2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

DexDLO: Learning Goal-Conditioned Dexterous Policy for Dynamic Manipulation of Deformable Linear Objects

Sun Zhaole[1], Jihong Zhu[2] and Robert B. Fisher[1]

***Abstract*— Deformable linear object (DLO) manipulation is needed in many fields. Previous research on deformable linear object (DLO) manipulation has primarily involved parallel jaw gripper manipulation with fixed grasping positions. However, the potential for dexterous manipulation of DLOs using an anthropomorphic hand is under-explored. We present DexDLO, a model-free framework that learns dexterous dynamic manipulation policies for deformable linear objects with a fixed-base dexterous hand in an end-to-end way. By abstracting several common DLO manipulation tasks into goal-conditioned tasks, DexDLO can perform tasks such as DLO grabbing, DLO pulling, DLO end-tip position controlling, etc. Using the Mujoco physics simulator, we demonstrate that our framework can efficiently and effectively learn five different DLO manipulation tasks with the same framework parameters. We further provide a thorough analysis of learned policies, reward functions, and reduced observations for a comprehensive understanding of the framework.**

I. INTRODUCATION

Deformable linear object (DLO) manipulation, e.g. ropes, cables, and rods, is widely applicable in surgical theaters, offices, textile factories, and other industries [1], [2]. Current research in DLO manipulation largely relies on single or dual parallel pinch grippers or end-effectors attached to a fixed end-tip of the DLO [3], [4], [5], [6], [7], [8]. However, without task-specific customization, such end-effectors cannot provide sufficient dexterity for DLO manipulation tasks like in-hand DLO sliding and DLO weight pulling (see Figure 2 (a), (b), and (c) respectively). On the other 'hand', an anthropomorphic hand, as a versatile end-effector, has the potential to handle all the aforementioned tasks.

There are three common practices in the above-mentioned *traditional* DLO manipulation methods: 1) quasi-static DLO manipulation with a near zero velocity, 2) fixed grasp on the DLO, and 3) customized end-effectors to execute a specific DLO task. Compared to a two-finger gripper or DLO end-tip fixed end-effector, an anthropomorphic hand can avoid relying on these requirements during DLO manipulation, as shown in Figure 1, where the anthropomorphic hand grabs a DLO to fetch its end-tip. The major technical challenges of dexterous DLO manipulation are:

- **Dynamic manipulation**. Previous continuous control methods mostly manipulated a DLO in a quasi-static state, assuming the DLO's velocity is near zero [3], and this assumption excludes some scenarios with non-zero velocities and higher manipulation speeds. Though handling the complexity of dynamics during real-time manipulation of the DLO is very difficult, dexterous manipulation of DLOs can benefit, making it more adaptable to different tasks.
- **Changing grasping positions**. Preventing the DLO from slipping or falling from the hand during changing grasping positions is challenging for parallel jaw grippers. Dexterous hands offer a unique advantage for this task without having to place and regrasp the DLO, which is often unavailable without a supporting surface, e.g. when picking and placing a rope on a table [7].
- **General end-effectors**. Different end-effectors are often used for different DLO manipulation tasks, including specialized end-effectors, e.g., a gripper with tactile sensors [9] for DLO following[1] and a specially designed gripper for rolling flat cables [10]. Dexterous hands are general-purpose end-effectors suitable for various rigid object manipulation tasks. Chen et al. [11] have shown that reorienting long and thin rigid objects in hand is difficult. Thus, it is even more difficult to use dexterous hands to manipulate DLOs, which are *highly deformable* long and thin objects and have high DoFs.

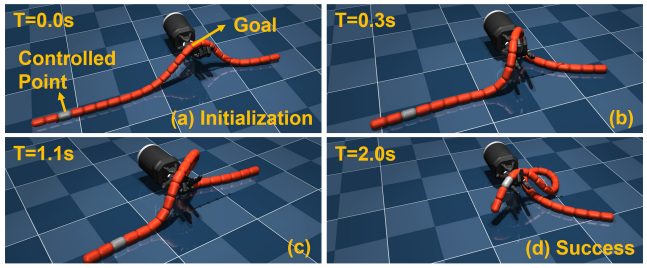

Fig. 1: An example of goal-conditioned dexterous manipulation of a deformable linear object. The *controlled point* (the grey segment) on the DLO is manipulated to minimize the distance to its *goal position* (the hand palm) and to finally reach the *goal position* using a base-fixed dexterous hand.

Some works have already explored one or two aspects, e.g., DLO following to change grasping positions within the gripper with tactile sensors [9] or specialized end-effectors [10] and DLO dynamic whipping with a fixed grasping position [4], [6], [12]. However, none have systematically studied the dexterous manipulation of DLOs to address all three challenges, as discussed in Section II.

