Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Implicit Neural-Representation Learning for Elastic Deformable-Object Manipulations

Minseok Song
KAIST
hjmngb@kaist.ac.kr

JeongHo Ha
KAIST
hajeongho95@kaist.ac.kr

Bonggyeong Park
KAIST
iampbk@kaist.ac.kr

Daehyung Park
KAIST
daehyung@kaist.ac.kr

***Abstract*—We aim to solve the problem of manipulating deformable objects, particularly elastic bands, in real-world scenarios. However, deformable object manipulation (DOM) requires a policy that works on a large state space due to the unlimited degree of freedom (DoF) of deformable objects. Further, their dense but partial observations (e.g., images or point clouds) may increase the sampling complexity and uncertainty in policy learning. To figure it out, we propose a novel implicit neural-representation (INR) learning for elastic DOMs, called INR-DOM. Our method learns consistent state representations associated with partially observable elastic objects reconstructing a complete and implicit surface represented as a signed distance function. Furthermore, we perform exploratory representation fine-tuning through reinforcement learning (RL) that enables RL algorithms to effectively learn exploitable representations while efficiently obtaining a DOM policy. We perform quantitative and qualitative analyses building three simulated environments and real-world manipulation studies with a Franka Emika Panda arm. Videos are available at http://inr-dom.github.io.**

I. Introduction

Deformable object manipulation (DOM), as shown in Fig. 1, presents a major challenge in automation and has attracted increasing attention in robotics over the past decade [9]. Researchers have investigated a variety of DOM tasks, such as grasping [39], folding [34], wearing [32], threading [24], winding [25], tangling [36], and bagging [4]. These tasks introduce challenges in the domains of perception, modeling, planning, and control [48], due to infinite degrees of freedom (DoF) and nonlinear interaction dynamics of deformable objects (DO). Further, dense but partial observations—often resulting from self-occlusions—also increase sampling complexity and uncertainty in policy learning.

In DOM, data-driven modeling approaches are increasingly gaining attention with their extensive representation capabilities for downstream tasks. For example, Lippi et al. reduce a high-dimensional space of DOs into a low-dimensional state graph to facilitate planning [20]. Researchers often model a DO as a particle-based interaction graph to describe detailed topological structures [17, 21]. Recent studies aim to capture comprehensive graph-based models by reconstructing complete geometries such as point clouds and meshes from partial observations [13]. However, the discrete nature of particle- or graph-based models often fails to consistently represent the flexible and smooth surfaces of DOs.

Meanwhile, signed distance fields (SDFs) are receiving increasing interest with their ability to represent complex, non-convex geometries of objects [47]. SDFs not only describe the surfaces of objects, but also provide spatial information at a distance, making them well suited to depict physical interactions and facilitating manipulation planning [49]. To capture continuous and fine details, researchers parameterize SDFs using neural networks (i.e., implicit signed distance function) [37, 3]. However, most studies that involve SDFs in DOM focus primarily on enhancing encoding for improved reconstruction [41], which may significantly distract the exploration process of reinforcement learning (RL)-based policy learning.

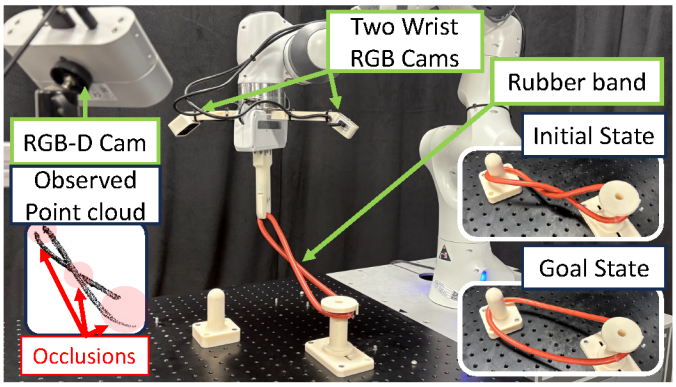


Fig. 1: A capture of deformable object manipulation task that requires disentangling elastic bands between two poles. The deformable and stretchable nature of bands increases the complexity of state representation. Further, the 360° twists create self-occlusions, significantly reducing the consistency of state embeddings. Our INR-DOM effectively captures the occlusion-robust implicit representation of the bands and efficiently generates real-world applicable manipulation policies.

We propose a novel implicit neural representation (INR) learning method for elastic DOM, which we call INR-DOM. Our method focuses on learning consistent and occlusion-tolerant representations of partially observable DOs and enhances task-relevant representations to optimize their manipulation policies efficiently. INR-DOM incorporates a two-stage learning process with two types of losses: 1) pre-training utilizes reconstruction and regularization losses to develop an occlusion-robust yet dense representation encoder, suitable for stretched or intertwined DOs; 2) fine-tuning employs contrastive learning and RL losses to refine representations, boosting exploitability and effectiveness in policy learning. The use of implicit SDF during pre-training allows for the reconstruction

of complete and implicit surfaces of highly elastic DOs. The integration of contrastive loss during fine-tuning further enables the encoder to effectively identify and handle complex, enclosed states, such as twisted rubber bands. We particularly introduce a temporal- and instance-wise key assignment method for time-series contrastive learning to better represent correlations between similar manipulation sequences.

We conduct quantitative and qualitative studies in both simulated and real-world manipulation environments. By constructing a 3D shape-recovery benchmark of nine elastic rubber bands, we show INR-DOM's consistent and occlusion-robust representation capabilities. Applying them to simulated environments, we then demonstrate the superior state-representation and policy-learning capabilities across three simulated environments (i.e., *sealing*, *installation*, and *disentanglement* tasks). INR-DOM excels in accurately recovering complete geometries from partial observations, surpassing state-of-the-art baseline methods, particularly in handling stretched or intricately enclosed states of rubber bands. Further, the fine-tuned representation captures both task-relevant and -irrelevant details, facilitating efficient policy convergence in RL and leading to significantly higher task success rates up to 41% higher than the next-best baseline approach. We also demonstrate the real-world applicability of INR-DOM with a Franka Emika Panda robot.

In summary, key contributions of this paper are threefold:

- We introduce an implicit neural-representation learning method that provides consistent and occlusion-robust representations from visual observation of deformable objects.
- We develop an effective representation-refining approach using a contrastive loss to capture task-relevant state information for RL.
- We demonstrate that INR-DOM significantly improves convergence stability in policy learning and success rate for DOM tasks, in both simulations and real-world settings.

II. Related Works

We investigate representation models and their learning or update methodologies for DOM tasks.

Representation models: The expressiveness of representations stems from the capacity of the underlying models. Most model-based approaches represent deformations using discrete structures composed of a finite number of elements, such as points or lines. Earlier approaches, including the finite element method (FEM), approximate complex, irregular geometries by partitioning objects into smaller elements that collectively represent the overall shape [18, 29]. However, their reliance on the mesh structures limits their ability to handle large deformations, particularly in elastic materials (e.g., bands). Alternatively, researchers often adopt particle-based, data-driven representations: point clouds [5] and voxels [6]. However, these approaches often struggle to capture continuum behaviors such as stretching or compression. Recently, studies build interaction graphs to leverage structural connectivity through graph-based representations [17, 19, 21]. Nonetheless, the inherently discrete nature of these models gives challenges representing continuous surfaces.

Alternatively, researchers often adopt model-free approaches that directly map raw observations to feature vectors—using 2D convolutional encoders for RGB images [43, 34] or PointNet [30] for 3D point clouds [16]. However, these methods lack the dense geometric representations required for downstream tasks such as precise tying or untwisting. To address this limitation, implicit neural representations have demonstrated superior capabilities in capturing dense correspondence and dynamics in deformable objects. Shen et al. reconstruct high-fidelity voxel geometries from simulated RGB-D images [35], while Wi et al. model the continuous representation of deformed geometries based on external forces and their locations [41]. Our approach not only provides dense descriptors but also distinguishes complex and overlapping shapes in real-world scenarios.

