Hierarchical DLO Routing with Reinforcement Learning and In-Context Vision-language Models

Mingen Li[1], Houjian Yu[1], Yixuan Huang[2], Youngjin Hong[1] and Changhyun Choi[1]

***Abstract*— Long-horizon routing tasks of deformable linear objects (DLOs), such as cables and ropes, are common in industrial assembly lines and everyday life. These tasks are particularly challenging because they require robots to manipulate DLO with long-horizon planning and reliable skill execution. Successfully completing such tasks demands adapting to their nonlinear dynamics, decomposing abstract routing goals, and generating multi-step plans composed of multiple skills, all of which require accurate high-level reasoning during execution. In this paper, we propose a fully autonomous hierarchical framework for solving challenging DLO routing tasks. Given an implicit or explicit routing goal expressed in language, our framework leverages vision-language models (VLMs) for in-context high-level reasoning to synthesize feasible plans, which are then executed by low-level skills trained via reinforcement learning. To improve robustness in long horizons, we further introduce a failure recovery mechanism that reorients the DLO into insertion-feasible states. Our approach generalizes to diverse scenes involving object attributes, spatial descriptions, as well as implicit language commands. It outperforms the next best baseline method by nearly 50% and achieves an overall success rate of 92.5% across long-horizon routing scenarios. Please refer to our project page: https://icra2026-dloroute.github.io/DLORoute/**

I. Introduction

Deformable linear objects (DLOs) are ubiquitous in daily life and industrial applications, yet they remain challenging for robotic manipulation. Their inherent flexibility and underactuated nature, stemming from unpredictable deformation, pose significant difficulties for both action-level control and modeling of deformable dynamics. Moreover, these properties impose heightened demands on high-level task reasoning when manipulating DLOs. For example, common routing tasks such as desk cable management require navigating cables through constrained and cluttered environments. This involves selecting appropriate entry points and accurately passing cables through holes, tubes, or clips in the correct sequence and orientation, ultimately arriving at destination sockets with suitable cable length and alignment. Similarly, industrial tasks such as automobile assembly require routing wires and signal lines along predefined paths within a vehicle frame while ensuring safety, avoiding sharp bends, and minimizing tension to prevent damage. Failures such as misalignment during insertion or undesirable DLO configurations can compromise task success, necessitating replanning and reevaluation of the entire routing process.

[1]The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 {li002852, yu000487, hong0745, cchoi}@umn.edu

[2]The authors are with Princeton University, Princeton, NJ 08544 yh1542@princeton.edu

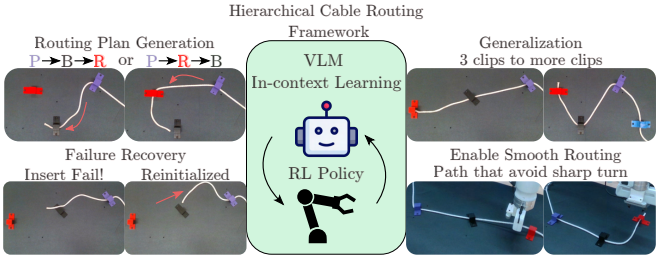

Fig. 1: **Hierarchical DLO routing framework.** Our framework combines high-level planning via a VLM with in-context learning and low-level control via an RL policy. The VLM generates routing plans and handles failure recovery, while the RL policy executes precise manipulation. This framework enables recovery from insertion failures through reinitialization, generalizes from three-clip to multi-clip routing, and produces smooth paths that avoid sharp turns.

Previous research has explored deformable object manipulation but has struggled with limited generalization, often focusing on single-skill settings or short-horizon scenarios. Early studies addressed fundamental tasks such as insertion and pulling under frictional environments for DLOs with varying physical properties [1, 2]. While these works advanced low-level control, they primarily targeted short-horizon tasks involving a single goal, without addressing multi-stage, long-horizon planning. Luo et al. [3], for example, proposed an imitation learning approach for routing cables through clips using human demonstrations. Although effective within the training domain, their method generalizes poorly to out-of-distribution scenarios, limiting its applicability to real-world DLO routing problems.

With the advent of VLMs, robotics has begun leveraging their powerful commonsense reasoning and scene understanding capabilities. These models can provide contextual knowledge, high-level task guidance, and even recovery strategies when a robot becomes stuck or encounters failure. For instance, VLM-PC [4] employs a VLM to guide a legged robot through unstructured environments by reasoning about progress and replanning dead-end cases, while relying on a locomotion controller for low-level actions. However, such approaches are unsuitable for DLO manipulation, where careless actions can damage the object—for example, pushing without regard to obstacles can lead to deformation or breakage. Thus, in addition to semantic reasoning for high-level planning, safe and reliable low-level control is essential for long-horizon DLO routing.

To address these challenges, we propose a hierarchical framework that integrates high-level planning via VLMs with reinforcement learning-based low-level control for DLO routing (Fig. 1). The objective is to route a DLO through

multiple clips in a specified or natural order based on a language prompt. We design three core low-level skills: **Insert** and **Pull** for clip routing, and **Flatten** for failure recovery. Pull and Flatten are inherently safe as they move the DLO away from the environment and can be initialized from predefined motion primitives. In contrast, insertion requires precise navigation near a clip to maximize insertion success while avoiding collisions, so we train this skill using reinforcement learning to balance safety and accuracy.

