Chopstick-Inspired Non-Prehensile Manipulation for Datacenter Automation

Harshit Khurana, Elliott Hogg, Freddie Hong, Iason Sarantopoulos, Fabian Otto, Iris Andrussow
Hugh Williams, Tim Deegan, David Sweeney, Andromachi Chatzieleftheriou
Microsoft Research, Cambridge, UK

***Abstract*—Automating repair and maintenance tasks in datacenters requires robots to operate in dense cable environments where overlapping cables occlude target components and must be displaced without damage. Traditional prehensile approaches are impractical in such settings due to the difficulty of isolating individual cables and the risk of entanglement. We present a non-prehensile cable parting approach using a chopstick-inspired tool that slides between cables and creates localized clearance around a target transceiver port. The tool and motion primitives are designed to enable safe cable displacement without requiring explicit force sensing or cable-specific planning. We evaluate the approach across randomized cable configurations in both a MuJoCo physics simulation and real-world experiments using a dual-arm UR5e system, where one arm performs cable decluttering and a second arm inserts a transceiver into the cleared port. Results are demonstrated across different cable environments, and successful end-to-end transceiver insertion is achieved in the integrated multi-robot pipeline.**

I. Introduction

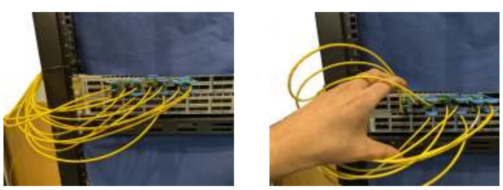


Fig. 1: (Left) Dense cable environment at the back of a datacenter network switch. (Right) A human operator manually parting cables to access a transceiver port.

Deformable Linear Objects (DLOs), such as cables and wires, are ubiquitous in industrial and domestic environments. Unlike rigid objects, cables possess theoretically infinite degrees of freedom, making their configuration difficult to perceive, model, and control [1]. Their thin, flexible geometry leads to frequent self-occlusion, overlap, and entanglement, resulting in knot formation or excessive bending [2]. A particularly relevant and high-impact instance of this challenge arises in datacenter automation. Modern datacenters contain densely packed cable bundles connecting network switches, optical transceivers, and server blades (Fig. 1). Automating maintenance tasks such as inserting or removing an optical transceiver requires a robot to operate within this clutter [3]. In contrast to structured manipulation tasks, the robot cannot simply avoid cables: it must often push aside a small subset of neighboring cables to gain access to a target component, much like a human uses their fingers to part wires temporarily (Fig. 1).

Dense cable environments introduce some fundamental challenges: *clutter*, *occlusion*, and *delicacy*. From a perception standpoint, overlapping cables severely occlude the target port and each other, making reliable state estimation difficult [4], [5]. From a manipulation standpoint, cables are deformable and fragile. Optical fibers, for example, can fracture if bent beyond a small radius or subjected to excessive force. Cluttered cables can easily become entangled depending on their configuration and the order of interactions [2]. This creates a central tension: physical contact with cables is unavoidable, yet unsafe interaction can cause damage.

In cable decluttering, some obstacles must instead be intentionally and safely displaced. Heuristic-based planners augment collision checking with domain-specific rules to guide a gripper through cable clutter [3], but fail in densely cluttered settings. Learning-based approaches train neural networks to predict safe grasping or pushing actions from visual observations [6], [7], but require large datasets and may lack safety guarantees. Dual-arm and tactile-reactive systems provide dexterous cable handling [8], [9], but introduce significant hardware complexity. Traditional prehensile manipulation, which relies on grasping individual cables, is often impractical in dense environments due to the risk of entanglement, the difficulty of isolating a single cable, and the complexity of dense clutter [6]. Consequently, we investigate *non-prehensile cable manipulation* as a more suitable abstraction for this problem.

In this work, we focus on local, task-driven cable parting rather than full cable disentanglement or global reconfiguration [2]. The objective is not to organize the entire cable bundle, but to temporarily create sufficient clearance around a target transceiver port to enable insertion or removal operations. To this end, we explore whether tool-mediated interaction, combined with structured motion strategies, can simplify manipulation in dense cable environments without relying on precise force sensing or complex feedback control.

We propose a chopstick-inspired tool designed specifically for gentle, non-prehensile cable manipulation. The tool geometry enables the robot to slide between cables, part them in controlled directions, and create localized openings while minimizing the risk of excessive bending or tangling. Building on this tool design, we introduce a set of modular motion primitives that can be composed into higher-level manipulation policies. This primitive-based approach draws inspiration from prior work on push-grasp synergies [7], [10] and motion primitives for DLO manipulation [11], but targets the distinct problem of non-prehensile cable parting. These primitives provide an interpretable and flexible way to reason about interaction with cables under uncertainty and variability in cable configurations.

We evaluate our approach in both a MuJoCo-based physics simulation [12] and real-world experiments in a dense datacenter cable environment. By systematically combining motion primitives into different policies, we demonstrate that effective cable decluttering strategies can be constructed without explicit cable grasping or detailed cable-specific planning heuristics.

