# Robotic Cable Routing with Spatial Representation

Shiyu Jin[1,2], Wenzhao Lian[1], Changhao Wang[2], Masayoshi Tomizuka[2], and Stefan Schaal[1]

***Abstract*—Cable routing is a challenging task for robotic automation. To accomplish the task, it requires a high-level path planner to generate a sequence of cable configurations from the initial state to the target state and a low-level manipulation planner to plan the robot motion commands to transit between adjacent states. However, there are yet no proper representations to model the cable with the environment objects, impeding the design of both high-level path planning and low-level manipulation planning. In this paper, we propose a framework for cable routing with spatial representation. For high-level planning, by considering the spatial relations between the cable and the environment objects such as fixtures, the proposed method is able to plan a path from the initial state to the goal state in a graph. For low-level manipulation, multiple manipulation primitives are efficiently learned from human demonstration, to configure the cable to planned intermediate states leveraging the same spatial representation. We also implement a cable state estimator that robustly extracts the spatial representation from raw RGB-D images, thus completing the cable routing framework. We evaluate the proposed framework with various cables and fixture settings, and demonstrate that it outperforms some baselines in terms of reliability and generalizability. Experiment videos and details are available at https://github.com/shiyujin0/cable_routing/blob/gh-pages/index.md.***

***Index Terms*—Manipulation Planning, Deep Learning for Visual Perception, Deformable Object Manipulation**

## I. Introduction

ROBOTIC cable manipulation has a wide range of applications, such as wire harnessing, thread packing, and surgical suturing [1]–[3]. While we have seen many studies in cable manipulation in recent years, robots can achieve only limited autonomy [4]–[6]. The main difficulty lies in the fact that cables have infinite degrees of freedom, hindering visual perception and planning. In addition, cables are under-actuated with a large action space and may deform to unexpected shapes during manipulation. These problems become more challenging for cable routing [7, 8] as it requires both high-level long-horizon planning and low-level contact-rich manipulation interacting with the environment objects such as fixtures.

In this paper, we consider a cable routing task (Fig. 1), where cables need to follow a designated path constrained by a set of fixtures on the table. Given randomly placed fixtures and a goal cable configuration, robots need to manipulate the cable from an initial configuration to the goal configuration.

Manuscript received: October, 22, 2021; Revised January, 20, 2022; Accepted February, 18, 2022.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

[1]Intrinsic Innovation LLC in CA, USA. {wenzhaol, sschaal}@intrinsic.ai.

[2]University of California, Berkeley in CA, USA. {jsy, changhaowang, tomizuka}@berkeley.edu. This research was conducted during Shiyu Jin's internship at Intrinsic.

Digital Object Identifier (DOI): see top of this page.

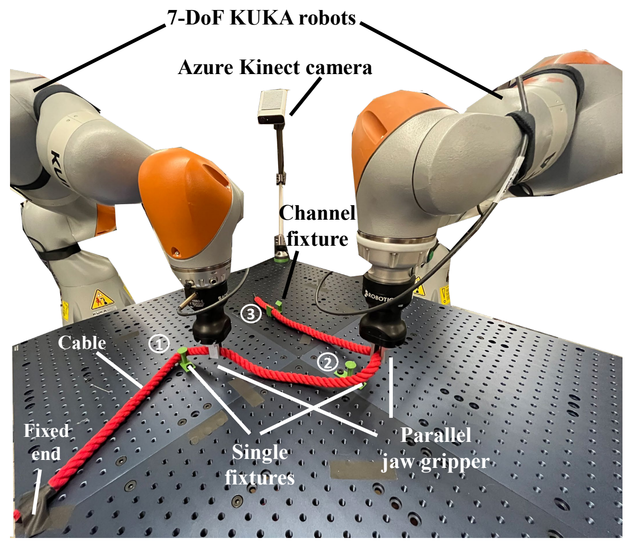


Fig. 1. Cable routing task setup. Two robot manipulators attempt to manipulate the cable to the goal configuration, which is constrained by several fixtures, through a sequence of picking, placing, and holding actions.

Sense-plan-act is an effective framework tackling the routing problem, which consists of 1) visual perception, 2) intermediate configuration planning, and 3) low-level manipulation planning and execution. However, there are multiple practical challenges preventing this method from being widely adopted. 1) For visual perception, a chain of connected nodes are commonly used to represent the cable state, where the node positions are estimated from the point cloud of the cable. The node estimation is significantly affected by the quality of the segmented cable point cloud and lacks robustness with conventionally used color filters [9]–[12] requiring manual tuning and susceptible to environment lighting changes. 2) For configuration planning, human demonstrated sequences composed of intermediate cable states are often needed, and the robots can finish the task following the predefined sequence [13]. Demonstrating the full routing sequence requires extensive human efforts and does not generalize when the cable configuration changes. 3) During manipulation, the cable can easily deform to unexpected shapes. An over-stretched cable may break the cable or fixtures, while a slack cable may fail to reach the desired configuration due to the under-actuated dynamics.