To address the mentioned challenges, we introduce **DexDLO**, a reinforcement learning-based framework for

[1]Sun Zhaole and Robert B. Fisher are with the School of Informatics, University of Edinburgh, UK. Corresponding author: zhaole.sun@ed.ac.uk
[2]Jihong Zhu is with School of Physics, Engineering and Technology, University of York, UK

[1]Here, 'following' is defined to mean sliding the DLO through the hand.

DLO dexterous manipulation, together with a pose-regularized reward function. DexDLO is an end-to-end framework designed to control a dexterous hand to minimize the distance between a selected point **X** on the DLO and a goal position **G** in a dynamic way without explicitly regrasping the DLO or moving the hand base (see Figure 1 for an example). Without changing any structure or parameters, DexDLO can solve a series of goal-conditioned DLO manipulation tasks by formulating these tasks into goal-conditioned forms.

The DexDLO framework is evaluated on five different goal-conditioned tasks [2]. This paper's contributions are:

- A general formulation of goal-conditioned deformable linear object (DLO) manipulation as applicable to several challenging DLO manipulation tasks (see Section III-A).
- The first general framework that learns manipulation policies that can perform a series of goal-conditioned dexterous DLO manipulation tasks (see Section III-B for the framework and Section IV-B for evaluation).
- An analysis of removing certain observations and reward terms on the learned policy, finding that the observation space can be further reduced and pose-regularized reward terms are essential for training (see Section IV-D).

II. RELATED WORK

Deformable linear object (DLO) manipulation has long been studied with various end-effectors on different tasks, including: 1) shape control by using a parallel-jaw gripper for pick-and-place shape control of the DLO [7], [13], and fixing two end-tips of the DLO to do 3D shape control [3], [5], 2) tying knots with dual grippers [14], [15] and a three-finger hand [16], 3) untangling DLOs lying on a table with discrete pulling actions by two grippers [17], 4) DLO insertion into a hole, by grasping the DLO with a gripper [18], [8], 5) whipping, by fixing one end of the DLO and fixing the other end to the robot arm [4], 6) following/sliding, moving along the DLO from one part to another (usually the end-tip), which requires a relative movement between the end-effector and the DLO, by either using a specially designed gripper [10] or tactile information [9].

In the referenced literature, frameworks are often designed for specific tasks and have limited versatility. However, by consolidating the formulations of various DLO manipulations into a goal-conditioned approach, we introduce a model-free, end-to-end framework based on reinforcement learning. The proposed single framework can handle a wide range of DLO manipulation tasks without needing customization for each specific task.

Some DLO manipulation works rely on model-based methods, and estimating a DLO model is a common practice [3], [5], [9], so that a control-based algorithm can manipulate the DLO into a target shape. Such methods need first to estimate the DLO's model parameters, such as radius, stiffness, and damping [5], [19] or collect DLO movement data [20], [3]. Different from these model-based methods which require parameter estimations, our framework can adapt to DLOs with a wide range of physical parameters.

[2]See the experiment video: https://youtu.be/wkLHlThNwDI

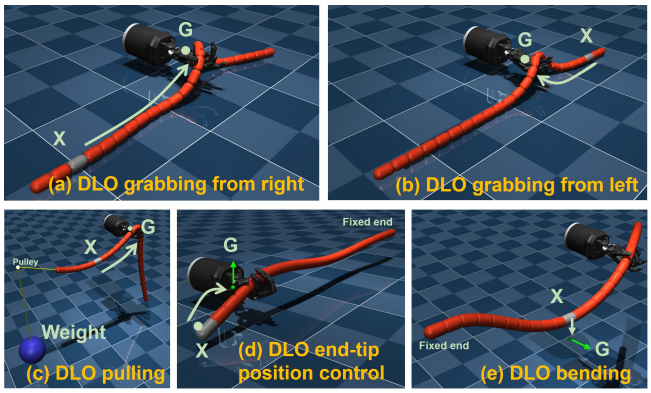

Fig. 2: Five example goal-conditioned DLO manipulation tasks. The grey segment on the DLO is the controlled point X, and the goal G is at the hand palm or highlighted as the green motion direction. (a) and (b) are two DLO grabbing tasks but from different directions. (c) is DLO pulling, where the weight is hung through a fixed pulley and connected to the end-tip of the DLO. (d) is DLO end-tip position control where the hand is fixed close to the end-tip, and another end-tip is fixed. (e) is DLO bending where the hand grasps near to one end-tip, another end-tip is fixed, and the middle point of the DLO is the controlled point.