Contributions of this paper:

- A chopstick-inspired tool for non-prehensile cable parting, tailored for dense and fragile cable environments.
- A set of modular motion primitives enabling safe, interpretable, and composable cable manipulation strategies.
- Systematic evaluation in both physics-based simulation and real-world datacenter experiments, demonstrating effective cable decluttering across varying cable configurations.

II. Problem Scope and Assumptions

Problem definition: Given a target transceiver port on a network switch surrounded by dense cable clutter, the objective is to displace neighbouring cables to create a *clearance region* around the target port.

Assumptions: We make the following assumptions. 1. The position of the target port on the switch is known. 2. Cables are assumed to be flexible and resilient to moderate lateral displacement. 3. No external disturbances or cable motion occurs during policy execution beyond the robot's own manipulation.

III. Related Work

Our work sits at the intersection of non-prehensile manipulation in clutter, deformable linear object (DLO) manipulation, and tool-mediated robotic interaction.

A. Non-Prehensile Manipulation in Clutter

Non-prehensile actions such as pushing and sliding have been widely studied for rigid object rearrangement. [10] proposed a planning framework for push manipulation under clutter and uncertainty, establishing that non-prehensile actions can be more effective than grasping when objects are densely packed. [7] demonstrated that pushing and grasping synergies can be learned end-to-end via deep reinforcement learning, where the robot discovers that pushing objects apart enables subsequent grasps. Similarly, [13] developed goal-oriented push-grasping policies with improved sample efficiency. [14] addressed object singulation in dense clutter using deep Q-learning, while [15] extended this to learn push-grasping strategies that generalize across object geometries. [16] incorporated occlusion reasoning into search-based retrieval in clutter. While these works target rigid objects, the core insight that non-prehensile actions can create access in cluttered environments directly motivates our approach. However, deformable cables introduce fundamentally different challenges: they lack fixed geometry, respond non-rigidly to contact, and risk damage from excessive bending.

B. Deformable Linear Object Manipulation

DLO manipulation has received growing attention, spanning modeling, control, and task-level planning. [1] presented dynamic modeling and control frameworks for single-arm and dual-arm DLO manipulation. [17] introduced a coherent point drift framework for shape-based DLO manipulation from demonstrations. For model-based approaches, [18] developed global model learning for large-deformation control, while [19] proposed online parameter estimation for adaptive DLO manipulation. [20] decoupled prediction, planning, and control for deformable object manipulation, enabling robust task execution. [21] combined learning-based model predictive control with safety filters for constrained DLO tasks.

On the learning side, [22] used self-supervised learning for DLO state estimation, [23] demonstrated sample-efficient real-world DLO manipulation through self-supervision, and [24] learned dynamic cable manipulation policies for fixed-endpoint cables. [25] trained rope manipulation policies using dense object descriptors from synthetic depth data. For cable routing specifically, [26] applied hierarchical imitation learning for multistage cable routing tasks. [27] explored sim-to-real transfer for physics-based robotic deployment of DLOs.

However, these works consider isolated or sparsely arranged cables, where individual cable state can be observed and tracked. In contrast, our work targets *dense cable clutter*, where cables severely occlude each other, individual cable tracking is infeasible, and the goal is local clearance rather than precise shape control.

C. Cable Manipulation in Clutter

Relatively few works have addressed cable manipulation in truly cluttered environments. [3] formulated transceiver access in cluttered cable environments as a heuristic search problem, using a variant of A* in the gripper's configuration space with domain-specific collision rules (e.g., permitting downward cable displacement while penalizing upward pushes). While effective and interpretable, these hand-crafted heuristics may not generalize across diverse and complex cable configurations. [6] trained a collision-aware neural network (CG-CNN) to grasp individual cables from clutter without disturbing neighbors. While this addresses collision avoidance in dense cables, it targets prehensile grasping of a single cable rather than creating clearance for port access. [2] tackled the problem of autonomously untangling long cables, but in a tabletop setting with full observability. [28] presented cable segmentation methods for moving cables, but focused on perception rather than manipulation planning.

D. Tool-Mediated and Tactile Cable Manipulation

Specialized tools and tactile sensing have been explored to improve cable handling. [8] developed a tactile-reactive gripper that uses GelSight sensors to manipulate cables with force feedback, enabling precise cable following and manipulation. However, such systems require sophisticated sensing hardware and are designed for single-cable interaction. [11] combined tactile-driven motion primitives for cable routing and assembly, demonstrating that decomposing cable manipulation into modular primitives improves robustness and interpretability. Dual-arm approaches [1], [9] provide additional dexterity for DLO manipulation but substantially increase system complexity. Nokia Bell Labs demonstrated a mobile robot for fiber patching [29], but in sparse, structured environments unlike the dense clutter we address.