To address the challenges in planning and manipulation, we propose a simple yet effective representation, called spatial representation, to model the spatial relations between the cable and environment objects (namely fixtures). The core insight of this representation comes from an empirical observation: the spatial relation between the cable and fixtures, instead of their accurate positions, contains the full information relevant to the routing task. With the proposed spatial representation, configuration planning can be efficiently achieved by searching a path from the initial to the goal state without the need for human demonstration. In addition, this representation enables efficient data collection and model learning for low-level manipulation. We design three low-level manipulation primitives, `stretch`, `cross`, and `insert`, to manipulate the cable from one spatial representation to another. A manipulation primitive takes the cable state, fixture positions, and the target spatial representation as input, and outputs picking and placing targets for robots. The `stretch` primitive stretches the cable to another configuration without changing the spatial representation. The `cross` and `insert` primitives change the cable configuration between two spatial representations. As will be shown in Sec. V, cable routing with the learned primitives outperforms rule-based heuristics and achieves reliable performance with diverse cable and fixture settings.

Besides, to address the robustness and generalization challenge in visual perception, we propose a cable state estimator composed of a neural network and non-rigid registration, which is robust to different cable colors and backgrounds. Particularly, the cable segmentation neural network is trained with collected real images and data augmentation without human annotation. Experiments show the cable state estimation is robust to different cable colors and backgrounds as long as the cable and the background have contrasting appearances.

The main contributions presented in this work are summarized as below:

- We proposed a novel cable routing framework built on a simple yet practical insight: the spatial relation between the cable and fixtures encodes sufficient information relevant to the cable routing task. With this novel insight, we put forward a spatial state representation shared across all components in our framework.
- We designed multiple learnable manipulation primitives which, once trained, are able to generate robot commands to manipulate the cable from one spatial representation to another. In the meantime, the learned manipulation primitives are generalizable across different cables.
- To reliably estimate the spatial representation, we presented a cable mask segmentation neural network followed by a designed non-rigid registration step. In addition, we proposed a self-supervised data generation method that efficiently synthesized labeled images for segmentation network training without human annotation.

## II. Related Work

### A. Vision-based State Estimation of Deformable Linear Objects

Cable state estimation in a cluttered environment is a challenging problem as cables, unlike rigid bodies, have infinite degrees of freedom. Different approaches have been developed such as sensor-based estimation [14] and model-based methods [15, 16]. In this section, we focus on vision-based state estimation approaches. One line of work applies end-to-end methods without extracting the structure of the cable, but the representation is not informative for downstream planning [17]–[20]. Another commonly used approach is color filtering, which relies on manual tuning and is sensitive to environmental changes such as lighting conditions [9]–[12]. Though deep neural networks (DNNs) have been widely used in rigid object detection [21]–[24], few works deploy DNNs to cable detection. One major reason is that, unlike rigid bodies, labeling deformable cables is time consuming. [25, 26] learn the cable detection using synthetic data generated in simulation. Although their methods do not require a manually tuned color filter to perceive the cable, there is a large sim-to-real gap when deployed to real-world tasks. Different from above, we train a cable segmentation neural network with collected real images adopting various data augmentation techniques while still avoiding the need of human annotation.

### B. Deformable Linear Object Manipulation

Deformable linear object manipulation has been studied for decades. A randomized algorithm is proposed to plan a collision-free path for elastic objects [27], but the object is not allowed to touch the obstacles in its environment. Minimal-energy curves are applied to plan paths for deformable linear objects in stable configurations [28], but it is difficult to find minimal-energy curves with contact. There are also a lot of works on knot planning [13, 29, 30] and untangling knots [31, 32], which deals with self-contact but does not consider the interaction with the environment.

Different from the above works that do not consider the environment constraints in the workspace, cable routing requires the cable to establish contact with environment objects such as fixtures. [33] formulates a trajectory optimization problem for belt drive unit assembly. [34] generates a visual plan for cable routing tasks via a casual InfoGAN, but the method cannot generate discontinuous actions such as the cable crossing a fixture. [35] proposes a tethered simultaneous localization and mapping method to estimate the state of a robot and any intermediate anchor points resulting from the tether coming into contact with obstacles. [36] analyzes the contact mobility for cable routing around fixtures, but their method requires a customized end-effector that allows the cable to slip. [37] builds a state machine for cable routing, which requires extensive human efforts to choose the state machine parameters and lacks generalization to unseen scenarios. Inspired by [29, 38], which combine topological planning and manipulation primitives to solve long-horizon cable knotting tasks, we introduce spatial representation for cable routing, bridging high-level configuration planning and low-level primitive execution.