Recently, various simulation environments have been designed to simulate DLOs, including MuJoCo [21], Unity with ObiRope asset [22], Reform [23], and SoftGym [24]. These environments support DLO data collection for model-based and model-free reinforcement learning.

Goal-conditioned manipulation achieves different goals in particular manipulation task scenarios without considering intermediate partial-solution states [4]. It is a popular approach to dexterous manipulation and deformable object manipulation. Previous works focused on dexterous manipulation. These include in-hand object reorientation to a goal pose [11], [25], [26], solving Rubik's cube [27], tossing cubes to goal positions [26], using tools to finish goal tasks [28], [29]. These works also include deformable object manipulation, including rope whipping to hit a goal in the air [4], [12], [6], shape control [7], [3], [13], insertion [18], and DLO following [9]. A gap still exists in the integration of goal-conditioned DLO manipulation with dexterous manipulation.

III. APPROACH

This section formulates goal-conditioned DLO Manipulation and then briefly describes the DexDLO framework.

A. Goal-conditioned DLO Manipulation

Define a DLO as n rigidly connected keypoints $P_0, P_1, ..., P_{n-1}$. Here, $n = 25$. All $P_i \in R^3$ are point positions along the DLO. A goal-conditioned policy is a point-to-point policy that moves a designated point $X = P_i \in R^3$ $(i = 0, 1, 2..n-1)$ to a specified goal position $G \in R^3$. In each task, X is a specific keypoint that remains

fixed during the given task. A successful action completion occurs when $||X - G||_2 < d$, where d is a fixed threshold.

Five different DLO tasks are introduced.

DLO Grabbing. The goal is to grab a specified point $X = P_i$ on the DLO by the base-fixed Shadow right hand. The DLO is initially placed on the upper area of the hand palm and falls freely, where P_0 is on the right-hand side and P_{24} is on the left-hand side. The hand needs first to grasp the DLO and use its motions to move the point X into the goal position G in the hand palm without dropping the DLO. There are two DLO grabbing tasks, specified by setting X to be either P_2 or P_{22}, which the hand needs to then slide from the right to the left (or the opposite way), as shown in Figure 2 (a) and (b). The success threshold is set to $d = 0.08m$. These two tasks are inspired by previous cable following work [9] and the special end-effector designed to grab the DLO [10].

DLO Pulling. The DLO is pulled over a pulley with a weight attached to the DLO at the right-hand side of the hand, as shown in Figure 2 (c). The hand must pull the DLO to counter the external force caused by gravity on the weight. X is set to P_4 and G is set to the hand palm. The pulley is abstracted to be an anchor fixed at a certain place connecting to P_0. By pulling the DLO, P_5 moves to G, and the weight can be lifted. The maximum weight and success threshold are set to $0.4kg$ and $d = 0.08m$.

DLO End-tip position control. The controlled keypoint $X = P_0$ of the DLO is manipulated into the goal position G in 3D space. We assume that the end-tip of the DLO is grabbed, and the hand should further manipulate it to a certain position, a simplified version of the DLO insertion task [18]. The goal position G is sampled in a spherical space whose radius equals $0.1m$. The center of the spherical goal space is the middle point between the initial position of X and the hand palm. The hand is fixed $10cm$ beneath P_7. The left-hand side end-tip P_{24} is fixed. The success threshold is $d = 0.05m$.

DLO Bending. Although DLO shape control with one robot arm or a dual-arm system using fixed end-effectors, like grippers has been well-studied previously [3], [5], the research assumed that the grasping position on the DLO is fixed and the shape is controlled by moving the gripper. Here, this problem is solved from a different perspective in a simplified version. We assume the right-hand side end-tip P_0 of the DLO is fixed. The controlled keypoint $X = P_{11}$ near the middle of the DLO is bent at a goal position G. The hand is fixed $10cm$ beneath the initial position of P_{20} near the left-hand side end-tip P_{24}. Since the reachable places of the middle keypoint are unknown, we only consider if the height of G can be reached by X, where G is $0.1m$ to $0.4m$ beneath the initial position of X. The bending is successful once X reaches the height of G. The success threshold is $d = 0.05m$.

B. DexDLO Framework

To solve the problem of dexterous manipulation of DLOs, model-free reinforcement learning (RL) is used.Proximal

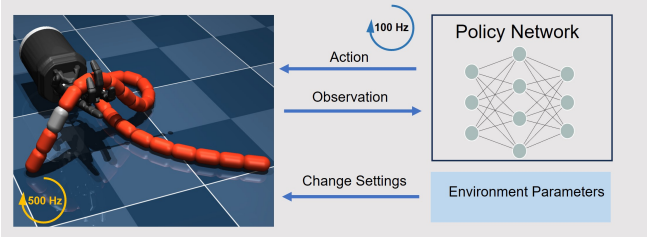

Fig. 3: Policy training. The policy network predicts actions at 100Hz, while the hand controller runs at 500Hz to execute actions.

