

# Learning Goal-oriented Bimanual Dough Rolling Using Dynamic Heterogeneous Graph Based on Human Demonstration

Junjia Liu<sup>1</sup>, Chenzui Li<sup>1</sup>, Shixiong Wang<sup>1</sup>, Zhipeng Dong<sup>1</sup>,  
 Sylvain Calinon<sup>2</sup>, *Member, IEEE*, Miao Li<sup>3</sup>, *Senior Member, IEEE*, and Fei Chen<sup>†1</sup>, *Senior Member, IEEE*

**Abstract**—Soft object manipulation poses significant challenges for robots, requiring effective techniques for state representation and manipulation policy learning. State representation involves capturing the dynamic changes in the environment, while manipulation policy learning focuses on establishing the relationship between robot actions and state transformations to achieve specific goals. To address these challenges, this research paper introduces a novel approach: a dynamic heterogeneous graph-based model for learning goal-oriented soft object manipulation policies. The proposed model utilizes graphs as a unified representation for both states and policy learning. By leveraging the dynamic graph, we can extract crucial information regarding object dynamics and manipulation policies. Furthermore, the model facilitates the integration of demonstrations, enabling guided policy learning. To evaluate the efficacy of our approach, we designed a dough rolling task and conducted experiments using both a differentiable simulator and a real-world humanoid robot. Additionally, several ablation studies were performed to analyze the effect of our method, demonstrating its superiority in achieving human-like behavior.

## I. INTRODUCTION

In the context of domestic scenarios, the manipulation of soft objects using dual arms is a common requirement. Numerous studies in the field of robotics have focused on bimanual activities to advance the application of robots in service-oriented tasks. Examples include folding clothes [1], coffee stirring [2], and cooking with stir-fry [3]. These tasks necessitate accurate state estimation of soft objects and the execution of coordinated bimanual movements. It is crucial to employ a reliable state representation method that can robustly capture the changes in the state of soft objects despite disturbances from environmental factors and sensor limitations. Additionally, an important practical consideration is the ability to transfer policies learned in simulation to real-world scenarios, bridging the significant gap between simulated and real environments. Such capability would

This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grant 14211723, 14222722, 24209021 and C7100-22GF.

<sup>1</sup>Junjia Liu, Chenzui Li, Shixiong Wang, Zhipeng Dong, and Fei Chen are with the Department of Mechanical and Automation Engineering, T-Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong (e-mail: jjliu@mae.cuhk.edu.hk, czli@mae.cuhk.edu.hk, sxwang@hkclr.hk, zhipengdong@cuhk.edu.hk, f.chen@ieee.org).

<sup>2</sup>Sylvain Calinon is with the Idiap Research Institute, Martigny, Switzerland (e-mail: sylvain.calinon@idiap.ch).

<sup>3</sup>Miao Li is with the School of Microelectronics and the Institute of Technological Sciences, Wuhan University, Wuhan, China (e-mail: miao.li@whu.edu.cn).

<sup>†</sup>Corresponding authors

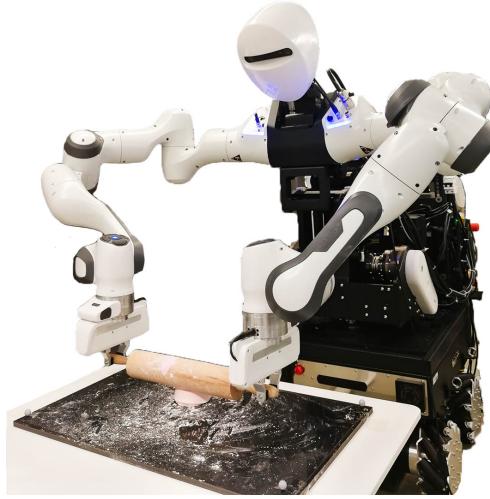


Fig. 1. Perform a bimanual dough rolling task with a rolling pin on a CURI robot by using the pre-trained dynamic heterogeneous graph model.

greatly facilitate the practical implementation of simulator-based learning methods in real-world applications. After obtaining a compact and concise state information, robots are required to learn the bimanual manipulation policy to achieve specific goals. However, the complexity of movements and the requirement for scene generalization make simple logic programming inadequate for such tasks. In recent decades, two mainstream solutions have emerged: Reinforcement Learning (RL) and Learning from Demonstration (LfD). However, both approaches have their limitations. Demonstrations provide direct examples of expert policies, enabling the robot to mimic behavior in a supervised manner. This approach restricts the robot's ability to learn object state representation and discover new skills due to the limited amount of available data. On the other hand, RL-based methods allow for extensive interaction with the environment but often struggle to learn complex or long-horizon tasks from scratch. In summary, state representation and policy learning are the primary factors that restrict robots' ability to acquire human-like skills, particularly in the domain of soft object manipulation. Therefore, this paper aims to address these two challenges simultaneously and in a natural manner by proposing a suitable approach.

