some prior work which performs full state estimation of the cable [11], we rely on learned perception methods, since the significant lengths of slack introduce occlusions that make full state estimation difficult. In addition, to initiate some manipulation primitives in Section IV-B, SGTM also relies on additional manipulation primitives to expose key areas of the cable, such as knots, to predict task-relevant untangling keypoints.

**1) Endpoint detection:** In the Reidemeister move and bi-manual physical tracing primitives, SGTM relies on detecting endpoints in the image. We collect and manually label a dataset of 700 images in the workspace of the cable either on the manipulation surface or in the gripper jaws during physical tracing. We train a fully-convolutional network [10] based on a ResNet 50 backbone to output a Gaussian heatmap over each endpoint as in [3, 20, 22].



**Fig. 6: Visual Cable Tracing:** Beginning from the detected endpoint keypoint (white), the algorithm traces the cable until it detects that it has reached a crossing, after which it backtracks to return the grasp point (green) and slide direction (blue) for subsequent physical cable tracing.

**2) Visual cable tracing:** SGTM uses *visual cable tracing* to identify good starting points for *physical* tracing (Figure 6). Given an RGB image of a segmented cable and an endpoint pixel location, algorithmic cable tracing uses a modified breadth-first search (BFS) to follow the cable mask in the image and stop at the first self-crossing. It detects crossings by monitoring the frontier of added pixels at each BFS iteration. If the bounding box tightly surrounding the frontier has side length over 12px, the algorithm terminates, as the search is now bleeding across different cable segments. After termination, the algorithm backtracks by 100px and returns a pixel location along the cable which is a safe distance from the crossing to grasp. SGTM uses this keypoint to expedite bimanual physical cable tracing by avoiding physically tracing segments from the endpoint prior to any crossings.

**3) Physical tracing stopping condition:** During physical cable tracing, SGTM pulls the cable through the sliding gripper in segments of 0.1 meters. After every segment, the stopping condition categorizes the next segment of cable near the sliding gripper as a knot, endpoint, or straight cable. After each segment, the robot takes an overhead image and evaluates the endpoint detector (Section IV-C1) to check if any endpoints are within a  $0.1 \times 0.1 \times 0.1m$  cube around the sliding arm grippers by using the depth values at each predicted endpoint. In order to suppress false positives, we ignore endpoint detections when over 0.4m of the cable remains to be traced. If an endpoint is found after this range, the stopping condition returns *ENDPOINT*.

If an endpoint is not found, the robot analyzes the point-cloud surrounding the gripper of the sliding arm to categorize whether the next segment of cable is a knot or a straight cable. The robot performs a  $3 \times 13 \times 6$  cm volume crop in front of and beneath the sliding gripper to capture the next segment of cable. This volume is separated into 1 cm thick horizontally-sliced cross sections, which are each analyzed for the number of connected components in a re-projected depth image (see Fig. 5). If any cross section contains multiple connected components, this could either indicate a loop that has not been