These low-level skills are provided to the VLM as in-context examples, each labeled as Insert, Pull, or Flatten, and each clip is annotated with spatial or attribute labels (e.g., color). We supply the VLM with task descriptions, skill definitions, and a scene image to produce a routing plan, including the clip order and insertion directions. During execution, the VLM receives both a full-scene image and a zoomed-in view of the current clip to infer task progress, select the next skill, and determine the target clip (Fig. 5). If repeated failures occur and no progress is made, the VLM recognizes the failure and triggers the Flatten skill to recover from the stuck state and resume the routing sequence.

The primary contributions of this paper are as follows:

- We introduce a hierarchical framework for long-horizon DLO routing through arbitrarily arranged clips with diverse attributes, integrating low-level skills learned via reinforcement learning with high-level in-context reasoning enabled by a vision-language model.
- We propose a failure-aware mechanism in which the vision-language model detects execution failures, reasons about their causes, and replans accordingly, substantially improving the overall task success rate.
- We demonstrate that our approach achieves high performance in long-horizon DLO routing, robustly handling diverse DLOs and clip configurations with varying poses and 3-clip and multi-clip settings.

II. Related Works

A. Deformable Manipulation

DLO manipulation has been explored in various applications, such as surgical tasks [5], rope shaping [6], and cable grasping [7]. Recent works such as [7, 8] address tasks like needle threading, pulling, and shaping. However, these approaches rely on coarse geometric models (e.g., capsules), which fail to capture DLO deformation accurately and cannot reliably predict object states.

To overcome these limitations, other research has focused on learning DLO dynamics and deformable–rigid contact interactions. This enables force-aware manipulation and improves generalization to deformable objects with varying physical properties [1, 9, 10]. Simulation platforms such as SoftGym [11] and IsaacLab [12, 13] provide more accurate deformable models and integrate them with reinforcement learning environments. Nevertheless, these works primarily focus on short-horizon tasks, leaving long-horizon DLO manipulation relatively underexplored.

One prior study proposes a hierarchical framework for cable routing by twisting a DLO through three clips [3]. However, their high-level controller is trained with imitation learning, which relies on limited demonstrations and struggles to generalize, leading to a substantial performance drop when extended to a four-clip setting. In contrast, our framework leverages robust low-level RL policies that accurately capture DLO dynamics with a high-level planner powered by a VLM. This combination addresses the generalization bottleneck and enables scalable, long-horizon DLO manipulation across diverse task configurations.

B. Vision-language Models for Long-horizon Planning

Vision-language models have demonstrated impressive performance in long-horizon planning, mainly by generating high-level task plans [14]–[17] or through reward generations [18]. Our proposed approach differs from prior work in that the high-level planner is explicitly designed to interleave with low-level skill execution. Recent studies [19]–[23] have also explored combining high-level reasoning with VLMs while simultaneously learning low-level control policies. However, their efforts primarily target rigid-object manipulation, limiting their applicability to the more challenging domain of DLO manipulation.

C. Failure Cases Detection and Recovery

Enabling robots to autonomously detect and recover from failures has become an important challenge in the robotics community [24]–[29], as it's a key step toward lifelong learning during deployment in diverse real-world environments. However, existing work has either concentrated solely on failure detection [24, 26]–[28] or on the manipulation of rigid objects [25, 29]. In contrast, our approach addresses failure reasoning and recovery within a hierarchical system, where high-level planning interleaves with low-level policies, with an emphasis on the challenging domain of DLO manipulation.

III. Method

A. Problem Formulation

The long-horizon DLO routing problem can be divided into two subtasks: (1) a high-level planner that reasons over scene knowledge and makes sequential decisions, and (2) a set of low-level skills that interact with the environment to perform DLO manipulation.

The planner is given an image of a scene I_{scene} with three or more clips placed at different orientations and locations, along with a text description indicating the order in which it should route the DLO through. As shown in Fig. 2, the planner must generate a routing plan consistent with the provided description. The clip may have different attributes like color or spatial relation. The order will describe the color attributes, spatial relation, or an implicit order. We also consider a 4-clip setting to evaluate long-horizon generalization. In order to achieve the routing goal, a skill set containing possible skills is required. The planner should decide the next low-level skill to execute, the target clips to route through, and reason about task completion, given the scene image and history plan. Details about the planner are in Section III-C

