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

DeformNet: Latent Space Modeling and Dynamics Prediction for Deformable Object Manipulation

Chenchang Li†, Zihao Ai†, Tong Wu, Xiaosa Li, Wenbo Ding*, and Huazhe Xu*

***Abstract*— Manipulating deformable objects is a ubiquitous task in household environments, demanding adequate representation and accurate dynamics prediction due to the objects' infinite degrees of freedom. This work proposes DeformNet, which utilizes latent space modeling with a learned 3D representation model to tackle these challenges effectively. The proposed representation model combines a PointNet encoder and a conditional neural radiance field (NeRF), facilitating a thorough acquisition of object deformations and variations in lighting conditions. To model the complex dynamics, we employ a recurrent state-space model (RSSM) that accurately predicts the transformation of the latent representation over time. Extensive simulation experiments with diverse objectives demonstrate the generalization capabilities of DeformNet for various deformable object manipulation tasks, even in the presence of previously unseen goals. Finally, we deploy DeformNet on an actual UR5 robotic arm to demonstrate its capability in real-world scenarios.**

I. Introduction

Manipulating deformable objects is crucial in robotics given its ubiquity in everyday activities, spanning from molding dough to tying knots in ropes. Previous research has proposed specific representations and techniques for different deformable objects, such as ropes [1]–[3], cables [4]–[6], clothes [7]–[9], fluid [10], [11], plasticine [12]–[14], clay [15], and gauze [16]. We instead ask the question: Can we model and manipulate such objects using only RGB-D observations without explicit inductive bias about physical properties, such as particle positions and mass distributions?

We argue that discovering a general state representation is at the heart of deformable object manipulation. Highly deformable objects, such as dough, pose challenges in capturing detailed deformations, as their shapes can exhibit significant variations. Hence, traditional convolutional neural networks (CNNs), which primarily capture 2D features, are unable to model these highly deformable objects. In recent years, neural radiance fields (NeRF) [17] have gained attention for their ability to precisely reconstruct 3D scenes even when the scenes are dynamic [11], [18], [19], making it suitable for various control tasks.

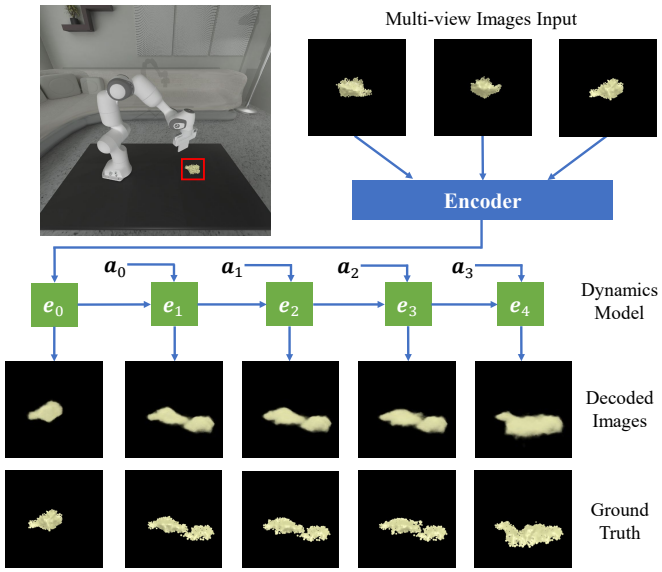


Fig. 1: Overview of DeformNet framework. For a given time step, an encoder initially transforms images captured from diverse viewpoints into a coherent embedding. Subsequently, a dynamics model is employed to forecast the resultant embedding following a set of sampled actions. Finally, a NeRF decoder is utilized to generate images conditioning on the forecasted embeddings.

With a reasonable representation in hand, another hurdle lies in the prediction of the dynamics, particularly when deformation is large. At first glance, some readers may wonder whether model-free reinforcement learning (RL) can circumvent the prediction by optimizing a manipulation policy directly. However, RL approaches usually struggle to find suitable reward functions and optimize actions in the face of high DoFs of deformable objects [20]. Recently, graph neural networks (GNNs) have been proposed to model the long-term dynamics of soft materials [13], [18], [21]. While GNNs have shown great promise in long-term prediction, their reliance on neighboring relations makes it difficult to effectively encode information such as rotations. Additionally, GNNs also require a predefined underlying structure, and the task of adapting these networks to dynamic graphs continues to present a research challenge. Some prior works have attempted latent space modeling for soft body manipulation [3], [11] with simple multilayer perceptron (MLP) as

† Contribute equally to this work, listed in random order. * Corresponding author.

This work was supported by Shenzhen Ubiquitous Data Enabling Key Lab under Grant No. ZDSYS20220527171406015, by Guangdong Innovative and Entrepreneurial Research Team Program (2021ZT09L197), by Shenzhen Science and Technology Program (JCYJ20220530143013030), by Tsinghua Shenzhen International Graduate School-Shenzhen Pengrui Young Faculty Program of Shenzhen Pengrui Foundation (No. SZPR2023005).

C. Li, Z. Ai, T. Wu, X. Li and W. Ding are with Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. E-mail: {li-cc21, azh21, wu-t23, lixs21}@mails.tsinghua.edu.cn and ding.wenbo@sz.tsinghua.edu.cn

W. Ding is also with RISC-V International Open Source Laboratory, Shenzhen, China, 518055.

H. Xu is with Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, Beijing, China. E-mail: huazhe_xu@mail.tsinghua.edu.cn

the dynamics model; however, these works have limited performance for objects with high DoFs. We train a recurrent state-space model (RSSM) [22] as a world model to learn the complex dynamics of deformable objects, which can simultaneously capture global features as well as internal relations.

