# Disentangling Dense Multi-Cable Knots

Vainavi Viswanath*[1], Jennifer Grannen*[1], Priya Sundaresan*[1], Brijen Thananjeyan[1], Ashwin Balakrishna[1], Ellen Novoseller[1], Jeffrey Ichnowski[1], Michael Laskey[2], Joseph E. Gonzalez[1], Ken Goldberg[1]

***Abstract*— Disentangling two or more cables requires many steps to remove crossings between and within cables. We formalize the problem of multiple cable disentangling and present an iterative, graph-based algorithm, Iterative Reduction Of Non-planar Multiple cAble kNots (IRON-MAN), that outputs moves to remove crossings from the scene. We instantiate it with a learned perception system, inspired by prior work in single-cable untying, to disentangle two cable twists, three cable braids, and knots of two or three cables, such as the overhand, square, carrick bend, sheet bend, crown, and fisherman's knots from image input. IRON-MAN keeps track of task-relevant keypoints corresponding to target cable endpoints and crossings and iteratively disentangles the cables by identifying crossings that are critical to knot structure and undoing them. Using a da Vinci surgical robot, we experimentally evaluate the effectiveness of IRON-MAN on the task of untangling a class of multiple cable knots present in the training data, as well as generalizing to novel classes of multiple cable knots involving two to three cables. Results suggest that IRON-MAN is effective in disentangling knots involving up to three cables with 80.5% success, with generalization to knots that are never seen during training on cables that are either distinct or uniform in color.**

## I. Introduction

Knots composed of multiple ropes and cables are found in many environments, including electronic cords in homes, offices, and concert stages, electrical wiring in warehouses, ropes in sailboats and ships, and cables in manufacturing settings [14, 18, 24, 25]. Furthermore, rope and cable disentangling can be critical in life-saving systems for search-and-rescue operations and disaster response [9, 15], where disentangling knots and tangles is crucial for task success. Designing robust systems for disentangling 1D deformable objects, which we refer to as "cables," is challenging. In this paper, we propose methods based on the graphical structure of knots to address this challenge.

This work considers cable *disentangling*, that is, separating 2 or more cables that are knotted or twisted together. While prior work has studied the problem of untying knots in a single cable [6, 13], disentangling multiple cables introduces new challenges. First, multi-cable knots can contain more crossings than single-cable knots, especially in cases where the cable is stiff and has a large turning radius: multiple cables can intertwine with each other with minimal turning, while single cables would require a smaller turning radius to achieve the same number of intersections. Second, perception is complicated by the presence of additional cable endpoints,

[1]AUTOLAB at the University of California, Berkeley
[2]Toyota Research Institute
*equal contribution

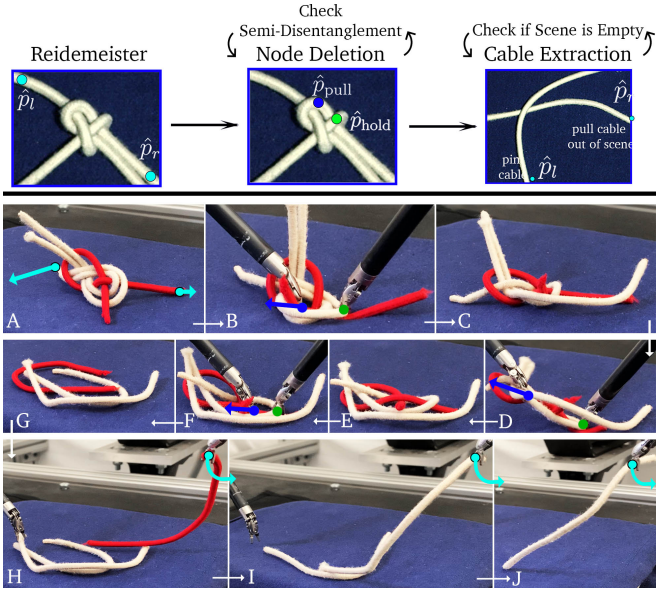


Fig. 1: **Overview of IRON-MAN**: IRON-MAN (Iterative Reduction Of Non-planar Multiple cAble kNots) is an algorithm for disentanglement of several knotted cables by sequentially employing Reidemeister (straightening), Node Deletion (loosening) and Cable Extraction (removal) moves. We present a sequence of moves planned by IRON-MAN on a three-cable Carrick Bend knot. Following an initial Reidemeister move (A) which pulls opposing cable endpoints apart, IRON-MAN takes several Node Deletion moves (B-C, D-E, F-G) to reduce inter and intra-cable crossings. Finally, we take three Cable Extraction moves (H-J) to isolate and remove each cable.

higher-density configurations with little space between adjacent cable segments, and a greater potential for occlusion. Third, multiple cable systems have more cable endpoints to track, making it difficult to track untangling progress along specific cables. Fourth, the mechanics of cable manipulation are also more complicated with multiple cables, as crowding of cables can impede reachability.

Prior methods for single-cable untangling cannot be straightforwardly adapted to the multi-cable setting due to differences in cable configurations, untangling actions, and perception [6]. Grannen *et al.* [6] present a geometric algorithm for untangling a single cable that iteratively undoes each cable crossing, starting from one endpoint of the cable and working toward the other. A naive approach to multi-cable untangling could consider each of the $n$ cables sequentially, and untie knots within each one by one. However, multi-cable systems have many endpoints, which cannot always be easily mapped to specific cables. Furthermore, the presence of inter-cable crossings across cables in multi-cable knots further complicates planning of disentangling actions compared to the single-cable setting.

We propose Iterative Reduction Of Non-planar Multiple

cAble kNots (IRON-MAN), an algorithm for disentangling multiple cables given a graphical representation of the knot structure. IRON-MAN distills full configuration state information by defining a disentangling hierarchy over cable crossings to generate bilateral disentangling actions on the cables. IRON-MAN then prioritizes disentangling crossings integral to the knot structure. To implement IRON-MAN from visual observations, we employ keypoint regression methods [6] to learn pull and hold keypoints. We plan manipulation actions over the learned keypoints with two manipulation primitives from knot theory and prior work [6, 13] and execute these actions with a learning-based controller from Sundaresan *et al.* [22].