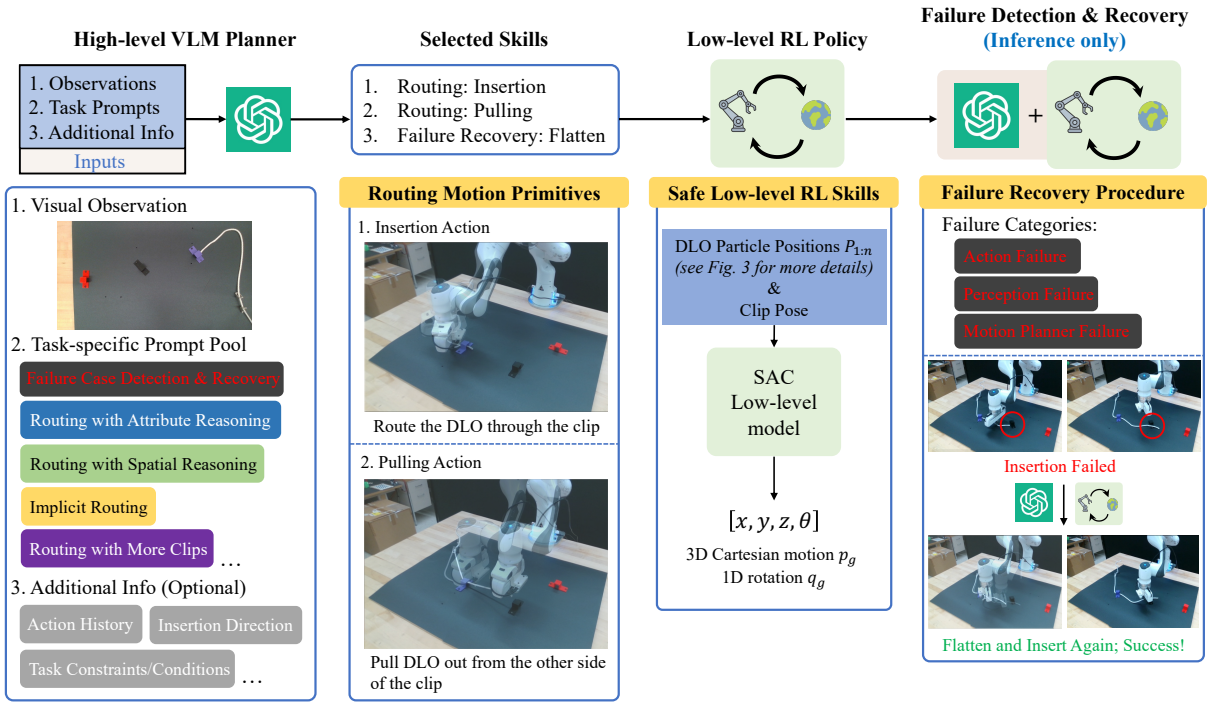

Fig. 2: **Pipeline of the proposed hierarchical DLO routing framework.** The high-level VLM-based planner processes top-down scene images, task prompts, and auxiliary information to select appropriate skills, including routing (insertion and pulling) and failure recovery (flattening). Insertion is performed by a safe, low-level RL–based parameterized motion primitive for precise manipulation. A failure detection and recovery module (inference only) monitors execution to identify action, perception, or motion-planning failures and triggers replanning, enabling robust recovery and successful long-horizon routing. Note that side-view images shown in the second row are for visualization only; the planner receives only top-down views.

Additionally, the planner must decide when to stop execution by reasoning over task progress and comparing it with the goal specified in the initial instruction.

Our low-level skills can be formulated as a Markov Decision Process (MDP) $(\mathbf{S}, \mathbf{A}, r, \gamma)$. The state $\mathbf{S}$ consists of the pose of the clip, and the positions $\mathbf{p}_{1:n}$ of a DLO, represented by n particles as shown in Fig. 3. The action space $\mathbf{A}$ includes 3D Cartesian motion $\mathbf{p}_g^t$, 1D rotation $\mathbf{q}_g^t$ of the robot gripper at time step t. Section III-B details the low-level skills setup and training of an RL policy for the DLO insertion skill.

B. Training Low-level Skills

Low-level skills should robustly achieve their own manipulation goal. We designed three low-level skills available to the planners: Insert, Pull, Flatten, shown in Fig. 3. Each of them requires a specified target clip. Among these, insertion is the most crucial for cable routing, as it directly accomplishes the task objective. Successful insertion must adapt to the dynamics of deformable objects and be reactive during execution in the contact-sensitive regions. In contrast, pulling and flattening serve as auxiliary skills that adjust the DLO configuration to create more favorable conditions for insertion. Additionally, pulling and flattening move the DLO away from the clips, whereas insertion requires precise motion close to the clip and must achieve a high success rate to minimize repeated trials. Thus, for pulling and flattening, we employ a predefined motion primitive, while insertion is a parametrized motion primitive with its parameters optimized by a reinforcement learning agent trained in IsaacSim [12]. This allows insertion to leverage adaptive, learned behaviors for complex contact scenarios, while auxiliary skills remain simple and efficient.

We design the insertion motion primitive as shown in Fig. 3. First, the insertion primitive will grasp the corresponding sampled DLO segment $\mathbf{p}_i$ given a predicted grasping index i. After grasping, the robot follows a waypoint trajectory, where the via-points are defined by gripper poses $(\mathbf{q}_g^t, \mathbf{p}_g^t)$ at each time step t. The insertion RL policy learns to iteratively improve the insertion condition and can be executed for multiple timesteps, enabling closed-loop control and maximizing the insertion success.

We employ Soft Actor-Critic (SAC) [30] for training, with multi-layer perceptron (MLP) networks for both actor and critic networks. The reward function r is defined as follows:

$$r = 0.5(rope\_in + rope\_out) + \beta(collide) \quad (1)$$

$$+ \gamma r_{hor} + r_{dist} + \eta r_{flat} \quad (2)$$

where $rope\_in$ and $rope\_out$ are binary indicators denoting a halfway-through insertion or full passage through the clip, following a reward design similar to [1]. The $collide$ indicates whether the robot is colliding with the clips or not. The r_{hor} is the current episode length and penalizes long episodes and encourages efficient insertions with fewer time steps.

