Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies

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

Abstract—Robot manipulation for untangling 1D deformable structures such as ropes, cables, and wires is challenging due to their infinite dimensional configuration space, complex dynamics, and tendency to self-occlude. Analytical controllers often fail in the presence of dense configurations, due to the difficulty of grasping between adjacent cable segments. We present two algorithms that enhance robust cable untangling, LOKI and SPi-DERMan, which operate alongside HULK, a high-level planner from prior work. LOKI uses a learned model of manipulation features to refine a coarse grasp keypoint prediction to a precise, optimized location and orientation, while SPiDERMan uses a learned model to sense task progress and apply recovery actions. We evaluate these algorithms in physical cable untangling experiments with 336 knots and over 1500 actions on real cables using the da Vinci surgical robot. We find that the combination of HULK, LOKI, and SPiDERMan is able to untangle dense overhand, figure-eight, double-overhand, square, bowline, granny, stevedore, and triple-overhand knots. The composition of these methods successfully untangles a cable from a dense initial configuration in 68.3% of 60 physical experiments and achieves 50% higher success rates than baselines from prior work. Supplementary material, code, and videos can be found at `https://tinyurl.com/rssuntangling`.

I. INTRODUCTION

Robot manipulation of 1D deformable objects, such as cables, ropes, and wires, can facilitate automation of tasks in industrial, surgical, and household settings [9]. Such manipulation tasks include organization of cables [7, 27], hoses [23], suture thread [31], and wires used in automotive and other electronic assembly [12], as well as reducing clutter and preventing injury in surgical, manufacturing, and home environments [2]. As robots automate more tasks involving 1D deformable objects, which we refer to as "cables," they will increasingly confront highly complex knots, either because the task itself requires knot untangling or because untangling cables is a prerequisite for a downstream task. Prior work has explored untangling manipulation plans consisting of high-level manipulation primitives (e.g., pick and pull points) [28, 19, 7], but robots often face challenges when attempting to execute these plans in physical manipulation tasks, due to the complexity of dynamics modeling and perception in real-world settings. This motivates the need for a low-level controller which can mitigate the gap between high-level planning and physical execution. This is relevant to any task involving
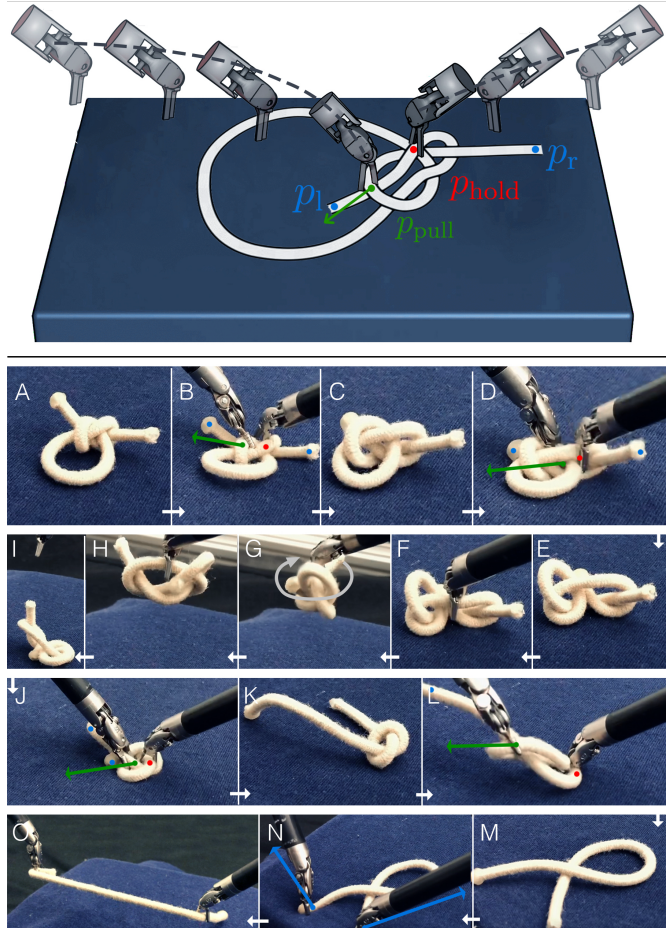

Fig. 1. Six-action sequence to untangle a non-planar square knot. We demonstrate the composition of **HULK** (Hierarchical Untangling from Learned Keypoints), **LOKI** (Local Oriented Knot Inspection), and **SPiDERMan** (Sensing Progress in Dense Entanglements for Recovery Manipulation), three algorithms that jointly perform sequential untangling of a dense, non-planar square knot (A). HULK learns to predict task-relevant keypoints for planning bilateral loosening (Node Deletion) and straightening (Reidemeister) moves. Node Deletion moves sequentially remove a crossing by pulling one cable segment (green) while holding another (red) in place (B-C, D-E, J-K, L-M). LOKI learns manipulation features to perform coarse-to-fine keypoint refinement and infers local cable grasp orientations for fine-grained control. SPiDERMan's recovery policy uses rotation and recentering (Re-Pose) actions that reposition a knot into a new pose at the workspace center for improved grasping (F-I). Finally, a Reidemeister move grasps at predicted endpoints (blue) and pulls taut to yield a linear configuration (O).

untangling or arrangement of cables where sensing grasping precision or recovering from manipulation failures is critical. We propose a low-level controller consisting of two algorithms that jointly improve the robustness of robot untangling.

*Equal contribution.
[1]AUTOLAB at the University of California, Berkeley, USA.
[2]Toyota Research Institute, Inc., USA.
Correspondence to {`priya.sundaresan, jenngrannen`}
`@berkeley.edu`

Untangling cables is a challenging task due to the infinite-dimensional configuration space, self-occlusions, and visual homogeneity of cables. These properties complicate the problems of both generating untangling manipulation plans and executing these plans effectively, especially in densely knotted configurations. Prior work for cable manipulation has primarily explored manipulation planning methods that learn full state information [28, 19] or task-specific features [7]. These methods then execute planned actions through a sequence of analytical, hand-engineered motion primitives, which can be brittle and imprecise under increasing knot complexity and density due to layered self-occlusions, friction at cable intersections, and the need for precision in grasping. Most prior work considers cable configurations that have sufficiently wide space between self-intersections for a gripper jaw to fit or that have only two cable segments at each self-intersection. These manipulation challenges are exacerbated for *non-planar* configurations, where configuration complexity increases to intersections with three or more segments.

Grannen et al. [7] propose HULK, a high-level vision-based planner for untangling of cables in *semi-planar* configurations, where self-intersections involve at most two segments of cable. In this work, we extend HULK with two novel algorithms for non-planar knots. The first algorithm, LOKI (Local Oriented Knot Inspection), learns manipulation features to generate high-precision grasping actions. LOKI refines the cable grasp location and orientation from a coarse keypoint by recentering and learning to grasp orthogonally to the cable's axis, as illustrated in Figure 1. The second algorithm, SPiDER-Man (Sensing Progress in Dense Entanglements for Recovery Manipulation) detects and recovers from 4 common failure modes: (1) consecutive poor action executions, (2) exceeding a maximum threshold of untangling actions, (3) the cable leaving the reachable workspace, and (4) the robot gripper jaws becoming wedged between two segments of dense cable. Unlike prior work which focuses on high precision for failure avoidance [22, 16], SPiDERMan acknowledges the likelihood of failures in real-world settings and plans recovery actions.