This paper contributes: (1) a formulation of the multiple cable disentangling problem; (2) Iterative Reduction Of Non-planar Multiple cAble kNots (IRON-MAN), a novel geometric algorithm for disentangling multiple cables, (3) an instantiation of Iterative Reduction Of Non-planar Multiple cAble kNots (IRON-MAN) from image input by extending the perception-driven single-cable planner from Grannen *et al.* [6]; and (4) physical cable disentangling experiments on multi-cable knots of 3 difficulty tiers consisting of combinations of different types of multi-cable knots, twists, and braids. Experiments suggest that the physical implementation of IRON-MAN can completely disentangle all cables in scenes containing up to three cables with 80.5% success.

## II. Background and Related Work

Deformable manipulation has gained significant traction in the robotics research community in recent years [1, 7, 18]. Deformable manipulation poses a number of challenges, as deformable objects can have complex dynamics, visually uniform appearances, and a high potential for self-occlusion. These properties introduce manipulation challenges that are exacerbated with multiple deformable objects, as in the task of disentangling complexly knotted cables. Prior work has focused on developing long-horizon perception-driven planners for deformable manipulation, but tend to focus on handling a single object instance such as a cable, a bag, or a piece of fabric. In contrast, this work embeds awareness of both within and cross-instance geometry into a planner for disentanglement of multiple cables.

### A. Deformable Object Manipulation

A standard approach in robot manipulation of nonrigid objects relies on inference of plannable visual state representations. One such method explicitly performs partial or full state estimation of a deformable object. Yan *et al.* [26] and Lui *et al.* [13] infer rope representations as sparse point sets from learned and analytical methods, respectively. Similarly, Chi *et al.* [2] demonstrate cloth and rope tracking based on template shape registration. To generalize single-instance perception models to multi-instance scenes, prior approaches rely on instance segmentation, which is difficult to achieve in the setting of homogeneous, tangled cables. Florence *et al.* [4] propose mapping object images to a dense pixel-wise descriptor embedding with which to recover object pose in both single and cluttered multi-instance scenes for semantic grasping. Dense descriptors have proven effective in rope knot-tying [23] and cloth folding and smoothing [5], but global descriptors lack robustness to severe deformation and occlusion as in the case of multiple overlapping cables.

In contrast, other approaches circumvent state estimation by performing end-to-end visuomotor learning for goal-conditioned tasks. These include rope shape-matching and knot-tying by learning dynamics models [16, 17]; cable vaulting by behavioral cloning [27]; cloth smoothing and folding from video prediction models [3, 8], latent dynamics models [12], reinforcement learning [11], and imitation learning [20, 21]; and bag manipulation by inferring spatial displacements [19]. While general, these algorithms do not leverage the geometric structure specific to the cable manipulation problem, which makes them difficult to apply to highly complex tasks such as cable disentangling, in which fine-grained perception and manipulation is critical for success.

### B. Cable Untangling Methods

Prior work has studied the task of single-cable untangling from both loose [13] and dense [6, 22] initial configurations, where dense configurations lack space between crossings. Lui *et al.* [13] propose modeling a cable configuration via a graphical abstraction representing cable crossings and endpoints, and approximate this model from RGB-D input through analytical feature-engineering. This method assumes reliable segmentation of crossings to construct the graph, and as a result does not readily adapt to untangling dense, non-planar knots within or across cables. Grannen *et al.* [6] define a single-cable untangling algorithm, HULK (Hierarchical Untangling from Learned Keypoints), which rather than explicitly reconstructing a cable graph as in [13], learns to predict untangling actions from RGB image observations, given examples of actions taken by a graph-based algorithmic supervisor. In particular, HULK learns to infer actions from predicted pin and pull keypoints corresponding to the first under-crossing in a cable, sensed relative to the *rightmost endpoint*. However, while HULK addresses the semi-planar cable knot untangling problem, it does not account for disentangling, or separation of multiple cables in non-planar knots. HULK also does not apply out-of-the-box to the multi-cable setting due to ambiguity in selecting knot-loosening actions when there are many cable endpoints from which to trace under-crossings. In this work, we extend the graphical abstraction from [13] to accommodate inter-cable crossings and propose IRON-MAN. IRON-MAN extends HULK to resolve ambiguity in action selection and termination, thus reasoning about the many cable endpoints and developing a scheme for systematically disentangling multiple cables.

## III. Task Formulation

A bilateral robot aims to disentangle a knotted configuration of $n$ cables, where $n > 1$ in the initial configuration, by removing one crossing at a time by learning visual features to plan hold and pull actions given an input RGB image. The objective is to reach a fully disentangled state with

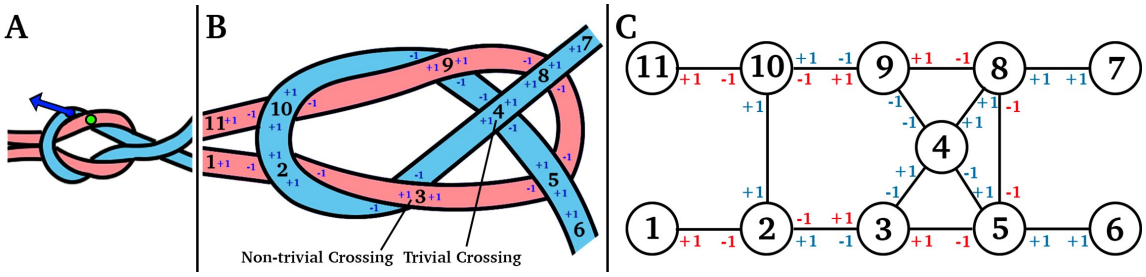


Fig. 2: **Graph Representation**: Provided a dense initial square knot (A), we take a Node Deletion move specified by hold (green) and pull (dark blue) keypoints, yielding a looser configuration shown in (B). We use a graphical abstraction to model the state of intertwined cables, extending previous work on modelling single cable configurations [6, 13, 22]. In this graph, endpoints and intra/inter-cable crossings constitute nodes, and edges denote over (+) and under (-) crossings. shown in (C). We prioritize removing crossings that are *non-trivial*, such as (3) rather than *trivial* ones such as (4), which can be easily undone by a Reidemeister move without substantially loosening the configuration.

no crossings (defined below). In this section, we define the workspace and notation for cable untangling (Section III-A), present a graphical abstraction to represent the state of multiple cables in the workspace (Section III-B), discuss the set of actions which can be used to manipulate the cables (Section III-C), and finally present the objective of the multiple cable disentangling problem (Section III-D).