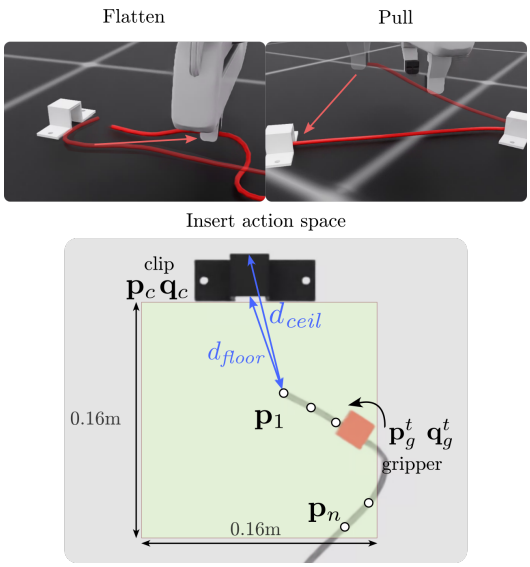

Fig. 3: **Illustration of the low-level action space for DLO routing skills.** The primitive set includes Flatten (top left) and Pull actions (top right), and the Insert action (bottom). For insertion skill, the gripper (in orange) operates within a $0.16m \times 0.16m$ space with orientation ($\mathbf{p}_g^t$, $\mathbf{q}_g^t$) conditioned on the clip (in black) state ($\mathbf{p}_c$, $\mathbf{q}_c$). The DLO is represented by particles $\{\mathbf{p}_1, ..., \mathbf{p}_n\}$.

The r_{dist} is a stage-specific reward, while r_{flat} encourages flattening the DLO to facilitate subsequent insertions. The β, γ, η are hyperparameters to be tuned. The r_{dist} is defined as follows:

$$r_{dist} = \begin{cases} 10 \cdot d_{floor} & \text{if } rope\_in \text{ and not } rope\_out \\ 20 \cdot d_{ceil} & \text{if } rope\_out \\ 1_{\mathcal{R}}(\boldsymbol{p}_1)/(4 + 80 \cdot d_{floor}) & \text{otherwise} \end{cases} \tag{3}$$

where d_{floor} and d_{ceil} are distances from the first DLO particle $\mathbf{p}_1$ to the clip center at the floor and ceiling (Fig. 3), $\mathbf{1}_{\mathcal{R}}(\mathbf{p}_1)$ refers to an indicator function whether the DLO head $\mathbf{p}_1$ lies within a predefined region ahead of the clip, discouraging DLO head $\mathbf{p}_1$ from being moved out of the insertion-possible region. The r_{flat} encourages the front segment of the DLO to stay straight in front of the clip, keeping insertion in a straightforward state as follows:

$$r_{flat} = 1/(1 + \frac{1000}{3} \sum_{i=0}^{3} ||\mathbf{p}_{i+3} - \mathbf{p}_i||_y). \tag{4}$$

To encourage a faster convergence, we apply curriculum learning, where the RL agent initially trains under simple randomization and is exposed to more complex scenes.

C. High-level In-context Learning

For a high-level VLM planner, we leverage in-context learning to adapt the robot to our task domain and output reliable low-level skill instructions. The planner takes as input a top-down scene image I_{scene}, and user-specified text about the routing order. In the first query, the planner outputs the predicted routing order (task plan). In subsequent queries, it outputs the predicted skills, target clips, and task progress information (skill-level execution). To achieve this,

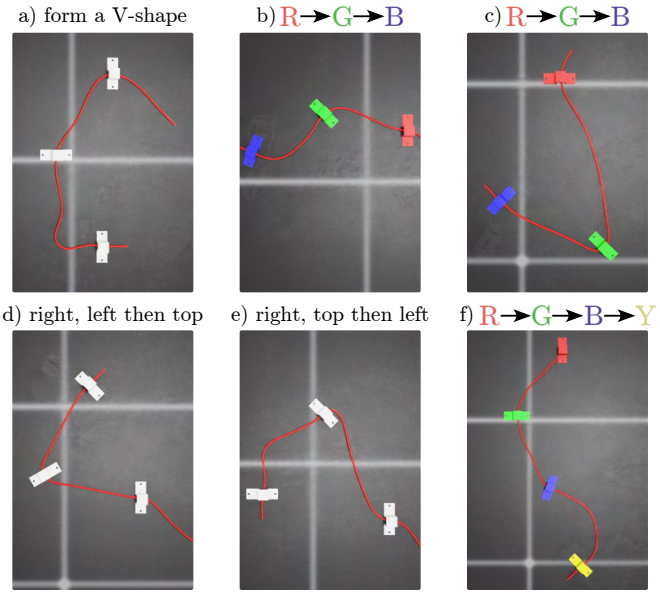

Fig. 4: **Long-horizon routing examples in simulation using our VLM with failure reasoning and RL-trained low-level policy.** (a) Implicit order forming a V-shape. (b–c) Fixed order based on color attributes (red, green, blue). (d) Fixed order determined by spatial reasoning. (e) Fixed order based on color attributes in a four-clip arrangement.

we instruct VLM to perform 1) Task progress reasoning, 2) low-level skills reasoning. Along with the user's text, we also provide a scene description that includes the attributes of the DLO, clips, and environment to help improve scene understanding. Since the location of the DLO front segments is crucial for the routing with DLO head manipulation, we ask the planner to track the DLO head throughout the task and to recognize occlusions caused by clips when the DLO has already passed through them. This enables more accurate reasoning about task progress, target clip selection, and completion status. To further enhance decision making, we employ the chain-of-thought (CoT) prompting [31] to ask the planner to output the zoom-in view analysis, thereby determining DLO–clip interaction before predicting the next skill or routing order.

Skill reasoning helps planners distinguish between available skills and make precise selections. An elaborated definition of each skill, along with verbal usage examples is given to VLM. Additionally, we provide two example images corresponding to insertion and pulling, illustrating the detailed scenario when these skills should be invoked.

