Robotic Untangling and Disentangling of Cables via Learned Manipulation and Recovery Strategies

Priya Sundaresan
Ken Goldberg, Ed.
Joseph Gonzalez, Ed.

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-121
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-121.html

May 14, 2021

Acknowledgement

(Abridged)

To Prof. Ken Goldberg, for the unmatched support and mentorship over the last three years.

To Brijen Thananjeyan and Ashwin Balakrishna for being my first advisors and role models in research.

To Jennifer Grannen for being a wonderful collaborator and an even better friend.

To Jeffrey Ichnowski and Ellen Novoseller for your selflessness and technical advice.

To all my fellow researchers for your collaboration and wisdom, especially Aditya Ganapathi, Vainavi Viswanath, Michael Laskey, Kevin Stone, Daniel Seita, Danyal Fer, Ryan Hoque, and Minho Hwang.

To Prof. Joseph E. Gonzalez for all the opportunities.

To Niky, Shubha, and Marcie for your endless support and encouragement.

To my friends at Cal for the memories.

To my family for all your love.

Robotic Untangling and Disentangling of Cables via Learned Manipulation and Recovery Strategies

by

Priya Anandhi Sundaresan

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Joseph Gonzalez

Spring 2021

The thesis of Priya Anandhi Sundaresan, titled Robotic Untangling and Disentangling of Cables via Learned Manipulation and Recovery Strategies, is approved:

Chair Ken Goldberg
Ken Goldberg (May 14, 2021 13:14 PDT) Date May 14, 2021

Joseph E. Gonzalez Date May 14, 2021

University of California, Berkeley

Robotic Untangling and Disentangling of Cables
via Learned Manipulation and Recovery Strategies

Abstract

Robotic Untangling and Disentangling of Cables
via Learned Manipulation and Recovery Strategies

by

Priya Anandhi Sundaresan

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ken Goldberg, Chair

Robotic untangling and disentangling of cables such as power cords, ropes, wires, and hoses has applications in decluttering and prevention of tripping hazards in the home and workplace. It can also serve as a prerequisite for downstream tasks such as knot-tying in robot-assisted surgical suturing or wire harness assembly in electrical and automotive manufacturing. However, one-dimensional deformable objects are difficult to model analytically and challenging for robots to visually track and manipulate. This is due to their infinite-dimensional state space, tendency to self-occlude, complex dynamics, and visually homogeneous appearance. Meanwhile, recent advances in deep learning, computer vision, and design of action primitives provide powerful means to handle increasingly complex manipulation tasks. Leveraging these advances, we present novel algorithms for single and multiple cable untangling and disentangling with learning-based perception.

First, we learn to untangle knots that involve multiple overlapping segments in single cables from RGB image observations. We build on a high-level planner [9] that parameterizes loosening actions via image keypoints and introduce a low-level controller for fine-grained manipulation. This controller (1) performs grasp sampling and refinement on cables and (2) monitors untangling progress to perform recovery interventions accordingly. Next, we extend this manipulation stack to disentangle multiple cables.

We evaluate the proposed systems on physical experiments using the da Vinci surgical robot. In single-cable experiments, the robot achieves successful untangling in 68.3% of single cable knots with dense, non-planar starting states across 60 trials, outperforming baselines by 50%. In the multi-cable setting, the robot disentangles cables with 80.5% success, while generalizing across unseen knots and both distinct and identically colored cables.

To the best parents, brother, and grandparents I could ask for.

Contents

List of Figures

List of Tables

Acknowledgments

I will forever be grateful to Professor Ken Goldberg for taking me into AUTOLAB even as a naive sophomore clueless about research, robotics, and computer science. Prof. Goldberg possesses many qualities that have inspired me over the years, but above all else I value his bold creativity, wealth of knowledge, and how much he cares about his students.

A huge thank you to Brijen Thananjeyan for being my first mentor in research and Ashwin Balakrishna for becoming a second advisor to me early on. The support, patience, friendship, and knowledge you have shown me has made the last three years of research a joy and you have undoubtedly shaped the trajectory of my career.

It is rare to find exceptional collaborators and even more rare to find true friendships. For that reason, I am incredibly grateful to have met Jennifer Grannen and will treasure all our fun times shared within and outside the lab.

This thesis would not have been possible without a host of other labmates. Thank you to Jeffrey Ichnowski and Ellen Novoseller for being so selfless with your time and technical advice. Extra thanks to Jeff for coming up with the best acronyms 😄. Thank you also to Aditya Ganapathi, Vainavi Viswanath, Michael Laskey, Kevin Stone, Daniel Seita, Danyal Fer, Ryan Hoque, and Minho Hwang for the wonderful collaborations. To everyone in AUTOLAB, thank you for your presence in the lab. I will miss the engaging discussions, group debugging sessions, game nights, pre-deadline panics made bearable together, and post-deadline celebrations.

Professor Gonzalez, I appreciate you volunteering as a second reader for this thesis. Thank you for your generous guidance and for all of the opportunities through RISELab.

I would like to thank the people who have made me feel immensely supported, fulfilled, and happy outside of research. Niky, Shubha, and Marcie — thank you for being my best friends, roommates, and support system all in one. Thank you to all the friends I have met at Cal for four years of laughs, memories, and adventures.

Finally, this would have been impossible without the unrelenting support of my parents, brother, and grandparents. Amma, Appa, Arun, Pattis, and Thatha — thank you for celebrating my best moments and guiding me through the worst ones.

Chapter 1

Introduction

Having to detangle power cords, wiring, tubing, suturing thread, and shoelaces is an all too common occurrence in homes, offices, concert stages, warehouses, operating rooms, and more [25, 38, 41, 30]. Rope disentangling can also be critical procedure in life-saving systems for search-and-rescue operations [26, 17]. Thus, untangling and disentangling 1D deformable objects, which we broadly refer to as *cables*, can greatly benefit from automation by mobile manipulators.

To realize this, robots will need to tackle knots with increasing complexity and cables in greater quantities, which is the central focus of this thesis. Autonomous untangling/disentangling would be fundamentally impactful as a stand-alone task but can also serve as a prerequisite for any downstream cable arrangement task such as knot-tying or coiling. In Chapter 2, we review relevant literature for the cable manipulation tasks considered in this thesis. There is a large body of related work that focuses on developing analytical controllers for deformable object manipulation. Such methods often require many hand-tuned specifications, significant system engineering, and rely on classical methods for state estimation via visual feature extraction or segmentation. These methods tend to suffer when applied to objects exhibiting self-occlusion, which is characteristic of tightly intertwined cables or complex knots. Meanwhile end-to-end methods have come a long way in developing generalizable, task-agnostic robot learning frameworks. In doing so, they tend to not leverage task/object geometry which can critically inform long-horizon planning for complex tasks like cable manipulation. In this thesis, we draw inspiration from both lines of work to develop an autonomous system for untangling. Like analytical controllers, the proposed system is modularized into individual sub-systems, each targeting a particular manipulation or perception challenge. We employ deep-learning for the parts of the system for which manually engineered solutions would be brittle or for which training supervision is cheaply available. As a result, I have worked on using geometric planning and learned perception to make progress towards robust, autonomous untangling and disentangling of cables throughout the course of my Master's studies.

The first project discussed in this thesis, Chapter 3, considers untangling of a single cable containing *dense*, *non-planar* knots from RGB image observations. Such knots lack space between self-intersections (dense) and contain crossings involving several overlapping cable segments (non-planar). These assumptions actually encompass a large class of knots that are common in our daily lives, from sailing knots to belaying knots in climbing rope to miscellaneous tangles found in headphones, yarn, and shoelaces. This chapter covers [35] which was accepted to RSS 2021. Prior work [9] has studied untangling semi-planar knots in single cables, where each crossing contains at most two overlapping cable segments. In Chapter 3, we extrapolate these methods to more complex knots involving crossings containing up to three cable segments, with fine-grained sensing and recovery policies in place to account for manipulation failures common in open-loop planning. The paper contributes an open-sourced compositional system with a high-level action planner derived from learned keypoints and a low-level controller for grasping and failure recovery. We also perform physical experimental validation on the da Vinci surgical robot.

The second project, discussed in Chapter 4, involves studying how to generalize the above approach to multi-cable knots. We present an iterative, graph-based algorithm for disentangling multi-cable knots instantiated with learned perception and the controller from Chapter 3. This chapter discusses [39] and is currently under review for IROS 2021.

Both projects are the result of a joint effort with my co-authors Jennifer Grannen [35, 39], Vainavi Viswanath [39], and close collaborators Brijen Thananjeyan, Ashwin Balakrishna, Jeffrey Ichnowski, Ellen Novosellor, Minho Hwang, Michael Laskey, Prof. Goldberg, and Prof. Gonzalez. In this thesis, I discuss my individual contributions as well as the larger context surrounding each of these projects.

Chapter 2

Related Work

Deformable object manipulation has garnered significant attention in the robotics research community in recent years [21, 43, 42, 34, 8, 13, 32, 33, 12, 30, 1], but choosing an appropriate state/action space for these tasks remains an open question. Prior work often focuses on generating manipulation plans for a single instance of a deformable object such as a rope, cloth, or bag [34, 8, 9, 13, 32, 4]. Carrying out these plans while appropriately sensing and accounting for possible failure modes is often left unaddressed, as is handling multiple object instances. In this thesis, we focus on learning controllers to robustly execute plans for untangling and disentangling, addressing failure modes specific to the tasks, and planning with awareness of both single and multiple cable instances.

2.1 Deformable Object Manipulation

Perception-driven strategies are standard in multi-step robotic manipulation of deformable objects. One strategy is to learn a representation for the state of a deformable object and leverage it for downstream planning. For instance, Lui and Saxena [23] and Chi and Berenson [3] propose using classical visual feature extraction to estimate the state of a rope as an ordered point set and cloth as a warped 2D grid, respectively. Sundaresan et al. [34] investigate object representation learning via dense object descriptors [31, 7] for rope knot-tying and arrangement, and Ganapathi et al. [8] extend this methodology to 2D fabric smoothing and folding. Chi and Song [4] also propose category-level pose estimation of garments by mapping image observations into a canonical space that fully parameterizes the 3D object state. Such global state representations are difficult to robustly infer in the presence of partial or self-occlusion, common in tangled cables or folded cloths. To bypass this, alternative perception-driven approaches explore learning a latent space representation for cloth [43] that abstracts away visual cues but provides a compressed representation for efficient planning. Another approach is tracking only a small set of task-relevant semantic keypoints for untangling [9] or dynamic cloth unfolding [10]. Applying aforementioned single-instance perception models out-of-the-box to multiple deformable object instances within the same scene would also require robust instance segmentation. This is nontrivial to learn or analytically infer in cluttered multi-cable settings. Thus, we learn a representation of cables via keypoints in a way that generalizes to single (Chapter 3) or multiple (Chapter 4) cable instances and circumvents the need to infer full object geometry.