## III. Problem Statement

Inspired by the cable routing/USB insertion task in the Robotic Grasping and Manipulation Challenge in IROS 2020 [7], we formulate a simplified and reconfigurable cable routing task as follows (Fig. 1). A cable is placed on a worktable with several fixtures. One end of the cable is rigidly attached to the table. Given a goal configuration, where the cable makes contact with the single fixtures or goes through the channel

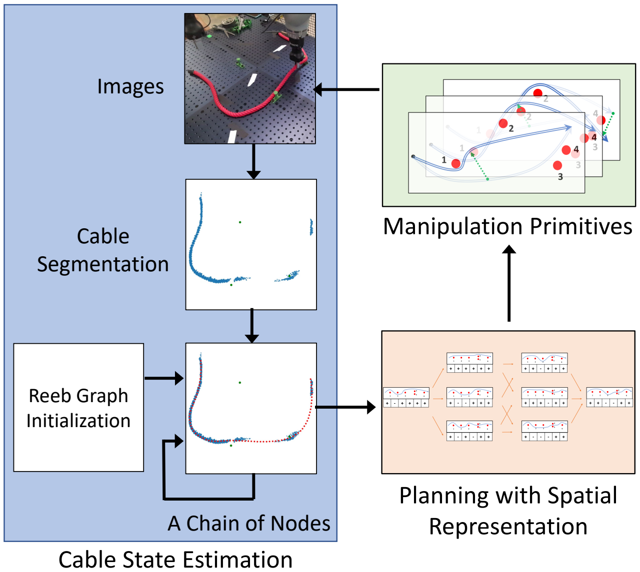


Fig. 2. The proposed cable routing framework consists of three modules: 1) cable state estimation, 2) planning with spatial representation, and 3) learning manipulation primitives.

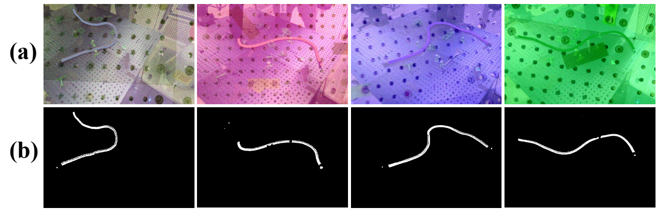


Fig. 3. (a) Synthesized training images with data augmentation. (b) Corresponding cable segmentation masks.

fixtures in a particular order (as illustrated in Fig. 1), two robotic manipulators attempt to manipulate the cable to the goal configuration through a sequence of picking, placing, and holding actions. We make the following assumptions on the task: 1) the number of fixtures and their positions are known in advance; 2) the parallel-jaw grippers can firmly grasp the cable during execution; 3) the cable is within the RGB-D camera's field-of-view throughout the task; 4) the cable is distinguishable by color contrast to its background.

Inspired by [29, 38], we decompose the cable routing problem into three subtasks. 1) Perception. A proper state representation of the cable needs to be extracted from the raw RGB-D image. The state extraction has to be robust to the cable color, background, environment lighting, and the cable being partially occluded. 2) Planning. Given the current cable state, positions of the fixtures, and the goal configuration, we need to generate the next intermediate state. This is expected to be generated autonomously without human demonstration. 3) Manipulation. Given the current and next intermediate states, robot commands such as picking and placing poses need to be inferred. To prevent the cable from moving to undesired states, a holding action is introduced after each cable placement as described in Sec. IV-C1.

## IV. Approach

We propose a cable routing framework with three modules: 1) cable state estimation, 2) planning with spatial representation, and 3) learning manipulation primitives (Fig. 2). 1) The cable state estimation module takes an RGB-D image as the input and outputs a chain of nodes representing the cable state. 2) The planning module generates a sequence of intermediate states based on the spatial relation between the estimated cable state and the known fixtures. These intermediate states define a path for the cable to reach the goal from the initial configuration. 3) The manipulation primitives generate different robot actions that can manipulate the cable from the current states to the next intermediate states.

### A. Cable State Estimation

*1) Segmentation neural network:* We propose a self-supervised data generation method that efficiently generates labeled images for cable mask segmentation without human annotation. The idea is to utilize a color filter manually tuned only once to automatically obtain the cable's segmentation mask from a pre-recorded video, in which a unicolor cable is manipulated randomly to different configurations. Then, the video frames and the corresponding masks are augmented with different colors, backgrounds, occlusion, and noise.

Concretely, we choose a cable with a contrasting color (e.g., a red cable) distinguished from the background and hand-design a color filter. We then record a video while an operator manipulates the cable to different configurations, changes the background, and adds occlusion. For each video frame, a cable segmentation mask can be generated automatically with the color filter. Afterwards we augment the data by randomly sampling different color distributions and impainting on the cable's mask. We apply standard augmentation techniques to increase robustness for the cable's mask and the remaining background, such as dropping out small patches and injecting pixel-wise noise [39]. Example training images and corresponding segmentation masks can be found in Fig. 3. Finally, the augmented data, which consists of image and segmentation mask pairs, is utilized to train a U-Net [21]. The trained U-Net for cable mask segmentation can be deployed with zero cost at inference time, removing the need of hand-tuning filters for each cable.

*2) Cable initialization with multi-resolution Reeb Graph:* To facilitate the downstream planning and manipulation, a chain of nodes is preferred to represent the cable state, as shown in Fig. 2. With the cable mask obtained via the segmentation neural network, the corresponding cable point cloud is retrieved from the depth image. We then construct the cable nodes from the point cloud with multi-resolution Reeb Graph [9, 40]. Specifically, the segmented points are grouped into different clusters, and the cluster centers, i.e., the cable nodes, are connected to form a smooth path while penalizing the path length and angle changes.