D. Hierarchical Planning & Failure Recovery

During the DLO routing rollout with only insertion and pulling skills, we observed that when pulling the DLO toward the next clip, the DLO head inevitably twisted or drifted far away from the clip opening. Under such challenging DLO states, insertion attempts will consistently fail. To address this problem, we introduce a failure recovery mechanism. The recovery module consists of a low-level rescue skill, Flatten, a simple yet effective failure recovery skill that

reintializes the DLO to an insertion-possible state. In addition to providing the definition and usage example of flattening skill, we set a counter of consecutive insertion attempts and feed this status summary to the planner. We then impose a step limit to prevent getting stuck by infinitely executing the same skill and incorporating this constraint into the CoT prompting. Failure detection and recovery are automatically predicted by the planner from observations and history, without human intervention. With this design, even if the DLO head aligns nearly parallel with the clip's long axis, the high-level planner will not get stuck and can invoke recovery, reorient the DLO, and resume insertion successfully.

IV. Experiments

We perform the experiments to answer the following research questions: **Q1**: Can our approach with a hierarchical framework excel at long-horizon deformable manipulation tasks? **Q2**: Does failure reasoning boost the performance during long-horizon reasoning? **Q3**: Would our approach generalize to unseen scenarios (e.g., more clips)? **Q4**: Does our approach outperform baselines using a visual baseline as the low-level policy? In this paper, we train our low-level RL policy in IsaacSim [12] and GarmentLab [32] and use GPT-5 [33] with low reasoning effort as the VLMs planner for our experiments.

A. Experiment Setup and Evaluation Metrics

In this section, we present experiments on closed-loop RL policy training for DLO Insertion, VLM planner setup, and a comparative performance analysis against several baselines.

1) Low-level Skills Setup: Low-level skills consist of the Insert, Pull, and Flatten, with the insertion skill designed as a parameterized motion primitive and trained using reinforcement learning. The Pull and Flatten primitives have predefined motion if the target clip is determined as shown in Fig. 3.

Metrics	Visual Baseline	RL Policy (Ours)
Success Rate (%) ↑	45	**87**
Avg. dis. (cm) ↑	1.24	**2.59**
Episode Length	2	3.86

TABLE I: Real-world evaluation on open-loop methods under different friction environments.

In both simulation and real-world experiments, we constrain the insertion manipulation trajectories to a 2D plane and 1D rotation with respect to the z-axis, resulting in a 3-dimensional waypoint for a motion primitive. One motion primitive consists of two via points ($\mathbf{p}_g^T \in \mathbb{R}^2$, $\alpha_g^T \in \mathbb{R}$, $T \in 1, 2$) and one indexed grasping location that grasp at DLO particle $\mathbf{p}_i$, totaling seven parameters for the insertion motion primitive. For the reward function, we use $\beta = -2$, $\gamma = -0.001$, and $\eta = 0.5$.

In the simulation, we randomized the DLO position and angle within a 10cm×5cm rectangle and between -10 deg to 10 deg. For curriculum learning, we randomize the clip size to control the task difficulty. The original clip has a 2.2cm opening. For the first 1600 training steps, the clip scale is randomized from 1.0 to 2.2, providing a maximum clip opening of 4.84cm wide. After the policy is well initialized, we then focus the training on a domain-specific clip scale from 0.9 to 1.5. The RL policy is trained for 6.2k steps, and the evaluation result is shown in Tab. I.

During long-horizon DLO routing tasks, the agent will encounter clips with vastly different orientations and poses. To alleviate this confusion for low-level policy, we always transform the DLO state observation back into the clip's local frame and execute the action in the local frame.

2) Evaluation Details: We evaluate the low-level policies using 100 different scenes of randomized DLO poses and clips with the original scale. Average distance (avg. dis.) is the average signed endpoint distance $\pm d_{ceil}$. It is positive when $rope\_out$ and negative otherwise. The success is recorded when the signed endpoint distance $\pm d_{ceil}$ is larger than +2cm after the motion primitive finishes.

As for long-horizon DLO routing, we evaluate our baselines with four different DLO routing strategies: implicit orders, fixed orders with spatial descriptions or color attributes, and fixed orders with color attributes under 4-clip settings, as shown in Fig. 4. Note that the 4-clip setting is created by adding one more clip to the front or back of the clip arrangement in the implicit routing order setting. The DLO must fully pass through each clip, and the DLO head must emerge from the other side. A success case is recorded when the DLO is routed through all clips in the scene with the correct order. Average episode length (Avg. Epi. Length) means the number of pick-and-place steps executed. Average number of clips inserted (Avg. #Clips Ins.) shows the average number of clips inserted when the episode terminates. An episode may be terminated and be considered a failure case when 1) a collision is detected, 2) the maximum timeout has been reached, or 3) early termination with some clips remaining not inserted.

B. Baselines

We conduct various open-loop and closed-loop low-level insertion experiments in our simulation. For our RL policy, the horizon of manipulation could take up to 7 steps to allow the agent to fully attempt the scene. The following baselines are open-loop, where the agent executes an entire trajectory and terminates the episode:

- **Visual baseline (VB)**: A heuristic approach that imitates a human expert's insertion strategy. We design a straightforward insertion action for different DLO shapes and clip sizes, where we first place the DLO in front of the clip opening and insert along the center line of the clip.

For long-horizon DLO routing evaluation, we employ several heuristic methods and conduct ablation studies under various challenging clip arrangements.