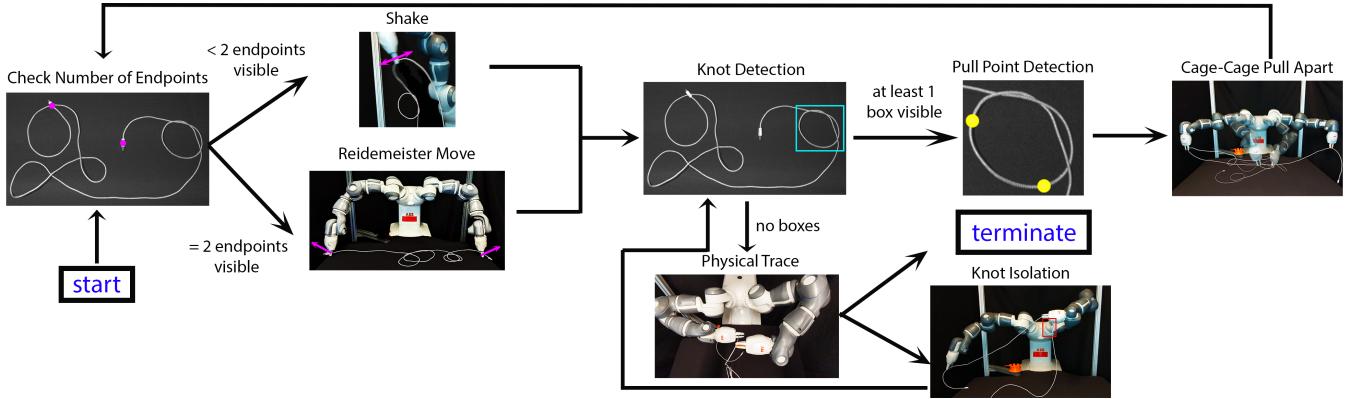


Fig. 7: **Sliding and Grasping for Tangle Manipulation (SGTM)**: SGTM begins by recognizing endpoints. If both are visible, it proceeds with a Reidemeister move. If not, it shakes. Afterwards, knot detection is performed after which it untangles if a knot is visible. If not, it performs physical tracing to ensure no knots remain. If a knot is found while tracing, the knot is isolated and placed in the workspace, and the algorithm returns to the knot detection step.

undone by gravity during physical tracing or a knot. If any cross section contains multiple connected components and the number of points in the volume crop is at least 1000 points, the stopping condition returns *KNOT*. If no cross sections contain multiple connected components, a knot could still exist if it is very tightly zipped together or self-occluded. Thus, if the volume crop contains at least 2000 points, the stopping condition also returns *KNOT*. If none of these conditions are satisfied, the stopping condition returns *STRAIGHT*.

4) *Knot detection*: We train an object detection network, based on the Mask R-CNN architecture and implemented using the Detectron2 codebase [26], to detect and classify figure 8 and overhand knots in cable images taken by the overhead PhoXi camera. The network is initialized with weights from a ResNet-50 FPN backbone pretrained on the COCO dataset [8] and trained on a dataset of 312 images with manually annotated knots in images containing both knots and loops.

5) *Pull point detection*: If the knot detection step locates a knot bounding box, it outputs a crop of the knot, which is passed through a two-stage cascading network. Both independently-trained stages use a Resnet34 backbone followed by a sigmoid activation to predict two pull points for performing a cage-cage knot untying action. The first network outputs a heatmap for both potential pull points. Of these, we select the location of the highest heatmap value, projected onto the nearest cable segment. The second network, trained separately, takes in the same cropped image with a fourth channel encoding a heatmap centered around the already-chosen first point, and it outputs a heatmap from which the second point is chosen the same way, using an argmax. This process is visualized in Figure 8.

We use the two-network cascading design because with a single network, we find that pick point multimodality often causes heatmap outputs to bleed together significantly, making it difficult to choose two distinct points. Conditional action prediction has also been previously studied in deformable manipulation by Wu et al. [25] for cable and fabric manipulation.

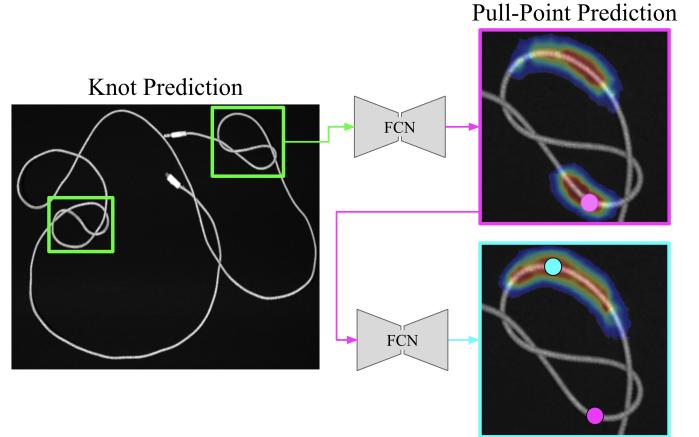


Fig. 8: **Learned Pull Point Prediction**: Left: predictions from the knot detection method on an image containing two figure 8 knots. Right: visualization of the cascading pull point detector. The top network outputs a heatmap over potential grasp points, and we select the point at which the heatmap has the highest value. The bottom network is conditioned on the first selected pull point (magenta) to predict a heatmap for the second pull point (blue).

