

# SGTM 2.0: Autonomously Untangling Long Cables using Interactive Perception

Kaushik Shivakumar<sup>\*1</sup>, Vainavi Viswanath<sup>\*1</sup>, Anrui Gu<sup>1</sup>, Yahav Avigal<sup>1</sup>, Justin Kerr<sup>1</sup>,  
Jeffrey Ichnowski<sup>1</sup>, Richard Cheng<sup>2</sup>, Thomas Kollar<sup>2</sup>, Ken Goldberg<sup>1</sup>

**Abstract**—Cables are commonplace in homes, hospitals, and industrial warehouses and are prone to tangling. This paper extends prior work on autonomously untangling long cables by introducing novel uncertainty quantification metrics and actions that interact with the cable to reduce perception uncertainty. We present Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0), a system that autonomously untangles cables approximately 3 meters in length with a bilateral robot using estimates of uncertainty at each step to inform actions. By interactively reducing uncertainty, Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0) reduces the number of state-resetting moves it must take, significantly speeding up run-time. Experiments suggest that SGTM 2.0 can achieve 83% untangling success on cables with 1 or 2 overhand and figure-8 knots, and 70% termination detection success across these configurations, outperforming SGTM 1.0 by 43% in untangling accuracy and 200% in full rollout speed. Supplementary material, visualizations, and videos can be found at [sites.google.com/view/sgtm2](https://sites.google.com/view/sgtm2).

## I. INTRODUCTION

Long cables, including electrical cords, ropes, and string, are ubiquitous in households and industrial settings [1], [2], [3]. These single-dimensional deformable objects can form knots that may restrict functionality or create hazards. Furthermore, as cable length increases, perception and manipulation of these objects become more difficult as the increased amount of free cable (which we refer to as *slack*) can cause the cable to not only fall into unreachable areas of the workspace, but also form complex knots and reach irrecoverable states. Further, retrieving full state information from image observations is especially challenging when slack occludes or falsely resembles true knots.

In our prior work, we approached partial observability by using manipulation primitives which attempt to simplify state for perception modules [4]. However, this approach lacked uncertainty awareness and took actions that were often exceedingly aggressive or conservative (see II-B). To address this limitation, this paper focuses on quantifying uncertainty to enable applying interactive perception [5], which involves physically manipulating objects in a scene to better understand it. By considering perceptual uncertainty, we are able to perform targeted interactive perception actions that clarify the state, allowing us to perform subsequent actions with higher confidence.

This paper makes the following contributions over SGTM 1.0 [4]:

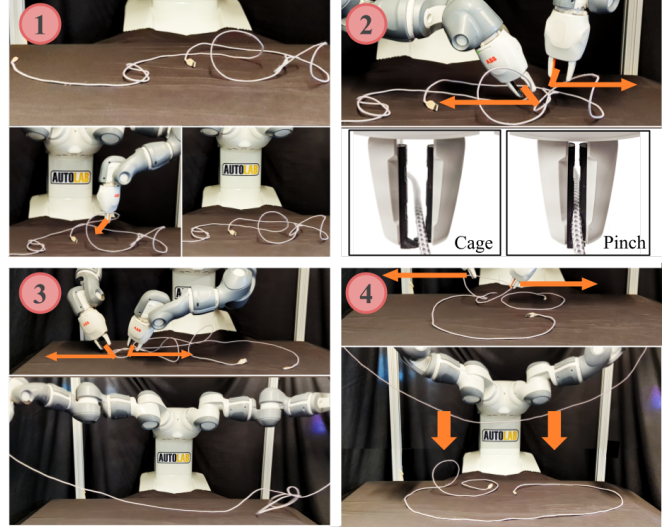


Fig. 1. **Overview of Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0):** SGTM 2.0 untangles a long cable with 2 figure-8 knots. (1) The system cannot perceive a clear path to a knot and performs an exposure move, bringing the endpoint cable segment back into the observable workspace. (2) SGTM 2.0 confidently untangles the figure-8 knot using cage-pinch dilation. (3) The system untangles the last figure-8 knot in the scene and does an incremental Reidemeister move. (4) SGTM 2.0 perceives a knot-like region and uses a partial cage-pinch dilation to disambiguate it. After another incremental Reidemeister move, the system terminates confidently having verified that no knots remain.

- 1) Novel perception-based metrics to estimate untangling-specific uncertainty in cable configurations, including tracing, network, and observational uncertainties as described in Section IV-B.
- 2) Novel primitives, including interactive perception actions, for cable slack management, untangling, and termination described in Section IV-C to reduce the probability of irrecoverable failures.
- 3) SGTM 2.0, an algorithm using uncertainty quantification to parameterize interactive perception actions for untangling described in Section IV-D (overview in Figure 1).
- 4) Data from physical experiments suggesting 43% improvement in untangling accuracy and 200% improvement in speed compared to SGTM 1.0, and data suggesting that interactive perception improves accuracy by 21% in complex cases.