- **VLM (w/o failure reasoning)**: Remove the failure reasoning and recovery description from the prompt text and remove the Flatten primitive. The VLM planner can only command insert and pull to complete the task.

High Level	Low Level	Eval. Metric	Implicit	Fixed (Spatial)	Fixed (Attribute)	4-Clip
VLM w/ Failure Reasoning	Visual Baseline	Success Rate (%) ↑	20	10	0	10
		Avg. #Clips Ins. ↑	1.7	0.7	1.1	1.3
		Avg. Epi. Length	9.6	7.6	7.4	9.2
VLM	RL policy	Success Rate (%) ↑	70	10	10	50
		Avg. #Clips Ins. ↑	2.5	1.5	1.3	3.1
		Avg. Epi. Length	14.1	21.5	20.4	27.7
Fixed Order	RL Policy	Success Rate (%) ↑	**80**	20	20	60
		Avg. #Clips Ins. ↑	2.6	1.6	1.5	3.2
		Avg. Epi. Length	10.1	9.6	8.9	13.7
VLM w/ Failure Reasoning	RL policy	Success Rate (%) ↑	**80**	**90**	**100**	**100**
		Avg. #Clips Ins. ↑	**2.8**	**2.9**	**3**	**4**
		Avg. Epi. Length	12.8	18.4	17.5	17.9

TABLE II: **Simulation result for long-horizon DLO routing.** Evaluation metrics include success rate (higher is better), average number of clips inserted (higher is better), and average episode length. Our hierarchical framework with failure reasoning achieves the best performance across settings.

- **Fixed order**: Given the ground truth insertion order, we adopt a heuristic method that repeatedly executes Pull and Insert actions until the task either succeeds or fails.
- **VB as low-level policy**: substitutes the low-level RL policy with the heuristic strategy that mimics human insertion but is less considerate of the environment, allowing us to evaluate performance under a simpler motion primitive.

C. Simulation Experiments

In this section, we will address the aforementioned questions using our simulation results summarized in Table II.

Q1: Can our approach with a hierarchical framework excel at long-horizon deformable manipulation tasks? The hierarchical planning has greatly improved the performance compared with the fixed order heuristic baseline. As shown in Table II, our method excels for most tasks, and shows a better success rate and higher number of clips inserted across all evaluation settings, particularly under challenging conditions such as Fixed Spatial, Fixed Attribute, and 4-Clip. While our method requires longer episode lengths, the improved robustness and task completion demonstrate that the hierarchical framework excels at long-horizon deformable manipulation tasks. The fixed order baseline achieves comparative performance in an implicit setting, primarily because insertion in this scenario is less challenging. In order to create cases for natural ordering, the clip opening angles within this experiment set are similar to each other to ensure mild turns between the clips. Such settings reduce the need for failure recovery, as the fixed order works well already. However, when it comes to challenging cases with sharper turns, the insertion needs more trials or re-initialization for the DLO to get through. Certain failure cases for ours are shown in Fig. 6, the high-level planner issues an early termination before the DLO has been fully inserted into the last clip.

Q2: Incorporating failure reasoning boosts our performance over the plain VLM policy. Across all evaluation settings, our method achieves higher success rates and more clips inserted, with the most substantial gains in challenging long-horizon scenarios such as Fixed Spatial, Fixed Attribute, and 4-Clip. Moreover, episodes are shorter in these cases, indicating that the agent recovers more efficiently from errors. In contrast, an agent without failure reasoning can only try several times under insertion-challenging scenarios and keep getting failure results. Additionally, scenarios such as Fixed Spatial and Fixed Attribute are relieved from the mild turn preset condition and contain over 90-degree turns between clips. This makes insertion much harder without recovering from failure. These results demonstrate that failure reasoning is a key factor in boosting long-horizon reasoning performance.

Q3: Our approach can extend beyond the current clip setting to a more challenging 4-clip setting. Our hierarchical framework with failure reasoning achieves a perfect success rate and consistently inserts all clips, demonstrating strong generalization to longer-horizon tasks. By contrast, both the VLM-only policy and the Fixed Order baseline suffer notable drops in success rate and efficiency. These results indicate that our approach scales robustly to scenarios involving more clips. Although this setting is extended from relatively simple routing scenarios, the added clip usually forms a large angle compared with the previous clip, making the setting much harder for simple insertion strategies.

Q4: Comparing against the visual baseline shows that the low-level policy plays a critical role in long-horizon deformable manipulation. While the heuristic insertion baseline quickly fails due to its lack of environmental awareness, the RL low-level policy adapts actions to task progress and scene dynamics, enabling precise and robust insertions. This distinction becomes most evident in the 4-Clip setting, where the visual baseline achieves only 10% success while our approach reaches 100%, demonstrating that task-aware low-level skill is essential for scaling to complex scenarios.