Our approach differs from prior work in several key respects: (1) we design a specialized chopstick-inspired *tool* that simplifies the manipulation problem through geometry (2) we focus on *non-prehensile cable parting*: temporarily displacing cables to create port access rather than grasping, routing, or untangling; and (3) we compose modular motion primitives into policies that can be systematically evaluated across varying cable configurations.

IV. Methodology

A. Chopstick Tool Design

Cable decluttering requires temporarily displacing cables without damaging them. Rather than relying on precise force sensing at the end-effector [8], which becomes increasingly difficult in dense environments with multiple simultaneous contacts, we simplify the interaction through tool design.

We designed an end-effector inspired by traditional chopsticks (Fig. 2), consisting of two parallel prongs actuated by a servo hinge. The chopstick design enables the robot to part cables gently and create the necessary clearance for transceiver insertion or removal. The key design features are:

- *Tapered tips:* The pointed ends allow the tool to slide between densely packed cables with minimal lateral displacement, reducing insertion force.
- *Length:* The prongs are long enough to reach into dense cable bundles without the robot body or wrist colliding with the switch, and short enough for the servo to open the chopsticks under the external contact forces.
- *Parallel opening:* The prongs are constrained to open with equal separation along their length.

(a) (b) (c)

Fig. 2: Chopstick tool design: (a) inspiration from traditional chopsticks, (b) closed configuration, and (c) open configuration.

B. Motion Primitives

One of the challenges of developing a motion policy in a dense cable environment is the high variability of cable configurations. Rather than planning over continuous cable state, we decompose manipulation into a set of modular motion primitives (refer Fig. 3.).

1) **Insert (d):** Inserts the closed chopsticks along the tool axis by distance d, sliding between cables.
2) **Rotate (θ):** Rotates the inserted tool by angle θ about the tool axis to orient the subsequent *Open* to displace cables in a desired direction.
3) **Comb (l):** Translates the tool laterally by distance l while inserted, separating overlapping cables around the target port.
4) **Open (α):** Opens the prongs to angle α, pushing cables apart on either side.

C. Policy Composition

We compose the motion primitives into sequential *policies* that achieves a decluttering objective. Every policy begins with *Insert* (to enter the cable bundle) and ends with *Open* (to create clearance). The intermediate primitives vary:

1) **P1:** Insert $\rightarrow$ Open
2) **P2:** Insert $\rightarrow$ Rotate $\rightarrow$ Open
3) **P3:** Insert $\rightarrow$ Comb $\rightarrow$ Open
4) **P4:** Insert $\rightarrow$ Comb $\rightarrow$ Rotate $\rightarrow$ Open

Adapting to port state: The target port may be in one of two states: *1. Empty Port:* The goal is to clear cables for transceiver insertion. *2. Occupied Port:* The goal is to clear surrounding cables and navigate around the attached cable for transceiver removal.

For occupied ports, the attached cable introduces an additional obstacle. This is handled by inserting extra primitives into the base policy (Table I): an *Open* to separate the chopsticks around the attached cable, followed by a *Comb* to reposition before the final opening. The parameters for these steps are set based on average observed cable diameter and protrusion length.

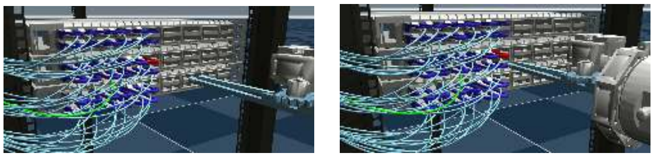
Port Type	**Manipulation Primitives**				
Empty Port	Insert	Comb	-	-	Open
Occupied Port	Insert	Comb	Open	Comb	Open
			added steps		

TABLE I: Policy adaptation for empty and occupied ports.

The modular primitive design enables composability (to create new policies), interpretability (failures can be attributed to specific primitives), allowing for structured data collection for studying cable interactions and training learned models in future work.

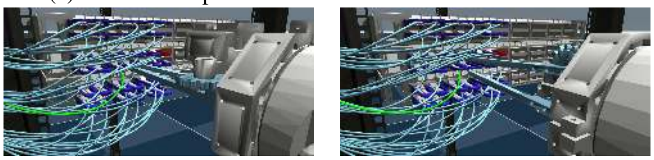
V. Simulation Evaluation

We evaluate the proposed cable manipulation policies in a physics-based simulation before deploying on hardware. The simulation allows us to test across a large number of randomized cable configurations and isolate the effect of each policy on clearance performance. Both scenarios of empty and occupied ports are evaluated in the simulation (refer accompanying video).

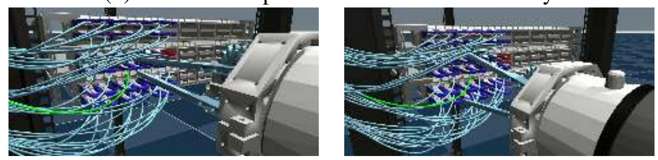


(a) Insert - Chopsticks are inserted into the cable bundle

(b) Comb - Chopsticks are translated laterally.



(c) Rotate - Chopsticks are rotated to orient the subsequent Open.