In this work, we propose a simple and unified framework named DeformNet which combines a learned world model with a conditioned NeRF structure to address the aforementioned challenges. More specifically, we first use a PointNet encoder to extract latent vectors from the point cloud converted from RGB-D images. Then, the encoded latent vectors are split into latent deformation vectors and latent appearance vectors, where the former are responsible for density predictions, and the latter are for prediction of RGB colors. In this way, our NeRF structure is capable of representing complex 3D shapes, which is essential for the world model to accurately predict the dynamics. Finally, we add the gradient descent step to sample efficient cross-entropy method [23] for trajectory planning, using reward prediction from the world model as the cost function. DeformNet achieves outstanding performance on complex manipulation tasks with various targets, including pinching plasticine to different shapes, writing on the clay, and towel manipulation.

The main contributions of this paper are tri-fold:

- We propose DeformNet, which augments the neural radiance field with a latent deformation vector and a latent appearance vector, enabling the accurate representation of largely deformed objects.
- By incorporating gradient-based planning with a learned world model, DeformNet exhibits substantial performance on complex manipulation tasks.
- DeformNet also showcases stability and generalization ability, demonstrating its potential for real-world applications in deformable object manipulation across various tasks and shapes.

II. Related Works

Representations of deformable objects. In recent years, data-driven methods have drawn lots of attention to represent deformable shapes in low-dimensional latent space. Latent vectors [3], [24]–[26], keypoint embeddings [27], [28] and mesh embeddings [29] have been used to represent deformable objects such as ropes [3], [24], [28], clothes [26], [27], [29] and plasticine [25]. Recently, neural radiance fields have been studied in non-rigid object representation due to its high-fidelity 3D reconstruction capability. Li et al. [11] leverage contrastive learning and neural radiance fields to learn latent representations of pouring scene. Driess et al. [18] propose to use a 3D feature encoder to obtain the latent vector of a rope and reconstruct the rope with learned conditional neural radiance fields. Park et al. [30] use a deformation field to model the transformation of complex facial expressions. However, existing works focus on representing only one type of deformable objects. By contrast, we explore the capability of neural radiance fields in representing different types of deformable objects such as ropes, plasticine, grains, and clay, with PointNet [31] as encoder.

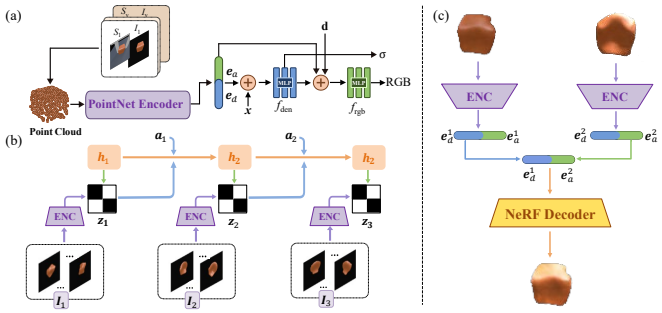


Fig. 2: Overview of the model architecture. (a) The PointNet encoder generates latent features. These latent features are split and utilized to enhance the NeRF. (b) We then employ RSSM to capture the dynamics within the trained latent space. (c) We examine the effect of latent deformation vectors and latent appearance vectors in a fixed dataset with various lighting conditions and object shapes.

Deformable object manipulation. The manipulation of deformable objects has been extensively studied, with various approaches proposed in the literature. Some works have focused on adaptive methods [32]–[34], while others have explored simulation-based approaches for actual manipulation [35]. However, the challenge of simulation-based method lies in accurately estimating the parameters, which creates a barrier in bridging the sim2real gap. Imitation learning is also leveraged to shape sand [33] or dough [36], but obtaining demonstrations for training can be expensive. In recent developments, model-based methods [37]–[39] have gained increasing attention. Matl and Bajcsy [14] utilize bounding box to represent dough and shape it to different lengths. Shi et al. [21] propose GNN-based dynamics and deform plasticine to more complex character shapes. Unlike methods utilizing manually designed features or numerous particles to represent deformable objects, DeformNet employs latent vectors. To accurately predict the deformation of complex shapes in latent space, we utilize RSSM [22] as the underlying dynamics model.

III. Method

A. Problem Formulation

Our research focuses on the utilization of a variety of actuators to manipulate or reposition a deformable object to a predefined position or configuration. Given the initial observation s_0 of the object and the target observation s_g, our robotic system needs to execute a sequence of actions $\mathbf{a}_{0,\dots,H-1} \in \mathcal{A}$ to bring the observation s_H close to the desired goal observation s_g.

To accomplish this, we employ the recurrent state space model (RSSM) [22] denoted as Φ to learn the transition model $\Phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ of the deformable object. At each time step t, the RSSM takes the current environmental observations s_t and a sequence of actions $\mathbf{a}_{t,\dots,t+H-1}$, and predicts future observations $s_{t+1,\dots,t+H}$, with H denoting the length of the prediction horizon. This predictive model

allows us to formulate our manipulation tasks as a model predictive control (MPC) problem. Considering the range of our goals, we adopt different cost functions to gauge the distance between the object's current observation and the goal observation. The action sequence to be carried out is chosen based on the cost function:

$$(\mathbf{a}_0,\ldots,\mathbf{a}_{H-1}) = \underset{\mathbf{a}_{0\ldots H-1}\in\mathcal{A}}{\arg\min}\, \mathcal{J}(\Phi(s_0,(\mathbf{a}_0,\ldots,\mathbf{a}_{H-1})),s_g). \quad (1)$$

B. 3D Representation Learning for Deformable Objects

DeformNet utilizes the encoder-decoder framework for 3D representation learning, as shown in Fig. 2(a).