Other methods learn visuomotor policies end-to-end instead of taking a perception-driven approach. Imitation learning [32], video prediction models [13], model-free reinforcement learning (RL) [20], and model-based RL [24] have proven effective for goal-conditioned fabric manipulation. Learning inverse dynamics models of deformable objects and leveraging them towards planning has also yielded promising results on template shape matching and knot-tying with rope [27]. Zhang et al. [44] learn to directly regress robotic configurations in joint space for robotic vaulting of cables over obstacles. While these approaches provide flexible frameworks that generalize across many tasks, they do not take advantage of geometric structure in the problem or provide high-precision solutions for manipulation. This makes them difficult to apply to tasks requiring extreme dexterity and geometric reasoning, such as cable untangling and disentangling. Thus, we use geometry-aware perception for high level planning and employ a learned controller only for precise grasping and failure recovery.

Lastly, several recent works have also learned deformable manipulation policies in simulation either for warm-starting learning in the real world [44] or directly deploying learned strategies via domain randomization [33, 34, 43, 8, 13, 33]. For tasks that require global reasoning about highly complex objects such as untangling and disentangling, we find that physical data is critical as it is difficult to realistically model the full state space of cables in simulation. However, local features of knots are feasible to represent with simple geometric models and simulate. As a result, we learn to perceive cables globally from supervised real-world examples and learn to infer structure about single and multi-cable knots using synthetic data and self-supervision.

2.2 Cable Untangling

Prior work has focused on untangling single cables containing loose [23] or *dense* [9] knots, where dense knots are those that lack space between adjacent cable segments. In particular, Lui and Saxena [23] present the first published study of robotic cable untying to our knowledge. Given RGB-D observations, they use manually specified visual features and particle filtering to approximate the state of a tangled cable by a linear graph. The graph consists of vertices that denote crossings/endpoints and edges corresponding to adjoining cable segments. This technique relies on adequate space amongst adjacent cable segments for segmentation, which is not guaranteed in dense single or multi-cable knots. In subsequent work, Grannen et al. [9] learn to infer the graphical cable representation of [23] from keypoints and apply it towards untangling of *semiplanar* knots, extending to a denser class of knots than [23]. Semiplanar knots consist of crossings involving at most two overlapping cable segments. The controllers in both [23, 9] are derived from analytical motion primitives that can lead to repeated failures which prolong the task horizon (missed grasps, poor loosening actions) or irrecoverable states (knocking the cable out of reach, cable wedged between grippers). In this thesis, we first aim to adapt method to a more challenging class of dense, *non-planar* knots. Non-planar knots relax the semiplanar assumption to handle up to three overlapping cable segments. In doing so, we employ coarse-to-fine refinement strategies for robust cable grasping and preemptive failure avoidance, drawing from recent work in designing precise controllers for surgical peg transfer [28] and peg insertion [19, 15]. Next, we propose closed-loop manipulation with recovery strategies. Prior work has explored contact-rich robotics tasks with error recovery [40, 6, 37] but does not make use of task geometry which is critical in cable untangling and disentangling. Finally, we extend the graphical formulation of [23, 9] to the multi-cable setting and apply the above two strategies towards disentangling of multiple cables.

Chapter 3

Learning to Untangle Dense Non-Planar Knots in Single Cables

3.1 Individual Contributions

This chapter is adapted from our paper "Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies" which will appear at RSS '21. This is joint work with Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Ellen Novoseller, Jeffrey Ichnowski, Minho Hwang, Michael Laskey, Prof. Ken Goldberg, and Prof. Joseph E. Gonzalez.

In this chapter, my contributions include developing learning-based (HULK) and analytical (SPiDERMan) components of the perception systems, creating simulation environments (LOKI), implementing several motion primitives involving grasping, pinning, and pulling detailed in Section 3.4, and running physical experiments (Section 3.5).

First, I would like to acknowledge Jennifer Grannen, who developed SPiDERMan's progress sensing functionality, split the implementation of several motion primitives, perceptual components, and experiments on the da Vinci with me (Section 3.4), and conducted analyses of alternative perception and manipulation methods in parallel. She is also responsible for Figure 3.5 and the digital knot drawings of Figure 3.2.

Brijen Thananjeyan, Ashwin Balakrishna, Jeffrey Ichnowski, and Ellen Novosellor played critical roles in scoping this project, formalizing the algorithmic contributions, and providing helpful feedback on the submitted version of the paper and this manuscript.

Minho Hwang's assistance with operation of the da Vinci surgical robot was invaluable, from helping us interface with and calibrate an overhead RGB-D sensor to providing expertise on tuning gripper parameters for grasping and manipulation.

Finally, I am grateful for the useful recommendations from Michael Laskey of TRI, most notably on designing the perception and simulation systems. I also thank Professors Ken Goldberg and Joseph E. Gonzalez for their valuable insights and suggestions throughout this project and on much of the work leading up to it.

3.2 Problem Statement

The contribution of this chapter is an iterative, bilateral manipulation policy for untangling dense, non-planar knots in a singular cable. The cable is deemed fully untangled if it reaches a crossing-free state.

Assumptions

We assume access to a bilateral manipulator capable of pinning and pulling cable segments simultaneously using two different arms. We also assume access to RGB image observations from an overhead camera mounted a fixed distance above the surface of manipulation. While we do not require the color of the cable or workspace surface to be pre-specified, we assume the two are sufficiently distinguishable. Within the workspace, we assume that the cable is within the kinematically feasible and reachable region of at least one robot arm, and that cable thickness is less than the gripper's maximum opening width.

This ensures that the robot is able to hold and pull cable segments with opposing arms. We find that due to the long task horizon, it is highly possible for the cable to be pushed to workspace extremities. Thus, these assumptions also allow for at least one arm to always reposition the cable in a central place where it is then within reach of both arms. To instantiate recovery and straightening actions, we also define three points, $\boldsymbol{w}_l$, $\boldsymbol{w}_c$, and $\boldsymbol{w}_r$, corresponding to the left, center, and right-most points in the workspace.

In this chapter, we make the following assumptions about cable starting states:

- *limited non-planarities*: we arrived at this assumption by first relaxing that of prior work [9]. While Grannen et al. [9] consider untangling of *semi-planar* cables, where each crossing involves at most 2 cable segments, we allow each self-intersection to involve up to 3 segments (*non-planar*).

- *visible endpoints*: Initially, both cable endpoints are unoccluded and in the field of view of overhead RGB images for planning purposes. We denote the endpoints as left and right based on thresholding their pixel coordinate x-values, breaking ties arbitrarily.

As humans, a common way to unknot a tangled cable is to trace from one endpoint up until the first crossing encountered, undo it, and repeat. The endpoint visibility assumption is based on this observation and provides a convention for action selection — we always undo the crossing relative to the right-most endpoint.

Configuration Graph

We formalize the problem of cable untangling by extending the graphical representation from [23] to handle non-planarities and use this as a representation for planning via an algorithmic supervisor BRUCE (Sec. 3.3). The deployed untangling system does not explicitly reconstruct or operate on this graph. Rather we use a combination of learned algorithms

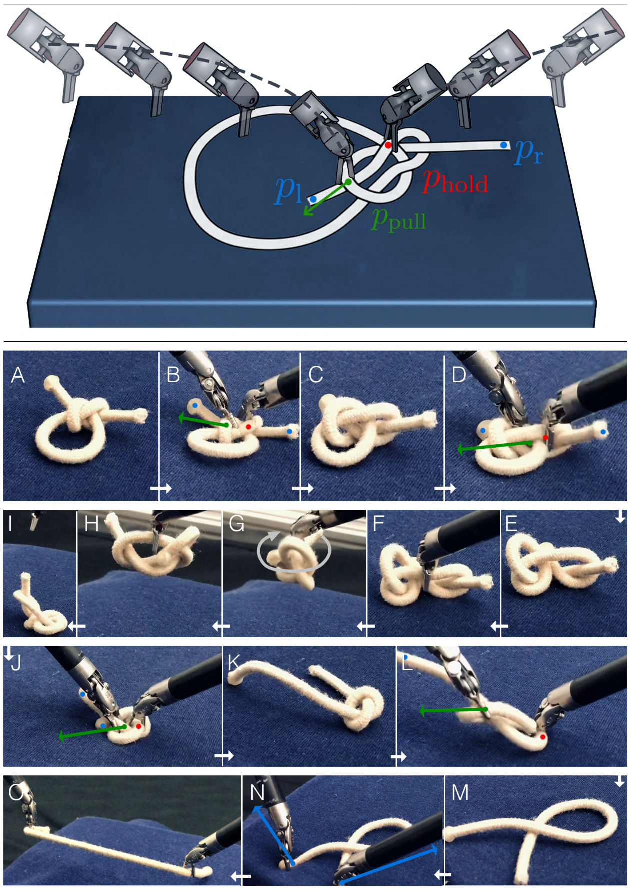

Figure 3.1: Six action sequence to untangle a non-planar square knot. We demonstrate the composition of **HULK** (Hierarchical Untangling from Learned Keypoints), **LOKI** (Local Oriented Knot Inspection), and **SPiDERMan** (Sensing Progress in Dense Entanglements for Recovery Manipulation), three algorithms 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).

—HULK (Sec. 3.3), LOKI, and SPiDERMan (Sec. 3.4)— to intelligently sense the relevant portions of the graph as features from RGB image observations and execute untangling actions accordingly.

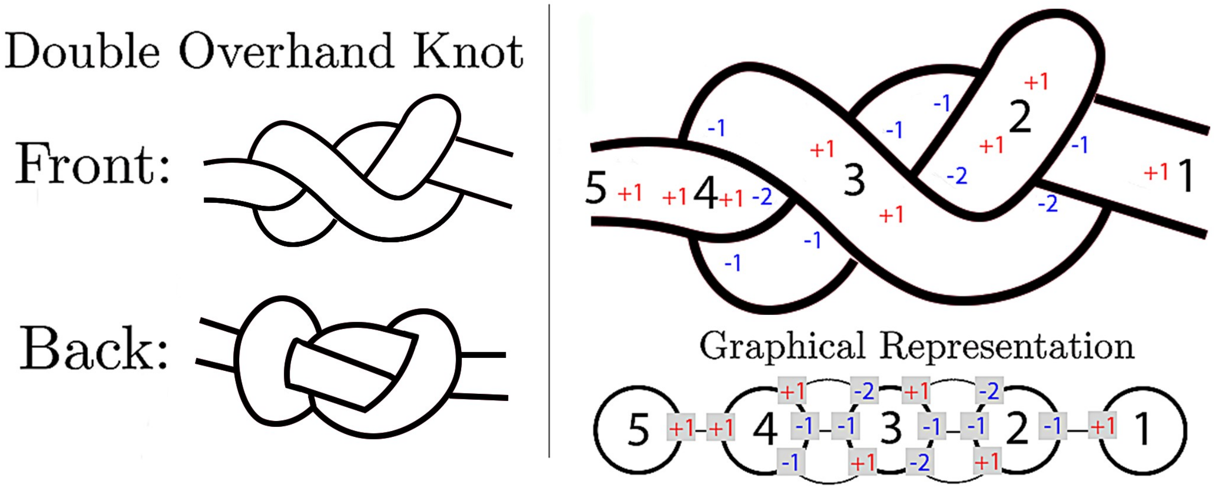

Figure 3.2: **Non-Planar Graphical Representation:** We model cable configuration with a graph as in Lui and Saxena [23] and extend this representation to non-planar configurations with an updated annotation method in Equation (3.1). We visualize a dense double overhand knot from two views to illustrate non-planarity, along with its corresponding annotations and graphical representation.