#### D. Sliding and Grasping for Tangle Manipulation (SGTM)

We now describe the Sliding and Grasping for Tangle Manipulation (SGTM) algorithm, which combines the manipulation primitives and perception subsystems from Sections IV-B and IV-C to untangle long cables while managing cable slack. An overview of the algorithm is displayed in Fig. 7. During each iteration, SGTM alternates between two phases: physical manipulation to increase knot visibility (*Active Knot Perception*) and predicting keypoints for a) the cage-cage knot untying maneuver to untie knots once they are detected or b) physically checking if the cable is untangled when no knots are detected (*Knot Untangling/Physical Tracing*).

- 1) *Active Knot Perception*: During execution, at each iteration, SGTM begins by detecting visible endpoints. If both are detected, it executes a Reidemeister move, and if none are detected it executes a shake action from the centroid of the cable cluster. If a shake action was previously performed and no endpoints are visible still, the shake action is executed again from a random location on the cable. All three types of moves serve to spread out the

cable and increase the chance of perceiving knots.

- 2) *Knot Untangling/Physical Tracing*: Next, if knot bounding boxes are detected, the robot executes a cage-cage pull apart action to attempt to untie the knot closest to an endpoint (found using the technique in IV-C2) based on grasp keypoints the perception system outputs. If SGTM does not detect knots, the robot initiates the physical tracing stage to confirm that the cable is untangled, either ending in termination or by detecting a knot and isolating it on the table for further untangling.

## V. PHYSICAL EXPERIMENTS

We evaluate SGTM on a set of physical cable untangling experiments using the bilateral ABB YuMi robot. The experiments evaluate whether SGTM can untangle different classes of initial configurations and terminate when it has completely untangled the cable.

### A. Difficulty Tiers and Performance Metrics

We consider several methods for initializing the cable state, with examples shown in Fig. 9. These are categorized into the following tiers of difficulty:

- **Tier 0**: Cable has no knots, but the slack is piled randomly within the workspace by holding both endpoints 1.5 meters above the workspace and dropping them. In this tier, the robot must successfully verify that the cable contains no knots before terminating. We report the number of trials that successfully detect that the cable is untangled (Untangled Detection Rate), average number of manipulation actions (Avg. Number of Actions), and average time to detect that the cable is untangled (Avg. Untangled Detection Time).
- **Tier 1**: Cable has a single overhand or figure 8 knot that is loose (12-14 cm diameter) or tight (6-8 cm diameter), located close to an endpoint (side), 0.75 meters from the endpoint (mid-center), or 1.5 meters from an endpoint (center). It is arranged with the knot isolated from the rest of the cable slack, and the slack is randomized by lifting both endpoints as high above the workspace as possible without lifting the knot itself from the workspace and dropped. We report the number of trials that successfully untangled the knot (Untangling Success Rate), the number of trials that both untangled the knot and detected that the knot was untangled (Untangled Detection Rate), the average time taken to untangle the knot (Avg. Time to Untangle), and the average time taken to successfully identify that the cable was untangled (Avg. Untangled Detection Time).
- **Tier 2**: Cable has two overhand or figure 8 knots (including mixed types) that are loose or tight, in series, located close to each other ( $< 0.75$  m apart) or far from each other ( $> 1.5$  m apart). Similar to tier 1, both knots are isolated from the rest of the cable slack and the endpoints are lifted as high above the workspace as possible without lifting either knot from the workspace and dropped. We record the number of trials that successfully untangle

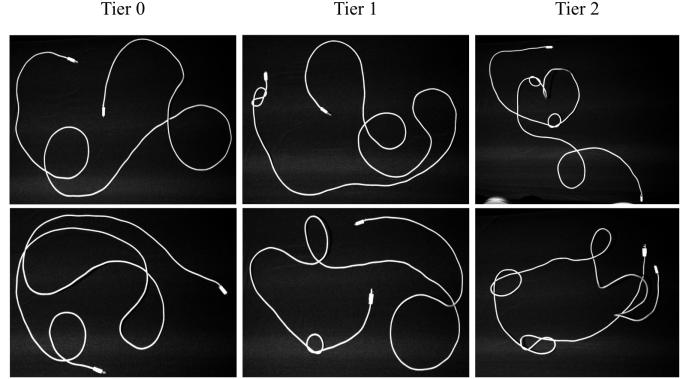


Fig. 9: **Cable Configurations**: Shown here are example configurations of the cable across the 3 tiers we run experiments on.