## II. RELATED WORK

### A. Deformable Object Manipulation

Autonomous deformable object manipulation is an open problem in robotics. Deformable objects have infinite-

<sup>\*</sup> Equal Contribution

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

<sup>2</sup> Toyota Research Institute (TRI)

dimensional configuration spaces, are prone to self-occlusions, and are subject to complex dynamics. An increasingly popular approach to these problems is end-to-end deep learning with imitation learning [6], [7], [8] or reinforcement learning [9], [10], [11]. Since large-scale physical data collection is difficult, one technique is training in simulation and deploying the learned policy on physical systems [9], [10], [12], [13], [14], [15], [16], [17], [18], [19]. However, there the sim-to-real gap remains large due to challenges in modeling deformable objects [12]. An alternative approach is to use self-supervised learning to collect the training data directly on the physical system [20], [11], [21].

In multi-step algorithmic pipelines, perception-based techniques have shown to be effective for deformable object manipulation. Prior work uses dense object descriptors [22] for cable knot tying [15] and fabric smoothing [23], as well as visual dynamics models for non-knotted cables [14], [24] and fabric [17], [14], [25]. However, robust cable state estimation and dynamics estimation remain challenging for self-occluded, knotted configurations. We build on prior keypoint-based work [4], [26], [27] with uncertainty-aware primitives and interactive perception for the task of autonomously untangling long cables.

### B. Cable Untangling

Early work by Lui and Saxena [28] uses traditional feature extraction to represent a cable's structure as a graph. Recent work [27], [29], [4] use learning-based keypoint detection to parameterize action primitives in an untangling pipeline. Specifically, we improve upon Sliding and Grasping for Tangle Manipulation (SGTM 1.0) [4], an algorithm built on action primitives for autonomous long cable untangling. However, SGTM 1.0 may proceed with untangling despite low confidence in the predicted actions, leading to irrecoverable states. It also relies on shaking, a randomized reset primitive, when progress is difficult. Finally, it also requires a time-consuming physical trace to achieve the necessary confidence for termination. SGTM 2.0 addresses these three issues through *interactive* perception, by taking untangling actions sensitive to uncertainty, making targeted moves to reduce uncertainty instead of generic recovery moves, and terminating only once sufficiently certain across multiple views that the cable is untangled.

### C. Active and Interactive Perception

In 1988, Bajcsy [30] defined *active perception* as a search of models and control strategies for perception. Strategies vary according to the sensor and the task goal, including controlling camera parameters [31] and moving a tactile sensor according to haptic input [32]. Recently, Bohg et al. [5] explore the differences between *active* and *interactive* perception, the latter of which specifically exploits environment interactions to simplify and enhance perception to achieve a better understanding of the scene [5], [33]. Within robotic manipulation, several works have focused on improving understanding of the environment through scene interaction. Tsikos and Bajcsy [34] propose interacting with

random heaps of unknown objects through pick and push actions for scene segmentation. Danielczuk et al. [35] present the mechanical search problem, where a robot locates and retrieves an occluded target object from a cluttered bin through a series of targeted parallel jaw grasps, suction grasps, and pushes. Novkovic et al. [33] propose a combination of camera motions with environment interactions to find a target cube hidden in a pile of cubes.

Interactive perception has also been applied to deformable manipulation. Willimon et al. [36] interact with a pile of laundry to isolate and identify individual clothing items. In our work, the robot interacts with the cable to reveal more information about the cable state.

## III. PROBLEM STATEMENT

As in [4], we consider untangling long ( $\sim 3$  m) cables from RGB-D image observations. We use a bimanual robot to execute manipulation primitives until the cable reaches a fully untangled state with no knots.

### A. Workspace Definition and Assumptions

The bilateral robot operates in an  $(x, y, z)$  Cartesian coordinate frame with two 6-DOF robot arms. The robot is equipped with cage-pinch jaws introduced in [4] to allow for both sliding along and tightly pinching the cable (Figure 1(2)). The workspace lies in the  $xy$ -plane and the only inputs to the algorithm consist of RGB-D images. The workspace contains a single incompressible electrical cable of length  $l_c$  and cross-sectional radius  $r_c$ . Cable state  $s \in \mathcal{S}$  can be described as a continuous path  $c_n(u) : [0, 1] \rightarrow (x, y, z)$  in the workspace, where  $u$  indexes the position along the length of the cable.  $c_n(0)$  and  $c_n(1)$  always refer to the position of the endpoints of the cable. We initialize the cable's state before  $n = 0$  with the procedures specified in Section V. One challenge in this problem is that parts of the cable may rest outside the reachable and observable workspace at any point in a rollout (defined as a single experiment aiming to remove all knots in the cable). Moreover, self-occlusions in the cable are possible due to only one overhead camera view. This partial observability motivates the need for actions that reveal more information about the cable state  $s$ .