Soft object state estimation and representation, a crucial challenge in the field of robotics, has garnered significant attention in the realm of Computer Graphics. Huang et al. introduced PlasticineLab, a state-of-the-art differentiable

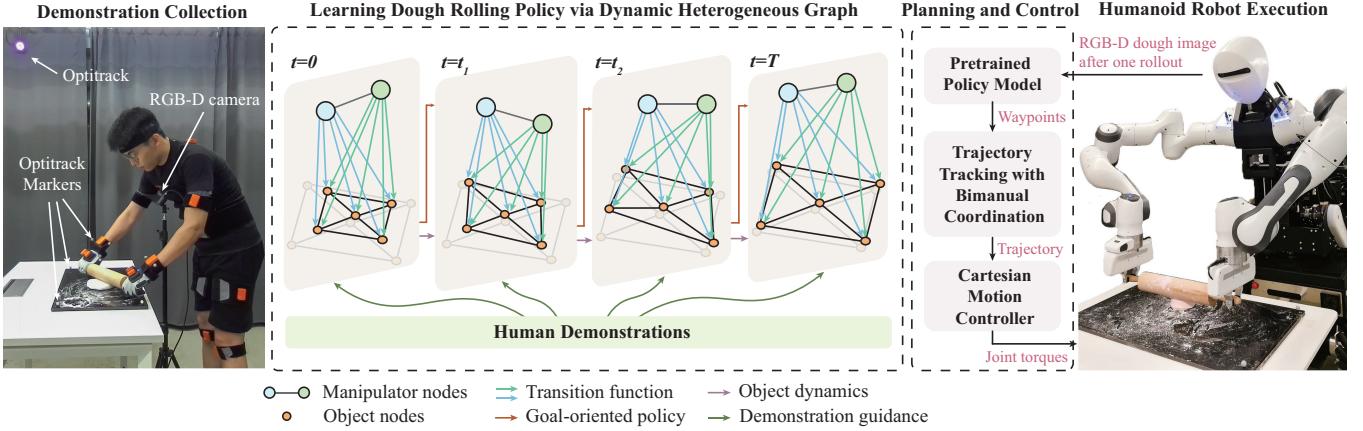


Fig. 2. The proposed pipeline of soft object manipulation, which contains two parts: dynamic heterogeneous graph policy learning in the simulator with the guidance of demonstration, and planning and control for deploying on the real-world humanoid robot.

physics benchmark, supported by the DiffTaichi system [4]. PlasticineLab utilizes the MLS-MPM algorithm to simulate plasticine-like objects. Notably, the differentiable physics framework provides analytical gradient information, enabling supervised learning with gradient-based optimization. The versatility of PlasticineLab has led to numerous studies focusing on soft object manipulation. For instance, Shi et al. proposed the utilization of graph neural networks (GNNs) to learn the particle-based dynamics of soft objects for model-based planning [5]. These advancements highlight the potential of employing differentiable physics simulations and graph neural networks to enhance soft object state estimation and manipulation strategies in the field of robotics.

Soft object manipulation policy research primarily revolves around two approaches: direct imitation from human demonstration and policy transfer from simulators. The former involves supervised learning techniques, such as Dynamic Movement Primitives (DMP) [6], TP-GMM [7], and deep learning models [8], to obtain optimal policies by fitting data from demonstrations. The latter leverages reinforcement learning (RL) to train robots in simulators like Habitat [9], Mujoco [10], pyBullet, DiffTaichi [4], and PlasticineLab [11]. While demonstration provides expert policies and mimics human behavior [3], it may have limitations in learning object state representation and discovering new skills due to limited data. Simulators bridge the gap with reality but struggle with policy transfer. Combining demonstration and simulation allows for the integration of their strengths and mitigates their weaknesses [12].

Relational learning offers a unified solution to the above challenges of soft object manipulation. Geometric information of soft objects can be represented as a graph, where robot end-effectors are abstracted as manipulator nodes to capture the relationship between manipulation and state change. By modeling continuous manipulation behavior as the evolution of dynamic graphs, we can learn robot behavior oriented towards object state changes. Demonstration data can also be represented as dynamic graphs, enabling seamless integration with simulator-based policy learning and overcoming differences between reality and simulation. This approach provides

a comprehensive framework for robust and adaptable soft object manipulation in dough rolling task.

This paper presents several key contributions. Firstly, we propose a **Dynamic heterogeneous Graph based model for learning Goal-oriented deformable manipulation policy (DGform)**. This model adopts a unified graph representation for both state information and policy learning, incorporating guidance from human demonstrations. Secondly, we provide a comprehensive pipeline for deploying the pre-trained policy on a real-world humanoid robot, specifically focusing on the task of dough rolling. Lastly, we conduct ablation studies and generalization analysis to demonstrate the superior performance of our proposed method in learning human-like bimanual skills.

## II. DOUGH ROLLING TASK

To evaluate our method, we tackle the challenging task of bimanual dough rolling. Previous dough rolling research includes Figueroa et al., who employed a hierarchical framework for automatic task segmentation and represented human demonstrations using action primitives [13]. However, such movement primitive-based methods have limitations in understanding the relationship between robot motions and object state changes. Calinon et al. discussed generalization in learning from few demonstrations, presenting a task-parameterized mixture model and comparing it with other approaches [14]. It primarily focused on determining the rolling direction rather than accomplishing the complete dough rolling task.