We perform physical untangling experiments with cables in dense, non-planar configurations more complex than the semiplanar arrangements considered in prior work and find that the combination of HULK, LOKI, and SPiDERMan achieves 50% higher success rates in comparison to analytical controllers.

II. RELATED WORK

Deformable object manipulation has seen a surge of interest recently in the robotics research community [18, 34, 33, 28, 6, 10, 25, 26], but modeling state and action spaces in such tasks remains challenging. Much prior work focuses on the complex task of generating plans for manipulating cables and higher-dimensional deformable objects. Executing the plans, especially in the presence of failure modes, is often left to future work. In this work, we focus on addressing untangling-specific failure modes by learning controllers to robustly execute plans for the task.

A. Deformable Object Manipulation

Several perception-focused techniques have proven effective in multi-step robotic manipulation of cables and other deformable objects such as cloth and bags. One approach performs state estimation and leverages the learned state representation for downstream planning. For instance, Lui and Saxena [19] and Chi and Berenson [3] propose using classical visual feature extraction to estimate the state of deformable rope and cloth, respectively, subject to partial occlusion. Sundaresan et al. [28] investigate object representation learning via dense object descriptors [24, 5] for rope knot-tying and arrangement, and Ganapathi et al. [6] extend this methodology to 2D fabric smoothing and folding. Alternative perception-driven approaches explore learning latent state spaces [34] for cloth manipulation, or using semantic keypoint perception [7, 33] for rope tracking. However, robust state estimation remains challenging for heavily self-occluded configurations, as are encountered in densely knotted cables. This work builds upon prior work on keypoint prediction for cable untangling, but unlike previous work, we couple perception with robust learned control policies for grasping and failure recovery.

Other methods learn visuomotor policies end-to-end. Imitation learning [25], video prediction models [10], model-free reinforcement learning (RL) [17], and model-based RL [20] have successfully been applied to goal-conditioned fabric manipulation. In other end-to-end approaches, Nair et al. [21] learn a dynamics model for performing template shape matching and knot-tying with rope. Zhang et al. [35] learn to directly regress robotic configurations in joint space for robotic vaulting of cables over obstacles. While these approaches are broadly general and can flexibly apply to many tasks, they can lack precision and do not leverage geometric structure in the problem, making them difficult to apply to dexterous manipulation tasks such as cable untangling, in which geometric reasoning and fine-grained manipulation are critical.

Several recent works have also explored learning robot policies for deformable manipulation entirely in simulation and transferring them to the real world via domain randomization [26, 28, 34, 6, 10, 26, 35]. By contrast, we train perception systems using a combination of data from simulation and the physical world. Crucially, we observe that physical data is critical for tasks that require global reasoning, as it is difficult to simulate the full distribution of configurations encountered during an untangling sequence. However, for tasks that only require local reasoning within specific knots, simulated data can provide sufficient information for robust perception.

B. Cable Untangling

To the best of our knowledge, Lui and Saxena [19] is the first published study of robot cable untying. The authors use RGB-D sensing and classical feature extraction to approximate a tangled cable as a graph consisting of cable crossings and endpoints. We primarily build upon the work of Grannen et al. [7], which (1) presents a geometric algorithm for cable untangling based on the graphical abstraction from Lui and Saxena [19] and (2) uses deep learning to detect cable crossings with

keypoint inference. These keypoints serve as input to a greedy action planner that iteratively reduces the number of crossings.

Lui and Saxena [19] and Grannen et al. [7] present algorithms for generating untangling manipulation plans for cables with semiplanar knots. However, their executed controllers are based on a set of analytical motion primitives, which make simplified assumptions about cable geometry that do not hold in highly deformed or occluded states.

In this work, we focus on the problem of learning a controller to robustly execute the manipulation plans generated by Grannen et al. [7]. First, in adapting the method to a more challenging class of knots, we employ coarse-to-fine refinement strategies to perform robust cable grasping. Several recent works have also studied coarse-to-fine controllers for manipulation tasks requiring precision, including surgical peg transfer [22] and peg insertion [16, 13]. Secondly, we explore recovery from manipulation-induced errors in the context of closed-loop untangling. Recent work has also considered recovery from manipulation errors in contact-based robotics tasks [32, 4, 30]. However, these works do not exploit task geometry, which is particularly relevant to cable untangling.

III. PROBLEM STATEMENT

This work considers untangling a densely knotted cable from RGB image observations. We use a bilateral robot to hold and pull cable segments until the configuration reaches a fully untangled state with no crossings. The cable color is assumed to be distinguishable from the workspace color.

We assume access to a bilateral manipulator and overhead RGB image observations from a camera that is a fixed distance above the workspace manipulation surface. We assume the camera is calibrated with a pixel-to-world transform using standard image registration methods. Within the workspace, we assume that the full cable is within the reachable limits of the robot and that the cable width does not exceed the gripper's maximum opening width. These assumptions ensure that both robot arms are able to hold and pull cable segments. We define three points, $\boldsymbol{w}_l$, $\boldsymbol{w}_c$, and $\boldsymbol{w}_r$, which are located respectively on the left side, center, and right side of the workspace for recovery action planning.

We make two assumptions about the initial cable configuration: (1) *limited non-planarities*: unlike prior work [7] which is limited to 2 segments in each *semi-planar* cable intersection, we allow for up to 3 segments (*non-planar*); and (2) *visible endpoints*: both cable endpoints are unoccluded and visible in overhead RGB images for planning purposes. We distinguish between the two endpoints as left and right based on their pixel coordinate x-values, breaking ties arbitrarily.

We formalize each cable's configuration by extending the graphical cable representation from Lui and Saxena [19] to non-planar configurations. Using this modelling approach, we plan untangling actions for an algorithmic supervisor, BRUCE (see overview in Sec. IV-A). The physical untangling algorithms—HULK (Sec. IV-B), LOKI, and SPiDERMan (Sec. V)—do not explicitly reconstruct this graph, but rather learn relevant features from RGB images to execute actions.
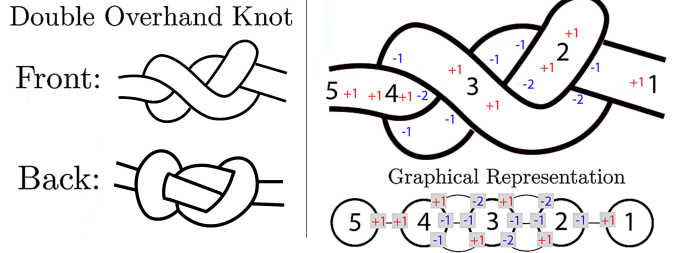

Fig. 2. **Non-Planar Graphical Representation:** We model cable configuration with a graph as in Lui and Saxena [19] and extend this representation to non-planar configurations with an updated annotation method in Equation (1). We visualize a dense double overhand knot from two views to illustrate non-planarity, along with its corresponding annotated graph.