*3) Non-rigid registration:* Although cable initialization with Reeb Graph can handle minor occlusion, it would fail in practice because of heavy occlusion by the robot arms during task execution. To tackle this issue, we find the node positions at the current step $X^t = [x_1^t, x_2^t, \ldots, x_N^t] \in \mathbb{R}^{N \times D}$ with non-rigid

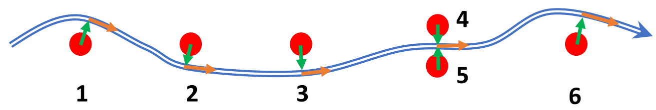


Fig. 4. An example spatial representation. Red dots 1,2,3,6 are single fixtures. Red dots 4 and 5 form a channel fixture. The cable from the fixed end to the loose end is traced following the orange arrows. The green arrows indicate the projection of fixtures onto the cable. The spatial state for each fixture is determined by the sign of the cross product between the green and orange arrows. In this particular example, the spatial representation is (+,-,-,-,+,+). Intuitively, the spatial representation means that the six fixtures are on the (right, left, left, left, right, right) side of the cable.

registration, given the node positions from the previous step $X^{t-1}$ as a prior, where $x_i^t$ is the position of *i-th* node at *t-th* frame, and $D$ is the dimension of node positions ($D=3$ in our case). $N$ is the number of nodes, which needs to be a constant throughout the registration steps ($N=50$ in our experiments). We initialize the cable nodes $X^0$ from the result of Reeb Graph as described in Sec. IV-A2. For each new frame, the cable nodes $X^t$ can be obtained from the current point cloud $Y^t=[y_1^t,y_2^t,...,y_M^t]\in\mathbb{R}^{M\times D}$ and $X^{t-1}$ via Coherent Point Drift (CPD) [41], where $y_i^t$ is the position of *i-th* point in the point cloud, $M$ is the number of points in the point cloud, and usually $M>>N$.

One of the problems with the registration cables using CPD is that the registered nodes do not have equal distances between neighboring nodes. [11] adds a regularization term in its optimization objective to maintain the local structure. Here we apply a simpler remedy by connecting the registered nodes and re-sampling along the connected path to obtain equally distributed nodes, which is found effective in our experiments.

### B. Planning with Spatial Representation

Empirically, we observed that matching the cable nodes exactly with their correspondences on the goal configuration is unnecessary. Rather, it is sufficient that the spatial relation between the cable and the fixtures matches the goal configuration.

We define the spatial representation for cable routing in the 2$D$ horizontal plane. The positions of cable nodes projected in the horizontal plane are denoted as $\tilde{X}=[\tilde{x}_1,\tilde{x}_2,...,\tilde{x}_N]\in\mathbb{R}^{N\times 2}$. The fixture positions are denoted as $P=[p_1,p_2,...,p_K]\in\mathbb{R}^{K\times 2}$, where $K$ is the number of fixtures. We define two directional vectors $\vec{v}_{i1}=\tilde{x}_j-p_i$ (green arrows in Fig. 4) and $\vec{v}_{i2}=\tilde{x}_{j+1}-\tilde{x}_{j-1}$ (orange arrows in Fig. 4), where $x_j$ is the closest nodes to the *i-th* fixture $p_i$. The spatial representation for each fixture is then defined with a plus/minus sign, as illustrated in Fig. 4. Formally, the spatial states are determined as $\mathbf{s}=[s_1,s_2,\cdots,s_K]\in\{-1,+1\}^K$, where $s_i=\text{Sign}(\frac{\vec{v}_{i1}\times\vec{v}_{i2}}{|\vec{v}_{i1}\times\vec{v}_{i2}|}\cdot\vec{e}_z)$ and $\vec{e}_z$ is the unit normal of the horizontal plane. Intuitively, if we trace the cable from the fixed end to the loose end, the spatial representation indicates whether each fixture is on the "left" or "right" side of the cable. The channel fixtures are treated as two single fixtures, e.g., fixture 4 and 5 in the example. Fig. 5 demonstrates a few example cable configurations along with their corresponding spatial states. Note that we do not consider the scenario where the cable circles around the fixtures.

Leveraging the proposed representation, a high-level planner for the long-horizon cable routing tasks can be easily implemented by only allowing single element changes in the spatial state $\mathbf{s}$ in each step[1]. Fig. 5 illustrates example paths and intermediate states searched connecting the initial to the goal spatial state. In practice, we select the path along which the spatial state $\mathbf{s}$ changes sequentially from the cable's fixed end to the tail since it facilitates the downstream manipulation.

### C. Learning Manipulation Primitives

In order to manipulate the cable to a desired spatial state, low-level robot commands such as picking and placing poses are needed. The low-level planner should be able to handle different cable configurations as well as diverse fixture locations.