In recent dough rolling research, there has been a growing reliance on differential simulators. Diffskill, for instance, introduced a method for learning skill abstraction in sequential dough manipulation tasks. It employed a differentiable simulator to generate simulated demonstrations and trained a supervised policy for each dough manipulation primitive [15]. Another notable work proposed a Generative Pre-trained Transformer (GPT) [16] based world model (Soft-GPT) for learning the dynamics of dough from a differential simulator [17]. This model facilitated efficient learning of diverse dough manipulation tasks including dough rolling,

but it only conducted experiments in the simulator.

### III. SOFT OBJECT MANIPULATION POLICY LEARNING VIA DYNAMIC HETEROGENEOUS GRAPH

In this section, we will introduce the key concepts and components used in our proposed methods. The definitions of important symbols are provided in Table I. The overall soft object manipulation pipeline is illustrated in Fig. 2, consisting of two main parts: (1) dynamic heterogeneous graph policy learning in the simulator with the guidance of demonstration, and (2) planning and control for deploying the pre-trained model on the real-world humanoid robot.

#### A. Represent soft objects and manipulator as a heterogeneous graph

Recent studies have started utilizing graph-structured data and graph neural networks to model soft object manipulation policy learning. For instance, some research has focused on predicting action effects of a bag using graph-based approaches [18], learning the dynamics of soft objects [5].

Graphs serve as efficient and compact data structures for representing non-Euclidean data, capturing both semantic and geometric information through their nodes and edges. To address the challenge of abstracting the state of a dough object, which lacks distinct characteristic points unlike soft objects such as clothes [1], we employ a robust solution that minimizes interference with subsequent policy learning. In this study, we utilize standard image processing techniques to achieve this abstraction. Initially, we segment the dough from the operation board using a color threshold and subsequently determine the center of the segmentation contour. From this contour center, we establish four lines with equal angular spacing and calculate eight focal points between them and the contour. These focal points serve as position features for the object's boundary nodes. Alongside the center node, we construct a 9-node  $\mathcal{V}_o$  graph to represent the object's state. Each node in the graph is associated with a position feature and a corresponding depth feature obtained from depth observations. To provide clarity, Fig. 2 illustrates an example with five object nodes. The graph includes edges connecting all object boundary nodes to the center node, while the boundary nodes are connected solely to their two immediate neighbors. This graph-based representation facilitates effective modeling of the dough object's state and its subsequent utilization in the research field of robotics.

As shown in the graph block in Fig. 2, except for the object sub-graph  $S\mathcal{G}_t$ , the dual end-effectors are added as *manipulator nodes*, together constructing the complete heterogeneous graph  $\mathcal{G}_t$ . Heterogeneous graph refers to graphs that contain different types of nodes and also different types of edges between nodes. Therefore, a heterogeneous graph representation allows us to learn more relationships as we expected. The manipulator node features are their poses with respect to the board coordinate system. The edges between the manipulator and object nodes are fully connected.

TABLE I  
DEFINITION OF SYMBOLS

Symbols	Definition
$D$	The dimension of state
$O$	The order of the controller
$T$	Time horizon, $t \in [0, T]$
$o_t$	The state of the object at time $t$
$o_g$	The goal state of the object
$a_t$	The Cartesian pose of robot dual end-effectors
$\mathcal{V}_m, \mathcal{V}_o$	Manipulator nodes and object nodes
$\mathcal{E}_{mo}, \mathcal{E}_{oo}$	Edges between $V_m$ and $V_o$ , and among $V_o$ s
$X_{\mathcal{V}_t}, X_{\mathcal{E}_t}$	The attribute matrices of nodes and edges at time $t$
$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$	The static graph at time $t$
$X\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, X_{\mathcal{V}_t}, X_{\mathcal{E}_t})$	The static graph at time $t$ with attributes
$S\mathcal{G}_t, SX\mathcal{G}_t$	The static object sub-graph at time $t$
$\zeta = [\zeta_1^T, \dots, \zeta_T^T]^T$	Demonstrated bimanual trajectory, each $\zeta_t = [\zeta_t^{(l)T}, \zeta_t^{(r)T}]^T \in \mathbb{R}^{2DO}$
$SX\mathcal{G}_{\zeta_t}$	Demonstrated static object sub-graph at time $t$

#### B. Learn state change as the evolution of the dynamic graph

After gaining the graph abstraction of object and manipulator states at each time step, we then consider describing the dynamic evolution of the state and the corresponding manipulation trajectory. Each static graph  $\mathcal{G}_t$  is a snapshot of the dynamic graph evolving from time  $0 \sim T$ . Thanks to the nature of the heterogeneous graph, we can define several different types of graph convolution operations on this graph. As shown in Fig. 2, we learn the goal-oriented policy (Equ. 1a), transition function and object dynamics (Equ. 1b) and value function (Equ. 1c) in a unified dynamic heterogeneous graph model.