Prior work has explored representations such as Bezier curves and splines to represent cables [34]. For tangled cables, we find that a graph structure streamlines specifying the key features of a cable's topology, particularly crossings and nested levels of self-occlusion. We use an undirected graph $G = (V, E)$ to model the state of a cable, where vertices $v \in V$ specify cable endpoints and crossing locations and edges $e = (u, v) \in E$ for $u, v \in V$ represent cable segments joining vertices. In the remaining sections, we will refer to a "vertex" as a "node." The left and right cable endpoints are denoted by nodes v_l and v_r, respectively. For any node $v \in V$ and edge $e \in E$ adjacent to v, we annotate the graph with $X(v, e) \in \{+1, -1, -2\}$ (Equation 3.1) to designate the cable segment hierarchy at a node:

$$X(v,e) = \begin{cases} +1 & \text{if } v \text{ is an endpoint or if } e \text{ crosses} \\ & \text{above all other edges at } v \\ -1 & \text{if } e \text{ crosses below only one edge at } v \\ -2 & \text{if } e \text{ crosses below two other edges at } v. \end{cases} \tag{3.1}$$

We parameterize an *under-crossing* by 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 physical cable. By this notation, the cable segment represented by e_i, e_j is occluded by one or two other cable segments, in line with the non-planarity assumption. 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 adjoining in the physical cable.

Under the non-planar assumption, each node has degree ≤ 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. We will later revisit this property to define a termination condition for the task when $|V| = 2$.

Action Space

At each time t, each gripper performs either a holding, pulling, or rotating action. We define all actions with respect to the 2D image frame. Due to the high degrees of freedom in a cable, there are many feasible action spaces and we specify actions in image space for simplicity.

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 z-axis rotation $\theta_{t,l}$, 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 3.2:

$$\begin{aligned} \boldsymbol{a}_{t,l} &= (x_{t,l}, y_{t,l}, \theta_{t,l}, \Delta x_{t,l}, \Delta y_{t,l}, \Delta\theta_{t,l}, \mathbb{1}_{\text{grasp}}) \\ \boldsymbol{a}_{t,r} &= (x_{t,r}, y_{t,r}, \theta_{t,r}, \Delta x_{t,r}, \Delta y_{t,r}, \Delta\theta_{t,r}, \mathbb{1}_{\text{grasp}}). \end{aligned} \tag{3.2}$$

This action space extends that of Grannen et al. [9], which only parameterizes grasp points and pull vectors.

3.3 Preliminaries

Grannen et al. [9] introduced BRUCE and HULK, two algorithms for untangling. In the remainder of this chaper, we discuss the relevant algorithmic modifications we make to both to accommodate non-planar knots. While BRUCE focuses on formalizing an algorithm for untangling based on the graph from Section 3.2, HULK discusses how to practically instantiate BRUCE from RGB images without reconstructing or employing the graph directly. Together, they lay the foundation of the algorithms discussed in the remaining parts of Chapters 3 and 4.

BRUCE: Basic Reduction of Under-Crossing Entanglements

Grannen et al. [9] propose Basic Reduction of Under-Crossing Entanglements (BRUCE), an algorithm to iteratively untangle semiplanar knots which operates on a cable's graph representation. BRUCE uses two manipulation primitives adapted from Lui and Saxena [23]: (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 cable slack from the corresponding under-crossing. Both of these primitives are practical instantiations of topological moves in knot theory [22].

First, BRUCE takes a Reidemeister move to disambiguate the cable state. This step straightens any rope slack that is not involved in knot. Next, BRUCE alternates between Node Deletion and Reidemeister moves. Node Deletion moves sequentially delete crossings and Reidemeister clear any accumulated cable slack to simplify the configuration and expose any remaining knots. In the context of the graph, Node Deletion moves locate the under-crossing edge (labeled -1) exiting the right-most node v_r, while holding a corresponding over-crossing edge ($+1$) at v_r in place.

To apply to non-planar knots, we extend Node Deletion moves to operate on under-crossing edges labeled as both -1 and -2 instead of just -1. 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. Each action reduces $|E|$ and $|V|$ by at least 2 and 1, respectively, terminating with $|V| = 2$: a fully linear cable with no crossings and two exposed endpoints.

HULK: Hierarchical Untangling from Learned Keypoints

In RGB image observations, the ground truth cable graph representation is not directly observable. One option would be to approximate the cable graph from images, and plan untangling actions accordingly. Instead, Grannen et al. [9] propose Hierarchical Untangling from Learned Keypoints (HULK). HULK directly predicts keypoints from images to execute BRUCE's manipulation primitives, bypassing the need for graph reconstruction. HULK learns a mapping from an image to four Gaussian heatmaps centered at task-relevant keypoints, $f : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times4}$. The argmaxes 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 right endpoint. This is the crossing for which, by tracing from the right endpoint towards the left endpoint along the cable length, the segment of cable contiguous with the right endpoint slips underneath a different part of the cable. This is a key convention used to disambiguate action selection, as there are many feasible crossings to be undone otherwise. HULK does not explicitly trace any part of the cable but rather learns contextual information to directly regress the right-most under-crossing as a keypoint.

In this chapter, we adapt the HULK training procedure from Grannen et al. [9], 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 [9] and train HULK on 200 pairs of images and hand-annotations for non-planar configurations. Initially, we focused on learning pin and pull keypoints entirely from self-supervision via a cable simulator developed in Blender 2.80 [18] (`https:`

//github.com/priyasundaresan/blender-rope-sim). The simulator renders photorealistic and domain-randomized cable images and saves ground truth annotations with which to train HULK. In practice, we found two main challenges with this approach: (1) developing a cable simulator with believable physics and (2) transferring the learned representation from simulation to real. Because of these challenges, we instead pivoted to training HULK entirely from real data while using data augmentation techniques to reduce the supervision burden. The full pipeline for labelling 200 pairs of images takes in total about about 20 minutes. Then, we apply various data augmentation techniques including image adjustments and affine transformations to generate a training dataset of 3,500 examples.

As in [9], we use a Gaussian standard deviation of 8 for the heatmap training data centered about each hand-labelled keypoint to train the model.

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 distance n to slacken the cable. In the work of Grannen et al. [9], 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 $\hat{\theta}_{\text{hold}} = \hat{\theta}_{\text{pull}} + 90°$:

$$\begin{aligned} \boldsymbol{a}_{t,l} &= (\hat{p}_{\text{pull},x}, \hat{p}_{\text{pull},y}, \hat{\theta}_{\text{pull}}, n_x, n_y, 0, 1) \\ \boldsymbol{a}_{t,r} &= (\hat{p}_{\text{hold},x}, \hat{p}_{\text{hold},y}, \hat{\theta}_{\text{hold}}, 0, 0, 0, 1). \end{aligned} \tag{3.3}$$

Note that we instead learn the grasp orientations $\hat{\theta}_{\text{pull}}, \hat{\theta}_{\text{hold}}$ instead of using heuristic approximations, further discussed in Section 3.4. 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 $\hat{\boldsymbol{p}}_r$:

$$\begin{aligned} \boldsymbol{a}_{t,l} &= (\hat{p}_{l,x}, \hat{p}_{l,y}, \hat{\theta}_l, w_{l,x} - \hat{p}_{l,x}, w_{l,y} - \hat{p}_{l,y}, 0, 1) \\ \boldsymbol{a}_{t,r} &= (\hat{p}_{r,x}, \hat{p}_{r,y}, \hat{\theta}_r, w_{r,x} - \hat{p}_{r,x}, w_{r,y} - \hat{p}_{r,y}, 0, 1). \end{aligned} \tag{3.4}$$

3.4 Methods

The main contribution of this chapter is two new algorithms, LOKI (Local Oriented Knot Inspection) and SPiDERMan (Sensing Progress in Dense Entanglements for Recovery Manipulation). These are combined to learn a robust controller to increase robustness of high-level manipulation plans provided by HULK (Section 3.3).

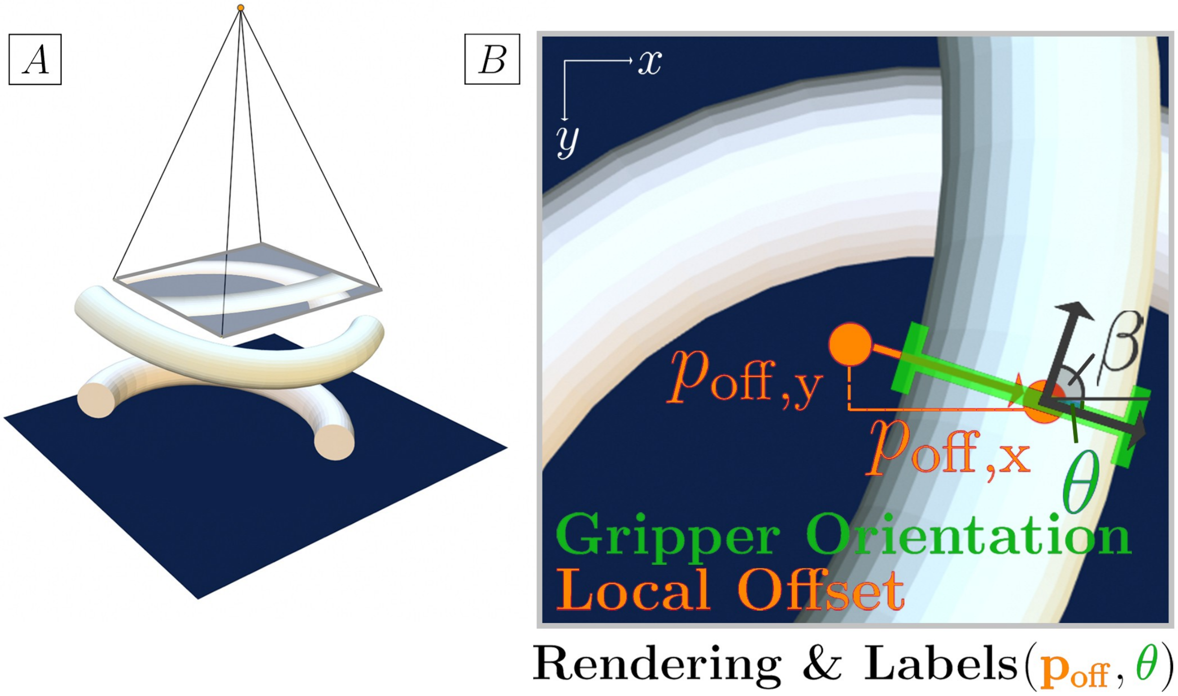

Figure 3.3: **LOKI Dataset Generation**: LOKI learns precise cable grasping from simulation. We simulate cable crossings in Blender 2.80 as overlapping, warped cylindrical meshes (A) and render RGB cable crops (B). Each 200 × 200 pixel crop is annotated with a ground truth optimal grasp angle θ orthogonal to the topmost cylinder orientation β and a local offset $(p_{\text{off,x}}, p_{\text{off,y}})$ to recenter the crop along the cable width. At test time, a real cable crop centered at a HULK keypoint is passed in and LOKI refines the keypoint via recentering and estimates the best gripper orientation for robust grasping.