(d) Open - Chopsticks are opened to create clearance.

Fig. 3: Cable parting primitives

A. Simulation Setup

The simulation environment is built using MuJoCo [12]. The datacenter rack is modeled as a rigid body, the network switch used is ARISTA 7050QX-32S, while cables are modeled as composite soft bodies consisting of connected segments that can bend and flex. Cable geometry is determined using a spline model. The chopstick tool and robot arm are modeled with the same kinematic constraints as the physical setup.

Randomized cable configurations. Since the motion primitives are designed to be independent of the specific cable layout, we evaluate across a range of randomized configurations by varying the following parameters:

- *Cable attachment points:* Vertical locations on the side of the switch where cables are secured to the rack via zip ties $\in (0, 10)cm$.
- *Cable bundling diameter:* How tightly the cables are grouped together at each attachment point $\in (1, 6)cm$.
- *Cable slack:* The amount of cable hanging freely under gravity between attachment points $\in (3, 10)cm$.

B. Evaluation Metric

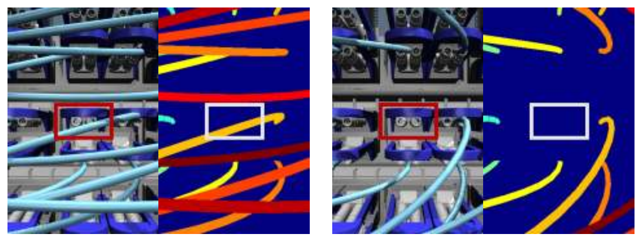
We evaluate each policy using the metric: *clearance rate*. *Clearance rate* is defined as the percentage of trials in which the target port area is fully cleared of cables after executing the cable decluttering policy.

To compute this, we define a rectangular target clearance region of width w and height h centered on the port (refer Fig. 4). The dimensions are set based on the transceiver size plus a safety margin to ensure sufficient clearance for insertion or removal. After policy execution, we perform image segmentation of the scene and check whether any cable segments lie within this region. A trial is considered successful if no cables remain within the clearance region.

Occlusion threshold: Cable occupancy within the clearance region is determined by rendering a segmentation mask from the scene. Clearance is considered a success if the proportion of cable pixels within the clearance region falls below a set occlusion threshold o_{th}.

C. Results

Fig. 4 shows a representative example of a successful decluttering trial. The target port and the clearance region is the area bounded by the red and white boxes (different colours used for contrast and clarity). The segmentation maps confirm that cables are displaced away from the target clearance region after policy execution.



(a) (b) (c) (d)

Fig. 4: Simulation example: cable configuration and segmentation before cable parting ((a) - (b)), and after cable parting ((c) - (d)).

Table II reports the clearance rate for each policy across both empty and occupied port scenarios. Each policy is evaluated over 500 randomized cable configurations.

Policy	Empty (%)	Occupied (%)
P1	93.8	76.8
P2	90.6	87.4
P3	90.8	83.4
P4	93.8	85.6

TABLE II: Clearance rate (%) for different policies in simulation.

All four policies achieve clearance rates above 90% for empty ports, confirming that primitive combinations can effectively create access in dense cable environments. For occupied ports, the presence of the attached cable increases difficulty, yet policies can be created with clearance rates of 87%. These simulation results validate the primitive-based approach in a controlled setting; however, real-world experiments (Section VI) reveal that simulation does not fully capture cable interactions such as friction-induced entanglement and knot formation, leading to larger performance differences.

VI. Real World Evaluation

We validate the proposed approach on physical hardware, evaluating both the chopstick decluttering policies and the transceiver insertion policy individually, as well as the full integrated task where one robot declutters cables and a second robot inserts a transceiver.

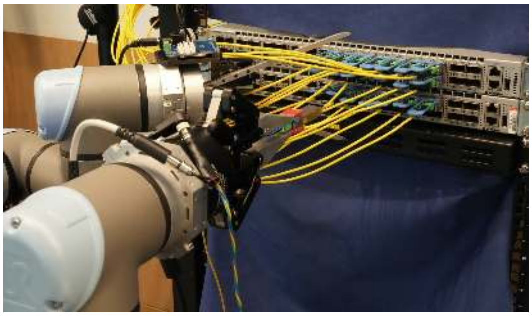


Fig. 5: Dual-arm experimental setup: two UR5e arms on a shared base, one with the chopstick tool (left) and one with a customized Robotiq gripper (right), positioned in front of an ARISTA network switch.

A. Hardware Setup

The experimental platform consists of a dual-arm system with two Universal Robots UR5e manipulators mounted on a shared base (Fig. 5). One arm is equipped with the chopstick tool (Section IV-A) for cable decluttering, while the other is fitted with a customized Robotiq 2F-85 gripper for transceiver insertion. The arms are positioned in front of a 7050QX-32S ARISTA network switch populated with QSFP transceivers and fiber optic cables. The empty port is in the middle of all the populated ports. All the cables are bundled together in the same direction leading to a very dense and chaotic cable environment as all the cables. This ensures that the decluttering policies are tested in the most challenging conditions. An RGB camera (Intel RealSense D405) is mounted on the arm with customized Robotiq 2F-85 gripper for clearance evaluation via image segmentation.