$$\hat{a}_{t+1} \leftarrow \hat{X}_{\mathcal{V}_{m,t+1}} = \pi_{\text{GraphConv}}(X_{\mathcal{V}_{m,t}}, X_{\mathcal{V}_{o,t}} | X_{S\mathcal{G}_g}) \quad (1a)$$

$$\hat{S}\mathcal{G}_{t+1} \leftarrow \hat{X}_{\mathcal{V}_{o,t+1}} = Tr_{\text{GraphConv}}(X_{\mathcal{V}_{m,t}}, X_{\mathcal{V}_{o,t}}) \quad (1b)$$

$$v \leftarrow V(\hat{X}_{\mathcal{V}_{m,t+1}}, \hat{X}_{\mathcal{V}_{o,t+1}}) \quad (1c)$$

The goal-oriented policy takes the graph abstraction of the goal state (the semitransparent graphs in Fig. 2) as a condition and learns to obtain the hidden feature of the manipulator node at the next time step through the graph feature of the current state. Similarly, another Graph Convolutional Network (GCN) [19] is set for predicting the next hidden feature of object nodes. These hidden features are transformed to the next action  $\hat{a}_{t+1}$  and next object state prediction  $\hat{S}\mathcal{G}_{t+1}$  by two additional feed-forward networks. The value of the current policy is estimated upon these hidden features as well. Specifically, we first perform a mean aggregation of these heterogeneous hidden features and then feed it to another feed-forward network.

Since several sub-tasks exist simultaneously, the loss function is composed of several parts. Referring to RL methods, we calculate the advantage function based on the value estimation and the actual return and generate the value loss and the action loss. Another term is a supervised loss for learning an accurate dynamic model.

$$\mathcal{L}_t(\theta) = \mathbb{E}[\mathcal{L}_t^{CLIP} + c_1 \mathcal{L}_t^{VF} + c_2 \mathcal{L}_t^{Dyn} - c_3 S[\pi](\hat{S}\mathcal{G}_t)] \quad (2)$$

### C. Demonstration guidance during the policy learning

The goal-oriented policy has shown the superiority of integrating the goal state in a compact dimension via graph representation. Another advantage is combining the fully explore-based methods with learning from demonstration methods, adding the demonstration as a policy learning guidance. This topic is studied in the offline RL field. The intention of CQL is to lower-bound the actual value of Q function by introducing a conservative term, avoiding the common value overestimation problem in offline RL. This term minimizes values under an appropriately chosen distribution over state-action tuples and then further tightens this bound by incorporating a maximization term over the data distribution [20]. Since DGform has not been extended to an actor-critic form, a state-action value estimation is not available. Thus, we adopt the current policy distribution to evaluate demonstration state-action pairs.

$$\mathcal{L}_t^{Imi}(\zeta|\theta) = - \sum_{\zeta} \pi_{\theta}(\zeta_t|SX\mathcal{G}_{\zeta_t}) \quad (3)$$

As we expect, demonstration guidance should not be a strong supervision constraint but should leave room for exploration. Therefore, we further add a self-adjusting term to convert Equ. 3 to a Lagrange version.

$$\begin{aligned} \hat{\theta}^{k+1} \leftarrow \arg \min_{\theta} \max_{\alpha} \mathbb{E} \left[ (\alpha \mathcal{L}_t^{Imi}(\zeta|\theta) - \tau) + \mathcal{L}_t^{CLIP} \right. \\ \left. + c_1 \mathcal{L}_t^{VF} + c_2 \mathcal{L}_t^{Dyn} - c_3 S[\pi](\hat{\mathcal{G}}_t) \right] \end{aligned} \quad (4)$$

where  $\tau$  is the imitation threshold,  $\alpha$  is a self-adjusting Lagrange coefficient of conservative term. More specifically, if the difference between the expected value and the threshold  $\tau$  is less than 0, then  $\alpha$  will tend to 0. Conversely,  $\alpha$  will increase, and the weight of the conservative term in the entire loss calculation will also increase. This version intends to adaptively change the supervision intensity according to the deviation between the rollout and demonstration actions while leaving room for exploration in the vicinity of the demonstration.

### D. Model-based rollout by the learned object dynamics

Although directly using image processing methods like segmentation to represent soft objects in a 2-dim graph simplifies our reliance on the visual front-end, it also brings some practical problems. It is challenging to represent the state of the dough by this method when occluding by other objects, like the rolling pin and robot hands. We certainly can also use methods similar to RoboCraft [5] to analyze the occlusion of objects by setting up cameras with multiple views in their experiments. However, information incompleteness is the most common in real scenarios. A more appropriate and robust solution is to predict actions over a period of time through the learned model when the state is unknown for part of the period. Formally, for a short horizon  $t \in t_k \sim t_k + H$ , since the step-by-step interaction with the environment is not available, the proposed model alternately generates the

features of object and manipulator nodes.

$$\hat{a}_{t+1} \leftarrow \hat{X}_{\mathcal{V}_{m,t+1}} = \pi(\hat{X}_{\mathcal{V}_{m,t}}, \hat{X}_{\mathcal{V}_{o,t}} | X_{S\mathcal{G}_g}) \quad (5a)$$

$$\hat{\mathcal{G}}_{t+1} \leftarrow \hat{X}_{\mathcal{V}_{o,t+1}} = Tr(\hat{X}_{\mathcal{V}_{m,t}}, \hat{X}_{\mathcal{V}_{o,t}}) \quad (5b)$$

Then we can get a sequence of predicted actions  $\hat{a} = [\hat{a}_{t_k} \dots \hat{a}_{t_k+H}]$  and their corresponding effect prediction. The simulator performs these actions sequentially to obtain the actual feedback (return and next state sequence). Furthermore, the loss function needs to be modified to a period form accordingly.