PointNet encoder. We use the PointNet [31] architecture as the encoder to capture the overall features of the object, as illustrated in Fig. 2(a). RGB-D image observation s_t from different viewpoints at time step t are converted into a single point cloud $\mathbf{P} = (\mathbf{c}_p^{n\times 3}, \mathbf{x}_p^{n\times 3})$ using camera parameters, where $\mathbf{c}_p^{n\times 3}$ is the RGB color and $\mathbf{x}_p^{n\times 3}$ is the 3D location. Subsequently, the PointNet encoder f_ω processes the input point cloud to generate an embedding $\mathbf{e} = f_\omega(\mathbf{c}_p^{n\times 3}, \mathbf{x}_p^{n\times 3})$. This embedding is then divided into two parts: a latent deformation vector $\mathbf{e}_d$ and a latent appearance vector $\mathbf{e}_a$. As detailed below, $\mathbf{e}_d$ contains the shape information and $\mathbf{e}_a$ denotes the lighting condition.

Neural radiance field. With 3D coordinate $\mathbf{x} \subseteq \mathbb{R}^3$ and viewing direction unit vector $\mathbf{d} \subseteq \mathbb{R}^3$ as input, NeRF learns a function that maps spatial location orientation to its RGB color value $\mathbf{c}(\mathbf{x}, \mathbf{d})$ and volume density $\sigma(\mathbf{x})$ [17]. The color of an image pixel from a particular viewpoint can be rendered based on volumetric rendering function:

$$\hat{\boldsymbol{C}}(\mathbf{r}) = \int_{\alpha_n}^{\alpha_f} T(\alpha)\sigma(\alpha)\boldsymbol{c}(\alpha)d\alpha,\ T(\alpha) = \exp\left(-\int_{\alpha_n}^{\alpha}\sigma(s)ds\right), \quad (2)$$

where α_n and α_f are near bound and far bound, $\mathbf{r} = \mathbf{o} + \alpha\mathbf{d}$ is the camera ray with origin $\mathbf{o} \in \mathbb{R}^3$ and the viewing direction $\mathbf{d}$, and $T(\alpha)$ denotes the accumulated transmittance along the ray from α_n to α_f. During the training phase, NeRF is optimized by ℓ_2 loss $\mathcal{L}_{\text{rec}}$ between the reconstructed color $\hat{\boldsymbol{C}}$ and ground truth color $\boldsymbol{C}$.

Conditional NeRF decoder. In the standard NeRF formulation, one model is trained to render one static scene. It has been proposed in the literature to use latent vectors as conditioned inputs to NeRF [30], [40]. However, previous works mostly used NeRF trained on conditioned inputs to render a scene where the variation is limited [30] or the objects are composed of rigid bodies [40].

Our task setting presents a key challenge as the deformable object may deform severely after applying a sequence of actions. Accurately manipulating highly deformable materials requires adequate 3D representation of the objects. When the object expands, it usually can be easily represented as this type of deformation causes changes in pixels. However, it is difficult to capture the changes in the dented surface during deformation, as the dented area may only manifest as a slight darkening of the pixels. Instead of using a single latent vector as input for NeRF, we use latent deformation vector $\mathbf{e}_d$ to represent the shape change and latent appearance vector $\mathbf{e}_a$ to capture the lighting condition. As in Fig. 2(a), we utilize $\sigma = f_{\text{den}}(\mathbf{x}, \mathbf{e}_d)$ to predict density and $\mathbf{c} = f_{\text{rgb}}(\mathbf{F}, \mathbf{e}_a, \mathbf{d})$ to predict color. Here, $\mathbf{F}$ represents the 256-dimensional feature generated by f_{den}. If we have two observations s_1 and s_2 with different shapes and lighting conditions, we can combine the deformation latent $\mathbf{e}_d^1$ encoded from s_1 with the appearance latent $\mathbf{e}_a^2$ from s_2, which is illustrated in Fig. 2(c). Subsequently, as we feed this resultant latent representation into the NeRF decoder, we observe a transfer of the lighting conditions from s_2 to s_1.

C. Learning Dynamics of Deformable Object via RSSM

RSSM [41] is a recurrent model that incorporates deterministic states $\mathbf{h}_t$ and categorical stochastic states $\mathbf{z}_t$ for long-term predictions. As demonstrated in Fig. 2(b), the components of our predictive model are as follows:

$$\begin{aligned}
&\text{Recurrent model:} && \mathbf{h}_t = f_\phi(\mathbf{h}_{t-1}, \mathbf{z}_{t-1}, \mathbf{a}_{t-1}) \\
&\text{Representation model:} && \mathbf{z}_t \sim q_\phi(\mathbf{z}_t \mid \mathbf{h}_t, \mathbf{e}_t) \\
&\text{Reward predictor:} && \hat{r}_t \sim p_\phi(\hat{r}_t \mid \mathbf{h}_t, \mathbf{z}_t, \mathbf{e}_g) \\
&\text{Transition predictor:} && \hat{\mathbf{z}}_t \sim p_\phi(\hat{\mathbf{z}}_t \mid \mathbf{h}_t) \\
&\text{Embedding predictor:} && \hat{\mathbf{e}}_t \sim p_\phi(\hat{\mathbf{e}}_t \mid \mathbf{h}_t, \mathbf{z}_t),
\end{aligned} \quad (3)$$