B. Environment Classes

To test generalization across different levels of cable complexity, we define three environment classes, and how we randomize the cable configurations within each class (refer Fig. 6):

- *Structured (S):* Each row of cables is bundled separately with moderate slack, resulting in organized, low density clutter. For changes in the environment, we allow same row transceivers to be randomly changed.
- *Semi-Chaotic (SC):* All cables are bundled together with moderate slack, creating denser overlap. For changes in the environment, we allow same row transceivers to be randomly changed.
- *Chaotic (C):* All cables are bundled together with moderate slack and no organization, producing the most challenging configurations. For changes in the environment, we allow different row transceivers to be randomly changed, leading to unpredictable knots and forces acting on the cables.

C. Standalone Decluttering Evaluation

We first evaluate the four decluttering policies (P1 - P4) from Section IV-C in real cable environments. A robot equipped with the chopstick tool executes each policy to clear a target port, while the *clearance rate* is measured as in simulation (Section V-B). Each policy is executed across 3 different bundling position and 5 randomized cable configurations per bundle position per environment class. So, for each environment class, we have 15 trials per policy, and a total of 45 trials per policy across all environment classes. The same clearance metric and occlusion threshold o_{th} from simulation (Section V-B) is used in the real-world evaluation. Since real images introduce noise from lighting, reflections, and cable appearance variation, the proportion of cable pixels within the clearance region is more sensitive to the choice of o_{th}. To validate robustness, we sweep o_{th} across a range of values and measure the resulting clearance rates for each policy (Fig. 7). P3 and P4 maintain high clearance rates across a wide range of o_{th}, while P1 and P2 are more sensitive, confirming that their lower performance is not an artifact of a particular threshold choice. We select a fixed o_{th} = 1% which creates a balance between sensitivity and the target port to be cleared.

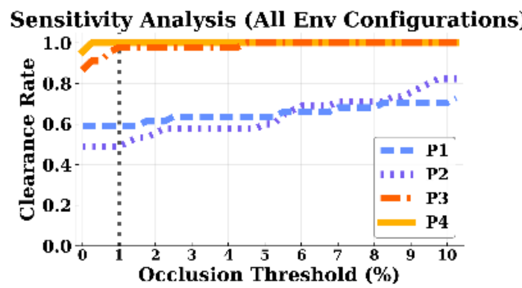


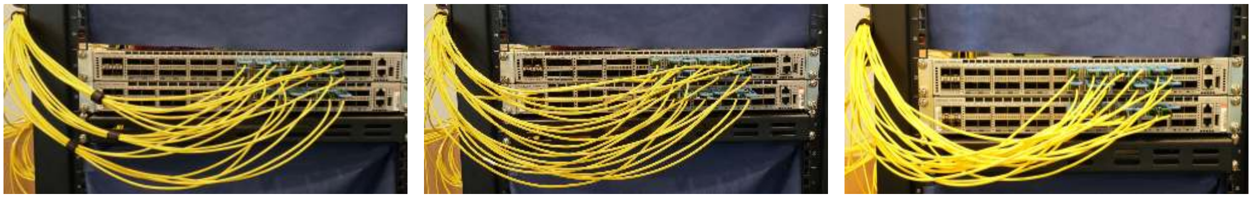
Fig. 7: Clearance rate vs. occlusion threshold o_{th} for each policy. P3 and P4 are stable across a wide range of thresholds.

Fig. 8: Segmentation masks imposed on the camera image before and after cable parting.

Policy	S %	SC %	C %	All %
P1	93	80	0	**57.67**
P2	73	67	7	**49**
P3	100	100	93	**97.67**
P4	100	100	100	**100**

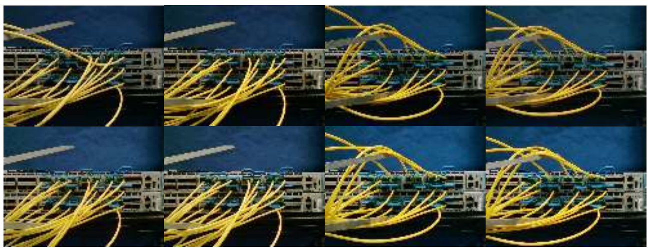
TABLE III: Standalone decluttering policies' clearance rate

The standalone decluttering results (Table III) show that policies P1 and P2, which lack the *Comb* primitive, fail almost entirely in chaotic configurations (0% and 7% respectively). Simply inserting and opening the chopsticks is insufficient when cables are densely interleaved. Adding *Rotate* alone (P2) does not properly resolve this, and in fact slightly reduces performance in structured environments (73% vs. 93% for P1). In contrast, policies that include *Comb* (P3 and P4) achieve near-perfect clearance across all environment classes, with P4 reaching 100% in all conditions. The *Comb* primitive is therefore the critical differentiator:



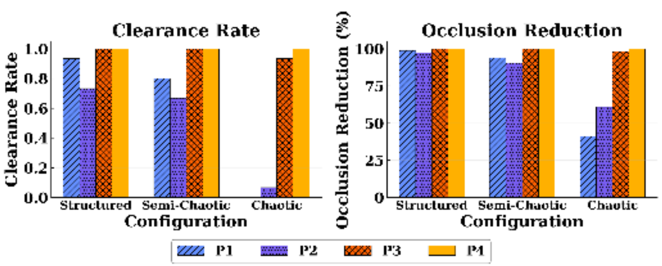
(a) Structured (b) Semi-Chaotic (c) Chaotic

Fig. 6: Example cable configurations for the three environment classes of increasing complexity.

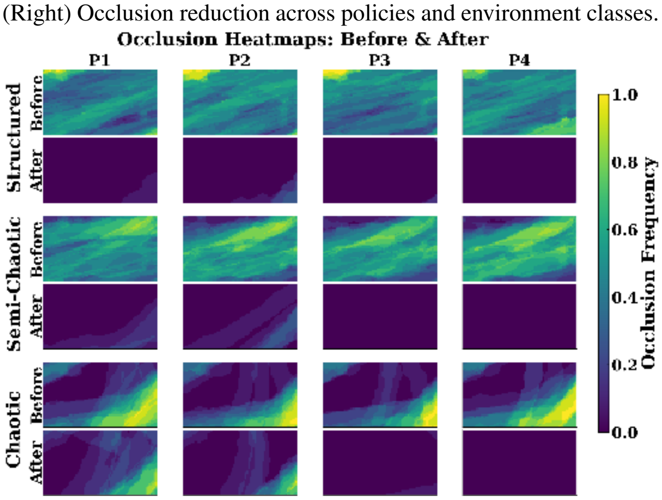


(a) P1 (b) P2 (c) P3 (d) P4

Fig. 9: Cable parting results for different policies in two instances of chaotic environments.



(a) (Left) Clearance rates of each policy across environment classes. (Right) Occlusion reduction across policies and environment classes.



(b) Heatmaps of cable pixel density in the clearance region before and after decluttering with each policy.

Fig. 10: Graphical visualisation of de-cluttering experiments

by laterally separating overlapping cables before opening, it ensures robust clearance even in chaotic configurations. These are shown in the bar graphs in Fig. 10a. The heatmaps in Fig. 10b show the normalised position of the cables before and after cable parting in all experiments and we can see the cable density in the clearance region to be a lot lower for the policies that include the *Comb* primitive (P3 and P4).

D. Standalone Insertion and Full Task Evaluation

Insertion Policy: Optical transceivers require sub-mm lateral precision for insertion. Such tasks are hard to implement without good old fashioned motion planning and controls [30]. We implement a force-feedback dependent Finite State Machine for insertion. The policy cycles through four states: `APPROACH` (forward force until contact), `SEARCH` (expanding lateral sweeps until the wrench along insertion axis drops, indicating the port opening), `ALIGN` (seat against port corner), and `INSERT` (force-guided drive with lateral wrench correction until a goal depth is reached or a jam is detected). The full insertion policy description and implementation details are out of scope for this paper, but we include it here for completeness and to show how we evaluate the decluttering policies in the context of a complete insertion task.

We first evaluate the finite state machine (FSM) insertion policy in isolation in two different environments 30 trials each to establish a baseline for transceiver insertion reliability. This isolates insertion performance from the decluttering task. The success rate is measured as the percentage of trials in which the transceiver is fully inserted. The two environments are:

- *Neighbours, Cables Clear:* The target port is surrounded by transceivers and cables, but they are manually cleared to ensure no physical obstruction.
- *Semi-Chaotic/Chaotic Cabling:* The target port is surrounded by dense cable clutter in semi-chaotic or chaotic configurations, without any manual clearing, representing the most realistic and challenging scenario.

Finally, we evaluate the complete multi-robot pipeline: one robot uses the chopstick tool to execute a decluttering policy, and a second robot subsequently inserts a transceiver into the cleared port after evaluating the clearance of the cable decluttering policy. We keep only the best performing policy from the standalone decluttering evaluation for this test. This evaluation is done on the same Semi-Chaotic and Chaotic environments (30 trials) from the standalone decluttering evaluation. This allows us to assess if we can bring the capability of inserting a transceiver to a realistic datacenter setting with dense cable clutter to similar success rates as the standalone insertion task. The success rate is measured as the percentage of trials in which the transceiver is fully inserted after the decluttering step.