### E. Bimanual Trajectory Planning

The action output of DGform policy is path waypoints of dual end-effectors, which is inadequate to perform directly on robots with high frequencies (1000Hz in control loop for the humanoid robot in this paper). Thus, a trajectory planning module is required. Besides, since the dual end-effectors has a fixed relative relationship, coordination needs to be considered in the trajectory planning. Here we use Bi-manual Relative Parameterization (Bi-RP) [21] for modeling the coordination and generating bimanual trajectories with Linear Quadratic Tracking (LQT) [22].

Formally, the coordination is defined as another frame that takes the trajectory of the other arm as dynamic task parameters and represents the relative relationship as GMM. The relative motion between dual end-effectors is described as  $\zeta^{(c)} = A_{c,t}^{-1}((\zeta^{(l)} - \zeta^{(r)}) - b_{c,t})$  and represented by  $\{\pi^{(k)}, \mu_c^{(k)}, \Sigma_c^{(k)}\}_{k=1}^K$ . For each end-effector, the task-specific GMM obtained by PoE

$$\mathcal{N}(\hat{\nu}^{(k)}, \hat{\Gamma}^{(k)}) \propto \prod_{j=1}^P \mathcal{N}(\nu_j^{(k)}, \Gamma_j^{(k)}) \cdot \mathcal{N}(\nu_c^{(k)}, \Gamma_c^{(k)}) \quad (6)$$

where  $\nu_c^{(k)} = A_{c,t}\mu_c^{(k)} + b_{c,t}, \Gamma_j^{(k)} = A_{c,t}\Sigma_j^{(k)}A_{c,t}^\top$ .  $A_{c,t}, b_{c,t}$  are transformation matrices.  $P$  is the number of reference frames. More details can be found in [21].

## IV. EXPERIMENTS

### A. Setup

**Simulation** The simulation environment is built on top of PlasticineLab [11], with a particle-based dough and a rolling pin. We set up cameras with multiple views to show how the rolling pin and dough change in 3-dim space, while the RGB-D observation for policy learning is obtained from the top-down view. To shorten the learning process while maintaining graph abstraction performance, the resolution of the camera was set to (128, 128).

**Real-world experiments** The pre-trained model was deployed on the Collaborative dUal-arm Robot manIupulator (CURI) robot, a self-designed bimanual human-like robot equipped with two 7 DoF arms and a 3 DoF torso. CURI utilizes a Cartesian motion controller to execute desired Cartesian space end-effector trajectories by generating joint torques. In real experiments, feedback is provided through first-person perspective observations captured by a single ZED 2i camera embedded in CURI's head.

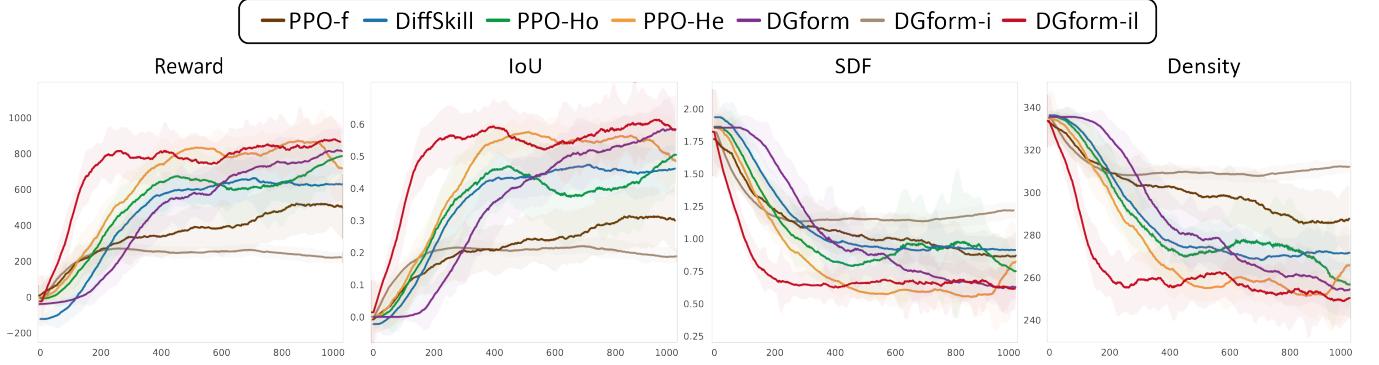


Fig. 3. The learning performance between baselines and DGform is compared from rewards to three evaluation metrics (SDF, density, and IoU) with the help of the differentiable simulator. The comparison between PPO-based methods demonstrate the effective of graph representation, while three DGform variants show the feasibility of learning policy via graph and the role of demonstration guidance and Lagrangian terms.