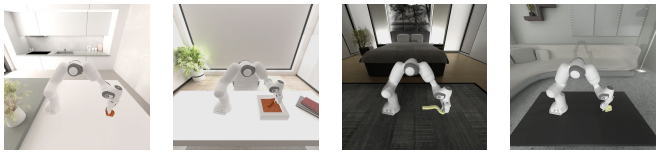
where $\mathbf{e}_g$ and $\mathbf{e}_t$ are the latent vectors encoded from goal images and current observation respectively, and ϕ describes the mutual parameters. At each step t, the RSSM computes the posterior stochastic state $\mathbf{z}_t$, incorporating information about the current embedding $\mathbf{e}_t$ obtained from the PointNet encoder. It also predicts a reconstructed embedding $\hat{\mathbf{e}}_t$, and a prior stochastic state $\hat{\mathbf{z}}_t$ that anticipates the posterior without access to the embedding. The reward predictor predicts the opposite value of the cost functions. The training process involves minimizing the embedding reconstruction loss, goal-conditioned reward log loss, and the KL loss of posterior and prior distributions. The loss function is defined as:

$$\begin{aligned}
\mathcal{L}_{\text{dym}} = \sum_{t=0}^{H-1} &\|\mathbf{e}_t - \hat{\mathbf{e}}_t\|_2^2 - \ln p_\phi(r_t \mid \mathbf{h}_t, \mathbf{z}_t, \mathbf{e}_g) \\
&+\text{KL}\left(q_\phi(\mathbf{z}_t \mid \mathbf{h}_t, \mathbf{e}_t) \,\|\, p_\phi(\hat{\mathbf{z}}_t \mid \mathbf{h}_t)\right).
\end{aligned} \quad (4)$$

During training, we first train the PointNet encoder and NeRF decoder by optimizing $\mathcal{L}_{\text{rec}}$, then we freeze the parameters of the encoder-decoder framework and train RSSM by minimizing $\mathcal{L}_{\text{dym}}$. To avoid leading to a poorly trained prior, we also employ KL-balancing [41]. This approach allows us to model the dynamics of the system and predict future states, providing a robust and efficient solution for target tasks.

D. Cost Functions

To quantify the spatial similarity between the distributions of MPM particles, we employ the following cost functions for our simulation environments including the Chamfer distance (CD), the earth mover's distance (EMD) [21], and the soft intersection over union (SIoU) [42]. We also adopt the discrete-to-continuous distance (D2CD). The D2CD is adapted from the average Chamfer distance [43], which gauges the distance between a discrete set of MPM particles



(a) Pinch (b) Write (c) Push (d) Pile

Fig. 3: Simulator visualizations. (a) The robot demonstrates its ability to manipulate plasticine through pinching actions, corresponding to the environment 'Pinch'. In the 'Poke' task, the gripper is replaced with a stick. (b) The robot showcases its writing skills by creating characters on a clay surface in environment 'Write'. (c) In the 'Push' task, the robot manipulates the towel by pushing it with a stick to achieve specific target shapes. (d) Within the 'Transport' and 'Pile' environments, the robot adeptly either relocates grains to desired locations or pushes them into predefined forms.

and a continuous target set, denoted as $\mathcal{O}$ and $\mathcal{S}_d$ in $\mathbb{R}^3$. It is defined as follows:

$$\mathcal{J}_{\mathrm{D2CD}}\left(\mathcal{O}, \mathcal{S}_d\right) = \frac{1}{|\mathcal{O}|} \sum_{\mathbf{x} \in \mathcal{O}} \min_{\mathbf{y} \in \mathcal{S}_d} \|\mathbf{x} - \mathbf{y}\|_2. \tag{5}$$

E. Closed-Loop Control

After obtaining the encoder and the dynamics model. We aim to optimize an action sequence $\mathbf{a}_{0,\ldots,H-1}$ in a designed action space. This sequence should make the final observation s_H as close as possible to the desired observation s_g, as shown in Equation (1). To achieve this, we adopt the sample efficient cross-entropy method (iCEM) [23], which utilizes the gradient generated by our predictive model to improve the selected actions. Intermediate observations are also used to correct model predictions as in standard MPC setting [21]. This planning process continues until either the predicted reward exceeds a predefined threshold or the action sequence reaches its maximum length.

During the closed-loop control, the cameras capture images of the workspace from various angles. These images are used as observations and to create a point cloud for the training of encoder and decoder. Our method takes as input the current observation and the final target, and calculates the best actions to achieve the goal. The robot interacts with the deformable object, adjusts its position to ensure a clear view for the cameras, and then plans its next move based on the visual feedback.