Policy Optimization (PPO) [30] is used to learn a policy $\pi_\theta(o)$, which takes observations $o \in O$ as input and predicts actions $a \in A$, to maximize the expected episodic discounted sum of rewards $max_\theta \ E_{\pi_\theta}(\sum_{t=0}^{T} \gamma^t r(s_t, a_t))$. We use the PPO implementation from Stable-Baseline3 [31] with the **default** setting without tuning any hyper-parameters except the MLP structure, which has 1024-512-128 nodes for each layer with ReLU as the activation function. Figure 3 shows the policy training model.

Observation Space O. The observation space is given in Table I. The dimension of the observation O_t is 393 at each timestep t. A four-frame stack with $\{O_t, O_{t-1}, O_{t-2}, O_{t-3}\}$ is the policy network input so that the temporal dynamics can be modeled.

The observation space is redundant. Removing some observations while maintaining similar performance is possible. These removable observations include 1) palm-DLO position and vector, 2) some DLO keypoint positions, and 3) hand joint velocities. Simplification is discussed in Section IV-D.

TABLE I: Observation Space

Observation	Dim	Observation	Dim
Hand joint positions	24	Hand joint velocities	24
DLO keypoint positions	3×25	Hand fingertip positions	3×5
Hand palm position	3	Goal position	3
DLO to palm vectors	3×25	DLO to palm distances	25
DLO to fingertip distances	5×25	Target keypoint to goal distance	1
Target keypoint to goal Vector	3	Actions	20

Action Space A. The Shadow hand contains 24 DoFs and 20 actuators that control the joint angles. The actor network predicts a 20-dim vector as the target joint angles. The Shadow hand's joints move to the target joint angles using a PD controller. Following Chen et al. [11], we add a joint angle change limit of 0.4 rad per timestep for each joint by clipping actions outside of the limit to avoid rapid motions.

Reward Function R. The reward function is important for guiding the agent to learn complicated manipulation skills. An intuitive, sparse goal-reaching reward may not provide enough guiding information when the search space is too large or complicated. For example, DexPBT [26] used a multiple-stage reward function to learn their policy and Sun et al. [32] used a two-term reward function for two primitives. For all tasks, we use a unified reward function: $r = r_{reach} + r_{disabled} + r_{success} + r_{failure}$, where the four reward terms are:

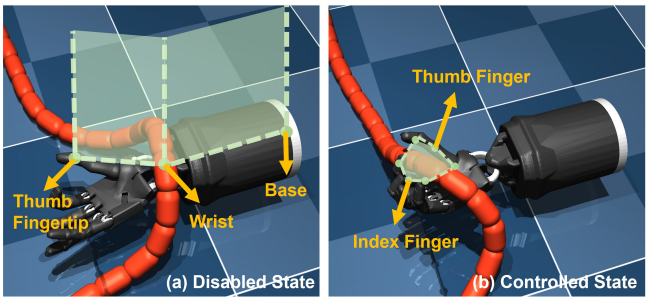

Fig. 4: Disabled state and controlled state. (a) The disabled state is set when the DLO penetrates the green area formed by the upper plane between the thumb fingertip, the wrist, and the base. (b) The controlled state is set when the DLO penetrates the green area, which consists of multiple triangles formed by the thumb and the index finger,

1. Goal-reaching-based reward r_{reach}, where $r_{reach}(\Delta \geq 0) = \lambda_1 \Delta \mathbb{1}_{controlled}$ and $r_{reach}(\Delta < 0) = \lambda_1 \Delta$. The reaching reward is based on the change of the distance $\Delta = ||X_{t-1} - G||_2 - ||X_t - G||_2$ between each timestep. The positive reaching reward is applied only if the DLO is in the controlled state, meaning $\mathbb{1}_{controlled} = 1$. A DLO held by the hand with the thumb and index finger is in the controlled state, as shown in Figure 4 (b). The controlled state encourages the DLO to be grasped in the closure between the thumb and index fingers, so that the thumb can perform its control action. We choose $\lambda_1 = 100$, where the length of the DLO is around 1.45m.

2. Pose-regularized reward $r_{disabled} = \lambda_2 \mathbb{1}_{disabled}$. The hand can barely control the DLO when the DLO is positioned above the wrist area. To avoid this location, we set an uncontrollable state as $\mathbb{1}_{disabled}$ and penalize the agent when the hand-DLO system reaches this state. $\mathbb{1}_{disabled} = 1$ indicates the hand-DLO system is in the disabled state. Like the controlled state, the DLO is in a disabled state when it penetrates the vertical surface from the thumb to the wrist and the base, shown in Figure 4 (a). We use $\lambda_2 = -1$.