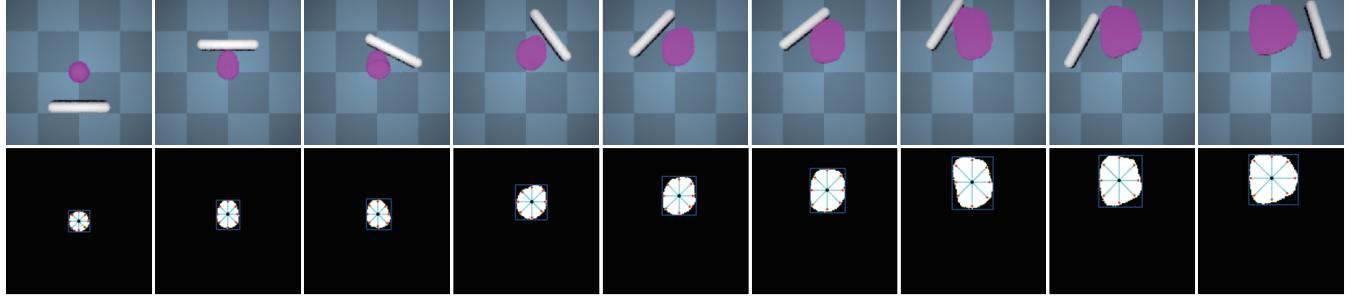


Fig. 4. An example of DGform inference in the simulator. The short horizon  $H$  in the simulation is set to be 50. The rolling pin will be reset to its initial state after each short horizon and inferred based on the current deformation and the goal state. The first row shows the sequential rolling pin movements sample, and the second is the corresponding visualization of object graph abstractions. Red nodes are boundary nodes, and black nodes are center nodes.

**Evaluation metric** Since the full state of particles is available in the simulator, we compare the Signed Distance Field (SDF), density, and Intersection over Union (IoU) between the current state and the target state.

**Network details** The model consists of two-layer heterogeneous Graph Convolutional Neural network and a two-layer MLP-based policy learning network combining CQL with PPO, as described in III-C.

The preprocessing of the multimodal demonstration data and the implementation of the LQT method depend on the Rofunc package, a full-process python package for robot learning from demonstration published by our lab [23]. It provides numerous interfaces for demonstration collection and processing, baseline LfD methods, planning methods, and Isaac Gym-based simulators.

### B. Demonstration collection

To incorporate human demonstration during policy learning, we adopt a comprehensive multimodal data collection approach, including visual and motion capture information. Since our focus is on generating bimanual Cartesian trajectories for the rolling pin handles, the whole-body human motion is simplified to dual hand motions. The spatial relationship between the demonstrator hands and the worktop was recorded using the Optitrack system. Specifically, two groups of markers are attached to the demonstrator’s wrists, while four markers are fixed on the board to indicate the position of worktop, as shown in the left side of Fig. 2. To

TABLE II  
AVERAGE PERFORMANCE OF DIFFERENT SCENES

Methods	Reward	IoU	SDF	Density
PPO-f*	-	-	-	-
DiffSkill	63.7	0.0733	1.5307	327.9
PPO-Ho	509.3	0.3883	0.9056	284.5
PPO-He	629.5	0.4203	0.6422	271.9
<b>DGform<sup>†</sup></b>	<b>756.1</b>	<b>0.5573</b>	<b>0.6223</b>	<b>266.4</b>

\* In any case, the full particle states are the same.

<sup>†</sup> The three variants of DGform share the same state representation, so here only the vanilla DGform is added as a representative.

capture the dynamic changes in the dough state, we employ ZED 2i cameras, which provide RGB-D data. By synchronizing and abstracting the data from these sensors as a sequence of heterogeneous graphs, following the techniques described in Section III-A, we create a demonstration dataset. This dataset is utilized in batch to calculate the imitation loss  $\mathcal{L}_t^{Imi}$  as defined in Equation 3. During the demonstrations, the demonstrator was instructed to roll the dough towards a large circular shape within 10 rollouts.

### C. Policy learning performance

An inference example of the DGform rolling policy is shown in Fig. 4, the first row shows the sample of sequential rolling pin movements, and the second row is the corresponding visualization of object graph abstractions.