The standalone insertion results (Fig 12) establish that the FSM insertion policy is reliable when the port area is

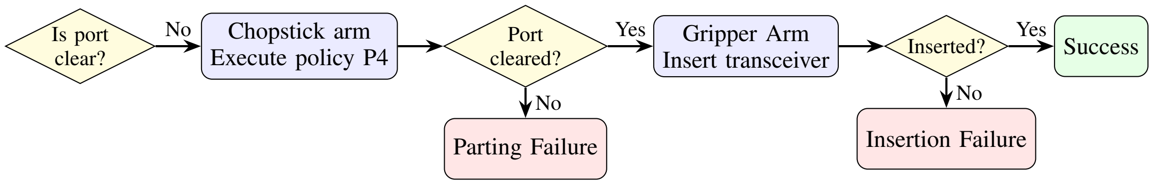


Fig. 11: Full task pipeline: the chopstick arm declutters the target port, then the gripper arm inserts the transceiver.

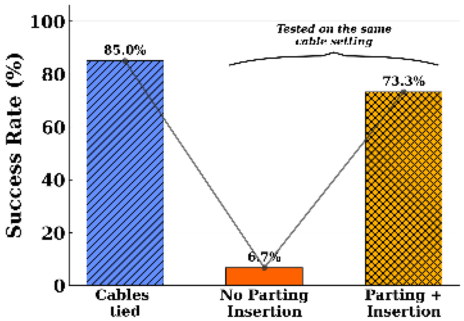


Fig. 12: Insertion success rates by scenario group.

Policy	Parting Clearance rate	Insertion % Success	Full Task % Success
P4	**93.33**	**78.57**	**73.33**

TABLE IV: Full task (declutter + insert)

unobstructed achieving 85% with neighbouring transceivers present and cables manually cleared. Without any cable clearing, insertion success drops to 6.67% in semi-chaotic and chaotic configurations. This dramatic decrease occurs because the the cables get entangled with the transciver and the gripper and the insertion process is manually aborted. This confirms that cable clutter is the primary barrier to reliable transceiver insertion and motivates the need for the decluttering step.

The full task evaluation (Table IV) demonstrates that the integrated pipeline recovers the majority of the insertion capability lost to cable clutter. Using P4 for decluttering, the parting success rate is 93.33%, and the conditional insertion success rate given a cleared port is 78.57%, yielding an overall full task success rate of 73.33%. This represents more than a $10\times$ improvement over attempting insertion directly in cluttered environments (6.67%). The gap between the full task insertion rate (78.57%) and the standalone insertion rate with cleared cables (85%) was observed to be caused by cables entangling with the gripper even after cable parting which suggests that residual cable displacement after decluttering occasionally leaves partial obstructions that interfere with the insertion policy, indicating tuning the segmentation threshold p_{th} is important.

Sim-to-real gap: Comparing the simulation results (Table II) with the real-world decluttering results reveals an important gap. In simulation, all four policies achieve similar clearance rates above 90% for empty ports, whereas in reality P1 and P2 perform significantly worse, particularly in chaotic environments. This discrepancy arises because the MuJoCo cable model does not fully capture real-world cable interactions such as friction-induced entanglement, knot formation, and variable cable stiffness.

VII. Discussion and Future Work

Several directions remain for future work. First, our cable parting policies currently execute open-loop; incorporating visual feedback from the onboard camera to adaptively select or re-parameterize primitives during execution could improve robustness in highly chaotic environments.

Second, beyond immediate task execution, the proposed primitive-based interaction framework offers a structured lens for studying cable physics. Accurate modeling of deformable linear objects remains challenging due to high-dimensional state spaces, complex contact interactions, and sensitivity to material parameters [18], [19]. High-fidelity analytical solutions do not operate in real-time conditions [31]. Unstructured manipulation actions often entangle multiple physical effects, making it difficult to interpret observed behavior or collect informative data [20]. In contrast, modular motion primitives induce repeatable and well-defined interactions such as localized bending, sliding, or separation that can be systematically analyzed. This structure enables the collection of targeted, interpretable datasets, which may support improved physical modeling, system identification, or learning-based approaches for cable manipulation in future work [23], [25].

Third, rather than reconstructing the full geometric state of every cable in the scene, a promising direction is to develop semantic scene understanding that captures higher-level properties of the cable configuration. This includes reasoning about where individual cables originate and terminate, how they are routed through the rack, where they are attached or zip-tied, the degree of slack between attachment points, and whether cables are knotted or looped around one another. Such understanding would enable the robot to anticipate how cables will respond to manipulation, for instance, predicting that a taut cable cannot be displaced laterally, or that a cable with significant slack can be combed away more aggressively. This could inform both primitive selection and parameterization, moving from a fixed policy to configuration-aware adaptive policies. To this end, we are interested in exploring domain-specific world models to develop an intuitive understanding of cable behaviour that is sufficient for decision-making. Such models would prioritise predicting likely outcomes and detecting when manipulation has failed or deviated from expectation, enabling the robot to trigger appropriate failure responses such as re-attempting a primitive with different parameters, switching to an alternative policy, or requesting human intervention. This failure-aware approach is more practical for real-world deployment.