We make the following assumptions: (1) the cable can be segmented from the background via color thresholding; (2) transformations between the camera, workspace, and robot frames are known; and (3) the cable start state contains overhand or figure 8 knots of dense (6-8 cm diameter) or loose (12-14 cm diameter) configurations in series.

### B. Task Objective and Metrics

The goal of the robot is to untangle the cable and terminate at time  $t < T_{\max}$ , specified in Section V. After each step of a rollout  $r$ , a new observation  $o$  of the cable state  $s$  is taken. Each primitive constitutes at least one step.

The goal of the robot over the course of each rollout is to untangle the cable and output a termination signal (DONE). We use  $H_{\text{DONE}}$  to represent a step function, with the step occurring when the robot outputs DONE. We define an

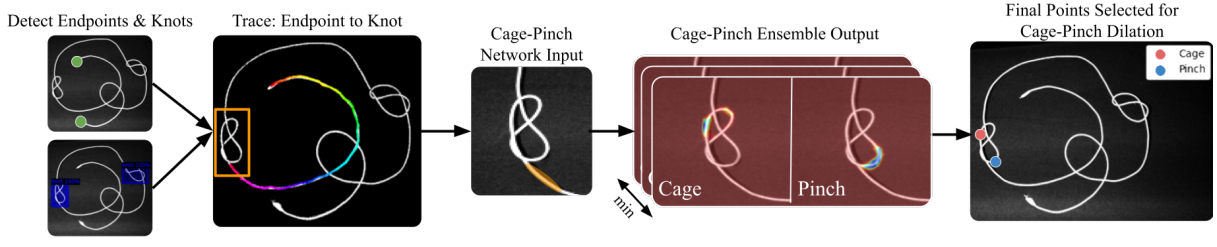


Fig. 2. **Perception system:** This is the pipeline used to determine which points to cage and pinch for a cage-pinch dilation move, the crucial action for untying a knot. First, we detect the endpoints and knots. Next, we trace from the endpoint to the first bounding box. If the trace is certain, we run the cage-pinch network ensemble on the cropped knot in the bounding box with the trace tail encoded into one of the channels. We take the pixelwise minimum across the cage-pinch network ensemble outputs, leading to 1 heatmap encoding “worst-case” untying success probabilities each for the cage and pinch point. We take the arg max of each of the two heatmaps to determine the final points to pinch and cage during the cage-pinch dilation action.

untangled cable as one that has no knots when its endpoints are grasped top-down and extended the maximum feasible distance, with knots defined identically to [4]. We use  $k_t^r$  to denote the number of knots in the cable at time  $t$  in rollout  $r$  and assume the cable is initialized with  $k_0^r$  knots.

We use the following metrics to measure performance, where  $0 < K \leq k_0^r$  refers to the number of knots untangled and  $R$  is the total set of rollouts:

- 1) *Untangling  $K$  Success Rate*, the percentage of rollouts that untangle  $K$  knots:  $\frac{1}{|R|} \sum_{r \in R} \mathbf{1}_{\{\exists t < T_{\max} : k_t^r \leq k_0^r - K\}}$
- 2) *Untangling Verification Rate*, the percentage of rollouts that untangle all knots and terminate successfully:  $\frac{1}{|R|} \sum_{r \in R} \mathbf{1}_{\{\exists t < T_{\max} : k_t^r = 0 \wedge \text{DONE}_t\}}$
- 3) *Average Untangling  $K$  Time*, the average time to untangle  $K$  knots across all applicable rollouts  $R_a$  where this occurs before  $T_{\max}$ :  $\frac{1}{|R_a|} \sum_{r \in R_a} (\min t : k_t^r \leq k_0^r - K)$
- 4) *Average Untangling Verification Time*, the average time to reach  $k_t^r = 0$  and declare termination across all applicable rollouts  $R_a$  like above:  $\frac{1}{|R_a|} \sum_{r \in R_a} (\min t : k_t^r = 0 \wedge \text{DONE}_t)$

## IV. METHODS

### A. Approach Overview

Unlike SGTM 1.0, SGTM 2.0 uses interactive perception primitives designed to better manage slack during untying or reveal additional information about the cable state  $s \in \mathcal{S}$ . As  $s$  is difficult to estimate from the provided observation  $o \in \mathcal{O}$ , SGTM 2.0 uses a policy  $\pi : \mathcal{O} \rightarrow \mathcal{A}$  built as described in Section IV-D from the components in Sections IV-B and IV-C to directly predict actions to execute. Unlike prior work, SGTM 2.0 contains perception components that lend themselves to probabilistic interpretation and manipulation primitives that are sensitive to perception uncertainty. We note that while the distribution of states the robot encounters may have high variance, SGTM 2.0 is sensitive only to variance that may affect the next untying action. Practically, this means that for example, even if much of the cable is bunched up and occluded, as long as the path from an endpoint to the first knot is clearly visible, the algorithm can still take an untying action with high confidence. SGTM 2.0 is designed to be sensitive only to uncertainty that affects the untying process.