**Baselines** The learning performance comparison is mainly divided into two categories, exploring the impact of state representation (full particle state, RGB-D and graph) and the

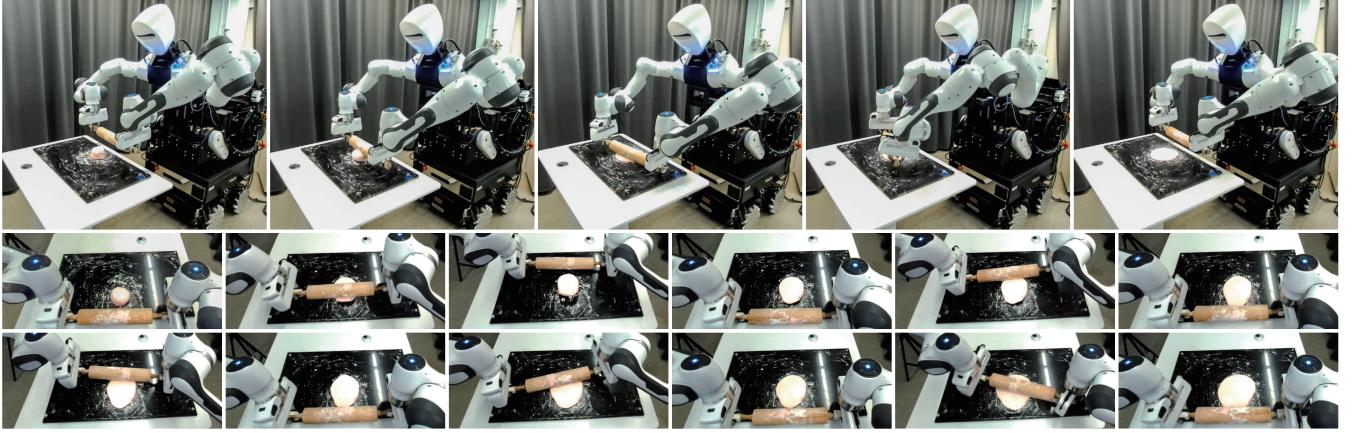


Fig. 5. The real robot experiment consists of five rollouts. Each time at the initial pose, the camera in CURI’s head captured an RGB-D image of the dough, then it was converted into a heterogeneous graph through graph abstraction and passed as input to the pre-trained DGform model. The model generated a path with 50 waypoints, which was then transformed into a smooth dual-arm trajectory with 10,000 points by bimanual LQT. The trajectories were executed by the Cartesian motion controller embedded in the CURI. The first row shows the recording from a third perspective, while the last two are observations from the first-person perspective camera.

impact of policy learning methods (RL, Lfd, or mixed via DGform).

- *PPO with the full state (PPO-f)*: Use full particle states as the observation of PPO.
- *PPO with RGB-D state (DiffSkill)*: Use RGB-D images from a top-down view as the observation of PPO, like the primitive skill learning model used in DiffSkill [15].
- *PPO with Homogeneous Graph state (PPO-Ho)*: Convert the RGB-D image to a homogeneous object subgraph, and use it as observations of PPO.
- *PPO with Heterogeneous Graph state (PPO-He)*: Convert the RGB-D image to a heterogeneous graph, and use it as observations of PPO.

Except for the PPO-based baselines, we provide three variants of DGform: vanilla DGform, DGform with imitation (*DGform-i*), and DGform with imitation and Lagrange (*DGform-il*). It is worth mentioning that all baselines do not have a model-based module and need to interact with the environment step-by-step. In contrast, DGform and its variants can rollout a trajectory in a period with the help of the learned model (Sec. III-D). Without immediate interaction, the performance might be influenced, but it will also be robust and practical in real-world scenarios.

From Fig. 3, we can figure that the performance is better when reducing the state dimension from full particle state or RGB-D to compact graph state. Among them, PPO with full particle state obtains the worst performance because it is intractable for the network to process tens of thousands of dimensional inputs. In the graph-based method, the heterogeneous graph adds the information of the manipulator. It thus has additional graph convolutional networks to acquire more relations implied in the soft object manipulation. Comparing these four PPO-based methods demonstrates that graph abstraction is sufficient to represent the deformation of objects and is an effective way of state representation.

On the other hand, policy learning with dynamic graph

(DGform) surpasses the PPO-based methods at a slow but robust pace. The inefficiency of this method is primarily affected by the non-interactive model-based rollout since there is little actual interaction with the environment. This makes sense for deploying real robots but also runs the risk of difficult convergence. Thus, we introduce the demonstration guidance, which shows a solid guiding effect in the early stage of learning, driving the policy in the direction of demonstration and avoiding wasting time in meaningless state-action pairs. However, DGform-i without Lagrange cannot converge to normal performance. This is likely constrained by demonstration data taken from different environments. This issue has been addressed by modifying it to a Lagrange version, evaluating the weight of demonstration guidance items in policy updates according to the actual policy dynamically. Finally, the Lagrange version DGform with demonstration guidance achieves the best performance most efficiently.

#### D. Generalization analysis

Another essential indicator for evaluating robot skill learning is whether it is easy to transfer to unseen scenarios. With the help of differentiable simulators, we designed generalization experiments with different initial states or camera positions. These two variables are generated randomly in a reasonable range. Their average values of evaluation metrics are shown in Table II. The results were inferred from pre-trained baselines and DGform models. It is easy to figure out that methods with graph state representation have better transfer ability. Moreover, the manipulation performance of DGform does not change much as the environment changes.