Representation Learning: Early approaches often rely on autoencoders with reconstruction losses to obtain low-dimensional features. While this enable training with large unlabeled datasets, the resulting representations are task agnostic and may distract explorations, leading to sample inefficiency in RL [46]. To address this issue, researchers introduce end-to-end learning approaches that jointly train the encoder and the action network [40]. In addition, contrastive learning [11] have emerged as another metric-learning technique for representation learning by promoting similarity constraints on the same or nearby data while maximizing separation from unrelated one [35]. Leveraging this contrastive learning, Srinivas et al. improve the sample efficiency in model-free RL [14] by integrating the contrastive loss as an auxiliary objective. However, the standard separation of values in time series often makes it difficult to preserve similarities between temporally correlated time-series instances in self-supervised learning [15]. In this work, we fine-tune the encoder using a contrastive loss and enhance the sample selection strategy by exploiting temporal structure in experienced trajectories, aiming to improve exploration efficiency and promote more generalizable latent representations.

III. Methodology

A. Overview

The objective of INR-DOM is to jointly learn consistent state embeddings ($\mathbf{s} \in \mathcal{S}$) from partial observations ($\mathbf{o} \in \mathcal{O}$) and an RL policy π for DOM. We formulate the policy learning problem as an MDP tuple $< \mathcal{S}, \mathcal{A}, T, R >$ with state space $\mathcal{S}$, action space $\mathcal{A}$, stochastic transition function T, and reward function R. The goal of the MDP is to find an optimal policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the cumulative discounted reward. Fig. 2 shows the overall architecture for the state-representation learning (SRL) integrated with policy learning. The input observation $\mathbf{o}$ is a combination of proprioception and exteroception information, including a point cloud $\mathbf{p} \in \mathcal{P}$ representing DOs, where $\mathcal{P}$ denotes the space of observable 3D point clouds. We derive a coherent, low-dimensional state

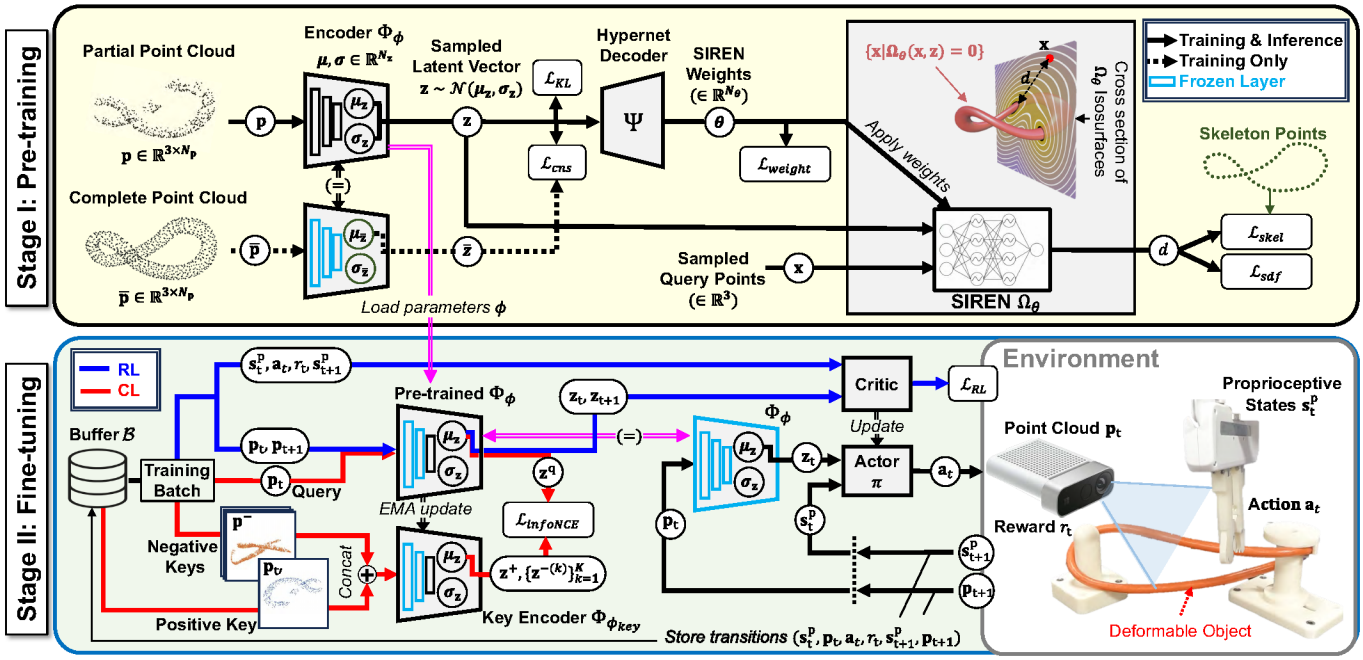


Fig. 2: An overview of INR-DOM framework that aims to train the occlusion-robust state representation encoder Φ_ϕ, parameterized by ϕ, of deformable objects (DOs) as well as the manipulation policy π. The training framework consists of two stages: 1) The first stage pre-trains a PointNet-based partial-to-complete variational autoencoder (Φ_ϕ, Ψ) that embeds a partial point cloud $\mathbf{p}$ of a target DO into a latent embedding $\mathbf{z}$ and recovers the parameters $\boldsymbol{\theta}$ of an implicit signed distance field (SDF) network Ω_θ. This stage predicts full geometries leveraging two loss functions $\mathcal{L}_{\text{SDF}}$, $\mathcal{L}_{\text{skel}}$, along with three regularization loss functions: $\mathcal{L}_{\text{KL}}$, $\mathcal{L}_{\text{weight}}$, and $\mathcal{L}_{\text{cns}}$. 2) The second stage then improves the task-relevant representation power of the encoder Φ_ϕ by jointly optimizing reinforcement learning (blue) with the loss $\mathcal{L}_{RL}$ and the contrastive learning (red) with the loss $\mathcal{L}_{\text{infoNCE}}$.

$\mathbf{s}$ by embedding the porint cloud $\mathbf{p}$ into a latent vector $\mathbf{z} \in \mathcal{Z}$ through a neural SRL function.

We design the neural SRL function, $\Phi : \mathcal{P} \to \mathcal{Z}$, as an encoder that embeds not only consistent but also task-relevant information of DOs from partial observations. To embed state representations for DOM policy learning, we introduce a two-stage training framework; Stage I) reconstruction-based pre-training (see details in Sec. III-B) and Stage II) contrastive-based fine-tuning (see details in Sec. III-C). Note that this work considers training in a physics simulation and transfers the learned SRL function to the real world. The following subsections detail the architecture and training procedures.

B. Reconstruction-based pre-training

The goal of the pre-training stage is to learn an occlusion-tolerant yet dense representation $\mathbf{z}$ from a partial point cloud $\mathbf{p}$ of elastic DOs. A key challenge lies in learning distinguishable representations of stretched or intertwined DOs. To address this, we detail the proposed SRL network architecture and its training process with the associated loss functions.

Network architecture. We design a reconstruction-based partial-to-complete variational autoencoder (VAE), capable of generating the parameters $\boldsymbol{\theta} \in \Theta$ of an implicit SDF network. Our proposed network consists of a PointNet-based encoder Φ parameterized by ϕ, a hypernetwork decoder Ψ, and an implicit neural SDF network Ω parameterized by $\boldsymbol{\theta}$,

$$\Phi_\phi : \mathcal{P} \to \mathcal{Z}, \quad \Psi : \mathcal{Z} \to \Theta, \quad \Omega_{\boldsymbol{\theta}} : \mathcal{X}, \mathcal{Z} \to \mathbb{R}, \tag{1}$$

where $\mathcal{Z}$, Θ, and $\mathcal{X}$ are the spaces of state embeddings, implicit network parameters, and Cartesian points ($\in \mathbb{R}^3$), respectively.

The encoder Φ is a modified PointNet that takes a point cloud $\mathbf{p}$ of a target DO, observed from a depth image, and returns a low-dimensional latent vector $\mathbf{z} \in \mathcal{Z}$. During training, we sample $\mathbf{z}$ using the reparameterization trick, while during testing, we use the predicted mean of a Gaussian distribution. The modification is the removal of T-Net in the original PointNet to distinguish the translation and rotation of point clouds. The vector serves as a global representation of the complete geometry, containing its continuous surface and kinematic structure information, despite the incomplete and occluded nature of the input point cloud. In this work, we use $\mathcal{Z} \subset \mathbb{R}^{N_\mathbf{z}}$ and $N_\mathbf{z} = 64$.