Untangled Detection Rate	8/12
Avg. Number of Actions	4.33
Avg. Untangled Detection Time (s)	266
Failures	A (1), B (1), C (2), D (0), E (0)

TABLE I: **Tier 0 results**: In 12 tier 0 experiments, SGTM must detect that a cable with no knots is untangled. We observe that SGTM is able to successfully terminate in 8/12 cases using an average of 4.33 manipulation primitive actions. See Section V-D for analysis of the failure modes A-E.

a single knot (Untangling 1 Success Rate), number of trials that successfully untangle both knots (Untangling 2 Success Rate), the number of trials that untangle both knots and detect that it is untangled (Untangled Detection Rate), average number of manipulation actions (Avg. Number of Actions), average time to untangle the first knot (Avg. Time to Untangle 1), average time to fully untangle the cable (Avg. Time to Untangle 2), and the average time to untangle both knots and detect that the cable is untangled (Avg. Untangled Detection Time).

For each tier 1 or 2 experiment, we record the average time until the cable has been successfully untangled. Because the robot may successfully untangle the cable but not successfully recognize that the cable is untangled (Section V-D), we separately record the average time to successfully detect the cable is untangled (when this is the case). We cap each experiment to 15 minutes in tiers 0 and 1 and to 20 minutes in tier 2. This is because tier 2 configurations are much more difficult and require more time to untangle. We analyze the observed failure modes of the algorithm in Section V-D.

### B. Experimental Setup

The workspace contains a bimanual ABB YuMi robot with two PC grippers. The manipulation surface is black and foam-padded to avoid end effector damage during any workspace collisions. The cable is a light-gray, braided 2.7 m micro-USB to USB cable that can be segmented from the manipulation surface via color thresholding. The workspace has an overhead PhotoNeo Phoxi Camera that captures depth and grayscale images of resolution 732 x 1142px.

### C. Results

In tier 0, SGTM successfully detects that 8/12 cases are untangled, taking an average of 4.33 manipulation primitive

	<b>Loose</b>	<b>Dense</b>
Untangling Success Rate	4/6	4/6
Untangled Detection Rate	1/4	2/4
Avg. Number of Actions	7.17	7.5
Avg. Time to Untangle (s)	154	139
Avg. Untangled Detection Time (s)	270	527
Failures	A (1), B (0), C (2), D (0), E (0), F (2)	A (1), B (0), C (1), D (1), E (0), F (1)

TABLE II: **Tier 1 results:** In 12 experiments, the cable starts off with a single knot in its initial configuration. SGTM is able to successfully untangle the cable 4/6 times in both the loose and dense cases of this problem. However, when the robot untangles the cable, it often fails to detect that it has successfully untangled the cable. This occurs because the robot forms loops that are not knots during execution, which are mistaken for knots during physical tracing (Failure Mode C).

	<b>Loose</b>	<b>Dense</b>
Untangling 1 Success Rate	5/6	5/6
Untangling 2 Success Rate	3/6	3/6
Untangled Detection Rate	0/3	1/3
Avg. Number of Actions	9.83	10.17
Avg. Time to Untangle 1 (s)	71	112
Avg. Time to Untangle 2 (s)	475	670
Avg. Untangled Detection Time (s)	N/A	1079
Failures	A (0), B (1), C (1), D (0), E (1), F (3)	A (1), B (1), C (1), D (1), E (1), F (0)

TABLE III: **Tier 2 results:** In this tier of 12 experiments for dense and loose knots, the cable starts off with two knots in its initial configuration. The robot is able to successfully untangle at least one of the knots 5/6 times in both the loose and dense versions of this tier. In 3/6 of both the loose and dense cases, SGTM successfully untangles both knots. However, similar to Tier 1, many loops and sometimes new knots are formed during untangling, which lead to timing out during a trial due to failure of untangled detection.