### B. Uncertainty-Aware Perception Systems

1) *Endpoint Detection:* We train a Faster R-CNN model with a Resnet-50 FPN (Feature Pyramid Network) backbone [37] on 305 labeled examples to detect cable endpoints. We discard all bounding boxes with lower than 99% confidence, achieving an average precision and recall for endpoint detection are 86.7% and 100% respectively.

2) *Knot Detection:* To identify all knots in the observable workspace, we use the same architecture as the endpoint model trained on 688 labeled images. The dataset is augmented with flip, contrast, brightness, rotation, saturation, and scale augmentations. We use a 99% detection threshold, which achieves an average precision of 91.3% and an average recall of 95.5%. We analytically filter out misclassified knots by checking if a simple loop fills the bounding box. Because the model’s output is dependent on the orientation of the cable, in certain cases, we use multiple observations of the underlying cable state  $s$  as described in Section IV-C.4. This allows SGTM 2.0 to be sensitive to what we define as *observational uncertainty*.

3) *Cable Tracing:* The objective of cable tracing is to follow the path of the cable from an overhead image. Given an RGB image and a start pixel (the center of an endpoint), the tracer we introduce in this work outputs a set of possible splines. It maintains a set of valid splines and iteratively expands it by exploring candidate successor points, preferring those that do not deviate sharply from the current spline’s trajectory. An example of candidate trace paths is shown in Figure 3.

Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0) uses this in 2 scenarios: (1) finding a knot to untangle and (2) achieving robust grasps near cable endpoints. When used for finding a path from endpoint to knot, the tracer terminates once the trace intersects the knot within a bounding box. For grasping endpoints, the trace terminates after traveling a fixed distance along the cable. At the termination of tracing, we fit a bounding box  $B_t$  to the ending points of all the final traces; if either dimension of  $B_t$  is greater than 24 pixels for knot tracing and 12 pixels for endpoint tracing (which requires more precision), the traces diverge and thus TRACE\_UNCERTAIN is returned; otherwise, TRACE\_CERTAIN is returned. Note that while traces with very different topologies may be returned, as long as they end near the same point (Figure 3, left and center),

the untangling-relevant uncertainty remains low.

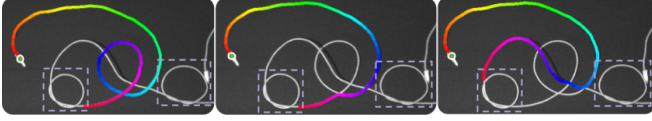


Fig. 3. **Cable Tracing:** multiple candidate trace paths returned by the tracer to find the closest knot from the top-left endpoint. The tracer finds the correct knot in all cases, but is unclear as to which side of the knot is attached to the free endpoint, and therefore returns `TRACE_UNCERTAIN`.

4) *Cage-Pinch Dilation Point Selection:* We use an FCN [38] with a ResNet-34 backbone trained on 568 knot crops to output two heatmaps: one for the cage point and one for the pinch point. We augment this data with random rotations, flips, shear, and synthetically added distractor cables to improve robustness. Approximately 250 of these images are gathered during rollouts to mitigate distribution shift in a style similar to DAgger [39]. The input is a 3-channel image, where one of the channels contains a Gaussian heatmap around the segment of cable entering the image determined by the cable tracing algorithm. This additional input conditions the network and breaks the symmetry between the cage and pinch points. We train the network to predict the cage point as the first graspable point beyond the undercrossing forming the knot, and the pinch point as the place to secure the cable to create an opening for the free end to slide through. Example cage and pinch points and the perception pipeline to obtain the points are shown in Figure 2.

The goal of this network is to model the probability  $P_p(U_s)$  per grasp pixel  $p$ , where  $U_s$  corresponds to untangling success on the cropped knot. We train an ensemble of 3 models on the same data but with different initializations. For the cage point, to sample from each of the ensemble heatmaps  $h^{\text{cage}} \in H^{\text{cage}}$ , we create a new heatmap as such:  $h'_{i,j} = \min_{h^{\text{cage}} \in H^{\text{cage}}} h_{i,j}^{\text{cage}}$ . The same procedure is used for the pinch point. We return the cage and pinch point as the  $\arg \max_{i,j} h'_{i,j}^{\text{cage}}$  and  $\arg \max_{i,j} h'_{i,j}^{\text{pinch}}$ , respectively. If  $\max_{i,j} h'_{i,j}^{\text{cage}} \max_{r,s} h'_{r,s}^{\text{pinch}} < \kappa$  where  $\kappa = 0.35$  (calibrated empirically), we output `NETWORK_UNCERTAIN`. Otherwise, we output `NETWORK_CERTAIN`.

The above methods help reveal whether the worst-case probability (across our predictive distribution modeled by an ensemble) of untangling success is high enough to proceed with untangling the cable at the specified points. If not, the network is too uncertain in its predicted points to proceed confidently as the next action may instead tighten the knot or lead the cable into an irrecoverable state.