Unlike conventional VAE, the decoder Ψ is a hypernetwork that takes the latent vector $\mathbf{z}$ and generates weights $\boldsymbol{\theta}$ for the implicit SDF network Ω. The network is a multi-layer perceptron (MLP) with three hidden layers, where each layer has 256 units with ReLU activations.

The implicit SDF network $\Omega_{\boldsymbol{\theta}}$ is a variant of SIREN Hyponetwork [37], parameterized by the decoded weights $\boldsymbol{\theta}$, which takes a Cartesian query coordinate $\mathbf{x}$ and a current latent

vector $\mathbf{z}$. The network then returns a corresponding signed distance d from the nearest object surface, where the positive sign indicates outside of the surface and the negative sign indicates inside. The nature of the implicit network allows for retrieving continuous and differentiable distances from a complete surface. In contrast to the original SIREN, we input the point-cloud embedding $\mathbf{z}$, which helps the decoder Ψ effectively learn the residual part similar to DeepSDF [28]. Our network consists of a three-layer MLP with 32 units per layer and sinusoidal activations.

Losses. We introduce an INR learning loss to train the proposed network through simulated DOM data and transfer the learned model to the real world. Our proposed loss $\mathcal{L}$ is a linearly weighted combination of shape and structure reconstruction losses as well as regularization losses,

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{SDF}} + \lambda_1 \mathcal{L}_{\text{skel}}}_{\text{reconstruction}} + \underbrace{\lambda_2 \mathcal{L}_{\text{KL}} + \lambda_3 \mathcal{L}_{\text{weight}} + \lambda_4 \mathcal{L}_{\text{cns}}}_{\text{regularization}}, \quad (2)$$

where λ_i are non-negative constants ($i \in [1, 4]$).

First of all, the reconstruction-based losses aim to complete a given partial point cloud while precisely representing its continuous and dense geometry. We introduce two loss functions:

1) $\mathcal{L}_{\text{SDF}}$: An SDF loss to precisely fit the generated SDF with a ground-truth complete point-cloud $\bar{\mathcal{P}}$. Let $\mathcal{Q} \in \mathbb{R}^3$ and $\bar{\mathcal{Q}}$ denote all sampled query points and their subset on the object surface, respectively. Based on the Eq. (6) of [37], we define the loss function as:

$$\begin{aligned}\mathcal{L}_{\text{SDF}} = &\int_{\mathbf{x}\in\mathcal{Q}} \left| \|\nabla_{\mathbf{x}} \Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\| - 1 \right| d\mathbf{x} \\ &+ \int_{\mathbf{x}\in\bar{\mathcal{Q}}} |\Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})| + (1 - \nabla_{\mathbf{x}} \Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \cdot \mathbf{n}(\mathbf{x}))\, d\mathbf{x} \\ &+ \int_{\mathbf{x}\in\mathcal{Q}\setminus\bar{\mathcal{Q}}} \exp\left(-\alpha \cdot |\Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})|\right) d\mathbf{x}, \end{aligned} \quad (3)$$

where $\mathbf{n}(\mathbf{x})$ is the surface normal vector at the query point $\mathbf{x}$, α is a constant, and $\mathbf{z}$ is the current latent vector. The first term in Eq. (3) constrains inconsistent SDF gradients since the gradients are mostly one, except on critical points such as medial-axis points. The second term penalizes when the on-surface points have non-zero distances and also their SDF gradient direction is not aligned with the ground-truth normal vector. Lastly, the third term regularizes off-surface points not to have large values. Note that we obtain $\mathbf{n}(\mathbf{x})$ by simulation at each time step. We also sample not only random points for $\mathcal{Q}$ but also near-surface points to accurately estimate the distance around the surface.

2) $\mathcal{L}_{\text{skel}}$: A skeleton loss that is the measure of how far the estimated medial-axis point of a DO is off from the ground-truth medial axes, where the medial-axis point exhibits the same distance to multiple boundaries (i.e., surface) on the object [33]. This is crucial to accurately recover the geometries of intertwined or occluded regions in a partial point cloud. Here, we formulate the loss as follows:

$$\mathcal{L}_{\text{skel}} = \int_{\mathbf{x}\in\mathcal{Q}_*} \log\Big(\max(\Delta\Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}), \epsilon)^{-1}\Big) d\mathbf{x}, \quad (4)$$

where Q_* are the ground-truth medial-axis points, $\Delta\Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ represents the Laplacian of the query point $\mathbf{x}$ given $\Omega_{\boldsymbol{\theta}}$, and ϵ means a small constant introduced to prevent division by zero. Around the medial-axis points, $\Delta\Omega_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \to \infty$, as these points correspond to the local minima of $\Omega_{\boldsymbol{\theta}}$.

Next, the regularization losses aim to constrain the divergence of latent vector space and weights. We introduce three loss functions:

1) $\mathcal{L}_{\text{KL}}$: A Kullback-Leiber divergence loss (i.e., a regularization loss) that is a negative divergence $-D_{KL}(\Phi(\mathbf{z}|\mathbf{p})||p(\mathbf{z}))$ from a prior $p(\mathbf{z})$, parameterized by $p_0(\mathbf{z}) = \mathcal{N}(0, 1)$, to the variational approximation $\Phi(\mathbf{z}|\mathbf{p})$ of $p(\mathbf{z}|\mathbf{p})$. Our encoder Φ estimates the mean $\mu \in \mathbb{R}^{N_\mathbf{z}}$ and standard deviation $\sigma \in \mathbb{R}^{N_\mathbf{z}}$ of the posterior distribution $\Phi(\mathbf{z}|\mathbf{p})$. Then, $\mathcal{L}_{\text{KL}} = -0.5 \cdot \big(\log(\sigma^2) + 1 - \mu^2 - \sigma^2\big)$.
2) $\mathcal{L}_{\text{weight}}$: A weight-regularization loss defined as $\frac{1}{N_{\boldsymbol{\theta}}}\|\boldsymbol{\theta}\|_2^2$, where $N_{\boldsymbol{\theta}}$ is the number of weight parameters including biases, adopted from [37]. This formulation promotes a low-frequency SDF solution. In this work, we use $N_{\boldsymbol{\theta}} = 41,857$, which includes the parameters of the original SIREN along with our modifications.
3) $\mathcal{L}_{\text{cns}}$: A consistency loss, defined as $\frac{1}{N_\mathbf{z}}\|\mathbf{z} - \bar{\mathbf{z}}\|_2^2$, where $\mathbf{z}$ and $\bar{\mathbf{z}}$ are the embeddings of the partial point cloud $\mathbf{p}$ and the complete point cloud $\bar{\mathbf{p}}$, respectively. This loss regularizes the embedding of the partial point cloud $\mathbf{p}$ to be close to that of the complete point cloud $\bar{\mathbf{p}}$. Therefore, INR-DOM enables various views of point clouds to be mapped onto the same region in the latent space.

C. Policy learning with fine-tuning

The objective of this fine-tuning stage is to learn sample-efficient representations while optimizing a DOM policy. To address this, we combine RL and contrastive learning, similar to [14]. Fig. 2 Stage II shows the overall architecture that updates the pre-trained encoder Φ_ϕ, parameterized by ϕ, through RL while contrasting experienced sequences in the replay buffer $\mathcal{B}$. To better represent correlations between similar sequences, we introduce temporal- and instance-wise representation selections (i.e., key assignments). In the following, we detail our proposed RL-based fine-tuning and contrastive learning methods.