D. Real-world Robot Experiments

We conducted real robot long-horizon routing tasks using a Franka Emika Panda robot. A wrist-mounted, calibrated Intel RealSense D415 camera captures top-down images of the tabletop scene at a resolution of 1280×720. The MoveIt planner [34] was used to generate continuous trajectories

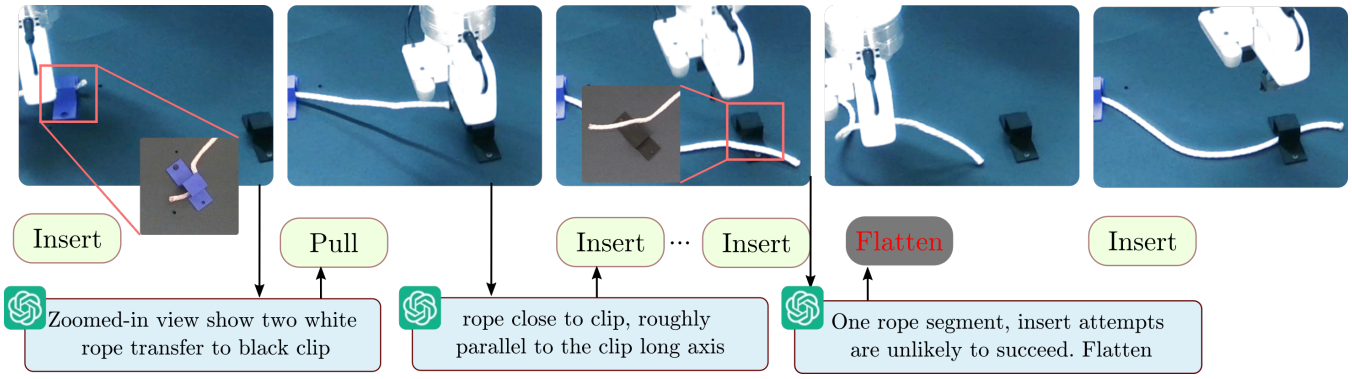

Fig. 5: **Real-robot execution of the proposed hierarchical DLO routing framework.** The robot receives both a whole-scene view and a zoomed-in view centered on the current clip. During normal execution, it alternates between insertion and pulling actions. When insertion becomes unlikely due to unfavorable DLO configurations (e.g., alignment along the clip's long axis or sliding past the clip without entering), the system detects the failure, executes a flatten action to reinitialize the DLO, and then resumes insertion successfully.

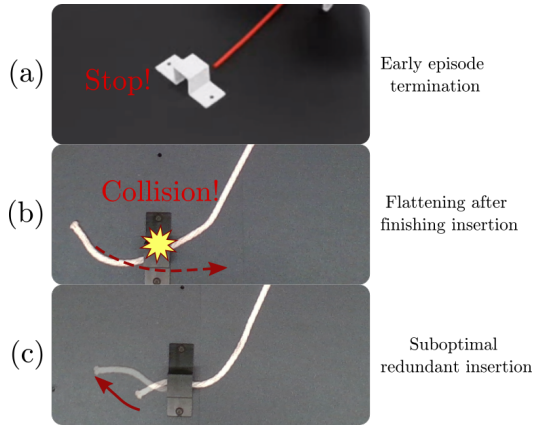

Fig. 6: **Representative failure cases observed in simulation and real-world experiments**: (a) early episode termination before completing last-clip insertion, (b) unintended collision when falsely flattening after an insertion, and (c) suboptimal redundant insertion where the VLM should choose pull for a more effective plan.

based on the motion primitives predicted by our model or predefined motion skills. The overall system was built using the Robot Operating System (ROS) framework [35], and clips and DLO are segmented using SAM2 [36] with an NVIDIA 5090 GPU. The reference points for DLO segmentation are initially set around the fixture and iteratively added during segmentation. Importantly, our policies did not require additional data for fine-tuning or domain adaptation during real robot testing. We compare our approach to the best-performing baseline in the 4 tasks mentioned in Tab. II and report the success rate and average number of clips inserted as shown in Tab. III. The routing process is shown in Fig. 5. Although the success rate in real experiments is lower than in simulation, our method still achieves strong performance, reaching 62.5% success despite challenges from calibration errors, perception noise, and sim-to-real transfer. In contrast, the Fixed Order baseline performs significantly worse, highlighting that real-world routing requires explicit failure reasoning and closed-loop decision making to handle unexpected variations and disturbances. The common failure cases in simulation and in the real world are summarized in Fig. 6. Faulty planning has been observed in a real experiment where the robot executes the 'flatten' command even though the DLO is already inserted into the clip, leading to a grasping failure and a collision with the clip. Another suboptimal plan observed in real and simulation is that the planner ignores the emerging DLO segment in the zoomed-in image, incorrectly concludes that insertion is incomplete, and issues another insertion action. While this behavior does not necessarily result in a collision since our RL policy can select an appropriate grasping location, it still unnecessarily prolongs the routing process and leads to a suboptimal plan.

Metrics	Fixed Order + RL	Ours
Success Rate (%) ↑	37.5	**62.5**
Avg. #Clips Ins. ↑	1.75	**2.625**

TABLE III: **Real-world evaluation for long-horizon DLO routing**

V. Limitations & Future Work

While our long-horizon cable routing framework provides a promising solution for DLO routing in diverse scenes, several failure cases remain unsolved. Addressing faulty or suboptimal planning is an important direction for future work. One potential approach is to fine-tune the VLM planner to improve planning stability. Another direction is to integrate new skills into the VLM planner automatically. This process could be automated by providing the planner with sequences of action images and letting it reason through the usage and definition of skills. Such an approach would enable large-scale skill integration and improve generalization to increasingly complex deformable manipulation tasks.

VI. Conclusion

In this work, we presented a hierarchical framework for long-horizon cable routing that integrates a high-level VLM planner with a key low-level RL skill. The VLM leverages in-context reasoning and failure recovery to generate reliable task plans and skill selections, while the RL insertion policy robustly handles complex DLO dynamics. Through extensive

simulation and real-world experiments, we demonstrated that our approach outperforms fixed-order and heuristic baselines, achieving strong generalization to 4-clip settings. These results highlight the importance of combining high-level reasoning with task-aware low-level control for scalable DLO manipulation.