### A. Workspace Definition

We define a Cartesian $(x,y,z)$ coordinate frame for the workspace and assume that the workspace contains a bilateral robot. Without loss of generality, we assume that the manipulation surface lies in the $xy$-plane. For planning purposes, we define three points, $w_l$, $w_r$, $w_c$, respectively located on the left bound, right bound, and center of the manipulation surface. This workspace setup resembles that in Grannen *et al.* [6], but differs from prior work in that rather than containing only a single cable, the scene now contains $n$ cables that can be knotted or twisted together. With these $n$ cables occupying the same workspace as a single cable in prior work, we now require improved cable slack management in the workspace. Let $c_i$ denote the $i^{\text{th}}$ cable, where $i \in \{1,\ldots,n\}$, and let $\rho_i > 0$ denote the radius of cable $c_i$. We assume the existence of a termination area to the right of the manipulation workspace, into which cables are relocated as they become fully disentangled. The termination area is centered at the point $p_{\text{term}} \in \mathbb{R}^3$ and can be reached by the robot's grippers.

We define *intra-cable crossings* to be crossings that only involve a single cable, while *inter-cable crossings* are crossings involving at least two cables. The structure of crossings between cables is not directly observable, and must instead be inferred from RGB images. At time $t$, an image $I_t$ is the input to the algorithm to generate manipulation actions.

### B. Configuration Graph

Previous works define a graph structure to concisely represent the configuration space of single-cable knots [6, 13, 22], and introduce algorithms that operate on this compressed representation to perform untangling. We extend this representation to model $n$-cable knots. The graph contains vertices (also referred to as nodes) $v \in V$, which represent any cable endpoints and crossings in the structure, and edges $e \in E$, which represent cable segments without crossings between vertices defined as $e = (u,v)$ for $u,v \in V$. While the graph representations defined in prior work limit crossings to 2 or 3 cable segments [6, 22], in this work, each vertex $v$ that represents a crossing of $k$ segments has a degree of $2k$, since there are $2k$ cable segments extending from the crossing. The graph vertices do not distinguish between intra-cable and inter-cable crossings. All vertices have a degree of $2k$, except endpoints, which have a degree of one. Additionally, we annotate every (vertex, edge) pair with a label $X(v,e) \in \{-1,\ldots,-(k-1)\} \cup \{+1\}$, where $k$ is the number of segments in the crossing at vertex $v$, according the definition below:

$$X(v,e) = \begin{cases} +1 & \text{if } v \text{ is an endpoint or if } e \text{ crosses over all other edges at } v \\ -m & \text{if } e \text{ crosses under } m \text{ edges at } v \end{cases} \tag{III.1}$$

Intuitively, $X(v,e)$ indicates the depth of edge $e$ at the crossing represented by vertex $v$. Observe that this graph can have multiple edges, as illustrated in Figure 2, but no two pairs of contiguous edges will have the same two annotations.

### C. Action Space

We define the possible actions at time $t$ in terms of global workspace coordinates:

$$\mathbf{a}_{t,\text{right}} = (x_{t,\text{r}}, y_{t,\text{r}}, \theta_{t,r}, \Delta x_{t,\text{r}}, \Delta y_{t,\text{r}}, \mathbb{1}_{\text{grasp}})$$

$$\mathbf{a}_{t,\text{left}} = (x_{t,\text{l}}, y_{t,\text{l}}, \theta_{t,l}, \Delta x_{t,\text{l}}, \Delta y_{t,\text{l}}, \mathbb{1}_{\text{grasp}}).$$

With $\mathbb{1}_{\text{grasp}} = 1$, gripper $k \in \{r,l\}$ grasps the topmost cable—defined as the cable with the highest $z_{t,k}$ at $(x_{t,\text{k}}, y_{t,\text{k}})$. We use a top-down grasp with orientation $\theta_{t,k}$ about the $z$-axis and with a 30° approach angle relative to the vertical. Upon grasping, the jaw moves by $(\Delta x_{t,\text{k}}, \Delta y_{t,\text{k}})$ and releases its hold. With $\mathbb{1}_{\text{grasp}} = 0$, the arms execute the same motions with the gripper jaws remaining open throughout, preventing a secure grasp. This motion is crucial for implementing Cable Extraction moves, which grasp the terminated cable with one arm and "pin" a different cable endpoint, if available, with the other arm for retrieval of only the target cable. We use $\mathbb{1}_{\text{grasp}} = 0$ for an effect of soft pinning, allowing slack to slip through the jaws, in the case that only cable is left in the scene and predicted endpoints lie on the same cable. This primitive is discussed in detail in Section V-B.2. The actions $\mathbf{a}_{t,\text{right}}$ and $\mathbf{a}_{t,\text{left}}$ are executed simultaneously by both arms, and single-arm actions can be performed by letting the other arm's action be null. We will describe specialized motion

primitives defined in terms of this general action definition in Section IV-C and Section V-B.

We assume access to a transformation between pixel coordinates $(p_x, p_y)$ and global positions $(x, y, z)$. Because the perception systems operate in pixel space, we will present positions in terms of pixels, overloading the action notation with pixel coordinates instead of workspace positions.