LOKI: Local Oriented Knot Inspection

Our insight is that fine-grained grasp planning for dense, non-planar knots requires nuance that neither HULK's primitives nor heuristic/analytical methods for cable grasp planning [34, 9] offer. Thus, we introduce a low-level controller called LOKI: Local Oriented Knot Inspection. 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 common to HULK (Section 3.3). Given a locally-cropped cable image, centered at one of the four HULK keypoints $\hat{\boldsymbol{p}} \in \{\hat{\boldsymbol{p}}_l, \hat{\boldsymbol{p}}_r, \hat{\boldsymbol{p}}_{\text{pull}}, \hat{\boldsymbol{p}}_{\text{hold}}\}$, LOKI outputs (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 [11] backbone that learns a mapping $g : \mathbb{R}^{200\times200\times3} \mapsto ([0, 180], \mathbb{R}^{200\times200})$. 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.

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 implement and collect but readily accessible from synthetic meshes. We implement a simulation environment in Blender 2.80 [18] that models cable self-intersections as two overlapping cylinder meshes slightly warped to emulate the curvature of real cables. We generate varied training data by positioning a synthetic overhead camera above, but not centered on, the topmost cylinder. Before warping, 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 $90^\circ + \beta$ 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 (u, v), where $\mathcal{I}$ is a 2×2 identity matrix.

Initially, we only focused on the problem of just regressing the gripper orientation from synthetic cable crops. However, the observation that HULK keypoints are occasionally off-center w.r.t. cable width, and that the cylindrical centers are available in sim, prompted the addition of (u, v) as a refined, re-centered keypoint output to the model.

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 $I \in \mathbb{R}^{640\times480\times3}$ 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:

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

where $\gamma = 60/200$ is a downscale factor to account for crop resizing and $(u_c, v_c) = (100, 100)$ is the upscaled crop center. 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 all grasps across the manipulation primitives.

SPiDERMan: Sensing Progress in Dense Entanglements for Recovery Manipulation

With an implementation of HULK alone, as described in Section 3.3, we observe 4 main manipulation failure-modes: (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. Our insight is that many of these failures are either recoverable, avoidable, or at least able to be sensed and planned around. Thus, we propose SPiDERMan. SPiDERMan uses both learned and analytical methods to sense untangling progress and common manipulation failures and employs two novel manipulation primitives to avoid and recover from these failures. In this section, we discuss SPiDERMan's detection of failure modes and response via two manipulation primitives for recovery.

Detection

SPiDERMan employs a hybrid of learned and analytical perception in its approach. To address failures (1) and (2), SPiDERMan compares workspace image observations $I_{t'} \in \mathbb{R}^{640\times480\times3}$ before and after executing actions to sense loosening progress or lack thereof. In this framing, t' denotes the number of Node Deletion actions executed thus far.

In practice, we find that specifying a continuous metric or reward function for untangling progress is difficult. One such metric could be a measure of cumulative area between crossings which directly quantifies density, but learning such a metric re-introduces the challenges of state estimation and segmentation. Thus, we instead learn to classify whether progress is being made with simple pairwise comparisons across the current observation and a history of observations. SPiDERMan uses a ResNet-18 backbone and learns a classifier $d : \mathbb{R}^{640\times480\times6} \mapsto \{0,1\}$, trained with a binary cross-entropy loss.

At test time, $d(I_1, I_2)$ uses a simple binary decision rule with a threshold of 0.5 to classify whether image I_1 corresponds to a denser (1) or less dense (0) state than I_2. If SPiDERMan classifies two consecutive actions as not making progress, given by the following condition:

$$d(I_{t'}, I_{t'-1}) \text{ and } d(I_{t'-1}, I_{t'-2}), \tag{3.5}$$

we conclude that the configuration is pathological and plan an intervention. This allows SPiDERMan to track untangling progress over *multiple* actions, allowing for an adaptive termination condition instead of a fixed action limit:

$$d(I_{t'}, I_{t'-5}) \text{ or } d(I_{\text{ref}}, I_{t'}) \tag{3.6}$$

Condition (3.6) evaluates to true when no progress has occurred over the last 5 actions, or if the cable reaches a fully-untangled state. This allows untangling to continue for as long as progress is being achieved.

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 3.4) to prevent termination false negatives from a biased preference of I_{ref}. We found that using a fully straightened cable as I_{ref} would delay untangling termination longer than necessary, since SPiDERMan is disinclined to classify any current cable state as *more* untangled than a fully linearized configuration.

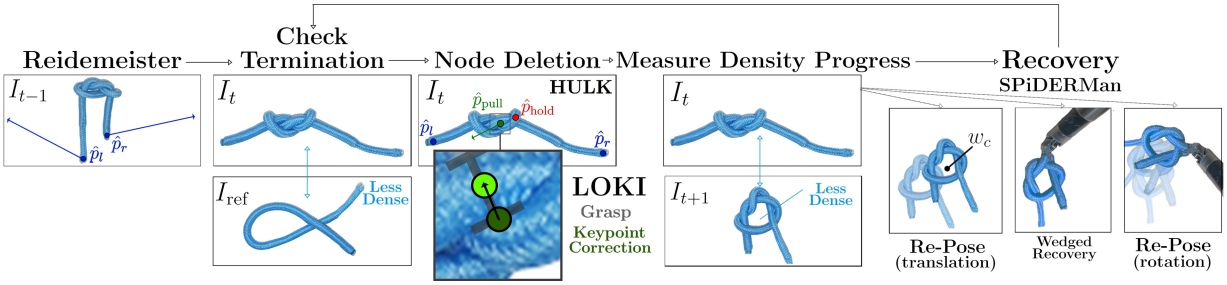

Figure 3.4: **System Overview**: We illustrate Algorithm 1 used to perform the cable untangling shown in Figure 3.1. Starting from the left, we untangle a cable from a non-planar initial configuration following the actions outlined by BRUCE: one initial Reidemeister move followed by successive Node Deletion 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 loop.

SPiDERMan also applies analytical contour-based perception to implement recovery manipulation primitives, relying on an earlier assumption that the cable and workspace are separable in terms of color. Given a 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. [36] in OpenCV [2]. 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 (3.7) 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:

$$\min\{||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{g}_l||, ||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{g}_r||\} < 20 \text{ px}. \tag{3.7}$$

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 $\hat{\boldsymbol{p}}_{\text{c}}$ is within a hand-tuned 200 px threshold from the predefined workspace center $\boldsymbol{w}_c$:

$$||\hat{\boldsymbol{p}}_{\text{c}} - \boldsymbol{w}_c|| > 200 \text{ px}. \tag{3.8}$$

Recovery

We define two novel manipulation primitives to recover once a failure is detected: (1) **Re-Posing** moves that 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. [9] 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, the right gripper then rotates by 180° to re-pose the configuration before releasing the grasp. We define a *Re-Posing move (Rotation)* to include this rotation, while a *Re-Posing move (Translation)* does not:

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

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 $\hat{\boldsymbol{p}}_{\text{hold}} + \hat{\boldsymbol{p}}_{\textbf{hold,off}}$ 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 $\boldsymbol{w}_l$:

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

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 full untangling algorithm, combining HULK (Section 3.3), LOKI (Section 3.4) and SPiDERMan (Section 3.4).

Algorithm 1 Untangling with LOKI and SPiDERMan

```
Input: RGB image of cable
Reidemeister move with $\underbrace{\hat{\boldsymbol{p}}_r}_{\text{HULK}} + \underbrace{\hat{\boldsymbol{p}}_{r,\text{off}}, \hat{\theta}_r}_{\text{LOKI}}, \hat{\boldsymbol{p}}_l + \hat{\boldsymbol{p}}_{l,\text{off}}, \hat{\theta}_l$
while NOT Eq. (3.6) do                                                  ▷ Termination check
    while Eq. (3.7) do                                                  ▷ Wedge check
        Wedge Recovery with $\hat{\boldsymbol{p}}_{\text{hold}} + \hat{\boldsymbol{p}}_{\text{hold,off}}, \hat{\theta}_{\text{hold}}, \boldsymbol{w}_c, \boldsymbol{w}_{l/r}$
    if Eq. (3.5) then                                                   ▷ Poor actions check
        Re-Posing (Rotation) with $\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{p}}_{c,\text{off}}, \hat{\theta}_c, \boldsymbol{w}_c$
    if Eq. (3.8) then                                                   ▷ Leaving workspace check
        Re-Posing (Translation) with $\hat{\boldsymbol{p}}_c + \hat{\boldsymbol{p}}_{c,\text{off}}, \hat{\theta}_c, \boldsymbol{w}_c$
    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
```

3.5 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.

Overview of Policies

We compare (1) the proposed method (Alg. 1) synthesizing HULK, LOKI, and SPiDERMan (H+L+S) against three baselines that ablate the effectiveness of LOKI or SPiDERMan individually and together: (2) HULK alone (H), (3) HULK with only LOKI (H+L), and (4) HULK with only SPiDERMan (H+S). Notably, baselines (2) and (4) employ analytical grasp planning described in Section 3.3. 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. 3.3. Baselines (2) and (3) do not use any recovery manipulation primitives. All policies employ the termination condition given by Eq. (3.6).

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 3.4. We also train LOKI from 3,000 200×200 synthetically-generated cable crops.

Tiers of Difficulty

All policies are tested on novel *dense* knots characterized by five tiers of difficulty, defined by the knot classes encountered and whether these knot classes are seen in training:

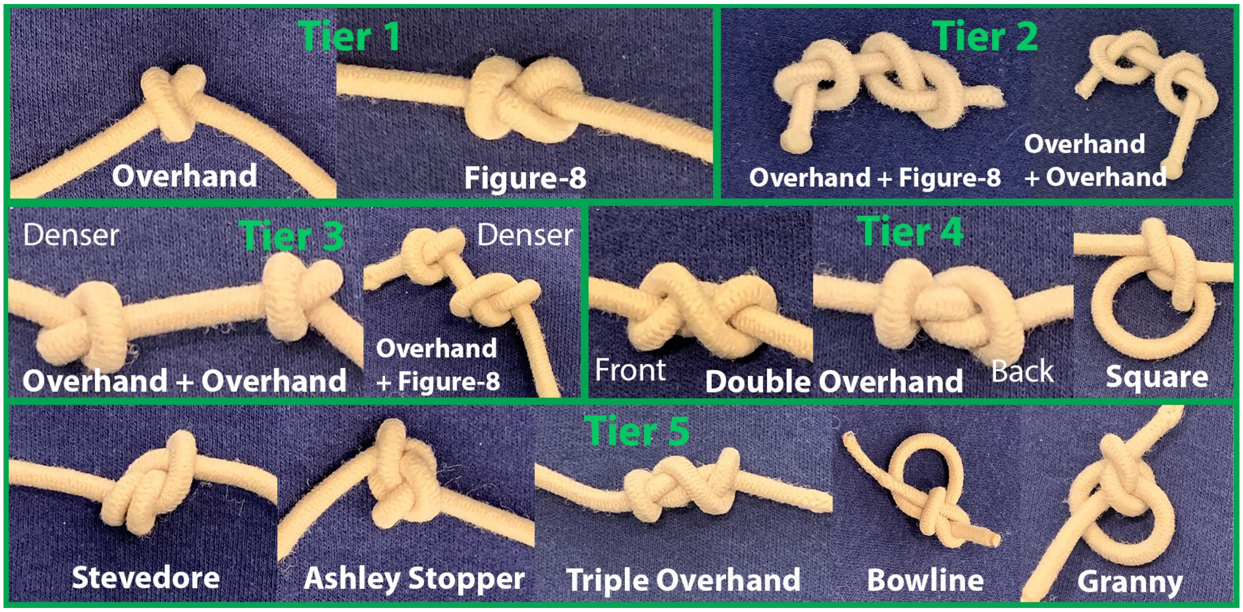

Figure 3.5: **Difficulty Tiers:** Examples of dense initial configurations used in physical untangling experiments. We categorize configurations into 5 tiers of difficulty based on configuration complexity (semi-planar vs. non-planar), knot density (qualitatively measured), number of knots, and whether these knots were present in the training data. Tiers 1 and 2 are semi-planar and are drawn from [9], while Tier 3 contain semi-planar configurations denser than those in [9] and Tier 4 and 5 consist of non-planar knots.