*1) Manipulation primitives:* We propose a low-level cable routing planner with three manipulation primitives, `stretch`, `cross`, and `insert`, as demonstrated in Fig. 6. Each primitive consists of a pick-move-place action sequence where the picking and placing actions are constrained to be vertical. Specifically, in `stretch`, a robot picks a point on the cable and stretches the slack cable to establish contact with the fixtures. In `cross`, a robot selects one point on the cable to pick and transports the cable from one side of the fixture to the other. In `insert`, a robot picks the cable and inserts it between two fixtures of the channel.

As shown in Fig. 6, `cross` and `insert` change the spatial state, while `stretch` does not. The purpose of `stretch` is to reshape the cable to robustify `cross` and `insert`. In practice, after each `stretch`, one robot is holding the cable at the placing location, while the other robot performs `cross` or `insert`. The holding action is crucial since it prevents the cable from moving to undesired states during `cross` or `insert`. By iteratively executing `stretch-cross` or `stretch-insert`, robots are able to manipulate the cable to the desired configuration following the planned path in the spatial state space.

*2) Learning manipulation primitives from labeled data:* Each primitive consists of a picking point and a placing point. We propose to learn the target points from real data collected by randomly configuring the cable and fixtures in the workspace. For each collected RGB-D image, the cable state is estimated as described in Section IV-A. With the known fixture locations, we synthesize an image plotting the cable nodes and fixtures, illustrated in Fig. 7. Based on the prompted target spatial representation and the current spatial state, humans will annotate the picking and placing points on the synthesized image.

Note that only the horizontal 2$D$ positions of the target points are learned while the height is assumed to be known. The orientation of the picking point is computed using the cable nodes' positions outputted from the cable state estimation, i.e., the yaw angle is the same as the estimated cable direction at the picking position. The orientation of the placing

[1]It is feasible, but not reliable to change the spatial relation between the cable and multiple fixtures with one step of robotic grasping and placing operations.

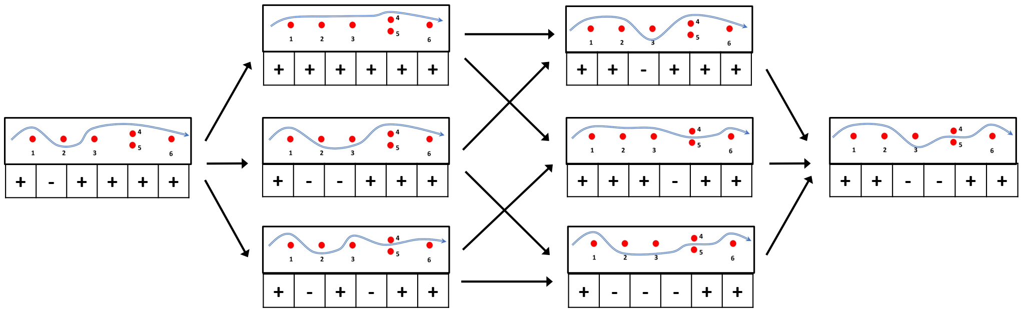


Fig. 5. Planning from the initial spatial representation to the goal spatial representation. Intermediate states are generated along each path, where the spatial representation vector changes in one and only one dimension at every step.

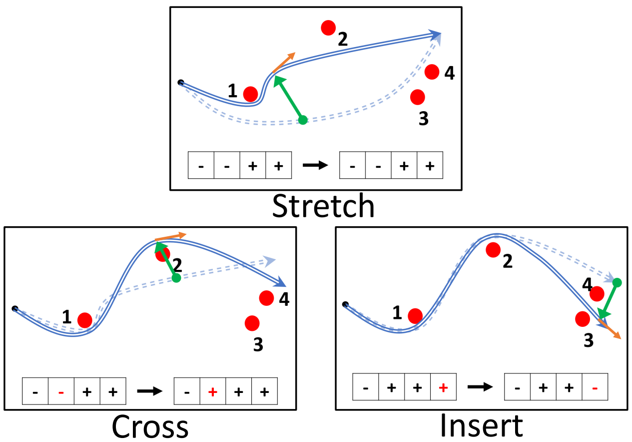


Fig. 6. Manipulation primitives: (a) In `stretch`, a robot stretches the slack cable to establish contact with the fixtures. (b) In `cross`, a robot transports the cable from one side of the fixture to the other. (c) In `insert`, a robot inserts the cable between two fixtures of the channel. `cross` and `insert` change the spatial state, while `stretch` does not. The green arrows represent the picking and placing actions for robots. The orange arrows represent the orientation of the placing point.

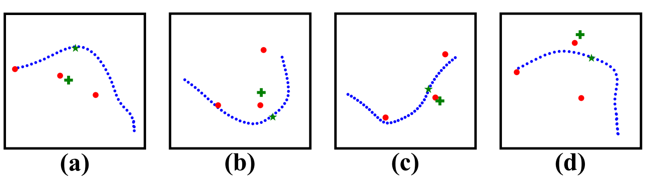


Fig. 7. Example labeled data for learning manipulation primitives. Cable nodes (blue dots) and fixtures (red dots) are plotted in a synthesized image. Humans annotate the picking point as a green star and the placing point as a green plus. (a) and (b) are demonstration examples for `stretch`. (c) and (d) are demonstration examples for `across`.

point, as shown in Fig. 6, is selected based on the desired spatial state. Specifically, the placing yaw angles are defined by the segment connecting the last and the next fixtures for `cross`, and by the segment connecting the placing position and the next fixture for `stretch` and `insert`. Learning the orientation and further, a 6-DoF pose is left to future work.