We model cable state via an undirected graph $G = (V, E)$ with cable endpoints and crossing locations represented as vertices $v \in V$ and cable segments between vertices represented as edges $e = (u, v) \in E$ for $u, v \in V$, where $u \neq v$. We use the term "node" and "vertex" interchangeably and label the nodes corresponding to the left and right endpoints as v_l and v_r, respectively. For a node and adjacent edge $(v, e) \in (V, E)$, we annotate the graph with $X(v, e) \in \{+1, -1, -2\}$ (Equation 1) to designate the cable segment hierarchy at each node:

$$X(v,e) = \begin{cases} +1 & \text{if } v \text{ is an endpoint or if } e \text{ crosses} \\ & \text{above all other edges at } v \\ -1 & \text{if } e \text{ crosses below only one edge at } v \\ -2 & \text{if } e \text{ crosses below two other edges at } v. \end{cases}$$
(1)

We define an *under-crossing* as a set of one node v and two incident edges $e_i, e_j \in E$, where $e_i = (v, v')$ and $e_j = (v, v'')$ for some $v', v'' \in V$, $X(e_i, v) = X(e_j, v) \in \{-1, -2\}$, and e_i, e_j are contiguous in the cable. Thus, the cable segment represented by e_i, e_j is occluded by one or two other cable segments. Similarly, an *over-crossing* is a set of one node v and two incident edges $e_i = (v, v')$ and $e_j = (v, v'')$ for some $v', v'' \in V$, where $X(e_i, v) = X(e_j, v) = +1$ and e_i, e_j are contiguous in the cable. Under the non-planar assumption, each node has a degree of at most 6, corresponding to 3 cable segments at a crossing. Edges with the same annotation $X(v, e)$ at a node v represent the same cable segment on either side of the crossing. A cable with $|V| = 2$ has only endpoint nodes and no crossings, and thus is fully untangled.

Each gripper performs either a holding, pulling, or rotating action at each time t. Below, we define all actions with respect to the 2D image frame. The left robot arm performs a pulling action, $\boldsymbol{a}_{t,l}$, by grasping the cable at the pixel $(x_{t,l}, y_{t,l})$ with a (vertical) z-axis grasping rotation of $\theta_{t,l}$, lifting by a fixed amount, pulling to $(x_{t,l} + \Delta x_{t,l}, y_{t,l} + \Delta y_{t,l})$, and releasing the cable. Holding actions are indicated via $(\Delta x_{t,l}, \Delta y_{t,l}) = (0, 0)$. We denote moving between points without grasping the cable via $\mathbb{1}_{\text{grasp}} = 0$. Lastly, $\Delta \theta_{t,l}$ (in degrees) indicates rotating a grasped cable about the z-axis. The right robot arm uses an analogous action formulation, $\boldsymbol{a}_{t,r}$. We show the full action representation for both arms in Equation (2):

$$\boldsymbol{a}_{t,l} = (x_{t,l}, y_{t,l}, \theta_{t,l}, \Delta x_{t,l}, \Delta y_{t,l}, \Delta \theta_{t,l}, \mathbb{1}_{\text{grasp}})$$
$$\boldsymbol{a}_{t,r} = (x_{t,r}, y_{t,r}, \theta_{t,r}, \Delta x_{t,r}, \Delta y_{t,r}, \Delta \theta_{t,r}, \mathbb{1}_{\text{grasp}}). \quad (2)$$

This action space generalizes that of Grannen et al. [7], which only parameterizes grasp points and pull vectors.

IV. PRELIMINARIES

In this section, we review the BRUCE and HULK algorithms introduced in Grannen et al. [7] and discuss relevant algorithmic modifications to accommodate non-planar knots.

A. BRUCE: Basic Reduction of Under-Crossing Entanglements

Grannen et al. [7] study semi-planar knot untangling given observations of the knot's graphical representation, and propose Basic Reduction of Under-Crossing Entanglements (BRUCE), an algorithm to iteratively untangle knots starting from the rightmost endpoint. BRUCE makes use of two manipulation primitives adapted from Lui and Saxena [19]: (1) **Reidemeister moves** pull the cable endpoints apart to reduce self-occlusions that are not part of a knot, and (2) **Node Deletion moves** delete a crossing in the graph by holding an over-crossing edge in place with one arm while the other arm pulls out the cable from the corresponding under-crossing.

BRUCE takes a Reidemeister move to disambiguate the cable state, followed by alternating Node Deletion and Reidemeister moves to sequentially delete crossings and attempt to linearize the cable. In semi-planar configurations, Node Deletion moves pull out the under-crossing edge (labeled -1) exiting the first node v traced from the right endpoint, while holding a corresponding over-crossing edge $(+1)$ at v in place.

This work extends Node Deletion moves to operate at non-planar crossings, which contain under-crossing edges labeled as both -1 and -2. In this setting, a Node Deletion move holds an over-crossing edge $(+1)$ and pulls out the under-crossing edge $(-1$ or $-2)$ traced from the right endpoint. This action reduces $|E|$ and $|V|$ by at least 2 and 1, respectively, until the algorithm terminates with $|V| = 2$: a fully linear cable with no crossings and two endpoints.

B. HULK: Hierarchical Untangling from Learned Keypoints

Grannen et al. [7] consider cable untangling from RGB image observations, where the ground truth cable graph representation is not directly observable. To infer untangling actions while bypassing the need for full graph reconstruction, Grannen et al. [7] propose Hierarchical Untangling from Learned Keypoints (HULK), which predicts keypoints from images to execute BRUCE's manipulation primitives. HULK learns a mapping from an image to four Gaussian heatmaps centered at task-relevant keypoints, $f : \mathbb{R}^{640 \times 480 \times 3} \mapsto \mathbb{R}^{640 \times 480 \times 4}$. The means of the four heatmap outputs, denoted by $\hat{\boldsymbol{p}}_l, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{p}}_{\text{pull}}$, and $\hat{\boldsymbol{p}}_{\text{hold}}$, indicate the predicted pixel locations of the left and right cable endpoints ($\hat{\boldsymbol{p}}_l$ and $\hat{\boldsymbol{p}}_r$) and the predicted pull and hold grasp locations for the next planned Node Deletion move ($\hat{\boldsymbol{p}}_{\text{pull}}$ and $\hat{\boldsymbol{p}}_{\text{hold}}$). HULK learns to predict the pull and hold keypoints at the first under/over-crossing pair relative to the rightmost endpoint.
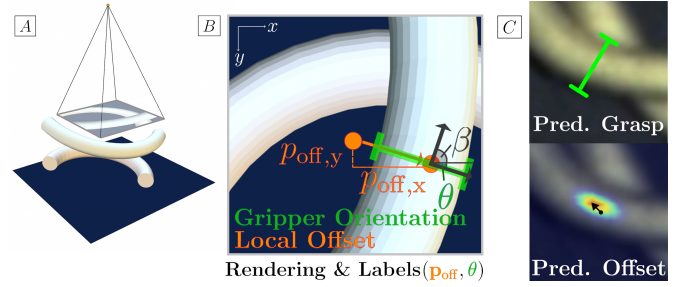

