

# SoftGPT: Learn Goal-oriented Soft Object Manipulation Skills by Generative Pre-trained Heterogeneous Graph Transformer

Junjia Liu<sup>1</sup>, Zhihao Li<sup>1</sup>, Wanyu Lin<sup>2</sup>, Sylvain Calinon<sup>3</sup>, *Member, IEEE*,  
Kay Chen Tan<sup>2</sup>, *Fellow, IEEE*, and Fei Chen<sup>†1</sup>, *Senior Member, IEEE*

**Abstract**—Soft object manipulation tasks in domestic scenes pose a significant challenge for existing robotic skill learning techniques due to their complex dynamics and variable shape characteristics. Since learning new manipulation skills from human demonstration is an effective way for robot applications, developing prior knowledge of the representation and dynamics of soft objects is necessary. In this regard, we propose a pre-trained soft object manipulation skill learning model, namely SoftGPT, that is trained using large amounts of exploration data, consisting of a three-dimensional heterogeneous graph representation and a GPT-based dynamics model. For each downstream task, a goal-oriented policy agent is trained to predict the subsequent actions, and SoftGPT generates the consequences of these actions. Integrating these two approaches establishes a thinking process in the robot’s mind that provides rollout for facilitating policy learning. Our results demonstrate that leveraging prior knowledge through this thinking process can efficiently learn various soft object manipulation skills, with the potential for direct learning from human demonstrations.

## I. INTRODUCTION

Soft object manipulation is a fundamental and common skill in human housework that has yet to be fully explored in robotics. Compared with deformable object manipulation that has been well studied [1] [2] [3], the shape change of soft objects is not elastic deformation, which makes it more difficult to predict the shape change outcome. Robots that can manipulate soft objects with the same dexterity and efficiency as humans have the potential to improve their usefulness in our daily lives significantly. They can help humans with tasks such as folding laundry, kneading dough, stir-frying, chopping, and prepping vegetables, which are more widespread than deformable object manipulation. However, soft object manipulation currently faces two significant challenges, including the representation of the shape change and the dynamics of soft objects. The shape change of soft objects often has a specific goal, so in order to measure the difference between the current shape and the target shape,

This work was supported in part by the Research Grants Council of the Hong Kong SAR under Grant 24209021, 14222722 and C7100-22GF and in part by CUHK Direct Grant for Research under Grant 4055140.

<sup>1</sup>Junjia Liu, Zhihao Li and Fei Chen are with the Department of Mechanical and Automation Engineering, T-Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong SAR (e-mail: jjliu@mae.cuhk.edu.hk, zhihaoli@mae.cuhk.edu.hk, f.chen@ieee.org).

<sup>2</sup>Wanyu Lin and Kay Chen Tan are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR (e-mail: wanyulin@comp.polyu.edu.hk, kctan@polyu.edu.hk).

<sup>3</sup>Sylvain Calinon is with the Idiap Research Institute, Martigny, Switzerland (e-mail: sylvain.calinon@idiap.ch).

<sup>†</sup>Corresponding authors



Fig. 1. The proposed pipeline of soft object manipulation contains two parts: a pre-trained SoftGPT model for providing prior knowledge about soft object manipulation and a goal-oriented policy for each specific task.

it is necessary to propose a compact representation that can cope with the shape and thickness variations of soft objects. Furthermore, although recent works have shown the potential of generating a world model with environment dynamics to efficiently learn robot manipulation skills online [4], these models still primarily focus on rigid objects. The dynamics of soft objects that produce shape change are more complex, and learning directly in real-world scenarios can be time-consuming due to the reset.