IV. Experiments

A. Tasks and Environments

We implement our simulation environment based on Maniskill2 [15], an MPM-based soft-body manipulation environment that facilitates real-time simulation, as visualized in Fig. 3. The performance of DeformNet is evaluated on six challenging deformable manipulation tasks namely 'Push', 'Pinch', 'Write', 'Pile', 'Transport', and 'Poke'. The 'Push' task involves pushing a soft towel on a 2D plane using a robot equipped with a stick. In the 'Pinch' task, the environment features a robot equipped with a gripper that manipulates plasticine into several predefined target shapes. For the 'Write' task, the robot employs a stick to write

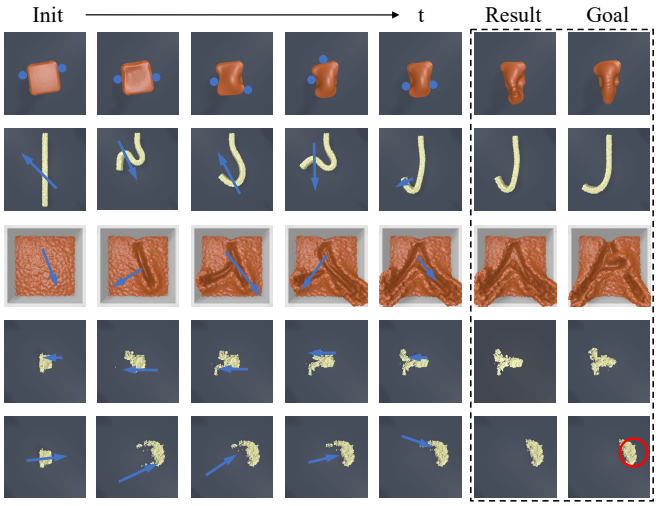


Fig. 4: Control results and example trajectories by DeformNet. The target configuration to be achieved is depicted in the image on the right. The planning process is illustrated in the left five columns, while the control results obtained from the algorithm are displayed in the sixth column. The five rows correspond to the 'Pinch', 'Push', 'Write', 'Pile', and, 'Transport' tasks, respectively. In the 'Pinch' task, the gripper's action is symbolized by a blue dot. In the 'Push' and 'Write' tasks, the actions executed by the stick are illustrated by blue arrows. In the 'Pile' and 'Transport' tasks, the arrows represent the piling direction and distance.

some characters on the clay surface. In the 'Pile' task, the robot's challenge is to pile grains on a table into specific target shapes using a board. Similar to the 'Pile' task, the 'Transport' task involves the same scenario but focuses on relocating grains on the table. Lastly, in the 'Poke' task, a robot with a stick prods plasticine into some target shapes. For simplicity, we assume a deterministic initial observation for all tasks. In the simulation experiments, all tasks are executed using a 7 DoFs Franka Emika robot with 8 cameras around the deformable object.

B. Action Space

We specify the action space for each task within a parameterized framework. In tasks such as 'Push', 'Write', and 'Transport', the action space is characterized by $\{x_0, y_0, x_1, y_1\}$, where $\{x_0, y_0\}$ denotes the initial position of the end effector, and $\{x_1, y_1\}$ denotes the final destination. The motion of the stick or board from $\{x_0, y_0\}$ to $\{x_1, y_1\}$ is at a fixed height.

For the 'Pile' task, the action space is defined as $\{x_0, y_0, \delta_d\}$, where $\{x_0, y_0\}$ denotes the initial position of the board, and δ_d represents the board's movement. In the 'Poke' task, we simplify its action space to $\{x_0, y_0\}$, indicating that the robot pokes the plasticine at a consistent height.

Lastly, for the 'Pinch' task, the action is parameterized as $\{x, y, z, r_z, d_g\}$. Here, $\{x, y, z\}$ denotes the center of the gripper, r_z denotes the rotation of the robot gripper along the z-axis, and d_g indicates the minimal distance between the gripper fingers. We apply the inverse kinematics library

	Push			Pinch			Write
	CD↓	EMD↓	SIoU↑	CD↓	EMD↓	SIoU↑	SIoU↑
Random Policy	13.28 ± 3.95	4.91 ± 2.50	0.13 ± 0.12	3.14 ± 0.55	1.27 ± 0.22	0.06 ± 0.08	0.19 ± 0.07
PPO	12.81 ± 3.79	3.76 ± 3.27	0.15 ± 0.09	2.53 ± 0.15	1.04 ± 0.03	0.15 ± 0.08	0.19 ± 0.04
NeRF-RL	9.50 ± 2.02	2.91 ± 1.29	0.12 ± 0.05	2.50 ± 0.13	1.01 ± 0.03	0.26 ± 0.04	0.17 ± 0.06
NeRF-dy	11.47 ± 2.55	3.09 ± 0.77	0.19 ± 0.13	2.25 ± 0.43	0.95 ± 0.20	0.21 ± 0.11	0.18 ± 0.07
Dreamer-V2	9.78 ± 0.01	2.75 ± 0.02	0.18 ± 0.09	2.53 ± 0.32	1.04 ± 0.07	0.22 ± 0.07	0.26 ± 0.06
DeformNet (Ours)	**6.86 ± 3.11**	**2.66 ± 1.45**	**0.27 ± 0.12**	**1.87 ± 0.27**	**0.80 ± 0.13**	**0.32 ± 0.09**	**0.53 ± 0.07**

	Poke			Pile			Transport
	CD↓	EMD↓	SIoU↑	CD↓	EMD↓	SIoU↑	D2CD↓
Random Policy	2.30 ± 0.15	0.94 ± 0.06	0.11 ± 0.05	5.68 ± 2.20	3.75 ± 1.80	0.05 ± 0.09	0.11 ± 0.03
PPO	2.28 ± 0.06	0.91 ± 0.01	0.08 ± 0.04	4.68 ± 1.94	2.69 ± 1.35	0.05 ± 0.04	0.10 ± 0.01
NeRF-RL	2.26 ± 0.10	0.90 ± 0.04	0.11 ± 0.07	3.83 ± 1.58	2.46 ± 1.24	0.06 ± 0.05	0.10 ± 0.01
NeRF-dy	2.05 ± 0.06	0.81 ± 0.11	0.17 ± 0.08	3.78 ± 1.46	2.65 ± 1.30	0.13 ± 0.15	-
Dreamer-V2	1.56 ± 0.03	**0.62 ± 0.07**	0.27 ± 0.04	5.14 ± 0.61	4.61 ± 0.50	0.08 ± 0.02	0.10 ± 0.03
DeformNet (Ours)	**1.46 ± 0.23**	0.63 ± 0.11	**0.45 ± 0.11**	**2.88 ± 0.47**	**2.37 ± 0.60**	**0.20 ± 0.09**	**0.07 ± 0.01**