Fig. 3. **LOKI**: LOKI learns precise cable grasping from simulation. We simulate cable crossings in Blender 2.80 as overlapping, warped cylindrical meshes (A) and render RGB crops (B). Each 200×200 pixel crop is annotated with a ground truth optimal grasp angle θ orthogonal to the topmost cylinder orientation β and a local offset $(p_{\text{off,x}}, p_{\text{off,y}})$ from a coarse grasp prediction to recenter the grasp location along the cable width. At test time (C), LOKI takes a real cable crop centered at a HULK keypoint, refines the keypoint via recentering, and estimates a robust gripper orientation for grasping.

In this work, we adapt the HULK training procedure from Grannen et al. [7], originally trained on semiplanar knots, to operate on dense, non-planar configurations. Since training HULK only requires light supervision in the form of keypoints, we follow the training procedure in [7] and train HULK on 200 pairs of images and hand-annotations for non-planar configurations. As in [7], we use a Gaussian standard deviation of 8 px for the heatmap training data centered about each hand-labelled keypoint, and we apply various data augmentation techniques including image adjustments and affine transformations to generate a training dataset of 3,500 examples. Once trained, HULK uses the predicted keypoints $\hat{\boldsymbol{p}}_l, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{p}}_{\text{pull}}$, and $\hat{\boldsymbol{p}}_{\text{hold}}$ to plan a bilateral move $(\boldsymbol{a}_{t,l}, \boldsymbol{a}_{t,r})$ at time t for the left and right arms, respectively, as follows.

For a **Node Deletion** move, the right arm holds the configuration at $\hat{\boldsymbol{p}}_{\text{hold}}$ while the left arm grasps at $\hat{\boldsymbol{p}}_{\text{pull}}$ and pulls in the direction away from $\hat{\boldsymbol{p}}_{\text{hold}}$ by some fixed vector $\boldsymbol{n} = (n_x, n_y)$ to slacken the cable. Both arms use coarse analytical grasp rotations $\hat{\theta}_{\text{pull}} = \arctan\left(\frac{\hat{p}_{\text{pull},y} - \hat{p}_{\text{hold},y}}{\hat{p}_{\text{pull},x} - \hat{p}_{\text{hold},x}} \right)$ and $\hat{\theta}_{\text{hold}} = \hat{\theta}_{\text{pull}} + 90°$:

$$\boldsymbol{a}_{t,l} = (\hat{p}_{\text{pull},x}, \hat{p}_{\text{pull},y}, \hat{\theta}_{\text{pull}}, n_x, n_y, 0, 1)$$
$$\boldsymbol{a}_{t,r} = (\hat{p}_{\text{hold},x}, \hat{p}_{\text{hold},y}, \hat{\theta}_{\text{hold}}, 0, 0, 0, 1).$$

For a **Reidemeister** move, the left and right arms simultaneously grasp at $\hat{\boldsymbol{p}}_l$ and $\hat{\boldsymbol{p}}_r$ and pull the endpoints to opposing workspace ends $\boldsymbol{w}_l$ and $\boldsymbol{w}_r$, with analytical grasp rotations $\hat{\theta}_l$ and $\hat{\theta}_r$ obtained as the angle between $(1, 0)$ and the vector orthogonal to the first principal component of a masked crop around $\hat{\boldsymbol{p}}_l$ and $\hat{\boldsymbol{p}}_r$:

$$\boldsymbol{a}_{t,l} = (\hat{p}_{l,x}, \hat{p}_{l,y}, \hat{\theta}_l, w_{l,x} - \hat{p}_{l,x}, w_{l,y} - \hat{p}_{l,y}, 0, 1)$$
$$\boldsymbol{a}_{t,r} = (\hat{p}_{r,x}, \hat{p}_{r,y}, \hat{\theta}_r, w_{r,x} - \hat{p}_{r,x}, w_{r,y} - \hat{p}_{r,y}, 0, 1).$$

V. METHODS

We present two new algorithms, LOKI (Local Oriented Knot Inspection) and SPiDERMan (Sensing Progress in Dense Entanglements for Recovery Manipulation), which are combined to learn a robust controller to increase robustness of high-level manipulation plans from Grannen et al. [7].

A. LOKI: Local Oriented Knot Inspection

To enable finer-grained grasp planning for the knot untying primitives of HULK, we introduce a low-level controller called LOKI. This controller jointly infers robust antipodal grasps that enclose a cable segment orthogonally and performs coarse-to-fine refinement of keypoints. These adjustments are designed to prevent near-miss grasps, a common failure mode in HULK [7]. While prior work in cable manipulation has employed analytical grasp planning [28, 7], such heuristic methods fail to generalize to dense, non-planar configurations. Thus, we propose LOKI: Local Oriented Knot Inspection. LOKI maps a locally-cropped cable image, centered at one of the four HULK keypoints $\hat{\boldsymbol{p}} \in \{\hat{\boldsymbol{p}}_l, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{p}}_{\text{pull}}, \hat{\boldsymbol{p}}_{\text{hold}}\}$, to (1) θ: an angle about the z-axis denoting the top-down grasp orientation and (2) $\boldsymbol{p}_{\text{off}} = (p_{\text{off},x}, p_{\text{off},y})$: a local offset in pixel space to recenter the keypoint along the cable width. We implement LOKI as a multi-headed deep neural network with a ResNet-18 [8] backbone that learns a mapping $g : \mathbb{R}^{200 \times 200 \times 3} \mapsto ([0, 180], \mathbb{R}^{200 \times 200})$. These outputs correspond to the predicted z-axis rotation (in degrees) and an unnormalized 2D heatmap centered at the refined keypoint $\hat{\boldsymbol{p}} + \hat{\boldsymbol{p}}_{\text{off}}$. We discuss the procedure for obtaining the heatmap and rotation training labels below.

1) Dataset Generation: We train LOKI in a self-supervised fashion from synthetic images. Simulation provides two advantages in this setting: (1) LOKI requires reasoning about local cable self-intersections which can be approximated with geometric models more readily than global knotted configurations, and (2) hand-annotation of the cable orientation is tedious to obtain from real data but readily accessible from synthetic meshes. We implement a simulation environment in Blender 2.80 [15] that models cable self-intersections as two overlapping cylindrical meshes slightly warped to emulate the curvature of real cables (see Fig. 3). We generate varied training data by positioning a synthetic overhead camera above, but not centered on, the topmost cylinder. This cylinder is parameterized by a pitch angle β denoting its orientation in the xy-plane and a translation projected to pixel coordinates (u, v). For each synthetic image rendered, we record the desired gripper orientation as $90° + \beta$ to model a grasp enclosing the cable orthogonally and a ground-truth Gaussian heatmap $\mathcal{N}((u, v), \sigma^2 \mathcal{I})$ over the network input image centered at the desired grasp location (u, v), where $\mathcal{I}$ is a 2×2 identity matrix. Empirically, we find that $\sigma = 5$ px yields stable training without producing high-entropy heatmaps. LOKI is trained from 3,500 such examples using mean-squared error in degrees for the orientation output's loss and binary cross entropy loss for the heatmaps.