Tier 1: A single, novel semiplanar knot (figure-eight or overhand) while knots of this class were present in training data, as in [9]

Tier 2: Two semiplanar knots (figure-eight or overhand) while knots of this class were present in training data, as in [9]

Tier 3: Two semiplanar knots (figure-eight or overhand) while knots of this class were present in training data, denser than those seen in [9]

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

Tier 5: A single non-planar knot (stevedore, bowline, ashley stopper, granny, or heaving line) while 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, and single double overhand and 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 knots not present in the training data.

Experimental Setup

We execute all experiments using the bilateral dVRK robot [16] 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 5mm by 15cm 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 procedure described in [14]. The complete experimental setup with 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.

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 3.1: **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 four tiers: (1) one semi-planar knot seen at train time, (2) two semi-planar knots both seen in training, (3) one non-planar knot seen in training, and (4) 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.

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 take 10 s to execute each, 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 3.1. 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.

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 (3.5);

(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. 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.

3.6 Discussion

In this chapter, we present the problem of untangling dense, non-planar knots in singular cables. Our approach builds on high-level action planning at the level of image keypoints (HULK) and instantiates a low-level controller (LOKI) to enable more secure grasping between cable segments. Finally, we address the failures in open-loop planning via SPiDERMan. SPiDERMan uses a combination of learned and heuristic perception strategies to counteract common manipulation failures. These include dislodging a cable that is wedged between grippers, re-centering a cable within the workspace, and placing the cable in a new stable pose upon sensing a lack of untangling progress to reveal new knot geometry.

With progress towards single cable untangling, one natural question is: how can we extend this approach to multiple cables? We consider this problem and the new challenges it introduces in Chapter 4.

Chapter 4

Learning to Disentangle Multiple Cables

4.1 Individual Contributions

This chapter covers our paper "Disentangling Dense Multi-Cable Knots" as submitted to IROS '21. This is joint work with Vainavi Viswanath, Jennifer Grannen, Brijen Thananjeyan, Ashwin Balakrishna, Ellen Novoseller, Jeffrey Ichnowski, Minho Hwang, Michael Laskey, Prof. Ken Goldberg, and Prof. Joseph E. Gonzalez.

In this project, my contributions include algorithmic and system design (Section 4.2), full-stack integration of perception and manipulation on the da Vinci surgical robot, infrastructure for experiments, and running physical trials (Section 4.4).

Building a robust system for cable disentanglement was a massive team effort that required the ability to explore and iterate on many ideas in parallel. This was made possible by my collaborators Vainavi Viswanath and Jennifer Grannen. Vainavi took the lead on the perception stack including HULK and detection of algorithmic termination (3.4). This is especially commendable considering the extent of what she managed to accomplish while working remotely due to COVID. She also created the figures in this chapter (Figures 4.1, 4.2, 4.3, 4.4). Jennifer Grannen's efforts on manipulation primitive design along with her work on deployment of perception systems was crucial to the development of this system. This chapter built substantially upon manipulation and perception stacks she developed in [9].

Brijen Thananjeyan, Ashwin Balakrishna, Ellen Novoseller, and Jeffrey Ichnowski played essential roles in this project from its initial scoping down to the experimental design. They were especially helpful in formalizing the topological graphical representation of multi-cable knots. They also provided very helpful suggestions on designing the primitives and order of operations for the manipulation stack required to tackle this complex task.

We thank Minho Hwang immensely for all of his expertise on the da Vinci surgical robot platform, which was foundational to implementing the systems of Chapters 3 and 4.

Professor Joseph E. Gonzalez provided one of the first motivating examples for autonomous cable disentangling in his discussion of wire harness assembly, and gave helpful feedback leading up until paper submission. Michael Laskey further assisted with project scoping in the context of home robotics, and Professor Ken Goldberg provided meaningful feedback at every stage of this project.

4.2 Problem Statement

In this chapter, we consider the setting of a bilateral robot attempting to disentangle a knotted configuration of $n > 1$ cables from RGB images, instead of a singular cable as discussed in Chapter 3. The objective is to remove all intra-cable and inter-cable crossings in the scene with a minimal number of actions while recognizing when cables become fully untangled and setting them aside.

Assumptions

We make the following assumptions, which naturally extend those of Chapter 3 to the multi-cable setting:

- *cables distinguishable* : we assume all cables are color-separable from the background but need not be distinguishable from each other;
- *visible endpoints*: at least two endpoints are visible in the initial cable configuration;
- *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.
- *non-planarities*: we allow non-planar knots, where more than two cable segments can be involved in each intersection.

Workspace Definition

We define a Cartesian (x, y, z) coordinate frame for the workspace with $\boldsymbol{w}_l$, $\boldsymbol{w}_r$, $\boldsymbol{w}_c \in \mathbb{R}^3$ denoting the left, right, and center points of the workspace for planning purposes. Next, we assume access to a bilateral robot and a flat manipulation surface that lies in the xy-plane. The primary difference in this setting is that rather than containing only a single cable, the scene now contains $n > 1$ cables that can be knotted or twisted together. The addition of several cables requires careful consideration of three new challenges: (1) slack management in the workspace, (2) appropriate handling of both intra and inter-cable crossings, and (3) separation of cables in addition to untangling.

To address (3), we assume the existence of a termination area to the right of the manipulation workspace, where we relocate cables as they become fully disentangled. The

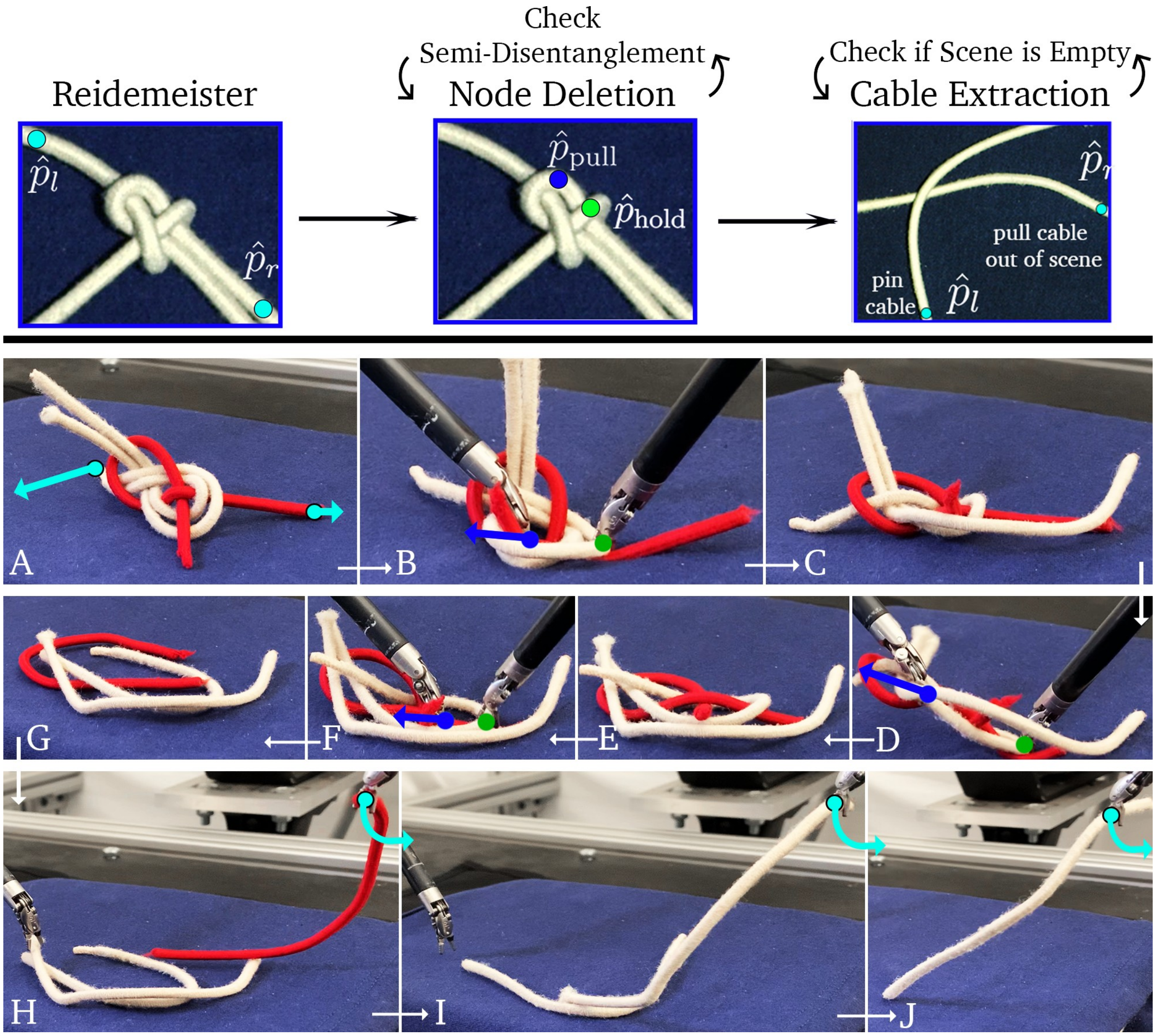

Figure 4.1: **Overview of IRON-MAN**: IRON-MAN (Iterative Reduction Of Non-planar Multiple cAble kNots) is an algorithm for disentanglement of several knotted cables. We present a sequence of moves planned by IRON-MAN on a three-cable Carrick Bend knot. Following an initial Reidemeister (straightening) move (A) which pulls opposing cable endpoints apart, IRON-MAN takes several Node Deletion (loosening) moves (B-C, D-E, F-G) to reduce inter and intra-cable crossings. Finally, we take three Cable Extraction (removal) moves (H-J) to isolate and remove each cable.

termination area is centered at the point $\boldsymbol{p}_{\text{term}} \in \mathbb{R}^3$ and is within reach of 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.