Two methods are implemented and compared to learn the target positions. In the first method, direct regression is applied using a Multilayer Perceptron (MLP) with two fully connected layers, whose inputs include cable nodes' positions, fixtures' positions, and the target spatial state. The output is the concatenated vector of the 2*D* picking and placing positions. In the second method, the cable nodes' and fixtures' positions are encoded in an image as the input to a U-Net [21], and similarly, the picking and placing target positions are encoded in a heatmap as the output. During training, the output heatmaps are constructed by the convolution of the point coordinates with a Gaussian kernel $\Phi = exp(-\frac{||p-p^*||^2}{2\sigma^2})$, where we select $\sigma = 4$ pixels. We hypothesize that outputting target point heatmaps allows for better spatial generalization than direct regression on point coordinates. A quantitative analysis on the two methods will be performed in Sec. V.

## V. Experiments

We aim to investigate three questions in our real-world experiments. First, we examine if the proposed framework based on spatial representations can solve cable routing tasks with various cables and fixture settings. Second, we evaluate whether the learned manipulation primitives outperform hand-designed policies and which learned model performs better. Third, we inspect whether the perception based cable state estimator provides reliable estimates for downstream planning and manipulation.

### A. Experimental Setup

As shown in Fig. 1, our system includes two 7-DoF KuKa IIWA robot manipulators, a Kinect Azure RGB-D camera, two Robotiq Hand-E grippers, several single and channel fixtures, and 1 non-stretchable deformable cable. We configured four routing scenarios with different goal configurations, as shown in Fig. 8. In each scenario, one end of the cable is rigidly attached to the table, and the fixtures are fixed on the table with known positions and orientations. We also conducted experiments on 7 different cables with varying colors, thicknesses, and physical properties, as shown in Fig. 9. We assume that the cable is within the robots' workspace throughout task execution, and collision-free robot motion sequences exist to manipulate the cable to desired configurations. Solving cable routing tasks with more dense fixtures and allowing 6-DoF picking/placing are left to future work.

### B. Implementation Details

*1) Rule-based manipulation baseline:* To evaluate the learned manipulation primitives, we implemented a rule-based

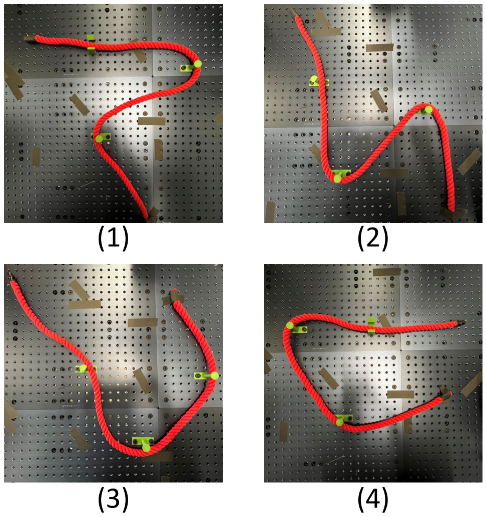


Fig. 8. Four experiment scenarios where the fixture setting and target cable configurations are different.

policy that is composed of the same set of primitives as in Sec. IV-C1, except that the picking and placing points are selected according to heuristics. Specifically, we compute the distance from all cable nodes to the relevant fixture and find the closest cable node $\tilde{x}_{closest}$ to the fixture. The node that is $c$ nodes away from $\tilde{x}_{closest}$ is selected as the picking location, and the placing location is chosen as $d$ distance away from the relevant fixture along the desired direction. By tuning $c$ and $d$, the rule-based method achieves decent performance in each specific scenario, but a fixed value pair is found difficult to generalize. We experimentally set $c = 5$ and $d = 0.05m$ in all scenarios.

*2) Model learning for manipulation primitives:* For each manipulation primitive, 100 human demonstrations are collected, with varying fixture locations and different start and end cable configurations. We then annotate the picking and placing locations for each demonstration. The annotated dataset is augmented with flipping, rotating, injecting noise, and node resampling along the cable. Resampling nodes is crucial to data efficiency as the exact node locations are irrelevant to the task while the spatial relation between nodes and the fixtures matters.

*3) Parameters for cable state estimation:* In the experiments, we use $N = 50$ nodes to represent approximately $0.8m$ long cables. For the cable initialization with multi-resolution Reeb Graph, voxels of $0.05m$ resolution are used to group the point cloud.