TABLE I: Quantitative comparisons between the DeformNet and the baselines on each task. In the 'Push', 'Pinch', 'Poke', and 'Pile' tasks, we evaluate DeformNet and the baselines using CD, EMD, and SIoU. Lower values of CD, D2CD, and EMD indicate better performance, while higher values of SIoU suggest better performance. Both CD and EMD are scaled by 100 in the table. In the 'Write' and 'Transport' tasks, we specifically compare the SIoU and D2CD respectively. NeRF-dy is omitted in the final column due to its reliance on the objective's latent representation, while the transport environment lacks a goal representation. DeformNet consistently outperforms all baselines in most testing scenarios, demonstrating its performance in generalization.

from robot toolbox [44] and default PD control from the Maniskill2 environment.

C. Baseline Methods

In order to provide a comprehensive comparison, we include the following baseline methods. **Random**: This baseline randomly samples actions uniformly within the action space. **PPO**: We employ the Proximal Policy Optimization (PPO) [45] as the model-free RL algorithm. **NeRF-RL**: Inspired by the original NeRF-RL paper [40], we utilize a 2D convolutional neural network (CNN) to encode images into a latent representation supervised by a NeRF decoder. PPO is then applied to the learned latent space. **NeRF-dy**: This method is an adaptation of NeRF-dy [11], which omits the use of time contrastive loss during training. Instead of the model predictive path integral (MPPI) planning method, we employ iCEM as the planning approach for consistency. **Dreamer-V2**: Following the structure of Dreamer-V2 [41], we adopt a 2D CNN encoder to encode images captured from different perspectives. These baseline methods serve as reference points for evaluating the performance of our proposed approach. For each baseline, we evaluate the result by sampling 100 trajectories in each task.

D. Training Details

To train our models, we follow a specific procedure for each task. Initially, we generate 100 random trajectories and train our representation model and dynamics model separately. The dynamics model is updated with the latest weights of the trained representation model every 50 epochs.

For the PointNet encoder, we utilize RGB-D images of sizes (8, 256, 256, 4) to generate the point cloud, while RGB images of sizes (8, 128, 128, 3) are used for conditioned NeRF training. After every 1000 epochs, we employ the gradient-based iCEM to generate additional training samples. The planning phase of iCEM consists of 50 iterations. The training process is performed on 4 NVIDIA A6000 GPUs. On average, each iteration of representation training takes approximately 1 second, while dynamics training requires around 5 seconds.

We employ the stable-baseline-3 framework [46] to train the PPO and NeRF-RL. Since we do not incorporate goal-conditioned input, we randomly select a single target for training and evaluation of the RL model. Additionally, we implement Dreamer-V2 using the default Actor-Critic algorithm. The Actor-Critic module, reward predictor, and terminal predictor within the Dreamer-V2 framework are conditioned on the goal-related information. In the 'Push', 'Pinch', 'Poke', and 'Pile' tasks, we use CD for training, and EMD and SIoU for evaluation. In the 'Write' task, the particle distances vary due to different writing orders for the same character. Thus, we just employ SIoU to train and evaluate the particle set's similarity to the goal. In the 'Transport' task, we only utilize D2CD rather than compare the similarity between two distributions of MPM particles.

E. Comparison with Baselines in Simulation

Our proposed approach demonstrates effective performance in handling complex deformable manipulation tasks, as presented in Table I. We first observe that NeRF-RL surpasses the vanilla model-free RL method (PPO), indicating the benefits of utilizing a latent representation supervised by NeRF. This improvement notably enhances the learning capabilities of the robot. Moreover, Dreamer-V2 also achieves satisfactory performance in the 'Push', 'Write', and 'Poke' tasks, highlighting the capabilities of

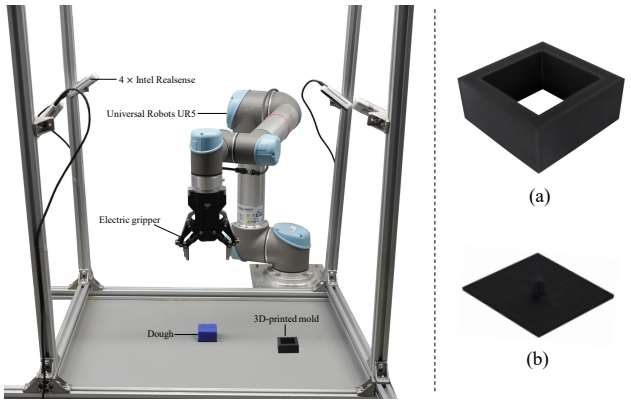


Fig. 5: Real world experiment setup. Left: an overview of the real robot experiment setup. Right: 3D-printed tools including (a) the mold for resetting, and (b) the platform to fix the dough.