### D. The Multiple Cable Disentangling Problem

The objective of the multiple cable disentangling problem is to remove all intra-cable and inter-cable crossings in the scene with a minimal number of actions while recognizing and sequentially removing fully untangled cables. In terms of the graphical representation of the knotted structure's state, the goal is to reach a configuration graph with two vertices per cable, one corresponding to each endpoint, where the two vertices belonging to each cable are connected to each other by an edge with positive annotations on both ends. At time $t$, the algorithm receives an image observation $I_t$ and outputs a linear, bilateral action $\mathbf{a}_t = (\mathbf{a}_{t,\text{right}}, \mathbf{a}_{t,\text{left}})$.

## IV. Preliminaries

### A. Assumptions

We make the following assumptions: 1) *cables distinguishable*: the cables are visually distinguishable from the background via color thresholding, but need not be distinguishable from each other; 2) *visible endpoints*: at least two endpoints are visible in the initial cable configuration; 3) *linear pull actions sufficient*: we assume all cables are within reachable limits of the robot, thus ensuring the robot can successfully perform grasping and pulling actions. Unlike Grannen *et al.* [6], who assume a semi-planar knot structure—i.e., at most two cable segments per crossing—we allow non-planar knots, where more than two cable segments can be involved in each intersection.

### B. Physical Disentangling System

In this section, we describe three methods used in prior work for single cable untangling: HULK, LOKI, and SPiDERMan. IRON-MAN modifies HULK to be well-defined for multiple cables and additionally adds new motion primitives. The planned grasps are executed using the grasp refinement steps from LOKI. IRON-MAN also modifies SPiDERMan to be defined for multiple cables.

*1) HULK—Hierarchical Untangling from Learned Keypoints:* HULK [6] senses four task-relevant keypoints in the scene that are used to plan motion primitives. Each keypoint is a pixel coordinate that corresponds to a semantically relevant point on the cable. For a single cable, HULK senses:

1) $\hat{p}_l$: The estimated pixel coordinate of the left endpoint.
2) $\hat{p}_r$: The estimated pixel coordinate of the right endpoint.
3) $\hat{p}_{\text{hold}}$: The estimated pixel coordinate of the topmost segment of cable at the first undercrossing $c$ from the rightmost endpoint.
4) $\hat{p}_{\text{pull}}$: The estimated pixel coordinate of the cable segment edge labeled $-i$ exiting the first undercrossing $c$ traced from the rightmost endpoint.

For each keypoint $\hat{p}$, HULK learns a mapping $f : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times1}$ which maps an RGB image to a heatmap centered at $\hat{p}$. HULK is trained from images, where, given a hand-specified keypoint annotation for an image, we compute a ground truth 2D Gaussian distribution centered at the annotated pixel and with $\sigma = 8$px as in [6, 22]. In Section V-A, we will redefine these keypoints analogously for the multi-cable setting.

*2) LOKI—Local Oriented Knot Inspection:* LOKI [22] is a low-level grasp planner that computes a robust grasp by refining a coarse grasp location input to center it on a cable and infer a grasp orientation that is orthogonal to the cable's path. These refinement prevent near-miss grasps. LOKI maps a local crop of an image in $\mathbb{R}^{200\times200}$ centered at one of the keypoints—$\hat{p}_r$, $\hat{p}_l$, $\hat{p}_{\text{pull}}$, and $\hat{p}_{\text{hold}}$—to 1) $\theta$: an angle about the $z$-axis for top-down grasp orientation; and 2) $(p_{\text{off},x}, p_{\text{off},y})$: a local offset in pixel space to recenter the keypoint along the cable width. For an input keypoint $\hat{p}$, we let $\tilde{p}$ denote its refined grasp location (in pixels) and $\tilde{\theta}$ (from $0^\circ$ to $180^\circ$ denote its grasp orientation.

*3) SPiDERMan—Sensing Progress in Dense Entanglements for Recovery Manipulation:* SPiDERMan [22] addresses four manipulation failures modes observed in Grannen *et al.* [6]. Here we implement SPiDERMan's recovery manipulations **Wedge Recovery** and **Re-posing (translation)** that are relevant to the task of disentangling multiple cables. **Wedge Recovery** detects when a gripper is wedged between cable segments. If the right gripper is wedged between cable segments, the right gripper brings the stuck cable to the center of the workspace $w_c$ and opens the gripper jaws. The left gripper pins the cable at the left endpoint while the right arm returns to its home position. The equivalent procedure is performed when the left gripper is wedged between cable segments with the roles of the left and right grippers reversed. When SPiDERMan detects that the cable mass is near workspace limits, it executes a **Re-posing move (translation)** to grasp the cable at the center of its mask and returns it to the center of the workspace $w_c$.

### C. Motion Primitives

We extend two motion primitives from Grannen *et al.* [6] to fully disentangle cables: Reidemeister moves and Node Deletion moves. To improve manipulation robustness, we use LOKI to center each grasp along the cable width with an improved grasping orientation.

*1) Reidemeister moves:* **Reidemeister moves** grasp the left endpoint at the corrected location $\tilde{p}_l$ with orientation $\tilde{\theta}_l$ and pull the cable to a predefined location $w_l$ at the left side of the workspace. The right gripper similarly grasps the right endpoint at $\tilde{p}_r$ with orientation $\tilde{\theta}_r$ and pulls the cable to a predefined location $w_r$ at the right side of the workspace. This move eliminates trivial crossings not involving a knot and disambiguates the configuration of the cable by spreading it apart. The actions are described as follows:

$$\mathbf{a}_{t,l} = (\tilde{p}_{l,x}, \tilde{p}_{l,y}, \tilde{\theta}_l, w_{x,l} - \tilde{p}_{l,x}, w_{y,l} - \tilde{p}_{l,y}, 1)$$
$$\mathbf{a}_{t,r} = (\tilde{p}_{r,x}, \tilde{p}_{r,y}, \tilde{\theta}_r, w_{x,r} - \tilde{p}_{r,x}, w_{y,r} - \tilde{p}_{r,y}, 1).$$