Reinforcement learning. We apply a deep RL framework to DOM as shown in Fig. 2 Stage II. Our framework aims to manipulate a DO, observed as a point cloud $\mathbf{p}_t$ at time step t, to a desired point-cloud status $\mathbf{p}_{\text{des}}$ using a six degree-of-freedom (DoF) robotic arm. We assume an RGB-D camera is available as shown in Fig. 2. Below, we describe our MDP details with the RL-based loss function $\mathcal{L}_{\text{RL}}$ to update the encoder Φ_ϕ:

- **State**: We define the state $\mathbf{s}_t$ as a tuple $(\mathbf{s}_t^{\text{p}}, \mathbf{p}_t)$ that consists of the arm and DO state information, where $\mathbf{s}_t^{\text{p}}$ is a proprioceptive state vector $[\mathbf{x}_{\text{ee}}, \mathbf{q}_{\text{ee}}, \dot{\mathbf{x}}_{\text{ee}}, \dot{\mathbf{q}}_{\text{ee}}, s_{\text{gripper}}]$ of the arm end-effector, s_{gripper} is the arm gripper's open and close state ($\in \{0, 1\}$).
- **Action**: We define the action $\mathbf{a}_t$ as a tuple $(\mathbf{a}_t^{\text{lin}}, \mathbf{a}_t^{\text{ang}}, \mathbf{a}_t^{\text{gripper}}) \in \mathbb{R}^7$, where $\mathbf{a}_t^{\text{lin}} \in \mathbb{R}^3$ and $\mathbf{a}_t^{\text{ang}} \in \mathbb{R}^3$

are the linear and angular velocities of the end effector, respectively. $\mathbf{a}_t^{\text{gripper}} \in \{0, 1\}$ indicates the gripper's open and close action.

- **Reward**: We define the reward function R as a linear combination of sparse and dense rewards. Let $\text{CD}(\cdot, \cdot)$ be a function that returns the Chamfer distance between two point clouds. Then,

$$R(\mathbf{s}_t, \mathbf{a}_t) = +\,100 \cdot \mathbb{1}_{\text{CD}(\mathbf{p}_t, \mathbf{p}_{des}) \leq \delta} - \alpha \cdot \text{CD}(\mathbf{p}_t, \mathbf{p}_{des}), \quad (5)$$

where δ is a distance threshold for success check and α is a constant.

In this work, we define the loss $\mathcal{L}_{\text{RL}}$ as a combination of actor, critic, and entropy losses in [10].

Our framework updates the encoder Φ_ϕ and its associated policy π employing off-policy RL, particularly soft actor-critic [10]. At each time step t, we embed a state $\mathbf{s}_t$ into a latent state $(\mathbf{s}_t^{\text{p}}, \mathbf{z}_t)$ using the encoder Φ_ϕ, and then take the embedding as input for the actor network π to determine an action $\mathbf{a}_t$. Simultaneously, we update the critic and encoder networks by back-propagating the loss $\mathcal{L}_{\text{RL}}$. During this update, we store transitions $(\mathbf{s}_t^{\text{p}}, \mathbf{p}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}^{\text{p}}, \mathbf{p}_{t+1})$ into the buffer $\mathcal{B}$, where r_t denotes the output from $R(\mathbf{s}_t, \mathbf{a}_t)$ and we use the buffer for sampling each batch $\mathcal{B}'$ for the off-policy updates.

Contrastive learning. To improve sample efficiency in RL, we adopt contrastive learning by adding an auxiliary task, inspired by CURL [14]. The task is to improve the discrimination capability of the *query* encoder Φ_ϕ while learning the DOM policy. The discrimination requires comparing query-key pairs to ensure that a query input $\mathbf{z}^{\text{q}}$ is close to a positive keys $\mathbf{z}^+$ and far away from negative keys $\{\mathbf{z}^{-(k)}\}_{k=1}^K$, where K is the number of negative keys. However, conventional key generation often does not capture the similarity of keys in time series [15]. To address this issue, we introduce a novel query-key selection strategy for time series and the update of the *query* encoder leveraging an information noise-contrastive estimation (InfoNCE) loss [27].

Consider a batch $\mathcal{B}'$ that contains sequences (i.e., episodes) of experience $[\mathcal{E}^{(1)}, ..., \mathcal{E}^{(|\mathcal{B}'|)}]$. Note that, for notational simplicity, we assume the batch $\mathcal{B}'$ is a set of point-cloud sequences $[\mathcal{E}_{\text{p}}^{(1)}, ..., \mathcal{E}_{\text{p}}^{(|\mathcal{B}'|)}]$. We represent the i-th sequence as $\mathcal{E}_{\text{p}}^{(i)} = [\mathbf{p}_1^{(i)}, ..., \mathbf{p}_T^{(i)}] \in \mathbb{R}^{(3\times N_p)\times T}$, where N_p and T denote the number of points and the sequence length, respectively. We then represent the sequence embedding as $\mathcal{E}_{\mathbf{z}}^{(i)} = \Phi_\phi(\mathcal{E}_{\text{p}}^{(i)}) = [\mathbf{z}_1^{(i)}, ..., \mathbf{z}_T^{(i)}] \in \mathbb{R}^{N_{\mathbf{z}}\times T}$.

For the contrastive-loss computation, we generate query-key pairs. For the query selection, we randomly sample a point cloud $\mathbf{p}_t^{(i)}$ to obtain $\mathbf{z}^{\text{q}} = \Phi_\phi(\mathbf{p}_t^{(i)})$ as a query input from a randomly selected episode $\mathcal{E}_{\text{p}}^{(i)}$. For the positive-key selection, we first sample the top-M similar episodes from the buffer $\mathcal{B}$ with respect to the episode $\mathcal{E}_{\text{p}}^{(i)}$. In this work, for computational efficiency, we determine the similarity based on the minimal start-and-goal embedding distance, $d_{\text{sg}}(\mathcal{E}_{\text{p}}^{(i)}, \mathcal{E}_{\text{p}}^{(j)}) = \mathbf{z}_1^{(i)} \cdot \mathbf{z}_1^{(j)} + \mathbf{z}_T^{(i)} \cdot \mathbf{z}_T^{(j)}$, where '$\cdot$' represents a dot product. We then select a point cloud $\mathbf{p}_{t'}$ in an episode with the minimum dynamic time-warping (DTW) distance from $\mathcal{E}_{\text{p}}^{(i)}$ to obtain $\mathbf{z}^+ = \Phi_{\phi_{\text{key}}}(\mathbf{p}_{t'})$,

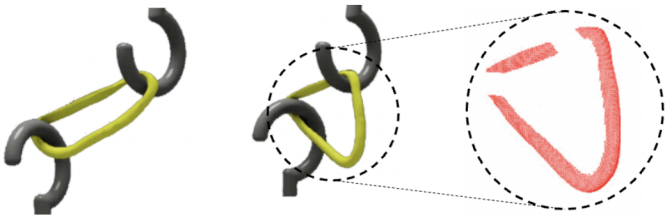


Fig. 3: Examples of randomly twisted and stretched rubber bands in simulation. The red points represent partial point clouds.

where $\Phi_{\phi_{\text{key}}}$ is the *key* encoder [12], initialized with ϕ. Note that t' is a time step matched to the time step t of $\mathcal{E}_{\text{p}}^{(i)}$ by DTW. For the negative key selections, we randomly sample the K number of point clouds from $\mathcal{B}' \backslash \{\mathcal{E}_{\text{p}}^{(i)}\}$ and embed them as $\{\mathbf{z}^{-(k)}\}_{k=1}^K$ using $\Phi_{\phi_{\text{key}}}$.

Given the query-key pairs, we compute the InfoNCE loss $\mathcal{L}_{\text{infoNCE}}$ to optimize the encoders Φ_ϕ and $\Phi_{\phi_{\text{key}}}$:

$$\log \frac{\exp(\mathbf{z}^{\text{q}} \cdot \mathbf{z}^+ / \tau)}{\exp(\mathbf{z}^{\text{q}} \cdot \mathbf{z}^+ / \tau) + \sum_{k=1}^K \exp(\mathbf{z}^{\text{q}} \cdot \mathbf{z}^{-(k)} / \tau)} \quad (6)$$

where τ is a temperature parameter ($\in \mathbb{R}^+$). As MoCO [12], we update $\phi_{\text{key}} = m\phi_{\text{key}} + (1 - m)\phi$ using the exponential moving average (EMA) method, where m is the momentum coefficient ($\in [0, 1)$). The encoder is then able to capture subtle distinctions between object configurations, which is critical for manipulation tasks involving deformable objects. In this work, we set $\tau = 0.1$.