### C. Manipulation Primitives for Interactive Perception

1) *Cage-Pinch Dilation:* To untangle an individual knot, the robot leverages the flexibility of the cage-pinch grippers introduced in [4] and depicted in Figure 1(2) to cage one point and pinch another point inside the knot and pull apart the arms to a distance determined by the length of the trace from the endpoint to the knot, while moving its wrist joint in a high-frequency sinusoidal motion. A major benefit of cage-pinch actions compared to cage-cage actions from [4]

is the ability to better manage slack, preventing accidentally tightening another knot. Following this action, the robot lays the cable down as far forward as kinematically feasible to isolate the newly untangled portion from the remaining cable.

2) *Partial Cage-Pinch Dilation:* This primitive is similar to the Cage-Pinch Dilation, but the distance the arms move apart is fixed to 5 cm beyond their starting separation. This is meant to perturb the state and to later retry perception rather than to completely untangle a knot.

3) *Reidemeister Move:* In this primitive, the robot uses tracing to find robust grasp points slightly down the cable from the endpoints. Next, the robot moves both arms outward horizontally and up, lifting the cable off the workspace, allowing for loops to fall away. Compared to prior work, we add (1) the vertical component of the action, forming a large letter “U” with the cable, and (2) the wide lay-out action, which places the cable on the workspace in a “U” shape.

4) *Incremental Reidemeister Move:* This primitive performs the exact same motions as a Reidemeister move, but uses a multi-stage, perception-based approach where the cable is observed at certain waypoints along the action. We use our knot detection network at these intermediate points to determine whether any knots remain. This can be interpreted as ensembling via perturbation of the observation of the same underlying cable topology. Being sensitive to observational uncertainty allows us to eliminate the time-consuming physical tracing action used in [4] for termination.

5) *Exposure Action:* When one or more endpoints are missing for an action, we uniformly at random sample a segment of the cable leaving the reachable workspace and pull it towards the center of the workspace. We do this also for unreachable knots that we wish to act on.

### D. Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0) Algorithm

Sliding and Grasping for Tangle Manipulation 2.0 (SGTM 2.0) ties together the aforementioned perception components and manipulation primitives to untangle cables. SGTM 2.0 alternates between the perception and manipulation components, using uncertainty from the former to determine whether to untangle or disambiguate the cable state. The algorithm is covered in detail in Figure 4.

## V. EXPERIMENTS

### A. Experimental Setup

For our experiments, we use the bimanual ABB YuMi robot with an overhead Phoxi camera, operating on a black foam-padded workspace of width 1.0 m and depth 0.75 m. Due to hardware constraints, we slightly extend the workspace with cardboard (by 0.1 meters on either side) not previously present in SGTM 1.0, but this does not make a significant difference as the cable mostly remains in the original foam-padded workspace. We use a 2.7 m-long white, braided electrical cable with USB adapters on both ends.

We evaluate SGTM 2.0 on 3 tiers of difficulty:

- 1) **Tier 1:** A cable with 1 overhand or figure-8 knot.