3. Success reward $r_{success} = \lambda_3 \mathbb{1}_{success}$. When the target point on the DLO reaches the goal position within the threshold distance d, as mentioned in Section III-A for each task, for more than 5 timesteps without leaving, then this episode is a success. We use $\lambda_3 = 100$.

4. Failure reward(penalty) $r_{failure} = \lambda_4 \mathbb{1}_{failure}$. There are two failure cases: 1) the DLO dropping away from the hand and 2) staying in the disabled state for over 50 timesteps. More specifically, a dropping failure is when the distance from the nearest DLO keypoint to the palm is further than $10cm$, or the highest DLO keypoint is less than $5cm$ above the floor. We use $\lambda_4 = -100$.

IV. EXPERIMENTS

This section presents the experiment setup, performance evaluation, learned policy analysis, and ablation studies on observation and reward designs.

A. Experiment Setup

MuJoCo [21] is the simulation environment and a simulated Shadow Right Hand is the dexterous hand, which is one of the most commonly used dexterous hands in previous works [11], [29], [27]. The Shadow Hand is an anthropomorphic robotic hand with 20 actuators and 24 DoFs. The DLO is modeled by 25 capsules connected with spherical joints. We only ran a single environment without parallelism in MuJoCo, and all experiments used a CPU having a clock speed of 3.5 GHz. The maximum episode length is normally 1500 steps (DLO pulling uses 3000 steps for the episode length). Each agent was trained with 1×10^6 steps, which take around 5 hours on a single CPU core. This can be further accelerated to spend around 30 minutes on a 10-core CPU with Sample Factory [33]. The agent's success criterion was evaluated every 1×10^4 steps. Each experiment is repeated three times, and the training curves in Figures 5 and 6 show the mean and variance.

TABLE II: DLO parameter randomization range. The stiffness unit is Nm/rad.

DLO parameter	Range
Radius (mm)	20 ± 3
Length (m)	1.45 ± 0.10
Stiffness of Soft DLO	0.020 ± 0.005
Stiffness of Medium DLO	0.2 ± 0.05
Stiffness of Stiff DLO	2 ± 0.5
Coefficient of Friction	0.5 ± 0.2
Mass (g)	250 ± 25

TABLE III: Hand-DLO Placement.

Hand-DLO Placement	Range
Hand Base Position (m)	$[0 \pm 0.04, 0 \pm 0.04, 0 \pm 0.04]$
DLO Initial Angles (rad)	0 ± 0.02

To introduce variation during training, environment parameters were different in each episode according to their sampling ranges, which include 1) DLO parameters, 2) environment setup, and 3) observation and action noise.

DLO parameters. DLO radius, length, three types of stiffness, friction, and mass were fixed at the range center until the 10-th successful episode, shown in Table II. Then, starting from the range center, an adaptive scheme expands the DLO parameter randomization range, depending on the number of successes. As successes accumulate, the sampling range extends linearly until, at 100 successes, it spans the full range given in Table II. For the two DLO grabbing and the DLO bending tasks, DLOs with medium stiffness are used. For DLO pulling, a soft DLO is used. The DLO end-tip position control uses a stiff DLO.

Hand-DLO placement. The hand-DLO placement is varied after the initial 10 successes, based on the DLO angles and hand base positions given in Table III. The hand's base default position is at the origin of the environment. The DLO is placed initially $10cm$ above the hand palm. For each task, the DLO is placed at different positions compared to the hand, see Section III-A, but the hand still has a random relative shift compared to the default position. The initial angles of each DLO spherical joint are sampled according to Table III, and overall the DLO is placed as a nearly straight line above the hand. The floor height is set to $-0.12m$ for DLO grabbing and DLO end-tip position control and