Given a global image $I \in \mathbb{R}^{640 \times 480 \times 3}$ and a HULK keypoint $\hat{\boldsymbol{p}}$, we take a 60×60 crop centered at $\hat{\boldsymbol{p}}$. We resize this crop to the LOKI network input dimensions, 200×200, resulting in a cropped image $\hat{I}$. LOKI yields outputs $[\hat{\theta}, H] = g(\hat{I})$, such that $\hat{\boldsymbol{p}}_{\text{off}}$ is given by the highest-probability point in H:

$$(\hat{p}_{\text{off},x}, \hat{p}_{\text{off},y}) = \gamma \left[\left(\underset{(u_i,v_i) \in \hat{I}}{\text{argmax}} H[u_i, v_i] \right) - (u_c, v_c) \right],$$

where $\gamma = 60/200$ is a downscaling factor to account for crop resizing, and $(u_c, v_c) = (100, 100)$ is the upscaled crop center (i.e., a coarse keypoint prediction). The refined keypoint $(\hat{p}_x + \hat{p}_{\text{off},x}, \hat{p}_y + \hat{p}_{\text{off},y})$ and predicted gripper orientation $\hat{\theta}$ are used for planning grasps across all manipulation primitives.

B. SPiDERMan: Sensing Progress in Dense Entanglements for Recovery Manipulation

To sense and recover from manipulation failures, we propose SPiDERMan. SPiDERMan uses both learned and analytical methods to sense untangling progress and employs two novel manipulation primitives to avoid and recover from failures. We address four manipulation failure modes from Grannen et al. [7]: (1) consecutive poor action executions due to high cable density, (2) task incomplete after exceeding the maximum number of untangling actions, (3) the cable leaving the workspace during manipulation, and (4) the high-density cable becoming wedged in the robot jaws.

We first discuss how SPiDERMan detects each failure mode from RGB image inputs and next define two manipulation primitives for failure mode recovery.

1) Detection: SPiDERMan employs one learning-based and two analytical perception methods to detect or prevent each of the four failure modes. To address the first two failures, SPiDERMan compares workspace image observations $I_{t'} \in \mathbb{R}^{640 \times 480 \times 3}$ before and after executing actions, where t' denotes the number of untangling, or Node Deletion, actions executed thus far. SPiDERMan is implemented with a ResNet-18 backbone and learns a classifier $d : \mathbb{R}^{640 \times 480 \times 6} \mapsto \{0, 1\}$, trained with a binary cross-entropy loss. To determine when a recovery action is needed due to lack of untangling progress at test time, $d(I_1, I_2)$ uses a simple decision threshold of 0.5 over the softmax of the classifier output to determine whether image I_1 corresponds to a denser (1) or less dense (0) state than I_2. When two successive actions have not effectively loosened the knot, the following condition evaluates to true:

$$d(I_{t'}, I_{t'-1}) \text{ and } d(I_{t'-1}, I_{t'-2}), \tag{3}$$

and we conclude that the configuration is pathological and a recovery action will be performed. In this way, SPiDERMan recognizes untangling progress over *multiple* actions, removing the need for a fixed untangling action limit. Instead, we define a termination condition:

$$d(I_{t'}, I_{t'-5}) \text{ or } d(I_{\text{ref}}, I_{t'}) \tag{4}$$

that checks both for untangling success against an untangled reference cable image I_{ref} and for progress every 5 actions to prevent termination as long as there is progress. Condition (4) evaluates to true when no progress has occurred over the last 5 actions, or if the cable reaches a fully-untangled state.

When $I_{t'}$ is fully untangled, both I_{ref} and $I_{t'}$ are equally dense. For this reason, we use an image observation of a knot-free cable with a single crossing for I_{ref} (Figure 4) to prevent termination false negatives arising from a bias toward I_{ref}.

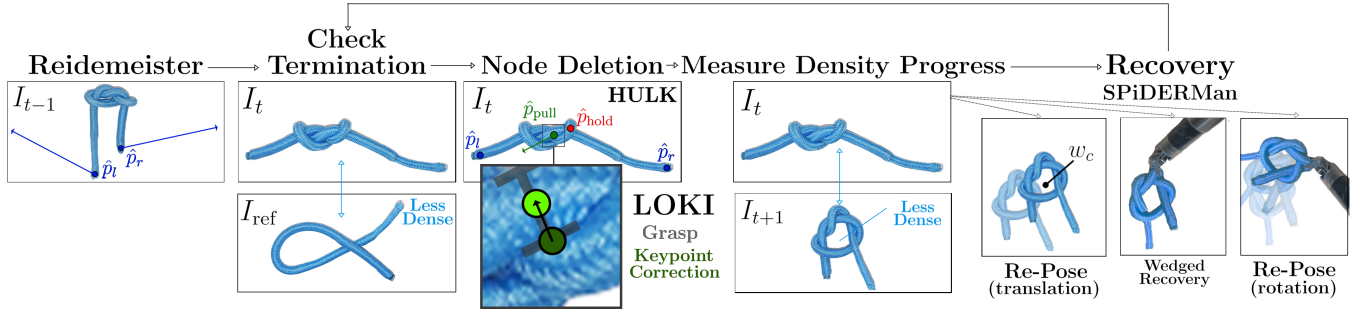SPiDERMan also applies analytical contour-based perception to implement recovery manipulation primitives. Given a

Fig. 4. **System Overview**: We illustrate Algorithm 1 used to perform the cable untangling shown in Figure 1. Starting from the left, we untangle a cable from a non-planar initial configuration following the actions outlined by BRUCE: one initial Reidemeister move followed by successive Node Deletion moves until no crossings remain. HULK instantiates BRUCE's Reidemeister and Node Deletion primitives by regressing 4 keypoints (left endpoint, pull, hold, and right endpoint) from RGB image inputs to define high-level action plans. LOKI refines each action by predicting a local offset to center each keypoint along the cable width and a gripper orientation rotation to grasp orthogonally to the cable direction. SPiDERMan senses action success by comparing configuration density in image observations before and after a Node Deletion action is performed. SPiDERMan also employs contour-detection methods (not shown) for sensing when the cable is approaching workspace limits or when the cable is wedged in the gripper jaws. When a lack of progress or a failure mode is detected, SPiDERMan performs one of two failure recovery manipulation primitives: a Wedged Recovery move or a Re-Posing move (Rotation or Translation). It checks when a cable is fully untangled by comparing its density to that of a pre-defined reference image I_{ref} with a single crossing.