### *C. Results*

Table I compares cable routing success rates on the red rope with differently acquired primitives in all four scenarios (Fig. 8). In each scenario, there are three single/channel fixtures. A trial is counted as success only if the goal configuration is achieved between the cable and all three fixtures. As seen in the table, the rule-based policy does not perform reliably in different cable and fixture configurations. The regression method achieves decent performance, but its predicted

TABLE I
COMPARISON OF METHODS WITH DIFFERENTLY ACQUIRED PRIMITIVES. FAILURE MODES: A(OVER-STRETCHING), B(SLACK AND FAILED TO CROSS), C(FAR-OFF PREDICTION)

| | Methods | Success rate | Failure modes |
|---|---|---|---|
| Scenario 1 | Rule-based | 2/5 | A(2), B(1), C(0) |
| | Regression | 4/5 | A(1), B(0), C(0) |
| | Heatmap | **5/5** | A(0), B(0), C(0) |
| Scenario 2 | Rule-based | 1/5 | A(1), B(3), C(0) |
| | Regression | **4/5** | A(1), B(0), C(0) |
| | Heatmap | **4/5** | A(0), B(0), C(1) |
| Scenario 3 | Rule-based | 1/5 | A(4), B(0), C(0) |
| | Regression | 3/5 | A(0), B(2), C(0) |
| | Heatmap | **4/5** | A(1), B(0), C(0) |
| Scenario 4 | Rule-based | 0/5 | A(3), B(2), C(0) |
| | Regression | 0/5 | A(0), B(5), C(0) |
| | Heatmap | **2/5** | A(0), B(3), C(0) |
| Overall | Rule-based | 4/20 | A(10), B(6), C(0) |
| | Regression | 11/20 | A(2), B(7), C(0) |
| | Heatmap | **15/20** | A(1), B(3), C(1) |

TABLE II
CABLE ROUTING WITH DIFFERENT CABLES. FAILURE MODES: A(OVER-STRETCHING), B(SLACK AND FAILED TO CROSS), C(FAR-OFF PREDICTION), D(WRONGLY ESTIMATED CABLE STATE)

| Cables | Success rate | Failure modes |
|---|---|---|
| Rope (red) | 5/5 | A(0), B(0), C(0), D(0) |
| Rope (pink) | 5/5 | A(0), B(0), C(0), D(0) |
| Rope (lime) | 3/5 | A(0), B(0), C(0), D(2) |
| Rope (orange) | 4/5 | A(0), B(0), C(1), D(0) |
| Rope (blue) | 5/5 | A(0), B(0), C(0), D(0) |
| Thin Rope (red) | 3/5 | A(0), B(1), C(1), D(0) |
| USB Cable (red) | 3/5 | A(0), B(2), C(0), D(0) |

target points have a large variance leading to over-stretching or slack cables. In addition, there are occasions where the cable or the fixture might break if not interrupted. By contrast, learning with encoded heatmaps achieves a higher success rate with few over-stretching or slack cable cases. This suggests that outputting heatmaps allows for better spatial generalization than directly regressing on point coordinates. Fig. 8 shows some example trials where cables are manipulated to the desired goal spatial states. It is worthwhile to note that one failure mode of heatmap prediction is “far-off prediction”, where the predicted heatmap has similar values across the image. Thus, applying *argmax* over the heatmap could result in predicting a pixel that is distant from the desired location. The reason is that this particular cable configuration is not close to the configurations seen in the training data. Increasing the demonstration data diversity can further improve the performance.

Table II shows the experimental results on different cables (Fig. 9) with heatmap-based learned primitives. Overall, a success rate of 28 out of 35 is achieved, showing promising performance of our cable routing method applied to various cables. In this experiment, we additionally evaluate the cable state estimation performance. A state estimate is labeled by human as “failed” if the chain of estimated nodes does not visually match the skeleton of the real cable. As summarized in failure mode D, the cable state estimation succeeds in 33 out of 35 trials. This illustrates the proposed cable state estimator is able to detect cables of different colors and thicknesses in diverse configurations, although only limited data with the red