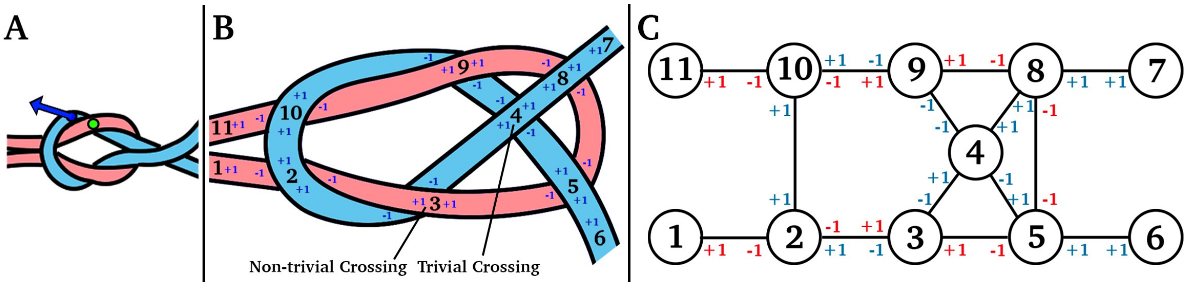

Figure 4.2: **Graph Representation**: Provided a dense initial square knot (A), we take a Node Deletion move specified by hold (green) and pull (dark blue) keypoints, yielding a looser configuration shown in (B). We use a graphical abstraction to model the state of intertwined cables, extending previous work on modelling single-cable configurations as in Chapter 3. 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 significantly loosening the configuration.

The toplogy of crossings within and amongst cables is not directly observable, and must instead be inferred from RGB images. At time t, each manipulation action is planned directly from an RGB image observation I_t.

Configuration Graph

BRUCE (Section 3.3) introduces an untangling algorithm operating on a graph structure that concisely represents the configuration space of single-cable knots [9, 23, 35]. This section extends the undirected graph $G = (V, E)$ representation to handle n-cable knots. Vertices $v \in V$ represent any cable endpoints and crossings in the structure, and edges $e \in E$ represent cable segments (with no crossings) joining two vertices, denoted as $e = (u, v)$ for $u, v \in V$. While the graph representation of Chapter 3 considers crossings with ≤ 3 cable segments, in this chapter, a crossing (vertex v) can have k segments for any $k \geq 1$, such that $2k$ cable segments extend from the crossing, and the corresponding vertex v has a degree of $2k$. The graph vertices makes no distinction between intra-cable and inter-cable crossings. Endpoint vertices differ in that they always have a degree of one. Additionally, we annotate every adjacent (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:

$$X(v,e) = \begin{cases} +1 & \text{if } v \text{ is an endpoint or if } e \text{ crosses over} \\ & \text{all other edges at } v \\ -m & \text{if } e \text{ crosses under } m \text{ edges at } v \end{cases} \tag{4.1}$$

Intuitively, $X(v, e)$ indicates the depth of edge e at the crossing represented by vertex v. For example, Chapter 3 considered an annotation scheme with labels $\{-2, -1, +1\}$ (Equation

3.1) for crossings involving up to 3 cable segments where an over-crossing occludes at most 2 cable segments. Equation 4.1 merely generalizes this annotation scheme to arbitrary levels of self-occlusion. Observe that this graph can have multiple edges, as illustrated in Figure 4.2, but no two pairs of contiguous edges will have the same two annotations. With this graph definition, the multiple cable disentangling objective is to reach a configuration graph with $|V| = 2$ per cable, one corresponding to each endpoint, such that the two vertices belonging to each cable are connected by an edge with positive annotations on both ends.

Action Space

Using the same action space as that of Chapter 3, we redefine action primitives for multi-cable disentangling and introducing specialized motion primitives for cable separation. At each time t, we consider linear, bilateral actions $\boldsymbol{a}_t = (\boldsymbol{a}_{t,\text{right}}, \boldsymbol{a}_{t,\text{left}})$. Each gripper grasps the cable with a top-down grasp orientation $\theta_{t,k}$ about the z-axis at a fixed 30° approach angle relative to the vertical, moves a displacement $(\Delta x_{t,k}, \Delta y_{t,k})$ while closing $(\mathbb{1}_{\text{grasp}} = 1)$ or leaving open $(\mathbb{1}_{\text{grasp}} = 0)$ the gripper jaws, and releases the cable (Equation 3.2):

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

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

Because the perception systems of operate in pixel space, we require access to a standard world (x, y, z) to pixel coordinate (p_x, p_y) transformation. In the remaining sections, we will describe positions in terms of pixels, overloading the action notation to use pixel coordinates as well as workspace positions.

4.3 Methods

To the best of our knowledge, prior work has only considered single-cable untangling [23, 9, 35], and the resulting algorithms cannot be directly employed for disentangling multiple cables. In Chapter 3, we cover three methods for single-cable untangling: a high level action planner HULK (Section 3.3) and a low-level controller composed of LOKI and SPiDERMan (Section 3.4) for precise grasping and recovery manipulation, respectively. In this chapter, we present a novel algorithm IRON-MAN for disentangling multiple cables using the graph representation of multi-cable knots defined in Section 4.2. Next, we discuss the perception methods and manipulation primitives used to instantiate IRON-MAN for physical disentangling. The proposed perception stack redefines HULK keypoints as described in Section 3.3 to accommodate multiple cables, and directly employs LOKI and SPiDERMan (Section 3.4) to account for grasp refinement and recovery actions during disentangling, respectively. We also adapt the motion primitives (Node Deletion, Reidemeister moves) of the single cable untangling work from Chapter 3 to the multi-cable setting and introduce a novel manipulation primitive for managing excess cable slack.

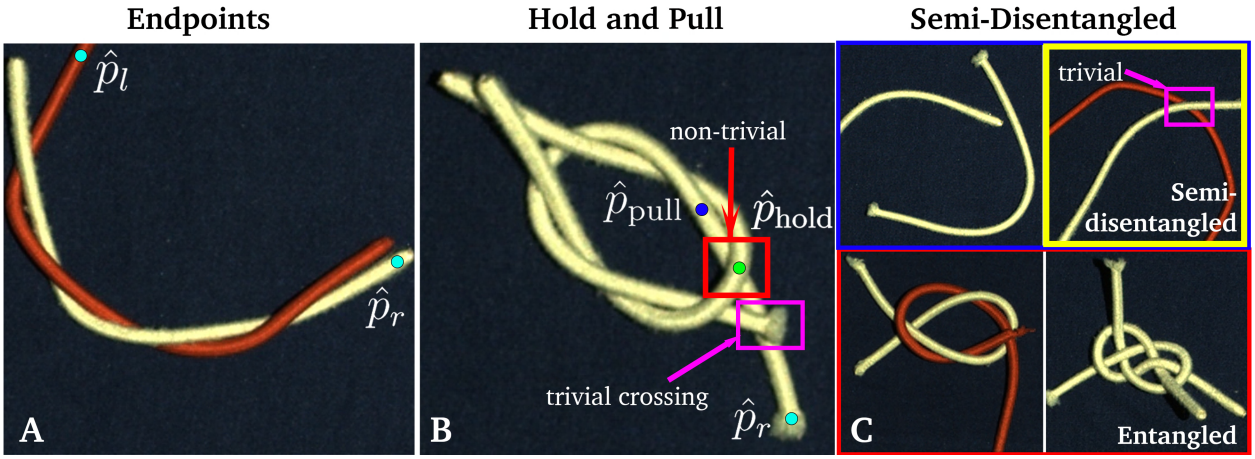

Figure 4.3: **Physical Implementation of IRON-MAN**: Image A depicts the endpoints used for Reidemeister moves: $\hat{\boldsymbol{p}}_r$ and $\hat{\boldsymbol{p}}_l$ on cables c_1 and c_2, respectively. The left endpoint annotation corresponds not to the leftmost endpoint in the scene (on cable c_1), but rather to the leftmost endpoint of cable c_2 because $\hat{\boldsymbol{p}}_r$ is on cable c_1. Image B depicts the hold and pull keypoints, $\hat{\boldsymbol{p}}_{\text{hold}}$ and $\hat{\boldsymbol{p}}_{\text{pull}}$, 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.

IRON-MAN

We present IRON-MAN, an algorithm for disentangling multiple cables knotted or twisted together. IRON-MAN operates on the implicit graph representation of a scene and disentangles n cables with repeated crossing removal. Compared to BRUCE (Section 3.3), IRON-MAN must reason about $2n$ endpoints and a greater number of potential crossings. This manifests in more challenging cable slack management in the practical instantiation of IRON-MAN.

Multi-cable Reidemeister Moves

In Chapter 3, Reidemeister moves (Equation 3.4) pull apart the left and right endpoints v_l, v_r of a singular cable to disambiguate the cable configuration. However, a scene with $n > 1$ cables contains n right endpoints and n left endpoints so the choice of v_l, v_r for instantiating a Reidemeister move is ambiguous. To resolve this, IRON-MAN first locates the rightmost endpoint node v_r, which belongs to some cable c_i, $i \in \{1, ..., n\}$. This is the rightmost endpoint in the scene across all possible cables. After defining v_r, the leftmost endpoint node v_l is defined as the leftmost endpoint belonging to some cable c_j, where $j \neq i$ and $j \in \{1, ..., n\}$ (Fig. 4.3). If only a single cable c_k remains in the scene so that $n = 1$, we define v_r and v_l to be the right and left endpoints of c_k, breaking ties arbitrarily. Reidemeister moves as defined in Equation 3.4 are then performed on the endpoints v_r and v_l.

Multi-Cable Node Deletion Moves

In the single cable untangling work of Chapter 3, operating on the first under-crossing relative to the right endpoint provided an unambiguous convention for Node Deletion moves. However, in the setting of multiple cables, the presence of multiple endpoints, a greater potential for accumulated slack, and inter and intra-cable crossings of varying degrees of entanglement complicates action selection. To address this, IRON-MAN first 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 those that come undone trivially by performing a Reidemeister move. For example, a twist is a set of trivial crossings, in which pulling the two cables apart via 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. Thus, IRON-MAN manages physical cable slack effectively and efficiently by prioritizing removal of non-trivial crossings. Reduction of trivial crossings is instead left to Reidemeister moves. IRON-MAN traverses the graph from the rightmost endpoint v_r and performs a Node Deletion move as defined in Equation 3.3 on the first non-trivial undercrossing it encounters.

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 no non-trivial crossings remain. As disentangling progresses, certain cables reach fully untangled states. These cables can be entirely removed from the scene in order to simplify the scene graph by minimizing the number of trivial crossings. For such cables, IRON-MAN uses a novel manipulation primitive called a Cable Extraction move (discussed further below) to remove fully untangled cables from the workspace.

Physical Disentangling

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 as in Chapter 3 for deployment on a physical robot. As in HULK (Section 3.3), 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 motion primitives for IRON-MAN, while Section 4.4 details the training dataset generation procedure.

Our approach uses a ResNet-34 backbone to separately learn three mappings for instantiating actions:

- $g_1 : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times2}$ maps an image to 2 heatmaps centered respectively at the keypoints $\hat{\boldsymbol{p}}_{\text{hold}}$ and $\hat{\boldsymbol{p}}_{\text{pull}}$, located at the first non-trivial intersection as defined in IRON-MAN. These predictions are used to perform multi-cable Node Deletion moves.

- $g_2 : \mathbb{R}^{640\times480\times3} \mapsto \mathbb{R}^{640\times480\times2}$ maps an image to 2 heatmaps centered respectively at the 2 keypoints $\hat{\boldsymbol{p}}_r$ and $\hat{\boldsymbol{p}}_l$, respectively located at the endpoints of cables c_i and c_j, $i \neq j$, to implement IRON-MAN's multi-cable Reidemeister moves. Note that $\hat{\boldsymbol{p}}_r$ corresponds to the rightmost endpoint in the scene, while $\hat{\boldsymbol{p}}_l$ is the leftmost endpoint in the scene that does not belong to cable c_i.