In summary, the fine-tuning stage of INR-DOM, inspired by the CURL framework, involves refining the latent space using contrastive learning to capture task-relevant features while maintaining generalization. This enhanced representation, coupled with RL, enables the model to learn effective manipulation policies more efficiently.

IV. Experimental Setup

Our experimental evaluation aims to answer two key questions: 1) Does the proposed representation provide consistent and occlusion-robust state information for DOM? 2) Further, does the proposed method improve the effectiveness of DOM in the real world? We perform quantitative evaluations through simulation and qualitative studies with a real robot.

A. Quantitative Evaluation through Simulation

We statistically evaluate the capabilities of **occlusion recovery** and **task completion** in INR-DOM.

Occlusion recovery: We assess the occlusion-tolerant reconstruction capability of INR-DOM through IsaacSim, a physics simulator from NVIDIA [26]. We build a random manipulation dataset for nine types of circular rubber bands. Each band has a pair of *inside diameter* (ID) d_{ID} and *cross-sectional diameter* (CSD) d_{CSD}, where $\{(d_{\text{ID}}, d_{\text{CSD}}) \mid d_{\text{ID}} \in \{6\,\text{cm}, 10\,\text{cm}, 14\,\text{cm}\}, d_{\text{CSD}} \in \{1\,\text{cm}, 2\,\text{cm}, 3\,\text{cm}\}\}$. As shown in Fig. 3, by randomly twisting and stretching them, we

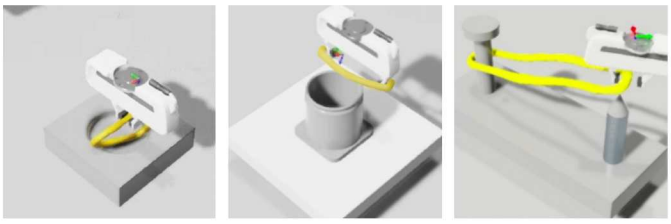


Fig. 4: Examples of three deformable-object manipulation environments: *sealing*, *installation*, and *disentanglement*.

collect a total of $90,000$ pairs of partial and complete point clouds, $(\mathbf{p}, \bar{\mathbf{p}})$, $10,000$ pairs each. Note that we restrict the randomization to a maximum of one twist in each direction, and a maximum stretch of twice the original length.

Given the nine classes of dataset, we perform a leave-one-out cross-validation that measures how representation models reconstruct the full geometry of rubber bands from a partial view of point clouds. During the pre-training stage of INR-DOM, we randomly sample $1,024$ points from each of the areas, on, near and off surfaces of the target object to compute $\mathcal{L}_{\text{SDF}}$ and $\mathcal{L}_{\text{skel}}$. To define the medial-axis points, we sample and track the 128 nodes spaced equally along the band and closest to the center of the cross section.

For comparison, we employ three baseline methods:

- PCN [45]: Point completion network; a shape-completion network with folding-based decoding that generates a dense and complete point cloud given a partial point cloud as input.
- PointTr [44]: A transformer-based encoder-decoder for point-cloud completion.
- Point2Vec [1]: A student-teacher framework of latent embedding and completion network.

After reconstructing the full geometry, we compute the earth mover's distance (EMD) and Chamfer distance (CD) between the reconstructed and complete point clouds. To handle SDF, we convert it into a mesh-based point cloud using the Marching Cubes algorithm [22].

Manipulation tasks: We evaluate the effectiveness of fine-tuned representations in DOM by building three tasks: *sealing*, *installation*, and *disentanglement* (see Fig. 4). For each task, we use a simulated parallel-jaw gripper with an RGB-D camera to manipulate a rubber band or an O-ring to achieve a desired state. We describe the task setup below.

- *Sealing*: A gripper seals a groove by inserting a rubber band after grasping a randomly placed band. To enhance sampling efficiency, we reduce the action space to $(\mathbf{a}_t^{\text{lin}}, \mathbf{a}_t^{\text{gripper}}) \in \mathbb{R}^4$ setting $\mathbf{a}_t^{\text{ang}}$ to zero.
- *Installation*: A gripper installs an O-ring onto the circular groove of a cylinder after grasping a randomly placed O-ring. The action space is $(\mathbf{a}_t^{\text{lin}.xz}, \mathbf{a}_t^{\text{ang}.y}, \mathbf{a}_t^{\text{gripper}}) \in \mathbb{R}^4$, where $\mathbf{a}_t^{\text{lin}.xz}$ and $\mathbf{a}_t^{\text{ang}.y}$ represent a two-dimensional velocity perpendicular to the direction of the parallel jaw's open/close axis and an angular velocity along the same axis, respectively.
- *Disentanglement*: A gripper untangles an entangled rubber band between two upright poles by grasping and repositioning it onto the poles. We initialize a band between two poles with up to two random entanglements in either of the two directions. The action space is $(\mathbf{a}_t^{\text{lin}}, \mathbf{a}_t^{\text{ang}.z}, \mathbf{a}_t^{\text{gripper}}) \in \mathbb{R}^5$, where $\mathbf{a}_t^{\text{ang}.z}$ represents an angular velocity constrained along the vertical axis.

For all tasks, we reduce the space of the partial point clouds to $\mathcal{P} \subset \mathbb{R}^{3\times 1024}$, through the iterative farthest point sampling of the input point cloud.

For training, we fine-tune INR-DOM with all types of rubber bands from the occlusion recovery study and update its policy network using an off-policy RL method, that is, SAC [10]. The policy network is a multilayer perceptron with three hidden layers consisting of 256, 128, and 64 nodes, respectively, each using ReLU activation. For testing, we assess the task completion performance of INR-DOM in 100 environments, including 30 randomly generated and configured rubber bands. The ID and CSD of the bands range from $[6\,\text{cm}, 14\,\text{cm}]$ and $[1\,\text{cm}, 3\,\text{cm}]$, respectively.

For comparison, we employ six baseline methods:

- **PCN+SAC**, **PoinTr+SAC**, and **Point2Vec+SAC**: RL-based DOM methods trained using the latent representations introduced in the occlusion-recovery study.
- **CFM** [43]: A model predictive control framework that leverages latent representations and dynamics models learned with contrastive estimation for DOM.
- **DeformerNet** [38]: A visual servoing method that minimizes the difference between the current and target point clouds of the deformable object.
- **ACID** [35]: A model-based planning method that calculates action costs based on the difference between the learned implicit representations.

We evaluate our method with these baselines over 100 trials, each with a maximum of 200 time steps.

B. Qualitative Evaluation

We demonstrate the fine-tuning and testing of our method in three real-world DOM environments using a Franka Emika Panda arm, as shown in Fig. 1. These environments are similar to those used in the quantitative evaluation. Each setup includes a rubber band with specific dimensions: 1) $d_{\text{ID}} = 10\,\text{cm}$ and $d_{\text{CSD}} = 0.6\,\text{cm}$ for *sealing*, 2) $d_{\text{ID}} = 6\,\text{cm}$ and $d_{\text{CSD}} = 1\,\text{cm}$ for *installation*, and 3) $d_{\text{ID}} = 20\,\text{cm}$ and $d_{\text{CSD}} = 1\,\text{cm}$ for *disentanglement*. To observe the band, we mount an Orbbec Femto Bolt RGB-D camera on the table capturing the point cloud $\mathbf{p}_t$ of the band by segmenting and tracking it with a pre-trained segmentation anything model 2 (SAM2) [31]. For more effective localization and grasping of the band, we employ two RealSense D405 RGB cameras mounted on the wrist, extending the latent state $\mathbf{s}'_t$ with a concatenated vector of RGB features processed by ResNet10 [8]. In this work, we acquire the segmented points $\mathbf{p}_t$ after outlier removal at $10\,\text{Hz}$ along with two RGB observations captured at $15\,\text{Hz}$.