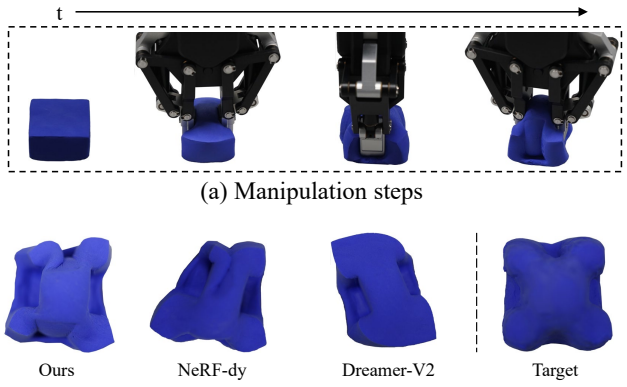


Fig. 6: Manipulation results. (a) The robot successfully molds the dough into the shape of the letter 'X'. (b) Left: the manipulation results of three baseline methods. Right: the target shape for reference.

	CD (cm) ↓	**EMD** (cm) ↓	**Human evaluation**↑
NeRF-dy	1.28 ± 0.11	0.76 ± 0.04	0.71 ± 0.16
Dreamer-V2	1.34 ± 0.04	0.79 ± 0.03	0.48 ± 0.07
DeformNet	**1.14 ± 0.09**	**0.69 ± 0.04**	**0.83 ± 0.14**

TABLE II: Quantitative comparisons between the DeformNet and the baselines in the real world scenario. In our evaluation, we compare the performance of DeformNet with baseline methods using metrics such as CD and EMD. For the human evaluation, we invite 50 humans to evaluate our results. The results highlight the superior performance of DeformNet.

the RSSM model. However, in tasks such as 'Pinch' and 'Pile', which involve a higher degree of deformation, all RL-based algorithms, including PPO, NeRF-RL, and Dreamer-V2, exhibit relatively lower performance compared to NeRF-dy. This can be attributed to the challenges posed by the highly deformable space, where RL-based algorithms struggle to achieve high rewards and may become trapped in local minima. In contrast, NeRF-dy, which utilizes iCEM for trajectory sampling with the learned model, achieves superior performance. DeformNet outperforms or closely matches all baseline methods across all metrics in the six challenging tasks.

F. Real-World Experiment

To evaluate the effectiveness of DeformNet in real-world robot systems, we conducted experiments utilizing a Universal Robots UR5 robot with 6 DoFs, as depicted in Fig. 5. The robot is connected to a computer with a NVIDIA 3080 GPU via socket communication. We control the electric gripper through the I/O interface. The experiment setup involves 4 Intel RealSense D415 RGB-D cameras at different positions around the dough. These cameras are calibrated and capture RGB-D images at a resolution of 1280×720 and a frame rate of 30 Hz. The initial shape of the plasticine is approximately a cuboid measuring 5×5×3 cm^3. Our goal is to manipulate the dough into the shape of the letter 'H' or 'X'. To ensure consistent starting conditions, we insert the dough into a 3D-printed mold, shaped as a cuboid cavity at the beginning of each episode.

In our real world experiment, we simplify the bounded action space into the parameterized set $\{x, y, z, r_z, d_g\}$. For data collection, we randomly collect about 300 pinching episodes for each step in the bounded action space. Then, we convert the RGB-D images captured during these episodes into point clouds with the camera parameters. To enhance the quality of the data, we apply the RANSAC algorithm [47] to remove the outliers in the point cloud data. For training the NeRF decoder, we set near bound $\alpha_n = 0.1$ and far bound $\alpha_f = 0.4$. Fig. 6 demonstrates that the resulting pinching behavior closely matches the desired shape. To provide a quantitative assessment, we executed a total of 50 trials after training the model. Among these, 36 trials yield a Chamfer distance measurement of less than 1.3 cm, corresponding to a success rate of 72%. Further quantitative verification is presented in Table II, affirming that DeformNet consistently achieved the lowest Chamfer distance and the Earth mover's distance. This real-world experiment validates the effectiveness and practical applicability of our proposed approach in deformable robotic manipulation scenarios.

V. Conclusion

This study presents DeformNet, a novel approach to learn representations of deformable objects from visual observations by employing an autoencoding framework enriched with a neural radiance field rendering module. In addition, we leveraged RSSM to capture the dynamics and facilitate accurate prediction. The effectiveness of DeformNet was demonstrated through its generalization capabilities across diverse deformable manipulation tasks. Experimental evaluations confirmed the superiority of our system compared to multiple baseline methods. Furthermore, our system demonstrated compatibility with an iCEM planner, emphasizing its practical application in real-world scenarios. The successful combination of deformable representations, dynamics learning and control provides a promising foundation for the manipulation of deformable objects.

References

[1] R. Antonova, *et al.*, "A Bayesian treatment of real-to-sim for deformable object manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5819–5826, 2022.

[2] R. Lee, *et al.*, "Sample-efficient learning of deformable linear object manipulation in the real world through self-supervision," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 573–580, 2021.

[3] W. Yan, *et al.*, "Learning predictive representations for deformable objects using contrastive estimation," in *5th Annual Conference on Robot Learning (CoRL)*, London, UK, Nov 2021.

[4] D. Sánchez, W. Wan, and K. Harada, "Tethered tool manipulation planning with cable maneuvering," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2777–2784, 2020.

[5] D. Seita, *et al.*, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, Jun 2021.

[6] Y. She, *et al.*, "Cable manipulation with a tactile-reactive gripper," *International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[7] H. Ha and S. Song, "FlingBot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *5th Annual Conference on Robot Learning (CoRL)*, London, UK, Nov 2021.