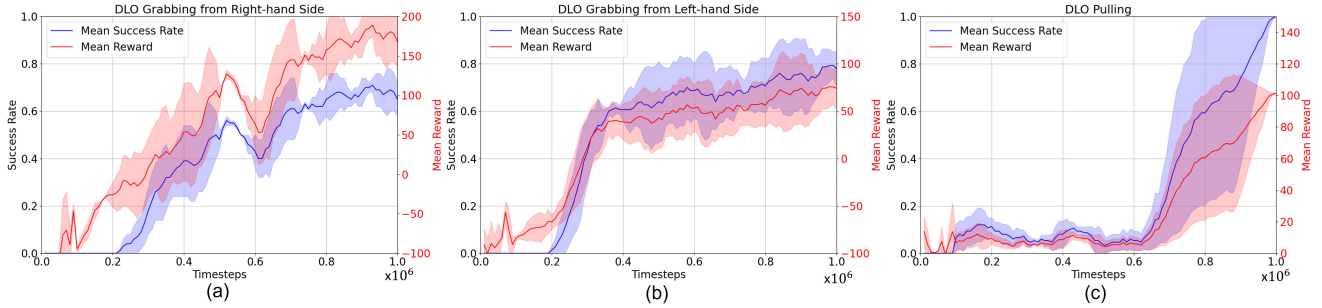

Fig. 5: Training curves for (a) DLO grabbing right, (b) DLO grabbing left, and (c) DLO pulling. Blue is the success rate, and Red is the mean reward per episode. The shaded area indicates the standard deviation of repeated trials.

TABLE IV: Std of Gaussian Noise Added to Observation and Action.

Observation & Action Noise	σ
Hand Joint Position (rad)	0.02
Hand Joint Velocity (rad/s)	0.02
DLO Keypoints Position (mm)	10
Fingertip Position (mm)	3
Action (rad)	0.1

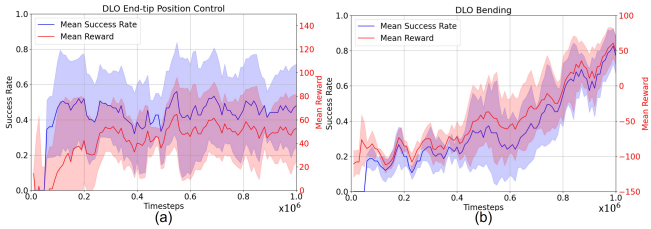

Fig. 6: Training curves for (a) DLO end-tip position control and (b) DLO bending. We use the same color as Figure 5.

$-0.6m$ for DLO pulling and bending. New environment setup parameters are uniformly sampled from the range after the initial 10 successes. The same adaptive range update scheme as for the DLO parameters was used to update the randomization ranges of hand-DLO placement. In addition, a special setting for DLO pulling initialized the weight to $0.01kg$ and added $0.01kg$ after each success until $0.4kg$.

Noisy observation. Noisy observations are important for easing Sim-to-Real implementation in the future. Table IV shows the variance of the observation noise of each type. A similar strategy for randomizing DLO parameters and environment setup parameters to the noise is used. Noise was added after the initial 10 successes and linearly increased from 1% to 100%, where each success leads to 1% increment on noise scales.

B. Experiment Results

Two evaluation success metrics were used: mean reward per episode in training and success rate of reaching the goal.

The performance of DexDLO was evaluated on three primary tasks 1) DLO grabbing from the right-hand side (DLO grabbing right for short), 2) DLO grabbing from the left-hand side (DLO grabbing left for short), and 3) DLO pulling. The results are displayed in Figure 5. The results show that DexDLO can achieve all three tasks. The agent achieved more than 60% success on DLO grabbing right, more than 80% success on DLO grabbing left, and almost

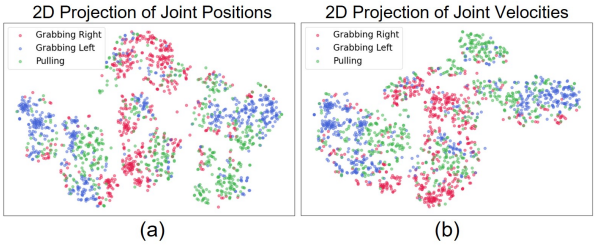

Fig. 7: 2D projection of hand joint positions (a) and velocities (b) from three tasks by applying t-SNE.

100% success on DLO pulling. The agents still showed an increasing performance trend when training was terminated.

DexDLO can also perform DLO bending and DLO end-tip position control, as shown in Figure 6. However, the performance on DLO end-tip position control is not as good as the other four tasks. DLO end-tip position control curves tend to be flat in the early stages of training, and the end-tip control does not increase in the rest of training. One possible reason is that the current framework finds it difficult to perform accurate goal-conditioned tasks. Another possible reason is improper goal setting. We do not guarantee the goal is reachable, and the controlled point may never reach some randomly sampled goal.

We further measured the speed of the policies. For the DLO grabbing right, it takes $1.9s$ to grab $0.6m$ on average. For the DLO grabbing left task, it takes $0.8s$ to grab $0.35m$ on average. For the DLO pulling with a $0.4kg$ weight, it takes $13.5s$ to pull $0.3m$ on average.

We show that an intuitive framework can solve a complicated control system where the actuators and the manipulated object both have very high DoFs, which is considered extremely challenging and has not been explored before.