Unlike simulation, we design an image-based reward classifier that assigns a high reward when the current RGB image closely matches the final scene from expert demonstrations;

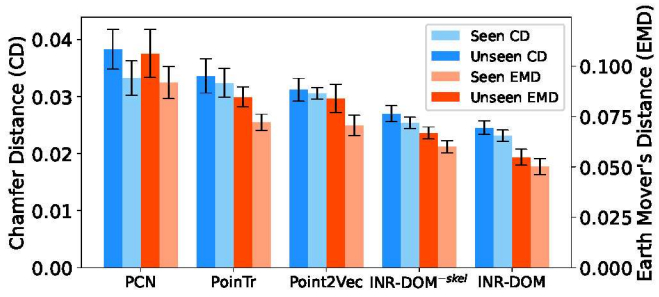


Fig. 5: Comparison of point-cloud reconstruction performance for both seen and unseen types of partially observable rubber bands. The blue and red bars represent the reconstruction errors measured by chamfer distance (CD) and earth mover's distance (EMD), respectively. Note that INR-DOM^{-skel} refers to a variant of INR-DOM that was not pre-trained using the $\mathcal{L}_{\text{skel}}$ loss.

TABLE I: Comparison of task success rates [%] across three simulated environments, based on the evaluation of 100 trials per environment. The superscripts, *-p* and *-cl*, indicate versions of the target method that were not trained with pre-training and contrastive learning, respectively.

Model	*Sealing*	*Installation*	*Disentanglement*
PCN[45] + SAC[10]	13	38	6
PoinTr[44] + SAC[10]	23	38	14
Point2Vec[1] + SAC[10]	14	40	22
Point2Vec^{-p}[1] + SAC[10]	2	1	1
CFM[43] + SAC[10]	41	47	23
DeformerNet[38]	41	53	19
ACID[35]	44	58	26
INR-DOM^{-p}	20	29	16
INR-DOM^{-cl}	58	61	54
INR-DOM	**85**	**89**	**75**

otherwise, it assigns a zero reward. To train this classifier, we collect 20 demonstrations by teleoperating the real robot using a 3-dimensional space mouse. We use these demonstrations to populate the replay buffer during the fine-tuning stage. Further, we adopt a sample-efficient robotic reinforcement learning (SERL) framework [23], which incorporates RL with prior data (RLPD) [2]. For RLPD, we construct training batches with half of the samples drawn from the demonstrations and the other half from the online replay buffer. We use 20 successful episodes as demonstrations for the prior data. Note that we set the update-to-data (UTD) ratio to 4. This allows leveraging human demonstrations to accelerate the learning process and minimize the sim-to-real gap in representation.

The robot operates with a Cartesian impedance controller running at 1 kHz while the policy network runs at 10 Hz. We perform all training on a Threadripper Pro 5975WX and an NVIDIA RTX A6000 GPU.

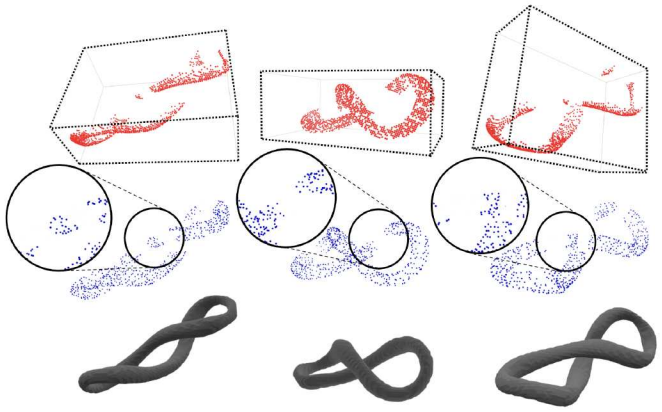


Fig. 6: Comparison of occlusion-robust reconstruction performance between INR-DOM and Point2Vec. (**Top**) Point-cloud inputs of partially observable elastic bands. (**Middle**) Point clouds reconstructed by Point2Vec. The larger circles highlight magnified views of regions where reconstruction failed, indicated by smaller circles. (**Bottom**) SDF-based Meshes from INR-DOM.

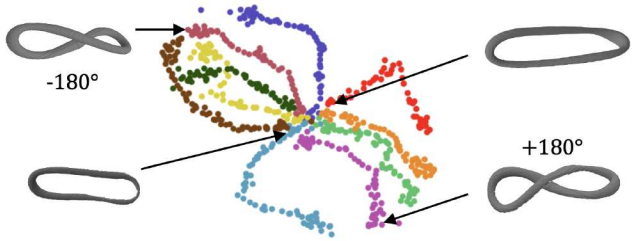
V. Evaluation Results

A. Evaluation through Simulation Studies

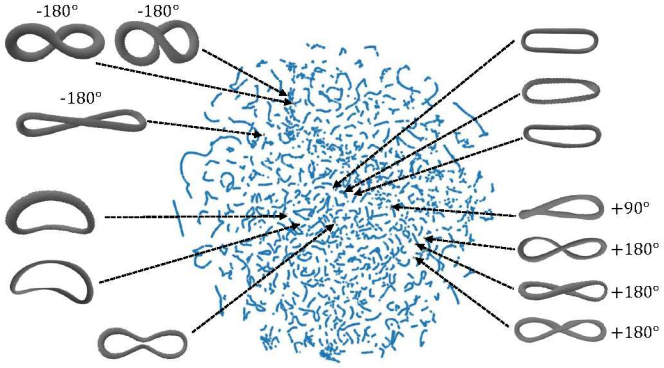
We statistically evaluate the occlusion-robust reconstruction capabilities of pre-trained INR-DOM on both seen and unseen partially observable bands. INR-DOM shows the lowest reconstruction errors using CD, even with previously unseen objects, as illustrated in Fig. 5. Remarkably, INR-DOM outperforms baselines on unseen objects more effectively than baselines do on seen objects. Further, INR-DOM achieves superior performance without structural prior $\mathcal{L}_{\text{skel}}$, exceeding all baselines. In addition, we assess performance using EMD, which better accounts for global distribution differences crucial for compressed or stretched DOs. The results with EMD align with those of CD, confirming that INR-DOM provides robust representations for tasks involving partially observable DOs.

To show the reconstruction quality, we present examples from INR-DOM and Point2Vec in Fig. 6. INR-DOM accurately reconstructs complete geometry from any observation angle, whereas Point2Vec struggles with occluded, particularly intertwined, parts. This highlights the effectiveness of our pre-training method for DOs.

Next, we assess the consistent representation capabilities of INR-DOM in the *disentanglement* task. Fig. 7 presents t-SNE visualizations of latent state vectors from the fine-tuned INR-DOM encoder Φ_ϕ, showing consistent mapping of similar configurations to adjacent regions and clear distinction of intertwined state directions. For example, in Fig. 7 (a), all disentanglement trajectories converge to the center region as the task completes, with $+180°$ and $-180°$ twists placed on opposite sides. This structured latent space allows the RL agent to recognize subtle yet crucial variations, such as twist counts, thereby improving success rates.



(a) Distribution of 10 trajectory embeddings from the *disentanglement* task



(b) Distribution of $2 \cdot 10^4$ embeddings from random manipulation dataset used in the pre-training stage

Fig. 7: t-SNE visualization of latent state vectors $\mathbf{z}$ from the fine-tuned encoder Φ_ϕ. We also visualize rubber band meshes corresponding to selected points to show how the latent state effectively captures the entanglements of the band. The signed numbers adjacent to the meshes represent twist angles and directions.

We evaluate the task-relevant representation capabilities of INR-DOM through three simulated DOM tasks. Table I shows INR-DOM outperforms all other methods, achieving the highest task-success rates, an average rate 40.3% higher than ACID, the next best approach. INR-DOM exhibits superior performance in the challenging *disentanglement* task, characterized by stretched and randomly intertwined states. Notably, INR-DOM^{-cl}, fine-tuned only via RL, ranks second, demonstrating the efficacy of contrastive learning in distinguishing state spaces for DOM. In contrast, INR-DOM^{-p}, which lacked pre-training, does not show performance gains, highlighting that the reconstruction-based pre-training is crucial for effective initialization of state-representation learning. In contrast, other recent DOM frameworks underperform, primarily due to their inability to capture local structures and their non-rigid transformations in DOs.