- $h : \mathbb{R}^{640\times480\times3} \mapsto \{0, 1\}$ is a binary classifier that detect when the cable c_i corresponding to the rightmost endpoint $\hat{\boldsymbol{p}}_r$ is semi-disentangled from the remaining cables. Cable c_i is what we call *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 (see Fig. 4.3). A cable in a *semi-disentangled* is ready to be removed from the scene, which we instantiate with a Cable Extraction move detailed below.

We train and implement LOKI per the procedure in Section 3.4. The only modification we make for LOKI in the multi-cable setting is to train on simulated cable crops consisting of both homogeneous and differently colored cables, since both are considered in this chapter (Assumption 4.2). By querying LOKI, we obtain refined keypoints, $\tilde{\boldsymbol{p}}_l, \tilde{\boldsymbol{p}}_r, \tilde{\boldsymbol{p}}_{\text{hold}}, \tilde{\boldsymbol{p}}_{\text{pull}}$, and their corresponding gripper orientations for executing grasps.

Manipulation

To execute the actions from IRON-MAN on a set of physical cables, we apply a novel manipulation primitive, a **Cable Extraction move**, along with **Reidemeister moves** and **Node Deletion moves**, to the task of multi-cable disentangling. We iteratively disentangle each cable in the scene with multi-cable Reidemeister and Node Deletion moves using newly-defined keypoints as in Section 4.3. During task execution, we use Cable Extraction moves to drop fully disentangled cables at $\boldsymbol{p}_{\text{term}}$, a predefined point within the termination area outside the manipulation region.

We define a **Cable Extraction move**, a novel manipulation primitive that fully disentangles and removes a semi-disentangled cable from a scene. This primitive is intended to preemptively de-clutter the workspace of fully disentangled cables to prevent unnecessary accumulation of cable slack. Given an image observation of the workspace I_t, we perform Cable Extraction moves when $h(I_t) = 1$, meaning the cable c_i corresponding to the rightmost endpoint $\tilde{\boldsymbol{p}}_r$ is detected as semi-disentangled. The right arm grasps c_i at its right endpoint $\tilde{\boldsymbol{p}}_r$, while the left arm *pins* $\tilde{\boldsymbol{p}}_l$, the leftmost endpoint from cable c_j $(i \neq j)$, against the workspace surface without closing the gripper jaws and grasping the cable. Crucially, we use *soft pinning*, which differs from *holding* in that a) in holding, the cable is not pushed against the workspace, and b) in soft pinning, the gripper jaws do not close to grasp the cable. Specifically, we use $\mathbb{1}_{\text{grasp}} = 0$ to achieve soft pinning. When only one cable is left in the scene, the predicted Reidemeister endpoints often lie on the same cable. This causes one arm to pull the right endpoint of the cable while pinning the left cable during a Cable Extraction move. Soft pinning crucially allows for the right arm to pull the remaining slack through the jaws of the left gripper, yielding a successful extraction. Finally, the right arm deposits cable c_i at a predefined termination point $\boldsymbol{p}_{\text{term}}$, reducing the number of trivial crossings in the scene:

$$\begin{aligned} \boldsymbol{a}_{t,r} &= (\tilde{p}_{x,r}, \tilde{p}_{y,r}, p_{x,\text{term}} - \tilde{p}_{x,r}, p_{y,\text{term}} - \tilde{p}_{y,r}, \hat{\theta}_r, 1) \\ \boldsymbol{a}_{t,l} &= (\tilde{p}_{x,l}, \tilde{p}_{y,l}, 0, 0, \hat{\theta}_l, 0). \end{aligned}$$

Finally, we implement a subset of manipulation recovery primitives from SPiDERMan (Section 3.4) for multi-cable disentangling. In particular, we implement the **Wedge Recovery** and **Re-posing (translation)** moves which rely on contour detection (Equation 3.8, Equation 3.7) and knowledge of $\boldsymbol{w}_l$, $\boldsymbol{w}_r$, $\boldsymbol{w}_c \in \mathbb{R}^3$. The implementation of SPiDERMan from Chapter 3 also included a binary classifier operating pairwise on current and past cable observations to sense loosening progress and termination. We omit this functionality for multi-cable disentangling for two reasons. First, sensing progress via pairwise comparisons of current and past observations is ambiguous when the number of crossings and cables considered changes drastically during task execution. Second, a more natural way of detecting termination in the multi-cable setting is to detect vacancy of the manipulation surface, since Cable Extraction moves sequentially deposit disentangled cables in a separated termination area.

Method Summary

The final manipulation stack consists of five manipulation primitives – **Node Deletion moves, Reidemeister moves, Cable Extraction moves, Wedge Recovery moves, and Re-posing (translation) moves** – instantiated with modified versions of HULK, SPiDERMan, and LOKI. The complete disentangling algorithm is presented in Algorithm 2. Starting with a Reidemeister move, IRON-MAN pulls endpoints $\tilde{\boldsymbol{p}}_r$ and $\tilde{\boldsymbol{p}}_l$ to opposite sides of the workspace, removing any initial trivial crossings in the process. Then, we successively take Node Deletion moves specified by $\tilde{\boldsymbol{p}}_{\text{hold}}$ and $\tilde{\boldsymbol{p}}_{\text{pull}}$ on the first non-trivial undercrossing localized relative to the $\hat{\boldsymbol{p}}_r$. Upon detecting a semi-disentangled cable c_i, we employ a Cable

Extraction move to remove c_i and any of its associated trivial crossings from the scene. This process repeats iteratively until all cables have been disentangled and deposited in the termination area, with occasional interventions by SPiDERMan (Figure 4.1).

Algorithm 2 Disentangling with IRON-MAN

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

4.4 Experiments

We evaluate IRON-MAN to disentangle knot configurations containing two or three cables and with three tiers of increasing difficulty. We implement the full system in experiments with the bilateral da Vinci surgical robot. We are not aware of any multi-cable disentangling algorithms that would provide a meaningful baseline, and single-cable algorithms are not well-defined in this setting.

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 7,020 examples via affine transforms, lighting shifts, and blurring. 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, augment to a dataset of 5,200 examples using similar augmentation techniques as mentioned above, and train with a binary cross-entropy loss. All datasets consist *only* of configurations with up to two cables, where the cables' colors are either both white or red and white. To reduce manipulation errors during experiments, we project all keypoint

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 4.1: **Physical Experiment 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). The IRON-MAN implementation disentangles two cables in Tier 1 and 2 configurations and three cables in Tier 3 configurations with an overall success rate of 80.5%. 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 the maximum number of disentangling actions, and (D) moving entangled cables to the terminated workspace.

predictions $\hat{\boldsymbol{p}} \in \{\hat{\boldsymbol{p}}_{\text{hold}}, \hat{\boldsymbol{p}}_{\text{pull}}, \hat{\boldsymbol{p}}_l, \hat{\boldsymbol{p}}_r\}$ onto the cable mask obtained by color thresholding from the background.

We train LOKI on 3,000 200 × 200 crops of images of configurations with one red and one white cable, generated in simulation via Blender 2.80 [18]. Offset heatmap labels are generated in simulation as 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 as described in Section 3.4.

Tiers of Difficulty

Across all difficulty tiers, the knot types considered are dense and depicted in Figure 4.4. The tiers are defined by the number of cables in the knot and whether the class of knots was present in the training dataset. In all three tiers, we consider both knots in which cables have contrasting colors (red and white) and knots in which all cables are of the same color (white).

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 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).

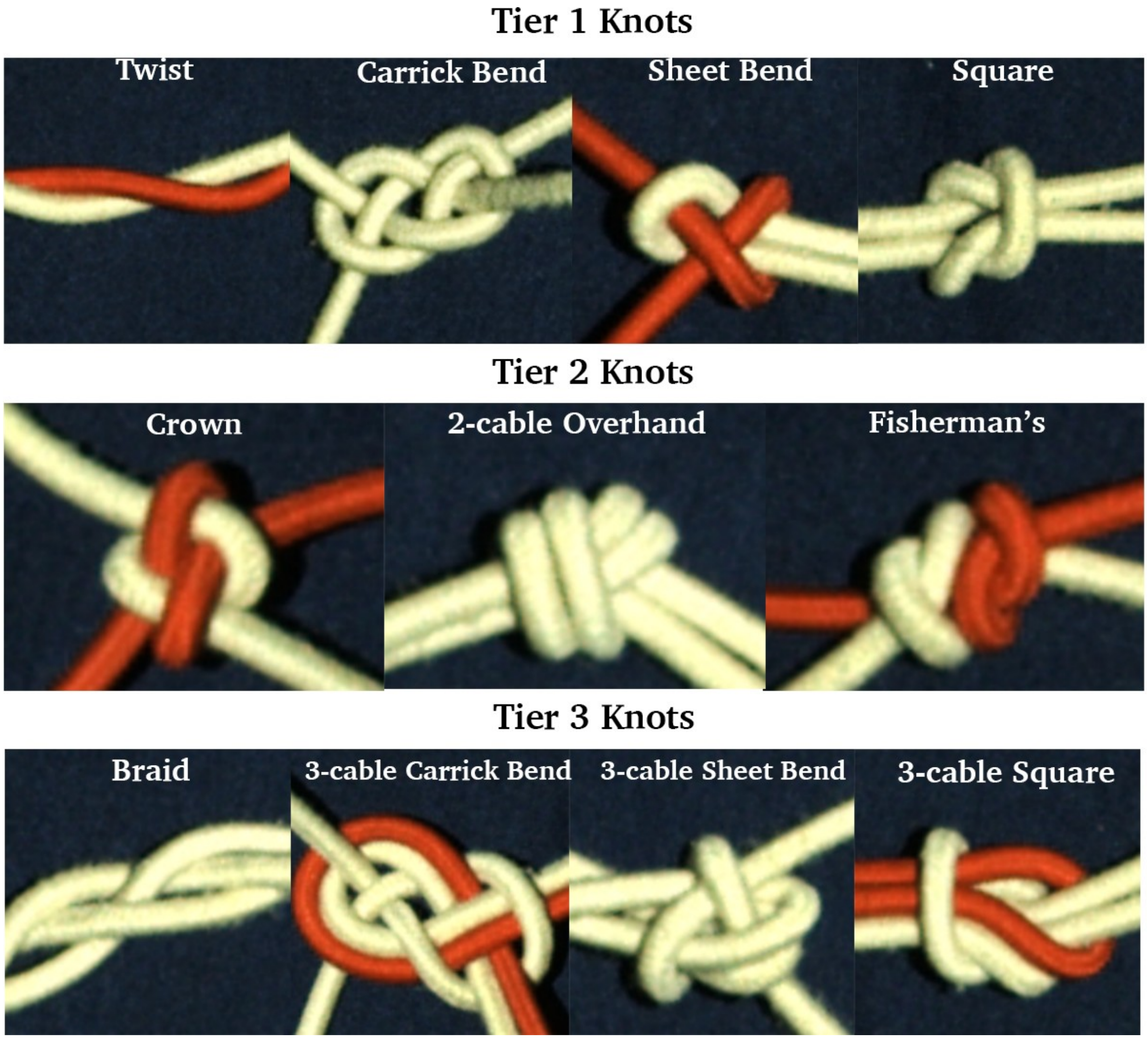

Figure 4.4: **Tiers of Configuration Difficulty**: The configurations considered in this chapter are organized into 3 tiers based on the number of cables in the knot and whether the class of knot was present in training data images. For each configuration, we consider both settings where cables have contrasting colors (red and white) and where cables are of the same color (all white).