C. Learned Policy Analysis.

We further studied if the learned policies of DexDLO on different tasks are similar or not. T-SNE [34], a popular tool to visualize high-dimensional data, was used to analyze three policies for DLO grabbing right, DLO grabbing left, and DLO pulling. Figure 7 (a) and (b) show the 2D projected hand joint positions and hand joint velocities from these three tasks. We observe that: 1) Both positions and velocities overlap in small areas for three tasks, indicating a common learned behavior among different tasks. 2) There is a large separation between DLO grabbing right and DLO

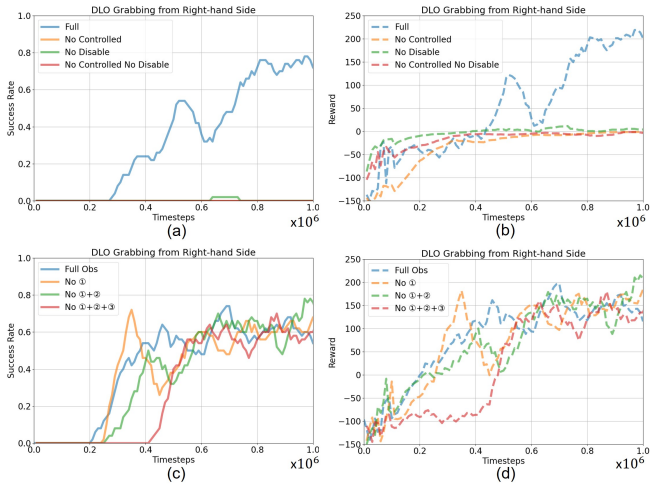

Fig. 8: Success rate and reward curves of different reward terms and lack of observations. (a) and (b) show success rate and reward curves with different reward terms. (c) and (d) show success rate and reward curves of removing certain observations. *Full obs* indicates using all observations. ① indicates hand palm positions and DLO-palm vectors. ② indicates hand joint velocities. ③ indicates additional 16 keypoints' positions based on existing 9 keypoints. The combination means removing multiple observations together.

grabbing left clusters. The different goals of the two tasks lead to different learned actions (note that the hand is not symmetric).

We speculate that the policies learned for different tasks have some common general dexterous manipulation behaviors and specialized goal-specific skills. However, drawing a solid conclusion requires further investigation.

D. Ablation Studies on Reward Function and Observations

The DLO grabbing right task was used for the ablation studies, which explored the effect of the different reward terms and observations.

Removing pose-regularized reward terms. Figure 8 (a) and (b) show that both reward terms $\mathbb{1}_{controlled}$ and $R_{disabled}$ (which also includes the failure condition) are essential during training the agent. By removing one or both pose-regularized reward terms, the agent has a near-zero success rate. $R_{disabled}$ can help the agent avoid useless explorations. $\mathbb{1}_{controlled}$ encourages the agent to learn more efficient manipulation skills by using the thumb and the index finger. The agent fails to learn the manipulation task when either reward term is removed.

Observation redundancy. As stated in Section III-B, there are many observation terms in the observation space, and we want to know if some of them are redundant and can be removed. We tested using fewer observations: 1) hand palm positions and DLO-palm vectors (103-d), 2) hand joint velocities (24-d), 3) evenly downsampled keypoints' positions (192-d, with only 9 keypoints rather than the full 25 keypoints, which also influences the dimensions of DLO-fingertip positions). Figure 8 (c) and (d) show the results of incrementally removing the observations. Even with fewer observations, the agent still learned the manipulation policy. This indicates that these observations are not decisive in the DLO grabbing task, and a potential exists to reduce the observation space in other tasks.

V. DISCUSSION AND CONCLUSION

A. Evidence supporting transfer to the real world

We highlight the importance of the following factors for Sim-to-Real transfer:

1. Observation space design: Previous works have investigated how to acquire the hand states, like fingertip positions and hand joint angles in the real world [25]. As for DLO state estimation, rather than using point clouds or RGBD images, the DLO state in the observation space is DLO keypoints' positions which can be detected and tracked using state-of-the-art algorithms that cope with with occlusions in real-time and in 3D space [35], [36], [37]. No DLO velocities in the observation space are needed. Additionally, in Section IV-D, we demonstrated that the agent can effectively learn the policy even when fewer keypoints are available.

2. DLO and dexterous hand simulation: We simulated the DLO in MuJoCo. The DLO simulated by a series of connected rigid capsule bodies has been used in Sim-to-Real [4], [6]. The contact between the DLO and the hand is between rigid objects, which simplifies the physics. The Shadow hand model is accurately modeled already [38] with tested realistic parameters. This makes the sim-to-real transfer more likely to proceed feasibly.