REFERENCES

[1] M. Li and C. Choi, "Learning for deformable linear object insertion leveraging flexibility estimation from visual cues," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5183–5189.

[2] M. Li, H. Yu, and C. Choi, "Routing manipulation of deformable linear object using reinforcement learning and diffusion policy," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 01–07.

[3] 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.

[4] A. S. Chen, A. M. Lessing, A. Tang, G. Chada, L. Smith, S. Levine, and C. Finn, "Commonsense reasoning for legged robot adaptation with vision-language models," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 12 826–12 833.

[5] Z. Yu, W. Xu, J. Ren, T. Tang, Y. Li, S. Yao, G. Gu, and C. Lu, "Precise robotic needle-threading with tactile perception and reinforcement learning," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=B7PnAw4ze0l

[6] F. Liu, E. Su, J. Lu, M. Li, and M. C. Yip, "Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3964–3971, 2023.

[7] S. Zhaole, J. Zhu, and R. B. Fisher, "Dexdlo: Learning goal-conditioned dexterous policy for dynamic manipulation of deformable linear objects," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 009–16 015.

[8] J. Lv, Y. Feng, C. Zhang, S. Zhao, L. Shao, and C. Lu, "Sam-rl: Sensing-aware model-based reinforcement learning via differentiable physics-based simulation and rendering," 2023.

[9] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[10] M. Van der Merwe, M. Oller, D. Berenson, and N. Fazeli, "Estimating deformable-rigid contact interactions for a deformable tool via learning and model-based optimization," *IEEE RA-L 2025*, 2025.

[11] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, 2020.

[12] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[13] Y. Wang, R. Wu, Y. Chen, J. Wang, J. Liang, Z. Zhu, H. Geng, J. Malik, P. Abbeel, and H. Dong, "Dexgarmentlab: Dexterous garment manipulation environment with generalizable policy," 2025. [Online]. Available: https://arxiv.org/abs/2505.11032

[14] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," 2024.

[15] A. Goldberg, K. Kondap, T. Qiu, Z. Ma, L. Fu, J. Kerr, H. Huang, K. Chen, K. Fang, and K. Goldberg, "Blox-net: Generative design-for-robot-assembly using vlm supervision, physics, simulation, and a robot with reset," in *2025 International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[16] J. Styrud, M. Iovino, M. Norrlöf, M. Björkman, and C. Smith, "Automatic behavior tree expansion with llms for robotic manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1225–1232.

[17] Y. Huang, C. Agia, J. Wu, T. Hermans, and J. Bohg, "Points2plans: From point clouds to long-horizon plans with composable relational dynamics," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[18] S. Patel, X. Yin, W. Huang, S. Garg, H. Nayyeri, L. Fei-Fei, S. Lazebnik, and Y. Li, "A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 8258–8266.

[19] R. Fox, R. Berenstein, I. Stoica, and K. Goldberg, "Multi-task hierarchical imitation learning for home automation," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1–8.

[20] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta, and A. Goyal, "HAMSTER: Hierarchical action models for open-world robot manipulation," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=h7aQxzKbq6

[21] C. Sun, S. Huang, H. Liu, J. Gong, and D. Pompili, "Retrieval-augmented hierarchical in-context reinforcement learning and hindsight modular reflections for task planning with llms," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1217–1224.

[22] K. Ryu, Q. Liao, Z. Li, P. Delgosha, K. Sreenath, and N. Mehr, "Curricullm: Automatic task curricula design for learning complex robot skills using large language models," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 4470–4477.

[23] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee, *et al.*, "Molmoact: Action reasoning models that can reason in space," *arXiv preprint arXiv:2508.07917*, 2025.

[24] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. Nesnas, and M. Pavone, "Semantic anomaly detection with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1035–1055, 2023.

[25] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," *arXiv preprint arXiv:2306.15724*, 2023.

[26] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo, "Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation," *arXiv preprint arXiv:2410.00371*, 2024.

[27] C. Agia, R. Sinha, J. Yang, Z. Cao, R. Antonova, M. Pavone, and J. Bohg, "Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=yqLFb0RnDW

[28] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," *arXiv preprint arXiv:2303.07280*, 2023.

[29] Y. Huang, N. Alvina, M. D. Shanthi, and T. Hermans, "Fail2progress: Learning from real-world robot failures with stein variational inference," *arXiv preprint arXiv:2509.01746*, 2025.

[30] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *CoRR*, vol. abs/1801.01290, 2018. [Online]. Available: http://arxiv.org/abs/1801.01290

[31] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," *arXiv preprint arXiv:2407.08693*, 2024.

[32] H. Lu, R. Wu, Y. Li, S. Li, Z. Zhu, C. Ning, Y. Shen, L. Luo, Y. Chen, and H. Dong, "Garmentlab: A unified simulation and benchmark for garment manipulation," in *Advances in Neural Information Processing Systems*, 2024.

[33] OpenAI, "Gpt-5 system card," https://cdn.openai.com/gpt-5-system-card.pdf, August 2025, openAI model documentation.

[34] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE robotics & automation magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[35] M. Quigley, J. Faust, T. Foote, J. Leibs, *et al.*, "Ros: an open-source robot operating system," in *IEEE International Conference on Robotics and Automation Workshop on Open Source Software, 2009*. IEEE.

[36] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714