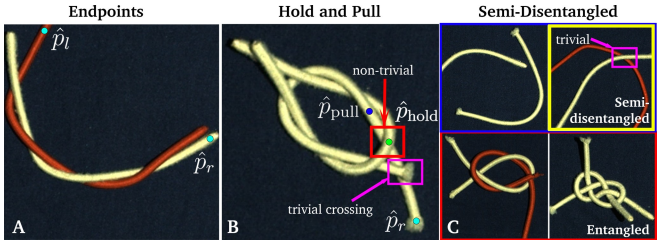


Fig. 3: **Physical Implementation of IRON-MAN**: Image A depicts the endpoints used for Reidemeister moves: $\hat{p}_r$ and $\hat{p}_l$ on cables $c_1$ and $c_2$, respectively. The left endpoint annotation corresponds not to the leftmost endpoint in the scene (on cable $c_1$), but rather to the leftmost endpoint of cable $c_2$ because $\hat{p}_r$ is on cable $c_1$. Image B depicts the hold and pull keypoints, $\hat{p}_{\text{hold}}$ and $\hat{p}_{\text{pull}}$, relative to the first non-trivial crossing from the right endpoint. The first crossing from the rightmost endpoint is a trivial crossing, and is skipped when traversing from the right endpoint to annotate a Node Deletion move. Image C presents semi-disentangled and entangled configurations. Although the image in the yellow highlighted box still contains a crossing, it is trivial and thus acceptable by our definition of semi-disentanglement.

*2) Node Deletion moves:* In a **Node Deletion move**, the right gripper grasps and holds a cable segment at $\tilde{p}_{\text{hold}}$ while the left gripper grasps a cable segment at $\tilde{p}_{\text{pull}}$ and pulls out the cable slack underneath $\tilde{p}_{\text{hold}}$. LOKI predicts each grasp rotation $\tilde{\theta}_{\text{hold}}$ and $\tilde{\theta}_{\text{pull}}$:

$$\mathbf{a}_{t,\text{hold}} = (\tilde{p}_{\text{hold},x}, \tilde{p}_{\text{hold},y}, \tilde{\theta}_{\text{hold}}, 0, 0, 1)$$
$$\mathbf{a}_{\text{pull}} = (\tilde{p}_{\text{pull},x}, \tilde{p}_{\text{pull},y}, \tilde{\theta}_{\text{pull}}, \tilde{p}_{x,\text{pull}} - \tilde{p}_{\text{hold},x}, \tilde{p}_{\text{pull},y} - \tilde{p}_{\text{hold},y}, 1).$$

After HULK identifies the first undercrossing traced from the rightmost endpoint and LOKI refines the grasps, a Node Deletion move attempts to pull out part of a cable segment underneath the topmost segment to eliminate undercrossings.

## V. Methods

To the best of our knowledge, prior work has only considered single-cable untangling, and the resulting algorithms cannot be directly employed for disentangling multiple cables. We present a novel algorithm, IRON-MAN, for disentangling multiple cables using the graph representation of the scene defined in Section III-B. We then discuss methods and a novel manipulation primitive that manages the excess slack present in multiple cable settings for instantiating IRON-MAN to physically untangle up to 3 cables.

### A. Graph-Based Disentangling Action Planning

We present IRON-MAN, an algorithm for disentangling multiple cables knotted or twisted together. IRON-MAN assumes access only to the knot's implicit graph structure and disentangles $n$ cables by removing crossings repeatedly. Multiple cable disentangling requires reasoning about the $2n$ endpoints, increased number of crossings, and complex cable slack management.

*1) Multi-cable Reidemeister Moves:* With $n$ cables in the scene, there are $n$ right endpoints and $n$ left endpoints. IRON-MAN first locates the rightmost endpoint $v_r$ belonging to some cable $c_i$. After defining $v_r$, the leftmost endpoint $v_l$ is defined as the leftmost endpoint belonging to some cable $c_j$, where $i \neq j$ and $i, j \in \{1, \ldots, n\}$ for any $n > 1$ respectively. When there is only a single cable $c_k$ remaining in the scene and $n = 1$, we define $v_r$ and $v_l$ to be the right and left endpoints of $c_k$, breaking ties arbitrarily. Reidemeister moves are performed on this newly defined set of endpoints.

*2) Multi-Cable Node Deletion Moves:* Due to the increased cable length in the scene, each crossing removal requires complicated slack management in physical disentangling, as we must perform multiple successful Node Deletion actions without introducing new crossings from cable slack being pulled through. IRON-MAN categorizes crossings as either *non-trivial* or *trivial* when determining which crossings to remove. *Non-trivial crossings* are integral to maintaining the knot structure, while *trivial crossings* are not integral to the knot structure and can be undone by performing a Reidemeister move or a variant of it. For example, a twist is a set of trivial crossings, in which pulling the two cables apart as in a Reidemeister move will undo all of the crossings. Removing a trivial crossing from a configuration does not change the number and types of knots present in the configuration, and as a result does not reduce the overall configuration density. IRON-MAN manages physical cable slack effectively and efficiently by only undoing non-trivial crossings, which cannot be undone by Reidemeister moves, while Reidemeister moves remove the trivial crossings. IRON-MAN traverses the graph from the rightmost endpoint $v_r$ and performs a Node Deletion move on the first non-trivial undercrossing.

*3) Algorithm Summary:* IRON-MAN disentangles $n$-cable knots given a graph representation of the knotted structure using multi-cable Reidemeister and Node Deletion moves, as defined above. First, IRON-MAN performs a Reidemeister move to remove trivial crossings such as intra-cable loops and disambiguate the knot configuration. Next, IRON-MAN successively performs Node Deletion moves on the first non-trivial crossing with respect to the rightmost endpoint in the scene until none remain. IRON-MAN then performs a variation of the Reidemeister move consecutively to remove all cables from the scene and any remaining trivial crossings (discussed further in the following section). This returns the cables to a fully unoccluded, disentangled state such that $|V| = 2n$ with a vertex for each endpoint in the scene.