Lastly, we analyze the lower performance of point-cloud completion approaches, such as PCN, PointTr, and Point2Vec. Their embeddings capture overall structure but struggle with continuous and fine deformations due to discrete sampling. In contrast, our SDF loss with gradient regularization and alignment not only allows adaptive resolution but also ensures high accuracy in surface completion. Although the latest approach, Point2Vec, successfully encodes local patches with contextual information, our target objects lack explicit contextual patches and instead require accurate positional and structural details. Due to its patch normalization, Point2Vec loses such information, resulting in collapsed or overlapped geometries.

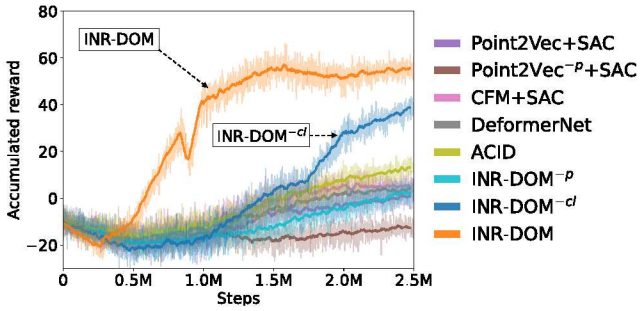


Fig. 8: Comparison of accumulated reward curves during training between INR-DOM and baseline models.

B. Learning Curve Analysis

Fig. 8 shows the comparison of the accumulated reward curves during training for the *disentanglement* task between INR-DOM with variants and baseline methods. INR-DOM shows superior learning efficiency, achieving the highest accumulated rewards compared to all baselines. INR-DOM^{-cl} particularly ranks second in terms of the accumulated rewards at the end, though it lags significantly behind INR-DOM. This underscores that contrastive learning enhances effective exploration in RL by fostering exploratory representations. In contrast, INR-DOM^{-p} achieves only a 15% success rate after 2.5 million training steps, indicating that exploratory representations are challenging to enhance over steps without a well-structured state space.

C. Evaluations in Real-World Settings

Finally, we demonstrate the real-world applicability of INR-DOM across three DOM tasks. Fig. 9 (Top) displays the Panda arm with a 3D-printed tip sealing a groove by dragging and inserting a randomly placed red rubber band, effectively targeting the protruding parts for pressing actions. Fig. 9 (Middle) shows the robot installing an O-ring onto a cylinder's groove, opening the ring, and leveraging one-sided contact for insertion. Fig. 9 (Bottom) illustrates the Panda arm with a long parallel-jaw gripper, tip-grasping and disentangling a randomly intertwined rubber band between two poles. INR-DOM successfully distinguishes the twisted direction as well as effectively manipulates the arm, achieving a 90% task-success rate in each task over ten trials with various initial conditions. This robust performance, coupled with successful learning from the fine-tuning of 20 episodes, confirms the applicability of INR-DOM to various real-world DOM tasks.

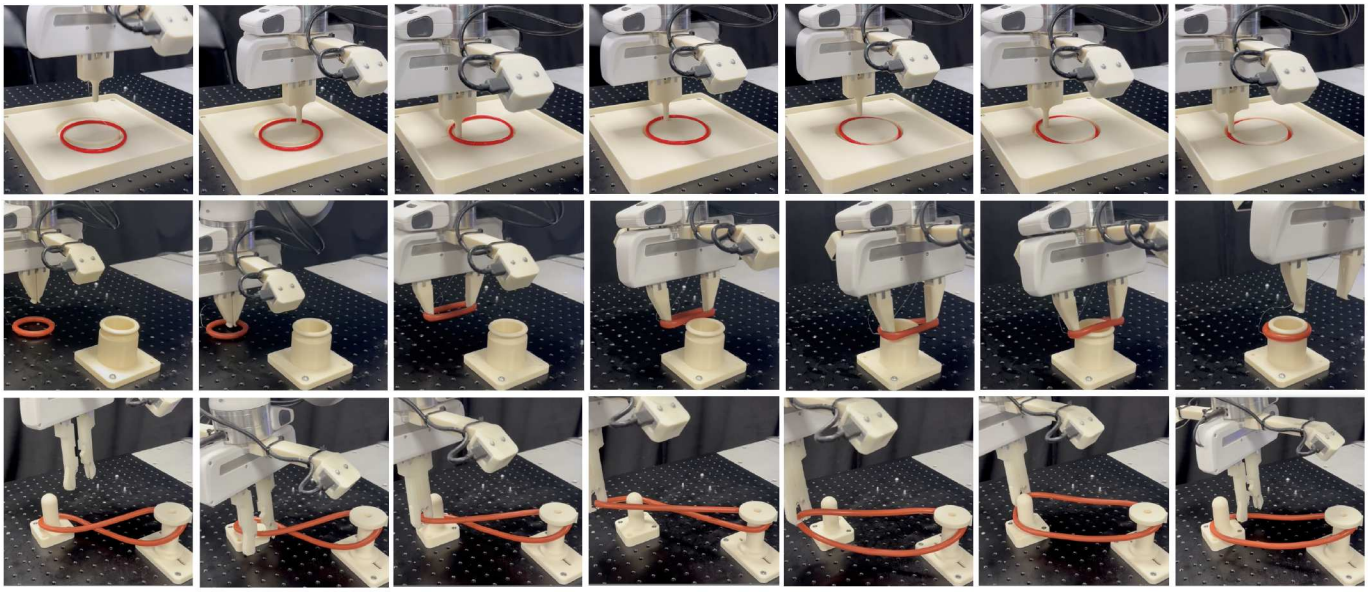


Fig. 9: Demonstrations of INR-DOM's manipulation capability in the three tasks. (**Top**) In *sealing* task, INR-DOM learns to drag and insert a rubber band into a groove. (**Middle**) In *installation* task, INR-DOM learns to install an O-ring onto a cylinder's groove. (**Bottom**) *disentanglement* task, INR-DOM learns to disentangle the band between two poles by precisely identifying the twisted status.

In addition, we compare INR-DOM with SERL, a representative image-based manipulation method on the real-world *disentanglement* task. To increase task difficulty, we extend the pole distance to stretch the bands, resulting in tighter intersections. SERL uses an RGB camera, whereas our method relies on a depth camera. As shown in Table II, INR-DOM successfully distinguishes between $\pm 180^\circ$ twist directions, while SERL fails due to visual ambiguities at the intersecting bands (see Fig. 10).

TABLE II: Comparison of task success rates [%] between an SERL image-based method and our method in the *disentanglement* task over 20 trials.

Model	Rate [%]	**Model**	Rate [%]
SERL	55%	INR-DOM	80%

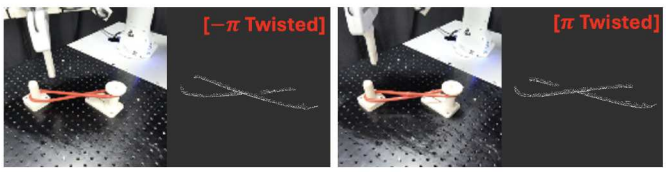


Fig. 10: Visual ambiguity in RGB vs. point cloud observations for $-\pi$ and π twist configurations.

VI. Conclusion

We introduced an implicit neural representation learning method for elastic deformable object manipulation (INR-DOM). INR-DOM models consistent and occlusion-robust state representations associated with partially observable elastic objects by learning to reconstruct a complete and implicit surface represented as a signed distance function. To obtain task-relevant state representations and manipulation policy, INR-DOM fine-tunes its internal encoder to have exploratory representations through RL-based contrastive learning. The statistical evaluation shows that our method outperforms state-of-the-art baselines in terms of reconstruction error and task success rate on novel objects and manipulation settings. We successfully demonstrate the proposed INR-DOM transfer into real-world DOM tasks.