3. Randomization in the simulation environment: We randomized the DLO parameters, shown in Table II, different environment setups in Table III, and noise in the observation, shown in Table IV. The ability to tolerate this variation in parameters demonstrates the robustness of our policy and further enhances the sim-to-real capability.

4. Training time: A short training time is important to directly train the agent on real-world robots with minimal wear-and-tear. Nagaband et al. [39] implemented their training on a real-world Shadow hand with four hours training. Our simulation runs in real-time, around 60 steps per second, matching the speed of the real world. The total training time is 1×10^6 steps for each task, which takes less than 5 hours in simulation. It should be possible to directly train the agent on real-world hardware.

B. Summary

This paper proposed a general formulation of goal-conditioned DLO dexterous manipulation to generalize a set of common DLO manipulation tasks. Based on this formulation, we presented a unified framework, DexDLO, that learns goal-conditioned policies to perform the dexterous DLO manipulation tasks. We have evaluated the performance of the learned policies on each task and run ablation studies to validate the importance of the proposed pose-regularized reward terms. We discussed the capability and potential of our DexDLO for real-world dexterous DLO manipulations. Future work will focus on bridging sim-to-real gaps and combining model-based methods and model-free reinforcement learning to produce better performance.

REFERENCES

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[3] M. Yu, H. Zhong, and X. Li, "Shape control of deformable linear objects with offline and online learning of local linear deformation models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1337–1343.

[4] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *arXiv preprint arXiv:2203.00663*, 2022.

[5] 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.

[6] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8282–8289.

[7] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, "Learning predictive representations for deformable objects using contrastive estimation," in *Conference on Robot Learning*. PMLR, 2021, pp. 564–574.

[8] W. Wang, D. Berenson, and D. Balkcom, "An online method for tight-tolerance insertion tasks for string and rope," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2488–2495.

[9] 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.

[10] J. Chapman, G. Gorjup, A. Dwivedi, S. Matsunaga, T. Mariyama, B. MacDonald, and M. Liarokapis, "A locally-adaptive, parallel-jaw gripper with clamping and rolling capable, soft fingertips for fine manipulation of flexible flat cables," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6941–6947.

[11] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning*. PMLR, 2022, pp. 297–307.

[12] H. Zhang, J. Ichnowski, D. Seita, J. Wang, H. Huang, and K. Goldberg, "Robots of the lost arc: Self-supervised learning to dynamically manipulate fixed-endpoint cables," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4560–4567.

[13] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.

[14] K. Suzuki, M. Kanamura, Y. Suga, H. Mori, and T. Ogata, "In-air knotting of rope using dual-arm robot based on deep learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6724–6731.

[15] M. Saha and P. Isto, "Manipulation planning for deformable linear objects," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1141–1150, 2007.

[16] Y. Yamakawa, A. Namiki, M. Ishikawa, and M. Shimojo, "One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 703–708.

[17] 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," *arXiv preprint arXiv:2207.07813*, 2022.

[18] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8973–8979.

[19] M. Yu, K. Lv, C. Wang, Y. Jiang, M. Tomizuka, and X. Li, "Generalizable whole-body global manipulation of deformable linear objects by dual-arm robot in 3-d constrained environments," *arXiv preprint arXiv:2310.09899*, 2023.

[20] 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.

[21] 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, 2012, pp. 5026–5033.

[22] V. Studio, "Obi-unified particle physics for unity 3d," 2019.

[23] R. Laezza, R. Gieselmann, F. T. Pokorny, and Y. Karayiannidis, "Reform: A robot learning sandbox for deformable linear object manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4717–4723.

[24] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 432–448.

[25] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," *arXiv preprint arXiv:2210.13702*, 2022.

[26] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk, "Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training," *arXiv preprint arXiv:2305.12127*, 2023.

[27] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[28] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang, "Towards human-level bimanual dexterous manipulation with reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5150–5163, 2022.

[29] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[31] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[32] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li, "Learning pregrasp manipulation of objects from ungraspable poses," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9917–9923.

[33] A. Petrenko, Z. Huang, T. Kumar, G. Sukhatme, and V. Koltun, "Sample factory: Egocentric 3d control from pixels at 100000 fps with asynchronous reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7652–7662.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[35] K. Lv, M. Yu, Y. Pu, X. Jiang, G. Huang, and X. Li, "Learning to estimate 3-d states of deformable linear objects from single-frame occluded point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7119–7125.

[36] J. Xiang, H. Dinkel, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl, "Trackdlo: Tracking deformable linear objects under occlusion with motion coherence," *IEEE Robotics and Automation Letters*, 2023.

[37] 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.

[38] M. M. Contributors, "MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo," 2022. [Online]. Available: http://github.com/deepmind/mujoco_menagerie

[39] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2020, pp. 1101–1112.