### B. Physical Disentangling

*1) Perception:* IRON-MAN operates on a graph representation, which is not directly observable. Therefore, we instantiate IRON-MAN to operate on image inputs using learned perception components inspired by prior work [6]. This makes it possible to instantiate IRON-MAN for cable disentangling from raw image input on a physical robotic system. As in HULK [6], we learn the newly-defined keypoints for holding and pulling actions and right and left endpoints from RGB image inputs to perform IRON-MAN's multi-cable Reidemeister and Node Deletion moves on physical knots. This section describes the networks used to implement IRON-MAN, while Section VI details the training dataset generation procedure.

As in HULK, our approach uses a ResNet-34 backbone to learn two mappings, each of which transform an RGB image

input to two heatmaps: (1) $g_1 : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times2}$ maps an image to 2 heatmaps centered respectively at the keypoints $\hat{p}_{\text{hold}}$ and $\hat{p}_{\text{pull}}$, located at the first non-trivial intersection as identified by IRON-MAN, and (2) $g_2 : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times2}$ maps an image to two heatmaps centered respectively at 2 keypoints $\hat{p}_{\text{l}}$ and $\hat{p}_{\text{r}}$, located at the endpoints of cables $c_i$ and $c_j$, where $i \neq j$ to implement IRON-MAN's multi-cable Reidemeister moves. Note that $\hat{p}_r$ corresponds to the rightmost endpoint in the scene, while $\hat{p}_l$ is the leftmost endpoint in the scene that is not part of cable $c_j$. By querying LOKI as described in Section IV-B.2, we obtain refined keypoints, $\tilde{p}_l, \tilde{p}_r, \tilde{p}_{\text{hold}}, \tilde{p}_{\text{pull}}$, and their corresponding gripper orientations for executing grasps. Cable $c_i$ is *semi-disentangled* from a cable configuration if the only crossings involving $c_i$ are trivial and $c_i$ can be fully disentangled when pulled apart from the other cables. We detect when the cable $c_i$ corresponding to the rightmost endpoint $\hat{p}_r$ is semi-disentangled from the remaining cables in the scene with a binary classifier, $h : \mathbb{R}^{640\times480\times3} \mapsto \{0,1\}$.

*2) Manipulation:* To execute the actions from IRON-MAN on a set of physical cables, we apply a novel manipulation primitive, **Cable Extraction moves**, along with **Reidemeister moves** and **Node Deletion moves** (Sec. IV-C), to the task of multi-cable disentangling. The multi-cable Reidemeister and Node Deletion moves are executed as described in Section IV-C with the newly-defined keypoints described in Section V-B.1. We iteratively disentangle each cable in the scene and drop fully disentangled cables at $p_{\text{term}}$, a predefined point within the termination area.

Motivated by the slack management difficulties of multi-cable disentanglement (Sec. V-A), we define **Cable Extraction moves**, a novel manipulation primitive to fully disentangle and remove a semi-disentangled cable from a scene. We perform Cable Extraction moves when the cable $c_i$ corresponding to the right endpoint $\tilde{p}_r$ is semi-disentangled. The right arm grasps the semi-disentangled cable $c_i$ at its right endpoint $\tilde{p}_r$, while the left arm holds $\tilde{p}_l$, the leftmost endpoint from a separate cable $c_j$ $(i \neq j)$, by *pinning* the cable at $\tilde{p}_l$ against the workspace surface without closing the gripper jaws and grasping the cable. *Pinning* differs from *holding* in that when holding, the gripper jaws close to grasp the cable but do not push the cable against the workspace. Then, the right arm pulls the cable $c_i$ to a predefined termination point $p_{\text{term}}$, removing all trivial crossings:

$$\mathbf{a}_{t,r} = (\tilde{p}_{x,r}, \tilde{p}_{y,r}, p_{x,\text{term}} - \tilde{p}_{x,r}, p_{y,\text{term}} - \tilde{p}_{y,r}, \hat{\theta}_r, 1)$$
$$\mathbf{a}_{t,l} = (\tilde{p}_{x,l}, \tilde{p}_{y,l}, 0, 0, \hat{\theta}_l, 0)$$

We start with a Reidemeister move to pull endpoints $\tilde{p}_r$ and $\tilde{p}_l$ to opposite ends of the manipulation workspace, removing any initial trivial crossings. Next, we perform successive Node Deletion moves with $\tilde{p}_{\text{hold}}$ and $\tilde{p}_{\text{pull}}$ at the first non-trivial undercrossing from the rightmost endpoint $\hat{p}_r$, which belongs to cable $c_i$. When we detect that $c_i$ is semi-disentangled, we perform a Cable Extraction move to undo any remaining trivial crossings involving $c_i$ and remove $c_i$ from the scene. This procedure continues until all cables are disentangled and deposited in the termination area.

**Algorithm 1** Disentangling with IRON-MAN

**Input:** RGB image of cable
Predict $\tilde{p}_l$, $\tilde{p}_r$, $\tilde{p}_{\text{hold}}$, $\tilde{p}_{\text{pull}}$
$c_i, c_j \leftarrow$ cables corresponding to $\tilde{p}_r, \tilde{p}_l$, respectively.
Reidemeister move with $\tilde{p}_r$ (cable $c_i$), $\tilde{p}_l$ (cable $c_j$)
**while** workspace not empty **do**
    Predict $\tilde{p}_l$, $\tilde{p}_r$, $\tilde{p}_{\text{hold}}$, $\tilde{p}_{\text{pull}}$
    $c_i \leftarrow$ cable corresponding to $\tilde{p}_r$
    Execute SPiDERMan recovery policy
    **if** cable $c_i$ is semi-disentangled **then**
        Cable Extraction move with $\tilde{p}_r$, $\tilde{p}_l$
    **else**
        Node Deletion move with $\tilde{p}_{\text{hold}}$, $\tilde{p}_{\text{pull}}$
**return** DONE

## VI. Experiments

We evaluate IRON-MAN for cable disentangling on initial cable configurations with three tiers of increasing difficulty. We implement the full system for experiments on the bilateral da Vinci surgical robot. Because this is the first work studying the multi-cable disentangling problem and because single-cable algorithms are not well-defined in this setting, we are not aware of any existing algorithms that would provide a meaningful baseline comparison in this setting.