#### E. Real robot dough rolling

The process of deploying a pre-trained simulated policy on a real-world robot is depicted in the right side of Fig. 2. At the start of the experiment, the initial state of the dough is captured by the embedded ZED 2i camera in CURI’s head,

which provides an RGB-D image. This information is abstracted as an object subgraph using the technique described in Section III-A. To determine the relative pose of the robot end-effectors with respect to the center of the worktop, we employ coordinate transformations between the robot hands and the camera. To ensure consistency with the simulator, the poses of the robot end-effectors are set to be the same as those used in the simulated environment, simplifying the impact of initial poses on the pre-trained model. As discussed in Section III-D, to mitigate occlusion of the end-effector and rolling pin from the first-person view camera during manipulation, a model-based rollout policy is adopted during training in the simulator. Consequently, the entire manipulation process is divided into multiple rollouts. After each rollout, visual feedback is captured and provided to the policy, which generates the bimanual trajectories for the subsequent rollout based on this feedback. The experimental results are presented in Fig. 5, showcasing the performance of the deployed policy on the real-world robot.

## V. DISCUSSION

It is worth noting that this paper only focuses on the two-dimensional shape change of dough. Point cloud information and three-dimensional graph neural network will be introduced, like SoftGPT [17], to realize the complete deformation manipulation on a real humanoid robot.

## VI. CONCLUSION

This paper proposes a dynamic heterogeneous graph-based dough manipulation policy learning model and elaborates its pipeline to deploy the model on a humanoid robot to achieve a dough rolling task with a rolling pin. We show the superiority of using a unified graph form, which can learn the goal-oriented policy and object dynamics and integrate the demonstration guidance simultaneously without the interference of environmental change. Compared with several baseline methods, the proposed DGform with a compact representation and demonstration guidance achieves a human-like dough rolling behavior. The generalizations of these methods are also analyzed, which show the robustness of DGform against environmental change. Finally, a real-world experiment was conducted on a CURi robot, aided by bimanual LQT with coordination.

## REFERENCES

- [1] H. Yin, A. Varava, and D. Kragic, “Modeling, learning, perception, and control methods for deformable object manipulation,” *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.
- [2] Z. Sun, Z. Wang, J. Liu, M. Li, and F. Chen, “Mixline: A hybrid reinforcement learning framework for long-horizon bimanual coffee stirring task,” in *International Conference on Intelligent Robotics and Applications*, pp. 627–636, Springer, 2022.
- [3] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, “Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5159–5166, 2022.
- [4] Y. Hu, L. Andersson, T. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, “Diffitaichi: Differentiable programming for physical simulation,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.
- [5] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, “Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks,” 2022.
- [6] S. Schaal, “Dynamic movement primitives-a framework for motor control in humans and humanoid robotics,” in *Adaptive motion of animals and machines*, pp. 261–280, Springer, 2006.
- [7] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent service robotics*, vol. 9, no. 1, pp. 1–29, 2016.
- [8] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in neural information processing systems*, vol. 29, 2016.
- [9] M. Savva, J. Malik, D. Parikh, D. Batra, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, and V. Koltun, “Habitat: A platform for embodied AI research,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 9338–9346, IEEE, 2019.
- [10] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033, IEEE, 2012.
- [11] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan, “Plasticinelab: A soft-body manipulation benchmark with differentiable physics,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [12] J. Liu, J. Shou, Z. Fu, H. Zhou, R. Xie, J. Zhang, J. Fei, and Y. Zhao, “Efficient reinforcement learning control for continuum robots based on inexplicit prior knowledge,” *arXiv preprint arXiv:2002.11573*, 2020.
- [13] N. Figueiroa, A. L. P. Ureche, and A. Billard, “Learning complex sequential tasks from demonstration: A pizza dough rolling case study,” in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 611–612, Ieee, 2016.
- [14] S. Calinon, T. Alizadeh, and D. G. Caldwell, “On improving the extrapolation capability of task-parameterized movement models,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 610–616, IEEE, 2013.
- [15] X. Lin, Z. Huang, Y. Li, J. B. Tenenbaum, D. Held, and C. Gan, “Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools,” *arXiv preprint arXiv:2203.17275*, 2022.
- [16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [17] J. Liu, Z. Li, W. Lin, S. Calinon, K. C. Tan, and F. Chen, “Softgpt: Learn goal-oriented soft object manipulation skills by generative pre-trained heterogeneous graph transformer,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4920–4925, IEEE, 2023.
- [18] Z. Weng, F. Paus, A. Varava, H. Yin, T. Asfour, and D. Kragic, “Graph-based task-specific prediction models for interactions between deformable and rigid objects,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5741–5748, IEEE, 2021.
- [19] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [20] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative q-learning for offline reinforcement learning,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.
- [21] J. Liu, H. Sim, C. Li, K. C. Tan, and F. Chen, “Birp: Learning robot generalized bimanual coordination using relative parameterization method on human demonstration,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 8300–8305, IEEE, 2023.
- [22] S. Calinon and D. Lee, “Learning control,” in *Humanoid robotics: A reference*, pp. 1–52, Springer Netherlands, 2017.
- [23] J. Liu, Z. Dong, C. Li, Z. Li, M. Yu, D. Delehelle, and F. Chen, “Rofunc: The full process python package for robot learning from demonstration and robot manipulation,” 2023.