VII. Limitations

- **Deformable linear object**: Our studies focus on evaluating deformable linear objects, such as rubber bands, though the usage is not limited to specific shapes. Although skeleton loss $\mathcal{L}_{skel}$ may reduce its applicability, 3D skeletonization is generalizable to any volumetric object. Extending this method to a variety of deformable object types remains a goal for future research.
- **Segmented observation**: The performance of INR-DOM depends on the accuracy of the point-cloud segmentation model. This dependency makes INR-DOM vulnerable to errors in the segmentation process, which affects the robustness and reliability of the task outcomes. Future improvements should involve processing the entire depth image and automatically focusing attention on the target deformable object.
- **Single task**: The fine-tuning process relies on a specific manipulation task which may limit the model's applicability to other scenarios. Moreover, INR-DOM's design focuses exclusively on a single deformable object manipulation and

does not account for scenarios with multiple interacting objects, common in complex environments. However, the architecture does not inherently restrict itself to single-task learning. The adoption of multi-task reinforcement learning [7] or few-shot policy generalization [42] extends beyond the scope of this work. Expanding to multitask RL approaches is our next objective.

- **Dynamic interaction**: This work does not extensively investigate the dynamics of interaction between the robot and deformable, particularly stretchable objects. The manipulation tasks focus primarily on the final outcomes of actions, rather than on modeling the interactive dynamics essential for tasks that require continuous adjustments based on real-time feedback. This limitation may restrict the applicability of INR-DOM to more interactive and dynamic tasks, where understanding and adapting to ongoing changes during manipulation are crucial.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2022-II220311, RS-2024-00509279, and RS-2024-00336738) and Samsung Electronics Co., Ltd, South Korea (No. IO220811-01961-01).

References

[1] Karim Abou Zeid, Jonas Schult, Alexander Hermans, and Bastian Leibe. Point2vec for self-supervised representation learning on point clouds. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2023.

[2] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1577–1594. PMLR, 2023.

[3] David Blanco-Mulero, Oriol Barbany, Gokhan Alcan, Adria Colome, Carme Torras, and Ville Kyrki. Benchmarking the sim-to-real gap in cloth manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 9(8):2981–2988, 2024.

[4] Lawrence Yunliang Chen, Baiyu Shi, Roy Lin, Daniel Seita, Ayah Ahmad, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. Bagging by Learning to Singulate Layers Using Interactive Perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[5] Siwei Chen, Yiqing Xu, Cunjun Yu, Linfeng Li, and David Hsu. Differentiable particles for general-purpose deformable object manipulation. *arXiv preprint arXiv:2405.01044*, 2024.

[6] BP Duisterhof, Z Mandi, Y Yao, JW Liu, J Seidenschwarz, MZ Shou, D Ramanan, S Song, S Birchfield, B Wen, et al. Deformgs: Scene flow in highly deformable scenes for deformable object manipulation. In *The 16th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2024.

[7] Jinyuan Feng, Min Chen, Zhiqiang Pu, Tenghai Qiu, Jianqiang Yi, and Jie Zhang. Efficient multi-task reinforcement learning via task-specific action correction. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–15, 2025.

[8] Jiaming Gong, Wei Liu, Mengjie Pei, Chengchao Wu, and Liufei Guo. Resnet10: A lightweight residual network for remote sensing image classification. In *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 975–978. IEEE, 2022.

[9] Feida Gu, Yanmin Zhou, Zhipeng Wang, Shuo Jiang, and Bin He. A survey on robotic manipulation of deformable objects: Recent advances, open challenges and new frontiers. *arXiv preprint arXiv:2312.10419*, 2023.

[10] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1861–1870. PMLR, 2018.

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

[13] Zixuan Huang, Xingyu Lin, and David Held. Mesh-based dynamics with occlusion reasoning for cloth manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[14] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5639–5650. PMLR, 2020.

[15] Seunghan Lee, Taeyoung Park, and Kibok Lee. Soft contrastive learning for time series. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[16] Chenchang Li, Zihao Ai, Tong Wu, Xiaosa Li, Wenbo Ding, and Huazhe Xu. Deformnet: Latent space modeling and dynamics prediction for deformable object manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 14770–14776. IEEE, 2024.

[17] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[18] Huan Lin, Feng Guo, Feifei Wang, and Yan-Bin Jia. Picking up a soft 3d object by "feeling" the grip. *International Journal of Robotics Research*, 34(11):1361–1384, 2015.

[19] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on robot learning*, pages 256–266. PMLR, 2022.

[20] Martina Lippi, Petra Poklukar, Michael C Welle, Anastasiia Varava, Hang Yin, Alessandro Marino, and Danica Kragic. Latent space roadmap for visual action planning of deformable and rigid object manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5619–5626. IEEE, 2020.

[21] Alberta Longhini, Marco Moletta, Alfredo Reichlin, Michael C Welle, David Held, Zackory Erickson, and Danica Kragic. Edo-net: Learning elastic properties of deformable objects from graph dynamics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3875–3881. IEEE, 2023.

[22] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *In Computer Graphics*, 21(4):163–169, 1987.

[23] Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. SERL: A software suite for sample-efficient robotic reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969. IEEE, 2024.

[24] Peter Mitrano and Dmitry Berenson. The grasp loop signature: A topological representation for manipulation planning with ropes and cables. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 10888–10894. IEEE, 2024.

[25] Masaki Murase, Kimitoshi Yamazaki, and Takamitsu Matsubara. Kullback leibler control approach to rubber band manipulation. In *Proceedings of the International Symposium on System Integration (SII)*, pages 680–685. IEEE, 2017.

[26] NVIDIA. Isaac Sim - Robotics Simulation and Synthetic Data Generation, 2024. URL https://developer.nvidia.com/isaac/sim. (accessed on Jan. 30, 2025).

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.

[29] Antoine Petit, Vincenzo Lippiello, Giuseppe Andrea Fontanelli, and Bruno Siciliano. Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot. *Robotics and Autonomous Systems*, 88:187–201, 2017.

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[31] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[32] Yu Ren, Ronghan Chen, and Yang Cong. Autonomous manipulation learning for similar deformable objects via only one demonstration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17069–17078, 2023.

[33] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. Chapter 1 - skeletonization and its applications – a review. In Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja, editors, *Skeletonization*, pages 3–42. Academic Press, 2017.

[34] Gautam Salhotra, I-Chun Arthur Liu, Marcus Dominguez-Kuhne, and Gaurav S Sukhatme. Learning deformable object manipulation from expert demonstrations. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):8775–8782, 2022.

[35] Bokui Shen, Zhenyu Jiang, Christopher Choy, Leonidas J. Guibas, Silvio Savarese, Anima Anandkumar, and Yuke Zhu. Acid: Action-conditional implicit visual dynamics for deformable object manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[36] Kaushik Shivakumar, Vainavi Viswanath, Anrui Gu, Yahav Avigal, Justin Kerr, Jeffrey Ichnowski, Richard Cheng, Thomas Kollar, and Ken Goldberg. Sgtm 2.0: Autonomously untangling long cables using interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5837–5843. IEEE, 2023.

[37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7462–7473, 2020.

[38] Bao Thach, Brian Y Cho, Alan Kuntz, and Tucker Hermans. Learning visual shape control of novel 3d deformable objects from partial-view point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 8274–8281. IEEE, 2022.

[39] Weifu Wang and Devin Balkcom. Knot grasping, folding, and re-grasping. *International Journal of Robotics Research*, 37(2-3):378–399, 2018.

[40] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[41] Youngsun Wi, Pete Florence, Andy Zeng, and Nima Fazeli. Virdo: Visio-tactile implicit representations of deformable objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3583–3590. IEEE, 2022.

[42] Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 24631–24645. PMLR, 2022.

[43] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. In *Conference on robot learning*, pages 564–574. PMLR, 2021.

[44] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[45] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018.

[46] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[47] Xinyang Zheng, Yang Liu, Pengshuai Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Computer Graphics Forum*, volume 41, pages 52–63. Wiley Online Library, 2022.

[48] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, et al. Challenges and outlook in robotic manipulation of deformable objects. *Robotics & Automation Magazine*, 29(3):67–77, 2022.

[49] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *International Journal of Robotics Research*, 32(9-10): 1164–1193, 2013.