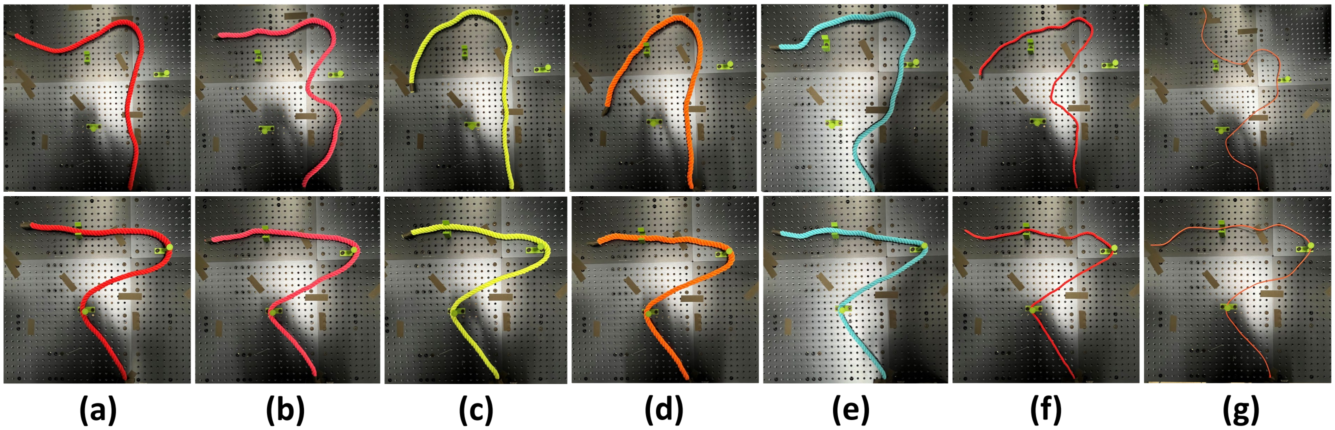


Fig. 9. Cable routing with different cables for the same target spatial state (scenario 1). First row: initial state. Second row: final state. (a) red rope. (b) pink rope. (c) lime rope. (d) orange rope. (e) blue rope. (f) red thin rope. (g) red USB cable.

rope was accessible during training. The inferred segmentation mask sometimes contains wrong predictions, especially when the robot partially occludes the cable. Even with a lower quality mask prediction, the cable state estimation remains accurate most times thanks to the cable initialization and node registration being robust to occlusion and outliers, as shown in the left plot of Fig. 2.

Note that Table II also reports the performance of the proposed method with cables of different physical properties, such as thinner/softer ropes and a USB cable. Although the dynamics of the unseen cables are different from the trained thick rope, the `stretch` primitives and holding actions with reasonably predicted target points alleviate the effect of cable dynamics. To further improve the reliability, we suggest exploring real-time cable tracking and leveraging force/torque feedback. In addition, higher-bandwidth reactive planning and execution would help to detect failure modes early and recover from them.

## VI. Conclusion and Future Work

This work proposes spatial representation for cable routing, which bridges the high-level long-horizon configuration planning and the low-level manipulation primitive execution. A simple configuration planner is implemented with the proposed representation to achieve the desired spatial relationship between the cable and fixtures. In addition, multiple execution primitives are designed and learned from the collected data, including stretching, crossing, and inserting. Real-world cable routing experiments are conducted with multiple cables, varying in visual appearances, physical properties, and fixture settings, demonstrating the method's effectiveness.

It is noteworthy that our framework is evaluated in simplified scenarios where the cable is approximately 2D and there is no heavy occlusion. By virtue of our modular framework, more sophisticated cable state estimators robust to occlusions and manipulation primitives reliable in cluttered 3D environments can be adopted in more complex scenarios. For example, a cable state tracker modeling the temporal correlations and manipulation primitives that plan locally to navigate with real-time updated 3D point clouds, are left as future directions to explore.

## References

[1] 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.

[2] 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.

[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] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, X. Li, J. Pan *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *arXiv preprint arXiv:2105.01767*, 2021.

[5] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 479–484.

[6] S. Jin, C. Wang, and M. Tomizuka, "Robust deformation model approximation for robotic cable manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6586–6593.

[7] "https://www.nist.gov/el/intelligent-systems-division-73500/iros-2020-robotic-grasping-and-manipulation-competition."

[8] "https://www.nist.gov/el/intelligent-systems-division-73500/iros-2021-robotic-grasping-and-manipulation-competition."

[9] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1130–1137.

[10] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, "State estimation for deformable objects by point registration and dynamic simulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2427–2433.

[11] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, p. 0278364919841431, 2018.

[12] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6443–6450.

[13] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.

[14] J. E. Frank, R. Geiger, D. R. Kraige, and A. Murali, "Smart tether system for underwater navigation and cable shape measurement," May 7 2013, uS Patent 8,437,979.

[15] O. A. N. Eidsvik and I. Schjølberg, "Finite element cable-model for remotely operated vehicles (rovs) by application of beam theory," *Ocean Engineering*, vol. 163, pp. 322–336, 2018.

[16] D. S. D'antonio, G. A. Cardona, and D. Saldaña, "The catenary robot: Design and control of a cable propelled by two quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3857–3863, 2021.

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

[18] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," *arXiv preprint arXiv:1910.13439*, 2019.

[19] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for physical sequential fabric manipulation," *arXiv preprint arXiv:2102.09754*, 2021.

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

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[22] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[25] 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.

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

[27] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 188–208, 2001.

[28] M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 625–636, 2006.

[29] M. Yan, G. Li, Y. Zhu, and J. Bohg, "Learning topological motion primitives for knot planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9457–9464.

[30] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot planning from observation," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 3887–3892.

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

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

[33] S. Jin, D. Romeres, A. Ragunathan, D. K. Jha, and M. Tomizuka, "Trajectory optimization for manipulation of deformable objects: Assembly of belt drive units," *arXiv preprint arXiv:2106.00898*, 2021.

[34] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," *arXiv preprint arXiv:1905.04411*, 2019.

[35] P. McGarey, K. MacTavish, F. Pomerleau, and T. D. Barfoot, "Tslam: Tethered simultaneous localization and mapping for mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1363–1386, 2017.

[36] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2019.

[37] G. Arslan Waltersson, "Planning and control for cable-routing with dual-arm robot," 2021.

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

[39] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[40] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3d shapes," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 203–212.

[41] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.