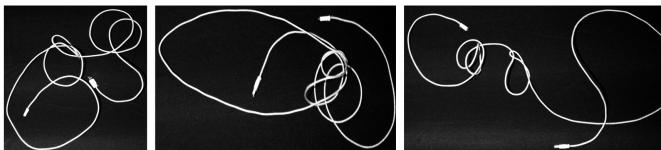


Fig. 10: **Failure Mode A Examples:** These are examples of complex knots or unseen knots that form while untangling, but fall out of the distribution of training samples for the knot detector, causing a failed knot detection.

actions and 266 seconds (Table I). In the cases where SGTM does not terminate successfully, the robot either moves the cable off the manipulation surface, creates a new knot during execution, or falsely detects loops as knots during physical tracing.

In tier 1, SGTM successfully untangles the cable in 4/6 trials in both the loose and the dense categories (Table II). The untangled detection rate is 1/4 for loose configurations and 2/4 for dense configurations, and the most common errors are detection of loops as knots during physical tracing and system errors with the YuMi robot.

In tier 2, SGTM is able to untangle a single knot in 5/6 trials in both the loose and dense cases (Table III). In both cases, the robot untangles both knots 3/6 times. The robot often has trouble detecting that it has successfully untangled the cable in this case, due to an accumulation of twists during untangling that lead to loops that look like knots during physical tracing.

#### D. Failure Modes

In experiments for SGTM, we observe the following failure modes:

- (A) Unseen or complex knots form while untangling and are not detected by Mask R-CNN for knot bounding box detection.
  - (B) The cable falls out of the reachable workspace.
  - (C) Repeated false positive knot detections during physical tracing result in hitting the time limit.
  - (D) Missed knots during physical cable tracing cause knots to tighten.
  - (E) Knot untangled, but not enough time to check termination.
  - (F) YuMi robot system or reset errors.
- (A) During execution, SGTM sometimes introduces new knots by pulling endpoints through existing loops on top of them as shown in Figure 10. Though unlikely, these occurrences are often difficult to recover from since knots formed in this way are visually distinct from the training set of knot bounding box detection model. This failure happens particularly often after no endpoints are detected and the robot must shake from a random cable location, after which endpoints are buried underneath slack.

(B) Because we shake near the edge of the table, sometimes the motion causes part of the cable to drop off the side, which can cause the rest of the cable to follow after it is released.

(C) One particularly confusing case for knot detection during physical tracing is tightly twisted loops which do not fall away due to gravity. Sometimes this happens multiple times in a rollout, wasting time and eventually causing the time limit  $T_{timeout}$  to be reached.

(D) If SGTM fails to detect a knot during physical cable tracing, the sliding actions can tighten the knot until it is too dense to fit grippers inside, resulting in failure.

(E) Sometimes, SGTM is able to recover from situations like failure mode (A), but this can happen very late in the experiment, resulting in not enough time to check termination.

(F) YuMi errors result from 1) the robot getting stuck at a singularity from which it is unable to reset or 2) the robot attempting to reach a point just outside of its reachable workspace and producing errors from which it cannot recover.

We observe that the physical tracing termination condition is low-recall but high-precision; that is, if it claims that the cable is untangled, it most likely is (only in one case out of 36 experiments did it output a false positive). However, while false positives terminate by presenting a tangled cable as untangled, a false negative results in the robot continuing to try to untangle an untangled cable. The latter is less severe, because the robot often eventually discovers that the cable is untangled; thus, we prefer SGTM to err on the side of false negatives.

## VI. DISCUSSION AND FUTURE WORK

For untangling a single long (up to 3m) cable, this work presents a formal problem definition, a novel jaw design, novel

perception-based primitives, and Sliding and Grasping for Tangle Manipulation (SGTM), an algorithm for untangling long cables. SGTM introduces 3 novel manipulation primitives for cable manipulation: shaking, physical tracing, and caged pull apart. These primitives aid in easing perception and managing slack in long cables. SGTM also introduces perception systems that guide the usage of these primitives. Experiments show that SGTM can untangle long cables with a 58.3% success rate overall across tiers containing one to two knots.