VIII. Conclusion

We presented a non-prehensile cable manipulation approach for dense datacenter environments using a chopstick-inspired tool and a set of modular motion primitives. By decomposing cable decluttering into four composable primitives: insert, rotate, comb, and open, we showed that sequential policies of increasing complexity can effectively clear target transceiver ports across structured, semi-chaotic, and chaotic cable configurations. The approach was validated in both MuJoCo simulation and real-world experiments on a dual-arm UR5e platform, demonstrating successful cable parting and end-to-end transceiver insertion without requiring force sensing, cable state estimation, or learned manipulation policies.

References

[1] N. Lv, J. Liu, and Y. Jia, "Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2341–2353, 2022.

[2] 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," 2022. [Online]. Available: https://arxiv.org/abs/2207.07813

[3] I. Sarantopoulos *et al.*, "Robust optical transceiver manipulation in cluttered cable environments using 3d scene understanding and planning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1–8.

[4] S. Zhaole, H. Zhou, L. Nanbo, L. Chen, J. Zhu, and R. B. Fisher, "A robust deformable linear object perception pipeline in 3d: From segmentation to reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 843–850, 2023.

[5] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, "Rt-dlo: Real-time deformable linear objects instance segmentation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 11, pp. 11 333–11 342, 2023.

[6] L. Zhang, K. Bai, Q. Li, Z. Chen, and J. Zhang, "A collision-aware cable grasping method in cluttered environment," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 2126–2132.

[7] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning Synergies between Pushing and Grasping with Self-supervised Deep Reinforcement Learning," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4238–4245, mar 2018. [Online]. Available: http://arxiv.org/abs/1803.09956https://ieeexplore.ieee.org/document/8593986/

[8] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021. [Online]. Available: https://doi.org/10.1177/02783649211027233

[9] M. Yu, K. Lv, C. Wang, M. Tomizuka, and X. Li, "A coarse-to-fine framework for dual-arm manipulation of deformable linear objects with whole-body obstacle avoidance," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 153–10 159.

[10] M. R. Dogar and S. S. Srinivasa, "A planning framework for non-prehensile manipulation under clutter and uncertainty," *Autonomous Robots*, vol. 33, no. 3, pp. 217–236, 2012.

[11] A. Wilson, H. Jiang, W. Lian, and W. Yuan, "Cable routing and assembly using tactile-driven motion primitives," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 408–10 414.

[12] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE. IEEE, oct 2012, pp. 5026–5033. [Online]. Available: http://ieeexplore.ieee.org/document/6386109/

[13] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong, "Efficient learning of goal-oriented push-grasping synergy in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6337–6344, 2021.

[14] I. Sarantopoulos, M. Kiatos, Z. Doulgeri, and S. Malassiotis, "Split deep q-learning for robust object singulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6225–6231.

[15] M. Kiatos, I. Sarantopoulos, L. Koutras, S. Malassiotis, and Z. Doulgeri, "Learning push-grasping in dense clutter," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8783–8790, 2022.

[16] W. Bejjani, W. C. Agboh, M. R. Dogar, and M. Leonetti, "Occlusion-aware search for object retrieval in clutter," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4678–4685.

[17] 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.

[18] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2023.

[19] A. Caporali, P. Kicki, K. Galassi, R. Zanella, K. Walas, and G. Palli, "Deformable linear objects manipulation with online model parameters estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2598–2605, 2024.

[20] D. McConachie, A. Dobson, M. Ruan, and D. Berenson, "Manipulating deformable objects by interleaving prediction, planning, and control," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 957–982, 2020. [Online]. Available: https://doi.org/10.1177/0278364920918299

[21] Y. Tang, X. Chu, J. Huang, and K. W. Samuel Au, "Learning-based mpc with safety filter for constrained deformable linear object manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2877–2884, 2024.

[22] 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.

[23] R. Lee, M. Hamaya, T. Murooka, Y. Ijiri, and P. Corke, "Sample-efficient learning of deformable linear object manipulation in the real world through self-supervision," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 573–580, 2022.

[24] 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)*, 2021, pp. 4560–4567.

[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] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine, "Multistage cable routing through hierarchical imitation learning," *IEEE Transactions on Robotics*, vol. 40, pp. 1476–1491, 2024.

[27] D. Tong, A. Choi, L. Qin, W. Huang, J. Joo, and M. K. Jawed, "Sim2real neural controllers for physics-based robotic deployment of deformable linear objects," *The International Journal of Robotics Research*, vol. 43, no. 6, pp. 791–810, 2024.

[28] O. Holešovský, R. Škoviera, and V. Hlaváč, "Movingcables: Moving cable segmentation method and dataset," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6991–6998, 2024.

[29] X. Xu *et al.*, "Automated fiber switch with path verification enabled by an ai-powered multi-task mobile robot," in *49th European Conference on Optical Communications (ECOC 2023)*, vol. 2023, 2023, pp. 1670–1673.

[30] S. Ye *et al.*, "World action models are zero-shot policies," 2026. [Online]. Available: https://arxiv.org/abs/2602.15922

[31] A. Tekinalp *et al.*, "Gazzolalab/pyelastica," 2024. [Online]. Available: https://doi.org/10.5281/zenodo.7658871