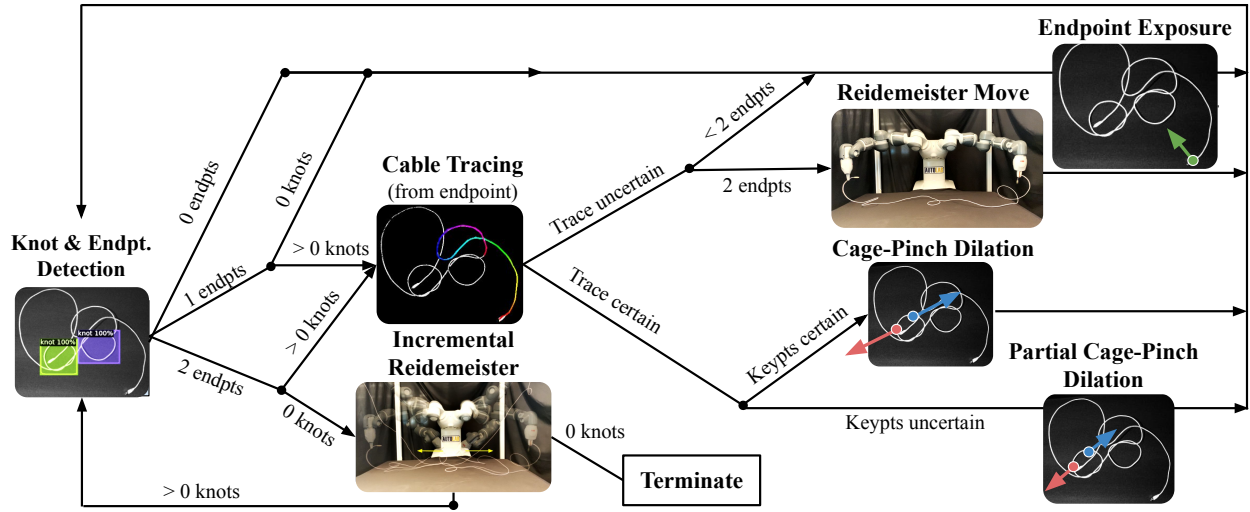


Fig. 4. **SGTM 2.0 Algorithm:** SGTM 2.0 first detects the number of knots and endpoints in the scene. If the endpoints are not visible, there is no way to verify any knot’s relative position to the endpoint. This is necessary because SGTM 2.0 only untangles knots adjacent to an endpoint to avoid knots colliding into each other and creating irrecoverable configurations. If fewer than two endpoints and no knots are visible, the algorithm is also unable to perform a termination check as that requires performing an incremental Reidemeister, which grasps the cable at the endpoints. In both these cases, SGTM 2.0 performs an endpoint exposure. If two endpoints are visible and no knots are visible, SGTM 2.0 proceeds to the incremental Reidemeister move. If one or two endpoints are visible and there are knots in the scene, it attempts to untangle, beginning by tracing from the visible endpoint(s). Here, if it is not able to confidently trace from either endpoint to a knot, SGTM 2.0 performs a Reidemeister move or endpoint exposure (based on the number of endpoints visible) to increase likelihood of unambiguous traces in future steps. Otherwise, it assesses the cage-pinch network uncertainty on the predicted points. If it is confident, it proceeds with a full cage-pinch dilation. Else, it performs a partial cage-pinch dilation to disambiguate the state.

- 2) **Tier 2:** A cable with 2 overhand and/or figure-8 knots.
- 3) **Tier 3:** A cable with 3 overhand and/or figure-8 knots.

The knots in all tiers are evaluated equally in both loose and dense configurations and evenly across positions along the cable (closer to an endpoint vs. closer to the middle). Example start configurations are shown in Figure 5. The cable is initialized by laying the knot(s) flat on the workspace, raising the endpoints as high as possible without lifting the knot(s), and then dropping the endpoints. We enforce a time limit of 15 minutes for all tiers. Note that the cable initialization procedures in Tiers 1 and 2 of this work are *exactly* the same as Tiers 1 and 2 in SGTM 1.0, the prior state-of-the-art [4].

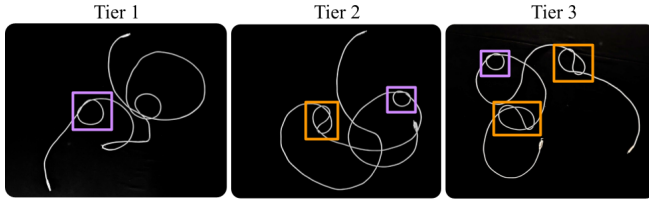


Fig. 5. **Example starting configurations for all 3 tiers:** Overhand knots are outlined in purple and figure-8 knots are outlined in orange.

For each tier, we report the average time to fully untangle the cable (for the rollouts that succeed in doing so) as well as the average time to correctly report that the cable is untangled (for the rollouts that succeed in doing so). Additionally, we report the success rates for untangling alone and untangling with termination detection. For Tiers 2 and 3, we present ablation results where SGTM 2.0(-U) represents SGTM 2.0 with the uncertainty-based components removed. For Tiers 1 and 2, we also report the speedup of SGTM 2.0 and SGTM 2.0(-U) from SGTM 1.0.

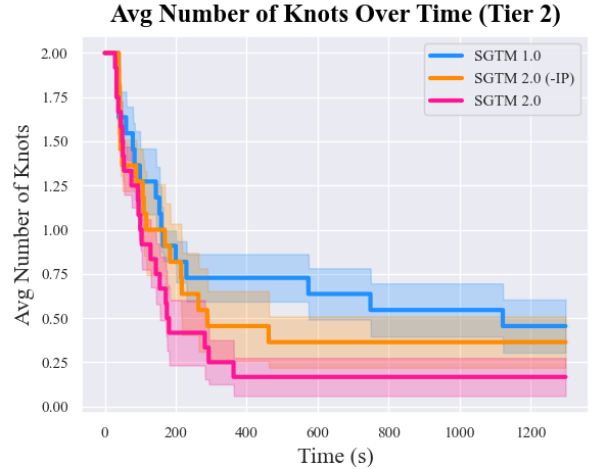


Fig. 6. **Average Knots over Time (Tier 2):** SGTM 2.0 most quickly reduces the number of knots on average. SGTM 2.0(-U) is less successful in untangling over the given time period than SGTM 2.0. Further, performance of even SGTM 2.0(-U) exceeds that of the prior state-of-the-art, SGTM 1.0, at 15 minutes, showing the effectiveness of improved untangling primitives such as cage-pinch dilations introduced in SGTM 2.0.