workspace image observation I_t, we preprocess the image by converting to grayscale and applying binary thresholding. The algorithm detects all contiguous contours in the resulting image using the open-source implementation from Suzuki et al. [29] in OpenCV [1]. We extract an approximate contour of the cable $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ via contour area filtering. Next, SPiDERMan approximates the cable center as the mean point $\bar{\boldsymbol{p}}$ of the contour, projected to the lie on the cable: $\hat{\boldsymbol{p}}_{\text{c}} = \arg\min_{\boldsymbol{p}_i \in \mathcal{P}} ||\boldsymbol{p}_i - \bar{\boldsymbol{p}}||$. Between actions, the left and right grippers move to known poses with gripper jaw pixel coordinates $\boldsymbol{g}_l$ and $\boldsymbol{g}_r$, respectively, outside of the field of view of the untangling workspace. Given $\hat{\boldsymbol{p}}_{\text{c}}$, $\boldsymbol{g}_l$, and $\boldsymbol{g}_r$, Condition (5) detects when the cable is wedged in either of the grippers by checking when the distance between the approximate cable center, $\hat{\boldsymbol{p}}_{\text{c}}$, and either gripper tooltip, $\boldsymbol{g}_r$ or $\boldsymbol{g}_l$, is below a hand-tuned threshold of 20 px:

$$\min\{||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{g}_l||, ||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{g}_r||\} < 20 \text{ px}. \tag{5}$$

To detect when the cable is leaving the workspace, we define a second condition over the contour-based perception method above to check when the center of the cable mask $\hat{\boldsymbol{p}}_{\text{c}}$ exceeds a hand-tuned 200 px threshold from the workspace center $\boldsymbol{w}_c$:

$$||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{w}_c|| > 200 \text{ px}. \tag{6}$$

2) Recovery: We define two novel manipulation primitives to recover once a failure is detected: (1) **Re-Posing** moves reorient and place a cable at the workspace center, and (2) **Wedged Recovery** moves detach a gripper jaw wedged between two cable segments.

For a **Re-Posing** move, we improve the Recentering primitive from Grannen et al. [7] to rotate the cable in addition to centering the configuration in the workspace. The right arm first grasps the cable at the center of the configuration $\hat{\boldsymbol{p}}_c$ with a grasp location $\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{p}}_{c,\text{off}}$ and grasp rotation of $\hat{\theta}_c$ predicted by LOKI. The right arm then lifts by a fixed amount and moves the cable to the predefined workspace center $\boldsymbol{w}_c$. When two successive poor actions are detected via Equation (3), the right gripper then rotates by $180°$ to re-pose the configuration before

releasing the grasp. A *Re-Posing move (Rotation)* includes this rotation, while a *Re-Posing move (Translation)* does not:

$$\boldsymbol{a}_{t,r} = (\hat{p}_{c,x}, \hat{p}_{c,y}, \hat{\theta}_c, w_{c,x} - \hat{p}_{c,x}, w_{c,y} - \hat{p}_{c,y}, \underbrace{\{0°, 180°\}}_{\text{optional rotation}}, 1).$$

A **Wedged Recovery** move is comprised of two successive actions. When detecting that the left robot gripper jaw is wedged between cable segments, the left arm first brings the stuck cable to the workspace center $\boldsymbol{w}_c$ and releases its grasp. Given the detected cable contour $\mathcal{P}$, the right arm grasps the cable at the far right point on the cable mask $\hat{\boldsymbol{p}}_{\text{hold}} = \arg\min_{\boldsymbol{p}_i \in \mathcal{P}} ||\boldsymbol{p}_i - \boldsymbol{w}_r||$ with a grasp location $\hat{\boldsymbol{p}}_{\text{hold}} + \hat{\boldsymbol{p}}_{\text{hold,off}}$ and grasp rotation $\hat{\theta}_{\text{hold}}$ given by LOKI and holds it down on the workspace while the left arm returns to its home position, a fixed height above the predefined point $\boldsymbol{w}_l$:

$$\boldsymbol{a}_{t,l} = (w_{l,x}, w_{l,y}, 0, w_{c,x} - w_{l,x}, w_{c,y} - w_{l,y}, 0, 1)$$
$$\boldsymbol{a}_{t+1,r} = (\hat{p}_{\text{hold},x}, \hat{p}_{\text{hold},y}, \hat{\theta}_{\text{hold}}, 0, 0, 0, 1)$$
$$\boldsymbol{a}_{t+1,l} = (w_{c,x}, w_{c,y}, 0, w_{l,x} - w_{c,x}, w_{l,y} - w_{c,y}, 0, 0).$$

When the cable is wedged in the right robot gripper, we execute the equivalent procedure with the roles of the right and left arms switched. In this case, $\hat{\boldsymbol{p}}_{\text{hold}}$ represents the far left point on the cable mask, and the right arm returns to the home position above the predefined point $\boldsymbol{w}_r$.

Algorithm 1 presents the untangling algorithm combining HULK from Grannen et al. [7] with LOKI and SPiDERMan, and Figure 4 illustrates the system.

VI. EXPERIMENTS

We experimentally evaluate the separate and joint effectiveness of HULK, LOKI, and SPiDERMan in performing physical untangling of dense semi-planar and non-planar knots.

A. Overview of Policies

We compare (1) the proposed method (Alg. 1) synthesizing HULK, LOKI, and SPiDERMan (H+L+S) against three baselines: (2) HULK alone (H), (3) HULK with only LOKI

Algorithm 1 Untangling with LOKI and SPiDERMan

Input: RGB image of cable

Reidemeister move with $\underbrace{\hat{\boldsymbol{p}}_r + \hat{\boldsymbol{p}}_{r,\text{off}}, \hat{\theta}_r}_{\text{HULK}}, \underbrace{\hat{\boldsymbol{p}}_l + \hat{\boldsymbol{p}}_{l,\text{off}}}_{\text{LOKI}}, \hat{\theta}_l$