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 to avoid end effector damage under collisions with the manipulation surface and to create friction with the hairties to prevent them from sliding out of the workspace. We collect 1920×1200 overhead RGB images for perception inference with a Zivid OnePlus RGBD camera.

Results

Table 4.1 presents the results from the physical trials. IRON-MAN succeeds in disentangling all cables with success rates of 91.6%, 66.6%, and 83.3% 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 Tier 2 and 3 cases, both the numbers of disentangling actions (Node Deletion and Cable Extraction moves) and SPiDERMan Recovery actions (Re-posing and Wedge Recovery moves) increase compared to Tier 1.

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. We allow fewer total actions in Tier 1 because the class of Tier 1 knots was seen in training.

(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 that the most common failure mode is (D), moving semi-disentangled (rather than fully disentangled) cables to the termination area with poor Cable Extraction manipulation. When the left arm does not effectively pin the remaining cables in the scene, high cable friction causes multiple cables to move 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).

4.5 Discussion

In this chapter, we build upon the strategies for single cable untangling from Chapter 3 and extend them to handle disentangling of multiple cables. We saw that dealing with multiple cables introduces new challenges, such as handling excess cable slack, tracking more endpoints and crossings, accommodating both identical and differently colored cables, and separating fully disentangled cables from remaining entanglements. To address these issues, we formalize a graph-based disentangling algorithm IRON-MAN with sensitivity to both intra and inter-cable crossings. We do not aim to infer or reconstruct full state knowledge of multiple cable instances in the scene. Instead, we reduce the problem to learning which crossings and endpoints to attend to at any point in time during task execution. We embed this awareness in the training procedure for HULK, and combine it with LOKI, SPiDERMan, and additional Cable Extraction primitives to instantiate IRON-MAN for robust multi-cable disentanglement.

Chapter 5

Conclusions

We present robot algorithms for untangling and disentangling single-cable and multi-cable dense, non-planar knots from RGB image observations. This thesis details the design of an iterative algorithm for intra and inter-cable crossing removal, perception of knot structures with global and local awareness, design of loosening and recovery motion primitives, and deployment of the resulting system onto a da Vinci surgical robot. A key focus of this thesis is understanding the interplay between precise controllers and recovery policies. While learning to sense and act with precision is important for proactive failure avoidance, errors are inevitable during any robotic manipulation task. Consequently, we additionally focus our efforts on anticipating or reducing the severity of failures.

In Chapter 3, we present LOKI and SPiDERMan, two algorithms that comprise a low-level controller for manipulation. LOKI and SPiDERMan jointly introduce precision and recovery into a previous high-level planner, HULK, for performing robust untangling of dense, non-planar knots in singular cables. Empirically, we find that the combination of HULK, LOKI, and SPiDERMan collectively outperforms baselines without recovery or refinement strategies. The combination achieves 72.8% success and 50% success on untangling seen knots and unseen knots, 66.7% success on non-planar knots not studied in prior work, and averages ≤ 9 actions over successful trials. In Chapter 4, we formalize the problem of disentangling multiple cables with IRON-MAN, an algorithm for sequentially removing crossings within and across cables. IRON-MAN — instantiated with HULK, LOKI, and SPiDERMan — disentangles up to three cables with 80.5% success. Together, HULK, LOKI, SPiDERMan, and IRON-MAN crucially leverage as much structure in the tasks as possible, and use analytical methods, deep-learning, and simulation as tools to address particular manipulation or perception pitfalls.

The open-source implementations for this body of work are linked below and intended for use:

- **Cable Simulation Environment**
- **HULK Training/Inference**
- **LOKI Training/Inference**
- **SPiDERMan Training/Inference**
- **da Vinci Untangling/Disentangling Codebase**

Chapter 6

Future Work

We view these as promising initial results but acknowledge several limitations that must be overcome to realize general-purpose, fully autonomous cable untangling and detangling. To that end, we are excited about several opportunities for future work:

- **Further Method Optimization**: The motion primitives considered in Chapters 3 and 4 are instantiated with linear trajectories and fixed approach angles, which may limit the range of possible untangling actions. In future work, we would like to expand the action space to include trajectories for more nuanced grasping between cable segments. We can also consider dynamic untangling actions such as shaking, bilateral pulling, slack clearance, or in-air untangling. On the perception side, we can improve the search space of viable crossings to be undone. Algorithms such as HULK and BRUCE strive for deterministic action selection by using conventions such as undoing crossings relative to endpoints. Instead, we could consider learning a multi-modal distribution or ranking over pulls/pins instead of a single prediction.

- **Improving Generality**: We hope to test the limits of HULK and SPiDERMan's performance on a diversified set of cables of varying physical properties such as friction, texture, turning radius, and length. These properties can substantially impact knot tightness, slack management, and ease of manipulation. Learning to grasp orthogonal to the cable direction from images may be too simple of an assumption for non-uniform thickness cables like ribbons or rubber bands. In follow-up work, we can also consider a low-level grasping controller learned directly from self-supervised interaction. This framework may enable one-shot or few-shot generalization to a cable with new physics at test time.

- **Task Extensions**: We are encouraged by several application-oriented extensions. One avenue is search, untangling, and retrieval of a cable with a target adapter (USB, Thunderbolt, etc.) from clutter, building on recent work in mechanical search of rigid objects [5]. The ideas of using keypoints for planning with a low-level grasping controller are also applicable to other deformable manipulation tasks such as knot-tying, cloth folding/unfolding, or bag manipulation. It is interesting to consider untangling followed by tying. This could apply to a task like shoelace tying, where untying could be a prerequisite. Alternatively, we can pose this as an adversarial learning framework [29] where tying and untying are jointly trained adversarial policies with competing objectives.

This work presents learnings about cable disentangling and untangling over the last two years, but leaves ample room to further our understanding of deformable objects and how best to interact with them. I am eager to continue tackling problems within and beyond this domain during my PhD at Stanford.

Bibliography

[1] Aude Billard and Danica Kragic. “Trends and challenges in robot manipulation”. In: *Science* 364.6446 (2019).

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

[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)*. IEEE. 2019, pp. 6443–6450.

[4] Cheng Chi and Shuran Song. “GarmentNets: Category-Level Pose Estimation for Garments via Canonical Space Shape Completion”. In: *arXiv preprint arXiv:2104.05177* (2021).

[5] Michael Danielczuk et al. “Mechanical search: Multi-step retrieval of a target object occluded by clutter”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1614–1621.

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

[7] 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.

[8] Aditya Ganapathi et al. “Learning to Smooth and Fold Real Fabric Using Dense Object Descriptors Trained on Synthetic Color Images”. In: *arXiv preprint arXiv:2003.12698* (2020).

[9] Jennifer Grannen et al. “Untangling Dense Knots by Learning Task-Relevant Keypoints”. In: *Conf. on Robot Learning (CoRL)*. 2020.

[10] Huy Ha and Shuran Song. *FlingBot: The Unreasonable Effectiveness of Dynamic Manipulations for Cloth Unfolding.* 2021.

[11] Kaiming He et al. “Identity mappings in deep residual networks”. In: *European conference on computer vision.* Springer. 2016, pp. 630–645.

[12] Rafael Herguedas et al. “Survey on multi-robot manipulation of deformable objects”. In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2019.

[13] Ryan Hoque et al. “VisuoSpatial Foresight for Multi-Step, Multi-Task Fabric Manipulation”. In: *Proc. Robotics: Science and Systems (RSS)*. 2020.

[14] Minho Hwang et al. “Efficiently Calibrating Cable-Driven Surgical Robots With RGBD Fiducial Sensing and Recurrent Neural Networks”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5937–5944.

[15] Tobias Johannink et al. “Residual reinforcement learning for robot control”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6023–6029.

[16] Peter Kazanzides et al. “An Open-Source Research Kit for the da Vinci Surgical System”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2014.

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

[18] Thomas Lallemand. *Blender*. US Patent App. 29/074,220. 1998.

[19] Michelle A Lee et al. “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)*. IEEE. 2020, pp. 7505–7512.

[20] Robert Lee et al. “Learning Arbitrary-Goal Fabric Folding with One Hour of Real Robot Experience”. In: *Conf. on Robot Learning (CoRL)*. 2020.

[21] Xingyu Lin et al. “SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation”. In: *arXiv preprint arXiv:2011.07215* (2020).

[22] Charles Livingston. *Knot theory*. Vol. 24. Cambridge University Press, 1993.

[23] 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*. IEEE. 2013, pp. 837–844.

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

[25] Hermann Mayer et al. “A system for robotic heart surgery that learns to tie knots using recurrent neural networks”. In: *Advanced Robotics* 22.13-14 (2008), pp. 1521–1537.

[26] Jean-Pierre Merlet and David Daney. “A portable, modular parallel wire crane for rescue operations”. In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010.

[27] Ashvin Nair et al. “Combining self-supervised learning and imitation for vision-based rope manipulation”. In: *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2146–2153.

[28] Samuel Paradis et al. “Intermittent Visual Servoing: Efficiently Learning Policies Robust to Instrument Changes for High-precision Surgical Manipulation”. In: *arXiv preprint arXiv:2011.06163* (2020).

[29] Lerrel Pinto, James Davidson, and Abhinav Gupta. "Supervision via competition: Robot adversaries for learning tasks". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1601–1608.

[30] Jose Sanchez et al. "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey". In: *The International Journal of Robotics Research* 37.7 (2018), pp. 688–716.

[31] Tanner Schmidt, Richard Newcombe, and Dieter Fox. "Self-supervised visual descriptor learning for dense correspondence". In: *IEEE Robotics and Automation Letters* 2.2 (2016), pp. 420–427.

[32] Daniel Seita 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.

[33] Daniel Seita et al. "Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks". In: *arXiv preprint arXiv:2012.03385* (2020).

[34] Priya Sundaresan et al. "Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data". In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2020.

[35] Priya Sundaresan et al. "Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies". In: *tinyurl.com/nonplanar-cable* (2021).

[36] Satoshi Suzuki et al. "Topological structural analysis of digitized binary images by border following". In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46.

[37] Brijen Thananjeyan et al. "Recovery RL: Safe Reinforcement Learning with Learned Recovery Zones". In: *NeurIPS Deep Reinforcement Learning Workshop* (2020).

[38] Jur Van Den Berg et al. "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, pp. 2074–2081.

[39] Vainavi Viswanath et al. "Disentangling Dense Multi-Cable Knots". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 2021.

[40] 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)*. IEEE. 2019, pp. 1309–1315.

[41] Yuji Yamakawa et al. "One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors". In: *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE. 2007, pp. 703–708.

[42] Mengyuan Yan et al. "Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2372–2379.

[43] Wilson Yan et al. "Learning Predictive Representations for Deformable Objects Using Contrastive Estimation". In: *Conf. on Robot Learning (CoRL)*. 2020.

[44] Harry Zhang et al. "Robots of the Lost Arc: Learning to Dynamically Manipulate Fixed-Endpoint Ropes and Cables". In: *arXiv preprint arXiv:2011.04840* (2020).