## B. Results

Results show that across Tiers 1 and 2, SGTM 2.0 outperforms SGTM 1.0 not only on untangling and verification success rate, but also achieves statistically significant speedups in the untangling time in Tier 1, untangling time for both knots in Tier 2, and verification time in Tier 2. Tier 3 was unachievable in SGTM 1.0 due to algorithmic constraints, but is now possible with SGTM 2.0, which achieves 9/12 successes in untangling 2 out of the 3 knots in Tier 3. In V-C, we discuss difficulties that result in failures, leading to 3/12 untangling success on all 3 knots in Tier 3.

TABLE I  
RESULTS FROM PHYSICAL EXPERIMENTS (84 TOTAL TRIALS)

	Tier 1		Tier 2			Tier 3	
	SGTM 1.0	SGTM 2.0	SGTM 1.0	SGTM 2.0(-U)	SGTM 2.0	SGTM 2.0(-U)	SGTM 2.0
Knot 1 Success Rate	8/12	<b>10/12</b>	10/12	10/12	<b>12/12</b>	12/12	12/12
Knot 2 Success Rate	-	-	6/12	7/12	<b>10/12</b>	9/12	9/12
Knot 3 Success Rate	-	-	-	-	-	3/12	3/12
Verification Rate	3/12	<b>7/12</b>	1/12	5/12	<b>7/12</b>	0/12	<b>2/12</b>
Avg. # of Actions	<b>5.0±1.5</b>	5.6±1.7	12.0	<b>6.0±1.2</b>	7.7±1.6	N/A	<b>12.0±0.0</b>
Avg. Knot 1 Time (s)	189.8±69.4	<b>53.1±7.5 (3.6x)</b>	93.3±16.2	128.6±41.9 (0.7x)	<b>75.6±21.0 (1.2x)</b>	88.7±25.5	<b>69.8±15.1</b>
Avg. Knot 2 Time (s)	-	-	586.4±165.3	<b>160.9±26.4 (3.6x)</b>	180.9±26.2 (3.2x)	<b>177.3±28.3</b>	233.1±57.6
Avg. Knot 3 Time (s)	-	-	-	-	-	<b>417.7±104.1</b>	476.7±160.1
Avg. Verif. Time (s)	330.8±127.8	<b>295.9±61.4 (1.1x)</b>	1079.0	<b>359.6±74.6 (3.0x)</b>	406.9±22.0 (2.7x)	N/A	<b>704.5±13.5</b>
Failures	-	(A) 2, (B) 1, (C) 1, (D) 1	-	(A) 1, (B) 0, (C) 5, (D) 1	(A) 2, (B) 1, (C) 1, (D) 1	(A) 1, (B) 4, (C) 6, (D) 1	(A) 5, (B) 2, (C) 2, (D) 1

### C. Failure Modes

Across all 3 tiers, we observe the following failure modes.

**(A) Timeout, unable to determine termination:** This is the most common failure case of SGTM 2.0 across all tiers, especially in Tier 3. Causes include:

- 1) The system inadvertently manipulates the cable into a state that is difficult to perceive and manipulate, most common in Tier 3 due to higher complexity and a higher chance that a rare failure in disambiguation may accidentally tighten or complicate a knot.
- 2) A substantial portion of the cable leaves the workspace. The system repeatedly attempts exposure actions, but due to the weight of the endpoint or large mass of cable, the cable continually slips back down into its prior configuration.
- 3) Though the entire cable may enter a difficult configuration, the algorithm slowly disambiguates it and given more time, may have untangled and terminated.

**(B) Cable or knot leaves observable/reachable workspace:** This is the next most common failure mode of SGTM 2.0. While performing a cage-pinch dilation or Reidemeister move, one gripper may miss the grasp, causing the entire cable to slide to one side of the workspace and fall off entirely and irrecoverably. This failure mode shows that slack management can be improved in future work.

**(C) False termination due to missed knot detection:** False termination is the most common failure mode in SGTM 2.0(-U), mostly resolved by SGTM 2.0 with uncertainty-based components. This failure also occurs in SGTM 2.0, largely due to rarer cases where the knotted portion inadvertently lands outside the observable workspace during an incremental Reidemeister move, causing early termination. This can be addressed with improved motion primitives.

**(D) Irrecoverable YuMi system error:** These relatively rare issues result from the YuMi losing connection to the computer running the algorithm and freezing.

### D. Ablations

We run ablations on Tier 2 and Tier 3 to compare the performance of SGTM 2.0 to the performance of SGTM 2.0(-U), which uses the exact same algorithm as SGTM 2.0, but with the following uncertainty-based components removed:

- Reidemeister move due to tracing uncertainty.

- Ensemble network for keypoint predictions and partial cage-pinch dilation in the case of ensemble uncertainty in the cage-pinch dilation network.
- Intermediate views for incremental Reidemeister move.