while NOT Eq. (4) **do** $\qquad\triangleright$ Termination check
$\quad$ **while** Eq. (5) **do** $\qquad\triangleright$ Wedge check
$\qquad$ Wedge Recovery with $\hat{\boldsymbol{p}}_{\text{hold}} + \hat{\boldsymbol{p}}_{\text{hold,off}}, \hat{\theta}_{\text{hold}}, \boldsymbol{w}_c, \boldsymbol{w}_{l/r}$
$\quad$ **if** Eq. (3) **then** $\qquad\triangleright$ Poor actions check
$\qquad$ Re-Posing (Rotation) with $\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{p}}_{c,\text{off}}, \hat{\theta}_c, \boldsymbol{w}_c$
$\quad$ **if** Eq. (6) **then** $\qquad\triangleright$ Leaving workspace check
$\qquad$ Re-Posing (Translation) with $\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{p}}_{c,\text{off}}, \hat{\theta}_c, \boldsymbol{w}_c$
$\quad$ Node Deletion with $\hat{\boldsymbol{p}}_{\text{hold}} + \hat{\boldsymbol{p}}_{\text{hold,off}}, \hat{\theta}_{\text{hold}}, \hat{\boldsymbol{p}}_{\text{pull}} + \hat{\boldsymbol{p}}_{\text{pull,off}}, \hat{\theta}_{\text{pull}}$
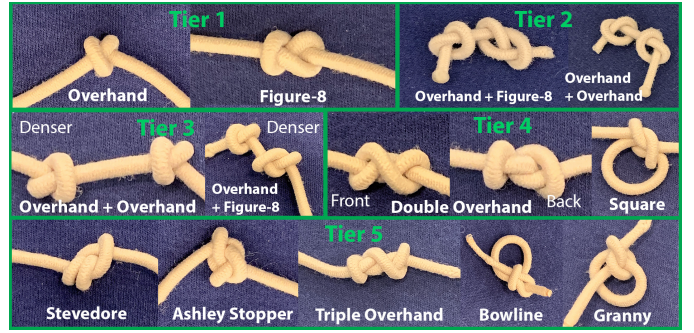
return DONE

Fig. 5. **Difficulty Tiers:** Examples of dense initial configurations used in physical untangling experiments. We categorize configurations into 5 tiers of difficulty based on configuration complexity (semi-planar vs. non-planar), knot density (qualitatively measured), number of knots, and whether these knots were present in the training data. Tiers 1 and 2 are semi-planar and are drawn from [7], while Tier 3 contains semi-planar configurations denser than those in [7], and Tiers 4 and 5 consist of non-planar knots.

(H+L), and (4) HULK with only SPiDERMan (H+S). Notably, baselines (2) and (4) employ analytical grasp planning [7]. In this strategy, Reidemeister grasps are planned orthogonally to the major axis given by principal component analysis on masked endpoint crops. Node Deletion grasps are also heuristically planned and are orthogonal to predicted action vectors as described in Sec. IV-B. Baselines (2) and (3) do not use any recovery manipulation primitives. All policies employ the termination condition given by Eq. (4).

We train HULK and SPiDERMan on the same dataset of 320 cropped, 640×480 real RGB workspace image observations augmented 8X and centered around detected cable contours according to Section V-B1. We also train LOKI from 3,000 200×200 synthetically-generated cable crops.

B. Tiers of Difficulty

All policies are tested on *dense* knots across five difficulty tiers (Fig. 5), categorized by the degree of non-planarity and whether the knot class appeared during training:

Tier 1: A single semiplanar knot (figure-eight or overhand); knots of this class were present in training data, as in [7]

Tier 2: Two semiplanar knots (figure-eight or overhand); knots of this class were present in training data, as in [7]

Tier 3: Two semiplanar knots (figure-eight or overhand); knots of this class were present in training data and denser than those of [7]

Tier 4: A single non-planar knot (double overhand or square); knots of this class were present in training data

Tier 5: A single non-planar knot (stevedore, bowline, ashley stopper, granny, or heaving line); knots of this class were *not* present in training data

HULK, LOKI, and SPiDERMan were trained on configurations containing up to two overhand and figure-eight knots, single double overhand knots, and single square knots. While Tiers 1-4 contain knots types that appear in the training dataset, Tier 5 tests the generalization capabilities of HULK, LOKI, and SPiDERMan to knot types absent in the training data.

C. Experimental Setup

We execute all experiments using the bilateral dVRK robot [14] equipped with two 7-DoF arms. The dVRK performs untangling of a cut elastic hairtie on a boxed, foam-padded surface to avoid end-effector damage and to prevent the cable from easily leaving the workspace. Given its small dimensions of 5 mm by 15 cm and its flexible material

properties, we find this elastic cable to be conducive to manipulation with the dVRK. The workspace is equipped with an overhead Zivid OnePlus RGBD sensor which captures 1900×1200 RGBD image observations, although only the RGB channels are used in HULK, LOKI, and SPiDERMan inference. The dVRK is calibrated with a standard pixel-to-world camera transformation using the fiducial registration procedure described in [11]. The complete experimental setup has workspace bounds $\boldsymbol{w}_l, \boldsymbol{w}_r$. Each grasp is executed with a $30°$ approach angle to avoid collisions caused by top down grasping.

D. Results

Given the trained networks, we instantiate the proposed policies and baselines and run 12 trials of dVRK cable untangling for each method and each of the four tiers. At the beginning of each trial, a human supervisor manually ties a dense configuration and places it at $\boldsymbol{w}_c$. Then, the dVRK executes the untangling procedure in Algorithm 1 without intervention. A successful trial is defined as one that terminates such that the cable has at most one crossing and no knots. This definition accounts for the natural tendency of the cable to lie in single-crossing stable poses due to its elastic material properties. On an Nvidia GeForce RTX 2080, HULK keypoint inference takes 314 ms, LOKI offset and rotation inference takes 260 ms, and SPiDERMan density comparison takes 18 ms. Node Deletion and Reidemeister moves each take 10 s to execute, while Re-Posing and Wedge Recovery moves take 7 s and 15 s to execute respectively.

We report the untangling success rate and median number of Node Deletion moves, Recovery moves (Re-Posing and Wedged Recovery), and total actions per success in Table I. These results suggest that the combination of HULK, LOKI, and SPiDERMan is effective in performing cable untangling, and exhibits higher empirical success with fewer actions than methods that do not jointly leverage all 3 algorithms. Untangling failures are distributed evenly across initial configuration type for Tiers 1, 3, and 4. For Tier 2, all 3 failures are with the Overhand + Figure 8 initial configuration. For Tier 5, all the

Tier	Policy	Success Rate	Node Deletion Actions	Recovery Actions	Total Actions	Failure Modes				
						A	B	C	D	E
1	H	6/12	6.5	–	6.5	1	3	0	2	0
1	H+L	6/12	5	–	5	2	2	0	2	0
1	H+S	6/12	5	3.5	8.5	0	0	0	6	0
1	H+L+S	**8/12**	6.5	1	8	1	1	2	0	0
2	H	4/12	5.5	–	5.5	1	6	0	1	0
2	H+L	7/12	5	–	5	0	4	0	1	0
2	H+S	8/12	5	0.5	7.5	2	0	0	1	1
2	H+L+S	**9/12**	4	2	6	1	0	1	0	1
3	H	0/12	–	–	–	0	7	0	3	2
3	H+L	6/12	4	–	4	1	5	0	0	0
3	H+S	4/12	8.5	1	9.5	3	0	1	3	1
3	H+L+S	**8/12**	5.5	0	6.5	1	0	1	2	0
4	H	1/12	6	–	6	3	6	0	1	1
4	H+L	5/12	6	–	6	3	4	0	0	0
4	H+S	4/12	10.5	3.5	14	4	0	0	3	1
4	H+L+S	**10/12**	9	3	12.5	2	0	0	0	0
5	H	0/12	–	–	–	1	10	0	1	0
5	H+L	0/12	–	–	–	1	9	0	1	1
5	H+S	3/12	5	2	7	1	0	2	6	0
5	H+L+S	**6/12**	8	2.5	10.5	2	0	0	4	0

TABLE I

PHYSICAL RESULTS: Success rate and efficiency (median number of actions per success) for untangling physical cables containing dense non-planar knots on the dVRK. We categorize initial cable configuration complexity into five tiers: (1) one semi-planar knot seen at train time, (2) two semi-planar knots both seen in training, (3) two semi-planar knots both seen in training and denser than those in (2), (4) one non-planar knot seen in training, and (5) one non-planar knot unseen in training. Untangling experiments are given a horizon of 5 node deletion actions to make progress. If the cable is equally or more dense after 5 untangling actions, we conclude that the configuration is pathological and terminate the trial.

initial configurations have one failure each, with the exception of the Ashley-Stopper knot, which had 2 failures in 2 trials.