### A. Training Dataset Generation

We train the Reidemeister and Node Deletion coarse keypoint prediction models $g_1$ and $g_2$ on a dataset of 270 real workspace images with hand-labeled keypoints, augmented to a dataset of 3,500 examples. We similarly train the semi-disentanglement classifier $h$ on a dataset of 170 real workspace images. For each image, we assign labels 0 or 1 by hand to indicate "semi-disentangled" or "entangled," respectively, and augment to a dataset of 5,200 examples before training with a binary cross-entropy loss. All datasets consist *only* of configurations containing up to two cables, where the cables' colors are either both white or red and white. These datasets are augmented via affine transforms, lighting shifts, and blurring.

To reduce manipulation errors during experiments, we project any keypoints predictions $\hat{p} \in \{\hat{p}_{\text{hold}}, \hat{p}_{\text{pull}}, \hat{p}_l, \hat{p}_r\}$ located off the cables onto the cable mask obtained by color thresholding from the background.

We train LOKI on 3,000 $200 \times 200$ crops of simulated images of crossings of red and white cables in Blender 2.80 [10]. Offset heatmap labels are generated in simulation by producing 2D Gaussian distributions centered along the cable width with a standard deviation of 5 px. SPiDERMan detects when to perform Recovery actions via analytical methods for sensing cable contours in the workspace [22].

### B. Tiers of Difficulty

Across all difficulty tiers, the knots considered are *dense* and depicted in Figure 4. The tiers are defined by the number of cables in the knot and whether the class of knots was present in the training dataset.

**Tier 1:** Two-cable knots where the class of knots was present in the training dataset (two-cable twists, Carrick Bend, Sheet Bend, and Square knots).

| Tier | Color | Success Rate | Disentangling Actions | Recovery Actions | Total Actions | Failure Modes |
|---|---|---|---|---|---|---|
| 1 | r-w | 12/12 | 7 | 0 | 7.5 | A (0), B (0), C (0), D (0) |
| 1 | w-w | 10/12 | 11.5 | 1 | 12.5 | A (0), B (0), C (1), D (1) |
| 2 | r-w | 7/12 | 19 | 0 | 20 | A (1), B (1), C (1), D (2) |
| 2 | w-w | 9/12 | 15.5 | 2 | 15 | A (0), B (1), C (2), D (0) |
| 3 | r-w-w | 11/12 | 16 | 1 | 17 | A (0), B (0), C (0), D (1) |
| 3 | w-w-w | 9/12 | 15 | 1 | 16 | A (0), B (1), C (0), D (2) |

TABLE I: **Physical Results:** We report the success rate and median number of actions to fully disentangle all cables in a scene using the physical implementation of IRON-MAN. We consider sets of cables that are all similarly colored (white-white) and differently colored (red-white). This implementation disentangles two cables in Tier 1 and 2 configurations and three cables in Tier 3 configurations with 80.5% success overall. We observe four failure modes: (A) one or more cables springing out of the manipulation workspace, (B) gripper collision in high-density configurations, (C) exceeding a maximum number of disentangling actions, and (D) moving entangled cables to the terminated workspace.

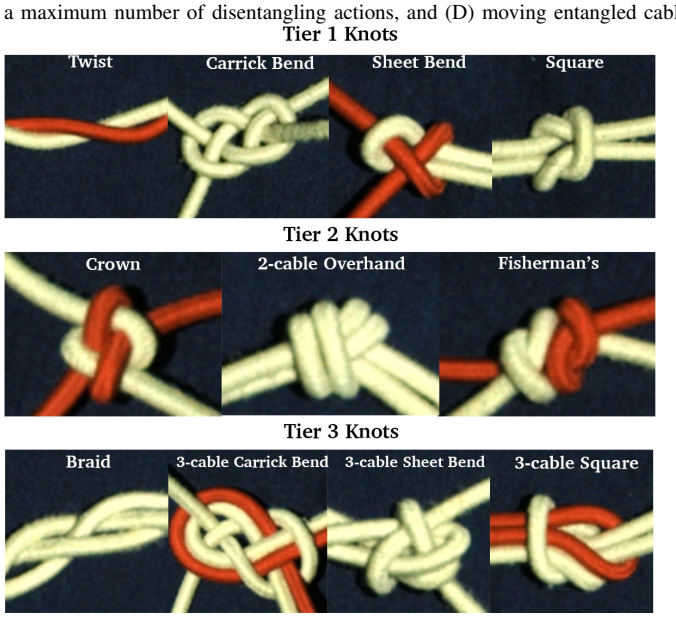


Fig. 4: **Tiers of Configuration Difficulty**: The configurations considered in this paper are organized into 3 tiers based on the number of cables in the knot and whether the class of knot was present in training data images. For each configuration, we consider both settings where cables have contrasting colors (red and white) and where cables are of the same color (all white).

**Tier 2:** Two-cable knots where the class of knots was not present in the training dataset (Crown, Fisherman's, and two-cable Overhand knots).

**Tier 3:** Three-cable knots where the class of knots was not present in the training dataset (braids, three-cable Carrick Bend, three-cable Sheet Bend, and three-cable Square knots).

### C. Experimental Setup

We use a da Vinci Research Kit (dVRK) surgical robot with two 7-DOF arms to untie 2 or 3 cut elastic hairties of diameter 5 mm and length 15 cm. We chose to disentangle hairties, as they have a smooth surface and fit well in the dVRK's end effector. We also use a foam padded stage on which the hairties rest during experiments. This helps to avoid end effector damage under collisions with the manipulation surface and creates friction with the hairties to prevent them from sliding out of the workspace. We collect $1920 \times 1200$ overhead RGB images for perception inference with a Zivid OnePlus RGBD camera.