[8] X. Lin, *et al.*, "Learning visible connectivity dynamics for cloth smoothing," in *5th Annual Conference on Robot Learning (CoRL)*, London, UK, Nov 2021.

[9] Y. Wu, *et al.*, "Learning to manipulate deformable objects without demonstrations," in *Robotics: Science and Systems (RSS)*, Corvalis, Oregon, USA, Jul 2020.

[10] C. Legaard, *et al.*, "Constructing neural network based models for simulating dynamical systems," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–34, 2023.

[11] Y. Li, *et al.*, "3D neural scene representations for visuomotor control," in *5th Annual Conference on Robot Learning (CoRL)*, London, UK, Nov 2021.

[12] J. Sanchez, *et al.*, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[13] Y. Li, *et al.*, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.

[14] C. Matl and R. Bajcsy, "Deformable elasto-plastic object shaping using an elastic hand and model-based reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, Oct 2021.

[15] J. Gu, *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," in *10th International Conference on Learning Representations (ICLR)*, Virtual Event, Apr 2022.

[16] B. Thananjeyan, *et al.*, "Multilateral surgical pattern cutting in 2D orthotropic gauze with deep reinforcement learning policies for tensioning," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, Jun 2017.

[17] B. Mildenhall, *et al.*, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *16th European Conference on Computer Vision (ECCV)*, Glasgow, UK, Aug 2020.

[18] D. Driess, *et al.*, "Learning multi-object dynamics with compositional neural radiance fields," in *6th Conference on Robot Learning (CoRL)*, Auckland, New Zealand, Dec 2022.

[19] S. Li, *et al.*, "Visual–tactile fusion for transparent object grasping in complex backgrounds," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3838–3856, 2023.

[20] X. Lin, *et al.*, "SoftGym: Benchmarking deep reinforcement learning for deformable object manipulation," in *4th Annual Conference on Robot Learning (CoRL)*, Cambridge, MA, USA, Nov 2020.

[21] H. Shi, *et al.*, "RoboCraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks," in *Robotics: Science and Systems (RSS)*, New York City, USA, Jul 2022.

[22] D. Hafner, *et al.*, "Learning latent dynamics for planning from pixels," in *36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, Jun 2019.

[23] C. Pinneri, *et al.*, "Sample-efficient cross-entropy method for real-time planning," in *4th Annual Conference on Robot Learning (CoRL)*, Cambridge, MA, USA, Nov 2020.

[24] T. Kurutach, *et al.*, "Learning plannable representations with causal infogan," in *32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, Montréal, Canada, Dec 2018.

[25] S. Chen, *et al.*, "DiffSRL: Learning dynamical state representation for deformable object manipulation with differentiable simulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9533–9540, 2022.

[26] M. Lippi, *et al.*, "Latent space roadmap for visual action planning of deformable and rigid object manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, Oct 2020.

[27] Y. Li, *et al.*, "Causal discovery in physical systems from videos," in *34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec 2020.

[28] X. Ma, D. Hsu, and W. S. Lee, "Learning latent graph dynamics for visual manipulation of deformable objects," in *International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, May 2022.

[29] Q. Tan, *et al.*, "Realtime simulation of thin-shell deformable materials using CNN-based mesh embedding," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2325–2332, 2020.

[30] K. Park, *et al.*, "Nerfies: Deformable neural radiance fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, Jun 2021.

[31] C. R. Qi, *et al.*, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul 2017.

[32] D. Navarro-Alarcon, *et al.*, "Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[33] A. Cherubini, *et al.*, "Model-free vision-based shaping of deformable plastic materials," *International Journal of Robotics Research*, vol. 39, no. 14, pp. 1739–1759, 2020.

[34] K. Yoshimoto, *et al.*, "Active outline shaping of a rheological object based on plastic deformation distribution," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, Sep 2011.

[35] A. Ganapathi, *et al.*, "Learning to smooth and fold real fabric using dense object descriptors trained on synthetic color images," *arXiv preprint arXiv:2003.12698*, vol. 3, 2020.

[36] N. Figueroa, A. L. P. Ureche, and A. Billard, "Learning complex sequential tasks from demonstration: A pizza dough rolling case study," in *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Christchurch, New Zealand, Mar 2016.

[37] A.-M. Cretu, P. Payeur, and E. M. Petriu, "Soft object deformation monitoring and learning for model-based robotic hand manipulation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 740–753, 2011.

[38] S. Li, *et al.*, "Contact points discovery for soft-body manipulations with differentiable physics," in *10th International Conference on Learning Representations (ICLR)*, Virtual Event, Apr 2022.

[39] X. Lin, *et al.*, "DiffSkill: Skill abstraction from differentiable physics for deformable object manipulations with tools," in *10th International Conference on Learning Representations (ICLR)*, Virtual Event, Apr 2022.

[40] D. Driess, *et al.*, "Reinforcement learning with neural radiance fields," in *36th Annual Conference on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, Dec 2022.

[41] D. Hafner, *et al.*, "Mastering Atari with discrete world models," in *9th International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2021.

[42] Z. Huang, *et al.*, "PlasticineLab: A soft-body manipulation benchmark with differentiable physics," in *9th International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2021.

[43] H. T. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *14th International Workshop on Algorithmic Foundations of Robotics (WAFR)*, Oulu, Finland, Jun 2021.

[44] P. Corke and J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python," in *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, Jun 2021.

[45] J. Schulman, *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[46] A. Raffin, *et al.*, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[47] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.