We observe 5 failure modes:

(A) gripper collision due to poor keypoint and/or grasp rotation predictions in high-density configurations;

(B) robot gripper jaws wedged between cable segments due to high configuration density and/or failed recovery actions;

(C) premature termination due to density comparison false positives in Equation (3);

(D) termination due to lack of untangling progress;

(E) the cable suddenly springing out of the reachable workspace where Re-Posing moves cannot grasp the cable due to poor keypoint predictions and the cable's elastic material properties.

Across policies that do not leverage SPiDERMan, the most common failure mode is the tendency of the gripper jaws to become wedged between cable segments (B). This failure mode is exacerbated with increasing density and non-planarity, though somewhat alleviated by LOKI's grasp refinement. We note a 22% decrease in HULK's performance on identical initial configurations from experiments in this work compared to those reported in Grannen et al. [7]: 4/12 (33% success) in Table I.2H versus 11/20 (55% success) from [7]. Six of the eight failures were wedged gripper failures (B), which we hypothesize result from the more frayed cable texture in this work compared to Grannen et al. [7]. This may be a consequence of elastic cable overuse over many trials and can be seen in the cables of Figure 5. HULK as reported in [7] is also trained exclusively on semiplanar configurations while the HULK baseline in this paper is trained on non-planar configurations, presenting a harder perceptual task. Lastly, we postulate that 12 trials of untangling experiments

per tier may have high variance, resulting in large differences in performance rates even across similar experimental setups.

SPiDERMan's sensing of configuration density change over time can either mistakenly or correctly detect a lack of untangling progress, resulting in premature (C) or justified rollout termination (D), respectively. Gripper collisions (A) and the cable springing to workspace extremities (E) account for the remaining manipulation-induced errors. Failure (A) is an artifact of high-density configurations and mispredicted grasp locations and orientations, while (E) can cause SPiDERMan's recovery moves to reach robot joint limits, yielding an irrecoverable state.

VII. CONCLUSION

We present LOKI and SPiDERMan, two algorithms that increase precision and recovery for robot untangling of dense, non-planar knots. When used in conjunction with HULK from prior work, LOKI and SPiDERMan anticipate, refine, and recover from errors due to HULK's coarse action planning. We experimentally evaluate the separate and collective impact of HULK, LOKI, and SPiDERMan on physical untangling of dense, non-planar knots across five difficulty tiers. We find that when combined, HULK, LOKI and SPiDERMan enable success rates of 72.8% and 50% on untangling knots of seen and unseen classes respectively and notably achieve a success rate of 66.7% on untangling non-planar knots not considered in prior work. In future work, we will consider cables of varying visual and material properties and explore how depth sensing or other sensing modalities can improve performance. We will also consider other grasp parametrizations and explore how ideas from LOKI and SPiDERMan can be applied more broadly to other deformable manipulation tasks.

REFERENCES

[1] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[2] Mitchell S Cappell. Injury to endoscopic personnel from tripping over exposed cords, wires, and tubing in the endoscopy suite: a preventable cause of potentially severe workplace injury. *Digestive diseases and sciences*, 55(4):947–951, 2010.

[3] Cheng Chi and Dmitry Berenson. Occlusion-robust deformable object tracking without physics simulation. In *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6443–6450. IEEE, 2019.

[4] Bruce R Donald. *Error detection and recovery in robotics*, volume 336. Springer, 1989.

[5] 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.

[6] 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.

[7] 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.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[9] Dominik Henrich and Heinz Wörn. *Robot manipulation of deformable objects*. Springer Science & Business Media, 2012.

[10] 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.

[11] Minho Hwang, Brijen Thananjeyan, Samuel Paradis, Daniel Seita, Jeffrey Ichnowski, Danyal Fer, Thomas Low, and Ken Goldberg. Efficiently calibrating cable-driven surgical robots with rgbd fiducial sensing and recurrent neural networks. *IEEE Robotics and Automation Letters*, 5(4):5937–5944, 2020.

[12] Xin Jiang, Kyong-mo Koo, Kohei Kikuchi, Atsushi Konno, and Masaru Uchiyama. Robotized assembly of a wire harness in a car production line. *Advanced Robotics*, 25(3-4):473–489, 2011.

[13] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[14] Peter Kazanzides, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon P. DiMaio. An open-source research kit for the da Vinci surgical system. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[15] Thomas Lallemand. Blender, June 9 1998. US Patent App. 29/074,220.

[16] Michelle A Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7505–7512. IEEE, 2020.

[17] 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.

[18] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. SoftGym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conf. on Robot Learning (CoRL)*, 2020.

[19] 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.

[20] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conf. on Robot Learning (CoRL)*, 2018.

[21] 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.

[22] Samuel Paradis, Minho Hwang, Brijen Thananjeyan, Jeffrey Ichnowski, Daniel Seita, Danyal Fer, Thomas Low, Joseph E Gonzalez, and Ken Goldberg. Intermittent visual servoing: Efficiently learning policies robust to instrument changes for high-precision surgical manipulation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.

[23] Daniel Sánchez, Weiwei Wan, and Kensuke Harada. Tethered tool manipulation planning with cable maneuvering. *IEEE Robotics and Automation Letters*, 5(2):2777–2784, 2020.

[24] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016.

[25] 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.

[26] 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.

[27] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. In *Proc. Robotics: Science and Systems (RSS)*, 2020.

[28] 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.

[29] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[30] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *Robotics and Automation Letters*, 2021.

[31] 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.

[32] Austin S Wang and Oliver Kroemer. Learning robust manipulation strategies with multimodal state transition models and recovery heuristics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1309–1315. IEEE, 2019.

[33] Mengyuan Yan, Yilin Zhu, Ning Jin, and Jeannette Bohg. Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE Robotics and Automation Letters*, 5(2):2372–2379, 2020.

[34] 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.

[35] Harry Zhang, Jeffrey Ichnowski, Daniel Seita, Jonathan Wang, and Ken

Goldberg. Robots of the lost arc: Learning to dynamically manipulate fixed-endpoint ropes and cables. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.