We find, as shown in table I, that SGTM 2.0(-U) achieves a lower success rate than SGTM 2.0 on Tier 2. While SGTM 2.0 and SGTM 2.0(-U) achieve the same success rate on Tier 3, the main failure case for SGTM 2.0 is timeout as higher complexity cases tend to require more time to disambiguate and untangle. In comparison, the main failure case for SGTM 2.0(-U) are false termination. In fact, the most common failure case in SGTM 2.0(-U) across Tier 2 and 3 is false termination, suggesting that sensitivity to observational uncertainty may be important for higher performance. Another implicit failure that results in more false terminations is over-tightening of knots to a diameter  $\leq 3cm$ . If the ensemble cage-pinch network has low confidence, SGTM 2.0 performs a partial cage-pinch dilation rather than a full cage-pinch dilation, preventing over-tightening knots in the case of poorly predicted cage-pinch points. Additionally, if the trace to a knot is uncertain, the Reidemeister move disambiguates the cable state, preventing poor grasps that may tighten or complicate the knots. The ablations suggest that these uncertainty-based primitives may prevent the over-tightening of knots and thus reduce false terminations.

## VI. CONCLUSION

In this paper, we significantly extend our prior work on SGTM 1.0 to present SGTM 2.0, with novel uncertainty-based and active perception actions. SGTM 2.0 achieves an average untangling success rate of 83% and average rollout time of 351 seconds on cables with 1 or 2 knots, outperforming the prior state-of-the-art, SGTM 1.0, in untangling success by 43% and in untangling speed by 200%. We find that introducing interactive perception – actively manipulating the cable to facilitate perception – improves untangling success on complex cases by 21%.

In future work, we will explore adding interactive perception and uncertainty modeling into other parts of the pipeline as well as generalization to more knot and cable types provided sufficient data. Another potential direction involves eliminating depth sensing and using interactive perception to achieve robust grasps with just RGB data.

## REFERENCES

- [1] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.
- [2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.
- [3] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2074–2081.
- [4] V. Viswanath, K. Shivakumar, J. Kerr, B. Thananjeyan, E. Novoseller, J. Ichnowski, A. Escontrela, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Autonomously untangling long cables," *Robotics: Science and Systems (RSS)*, 2022.
- [5] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.
- [6] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," 2020.
- [7] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," 2021.
- [8] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2146–2153.
- [9] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.
- [10] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," *Robotics: Science and Systems (RSS)*, 2019.
- [11] R. Lee, D. Ward, A. Cosgun, V. Dasagi, P. Corke, and J. Leitner, "Learning arbitrary-goal fabric folding with one hour of real robot experience," *Conference on Robot Learning*, 2020.
- [12] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.
- [13] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.
- [14] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, "Learning predictive representations for deformable objects using contrastive estimation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 564–574. [Online]. Available: <https://proceedings.mlr.press/v155/yan21a.html>
- [15] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9411–9418.
- [16] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali *et al.*, "Learning dense visual correspondences in simulation to smooth and fold real fabrics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 515–11 522.
- [17] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *Robotics: Science and Systems (RSS)*, 2020.
- [18] H. Zhang, J. Ichnowski, D. Seita, J. Wang, H. Huang, and K. Goldberg, "Robots of the lost arc: Self-supervised learning to dynamically manipulate fixed-endpoint cables," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4560–4567.
- [19] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8282–8289.
- [20] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," *International Conference on Intelligent Robots and Systems (IROS)* 2022, 2022.
- [21] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," in *International Symposium on Robotics Research*, 2022.
- [22] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," 2018.
- [23] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali *et al.*, "Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images," 2021.
- [24] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," *Robotics: Science and Systems (RSS)*, 2019.
- [25] X. Lin, Y. Wang, Z. Huang, and D. Held, "Learning visible connectivity dynamics for cloth smoothing," in *Conference on Robot Learning*. PMLR, 2022, pp. 256–266.
- [26] V. Viswanath, J. Grannen, P. Sundaresan, B. Thananjeyan, A. Balakrishna, E. Novoseller, J. Ichnowski, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Disentangling dense multi-cable knots," 2021.
- [27] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, M. Hwang, V. Viswanath, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Untangling dense knots by learning task-relevant keypoints," *Conference on Robot Learning*, 2020.
- [28] W. H. Lui and A. 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.
- [29] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, J. Ichnowski, E. Novoseller, M. Hwang, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Untangling dense non-planar knots by learning manipulation features and recovery policies," 2021.
- [30] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.
- [31] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.
- [32] K. Y. Goldberg and R. Bajcsy, "Active touch and robot perception," *Cognition and Brain Theory*, vol. 7, no. 2, pp. 199–214, 1984.
- [33] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8338–8344.
- [34] C. J. Tsikos and R. K. Bajcsy, "Segmentation via manipulation," *Technical Reports (CIS)*, p. 694, 1988.
- [35] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "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.
- [36] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1862–1868.
- [37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CVPR*, 2015.
- [39] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," 2011.