### D. Results

Table I presents the results from the physical trials. IRON-MAN succeeds in disentangling all cables 91.6%, 66.6%, and 83.3% of the time on Tier 1, Tier 2, and Tier 3 configurations, respectively. The success rates drop for the knots in Tiers 2 and 3, as they are not present in training data images. In successful cases, both the number of disentangling actions (Node Deletion and Cable Extraction moves) and the number of SPiDERMan Recovery actions (Re-posing and Wedge Recovery moves) also increase in these tiers.

### E. Failure Modes

In the disentangling experiments, the physical implementation of IRON-MAN encounters four failure modes:

(A) One or more cables springing out of the reachable manipulation workspace due to elastic cable physics.
(B) Robot gripper jaws colliding when executing Node Deletion moves in high-density cable configurations.
(C) Exceeding a maximum threshold of 20, 30, and 30 disentangling actions for Tiers 1, 2, and 3 respectively, due to repeatedly-poor action predictions and executions.The reason for the difference in the maximum allowable actions is because the class of Tier 1 knots was seen in training while Tier 2 and Tier 3 were not.
(D) Semi-disentangled cables dropped in the termination area rather than fully disentangled cables, due to poorly-executed Cable Extraction moves that do not effectively pin down remaining cables in the scene.

We observe the most common failure mode to be moving semi-disentangled (rather than fully disentangled) cables to the termination area with poor Cable Extraction manipulation (D). When the left arm does not effectively pin the remaining cables in the scene, high cable friction causes multiple cables to be moved with the grasped semi-disentangled cable into the termination area without removing trivial crossings. We also observe two manipulation failure modes relating to cable grasps. Due to the hairties' elastic properties, poor cable grasps occasionally cause one or more cables to spring out of the reachable manipulation workspace, which yields an irrecoverable state (A). In high-density configurations, disentangling actions often require gripper jaws to grasp adjacent cable segments that are very close together. These grasps may cause gripper jaw collision, which requires human intervention to reset the robot and is deemed a failure (B). We limit the number of disentangling actions to 20 actions for Tier 1 configurations and 30 actions for Tier 2 and 3 configurations to end rollouts when the robot is repeatedly unable to execute effective disentangling actions in pathological high-density configurations where gripper jaws cannot grasp between cable segments (C).

## VII. Discussion and Future Work

We formalize the problem of autonomously disentangling multiple cables and present IRON-MAN, a geometric algorithm for disentangling multiple-cable knots. IRON-MAN, iteratively undoes inter-cable and intra-cable crossings by operating on a graphical representation of the cable. We then build on the perception driven untangling approaches from prior work (HULK, LOKI, SPiDERMan) to instantiate IRON-MAN for disentangling cables on a physical robotic system using image inputs. Experiments suggest that the physical implementation of IRON-MAN can disentangle up to three cables with 80.5% success. In future work, we will extend IRON-MAN and its physical implementation to scenarios where one endpoint is fixed or where cables are tangled with rigid objects such as electrical appliances or tools.

## References


[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation", *Science*, vol. 364, no. 6446, 2019.

[2] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation", *arXiv preprint arXiv:2101.00733*, 2019.

[3] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control", *arXiv preprint arXiv:1812.00568*, 2018.

[4] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation", *arXiv preprint arXiv:1806.08756*, 2018.

[5] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali, *et al.*, "Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images", *arXiv preprint arXiv:2003.12698*, 2020.

[6] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, M. Hwang, V. Viswanath, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Untangling dense knots by learning task-relevant keypoints", *arXiv preprint arXiv:2011.04999*, 2020.

[7] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés, "Survey on multi-robot manipulation of deformable objects", in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2019.

[8] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation", *arXiv preprint arXiv:2003.09044*, 2020.

[9] F. Kempf and J. Fischer, *Fiber cable made of high-strength synthetic fibers for a helicopter rescue winch*, US Patent 7,866,245, Jan. 2011.

[10] T. Lallemand, *Blender*, US Patent App. 29/074,220, Jun. 1998.

[11] R. Lee, D. Ward, A. Cosgun, V. Dasagi, P. Corke, and J. Leitner, "Learning arbitrary-goal fabric folding with one hour of real robot experience", *arXiv preprint arXiv:2010.03209*, 2020.

[12] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation", *arXiv preprint arXiv:2011.07215*, 2020.

[13] W. H. Lui and A. Saxena, "Tangled: Learning to untangle ropes with rgb-d perception", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 837–844.

[14] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks", *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.

[15] J.-p. Merlet and D. Daney, "A portable, modular parallel wire crane for rescue operations", in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010.

[16] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation", in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2146–2153.

[17] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, "Zero-shot visual imitation", in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 2050–2053.

[18] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey", *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[19] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks", *arXiv preprint arXiv:2012.03385*, 2020.

[20] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, *et al.*, "Deep imitation learning of sequential fabric smoothing policies", *arXiv preprint arXiv:1910.04854*, 2019.

[21] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making", *arXiv preprint arXiv:1809.09810*, 2018.

[22] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, J. Ichnowski, E. Novoseller, M. Hwang, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Untangling dense non-planar knots by learning manipulation features and recovery policies", *tinyurl.com/nonplanar-cable*, 2021.

[23] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data", *arXiv preprint arXiv:2003.01835*, 2020.

[24] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations", in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010.

[25] Y. Yamakawa, A. Namiki, M. Ishikawa, and M. Shimojo, "One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors", in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 703–708.

[26] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[27] H. Zhang, J. Ichnowski, D. Seita, J. Wang, and K. Goldberg, "Robots of the lost arc: Learning to dynamically manipulate fixed-endpoint ropes and cables", *arXiv preprint arXiv:2011.04840*, 2020.