Future work will investigate refining these primitives to improve performance, generalizing this method to less structured starting configurations of cables and knots, as well as multiple cables. Further, active perception for cable manipulation is of interest, for example investigating ways of interrogating the presence of knots with finer precision via high frame rate videos and gripper perturbations. In addition, replacing the manually tuned components of SGTM, for example the knot detection threshold, with more data-driven methods and self supervised data collection may increase robustness and generalization.

## ACKNOWLEDGMENTS

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkley AI Research (BAIR) Lab, the CITRIS “People and Robots” (CPAR) Initiative, and the RealTime Intelligent Secure Execution (RISE) Lab. The authors were supported in part by donations from Toyota Research Institute and by equipment grants from PhotoNeo, Nvidia, and Intuitive Surgical.

## REFERENCES

- [1] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conf. on Robot Learning (CoRL)*, 2018.
- [2] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minho Hwang, Ryan Hoque, Joseph E Gonzalez, Nawid Jamali, et al. Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.
- [3] Jennifer Grannen, Priya Sundaresan, Brijen Thananjeyan, Jeffrey Ichnowski, Ashwin Balakrishna, Minho Hwang, Vainavi Viswanath, Michael Laskey, Joseph E Gonzalez, and Ken Goldberg. Untangling dense knots by learning task-relevant keypoints. In *Conf. on Robot Learning (CoRL)*, 2020.
- [4] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022.
- [5] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Visuospatial foresight for multi-step, multi-task fabric manipulation. In *Proc. Robotics: Science and Systems (RSS)*, 2020.
- [6] Thomas Kollar, Michael Laskey, Kevin Stone, Brijen Thananjeyan, and Mark Tjersland. Simmet: Enabling robust unknown object manipulation from pure synthetic data via stereo. In *Conference on Robot Learning*, pages 938–948. PMLR, 2022.
- [7] Robert Lee, Daniel Ward, Akansel Cosgun, Vibhavari Dasagi, Peter Corke, and Jurgen Leitner. Learning arbitrary-goal fabric folding with one hour of real robot experience. In *Conf. on Robot Learning (CoRL)*, 2020.
- [8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [9] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022.
- [10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [11] Wen Hao Lui and Ashutosh Saxena. Tangled: Learning to untangle ropes with RGB-D perception. In *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 837–844. IEEE, 2013.
- [12] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conf. on Robot Learning (CoRL)*, 2018.
- [13] Hermann Mayer, Faustino Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jürgen Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Advanced Robotics*, 22(13-14):1521–1537, 2008.
- [14] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2146–2153. IEEE, 2017.
- [15] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The Int. Journal of Robotics Research*, 37(7):688–716, 2018.
- [16] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, et al. Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.
- [17] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.
- [18] Yu She, Shaowiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 40(12-14):1385–1401, 2021.
- [19] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Michael Laskey, Kevin Stone, Joseph E Gonzalez, and Ken Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2020.
- [20] Priya Sundaresan, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Jeffrey Ichnowski, Ellen Novoseller, Minho Hwang, Michael Laskey, Joseph E Gonzalez, and Ken Goldberg. Untangling dense non-planar knots by learning manipulation features and recovery policies. *Proc. Robotics: Science and Systems (RSS)*, 2021.
- [21] Jur Van Den Berg, Stephen Miller, Daniel Duckworth, Humphrey Hu, Andrew Wan, Xiao-Yu Fu, Ken Goldberg, and Pieter Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *2010 IEEE International Conference on Robotics and Automation*, pages 2074–2081. IEEE, 2010.
- [22] Vainavi Viswanath, Jennifer Grannen, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Ellen Novoseller, Jeffrey Ichnowski, Michael Laskey, Joseph E Gonzalez, and Ken Goldberg. Disentangling dense multi-cable knots. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021.
- [23] Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. *arXiv preprint arXiv:1905.04411*, 2019.
- [24] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.
- [25] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. *arXiv preprint arXiv:1910.13439*, 2019.
- [26] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [27] Yuji Yamakawa, Akio Namiki, Masatoshi Ishikawa, and Makoto Shimojo. One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors. In *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 703–708. IEEE, 2007.

- [28] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. In *Conf. on Robot Learning (CoRL)*, 2020.
- [29] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.