In the domestic scenario, the ability to learn these complex skills directly from a few human demonstrations will be the key to the development of domestic service robots [5], which suggests that robots should have a fundamental judgment on the manipulation [6] [7] in addition to the demonstration data. Drawing inspiration from the hot topics in natural language processing, we propose to develop a Generative Pre-trained Transformer (GPT) [8] based model to establish a prior knowledge about the representation and dynamics of soft objects, supporting the efficient skill learning of soft object manipulation. Compared with the recent discussion about using ChatGPT as high-level function commands [9], we tend to find ways to integrate GPT technologies more closely with robotics. Thus, in this paper, we adopt a GPT-based model named SoftGPT to learn the relationship be-

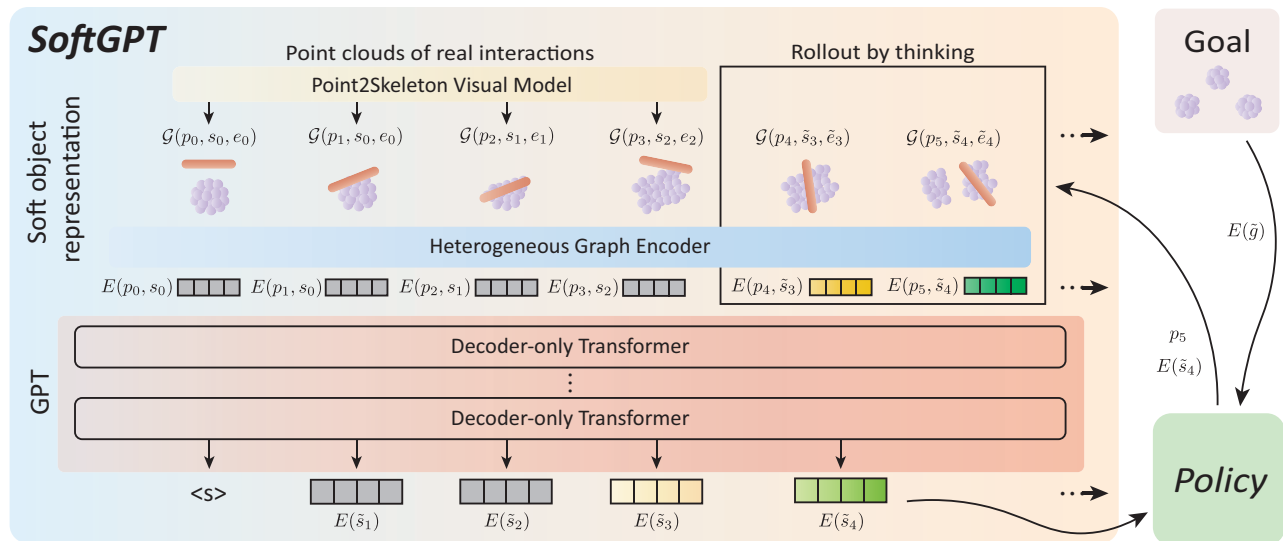


Fig. 2. SoftGPT is a pre-trained world model that learns the representation of soft objects and to predict the dynamics of environment. The model processes point clouds obtained from real interactions and represents them as three-dimensional compact graphs using the Point2Skeleton Visual Model. With the next robot pose provided by a goal-oriented policy, the model predicts the next state using several Decoder-only Transformer blocks. In this regard, the purple balls represent skeleton points, and the orange bars represent operation nodes, which illustrate a policy learning process for a cutting task with the aid of SoftGPT.

tween robot manipulation and soft object shape change, similar as how GPT works when answering questions. Generally speaking, it serves as a pre-trained world model in typical robotics research. As shown in Fig. 1, the SoftGPT with prior knowledge can estimate the manipulation consequence in advance and rehearse a rollout in robots' mind, and supports the skill learning of specific goal-oriented policies. Such a pre-trained model can significantly reduce the computational and time costs of training models from scratch.

Despite the help of differential simulators, learning soft object manipulation from scratch directly using reinforcement learning (RL) has proven difficult. PlasticineLab, a differentiable elastic and plastic deformation benchmark utilizing the DiffTaichi system [10], is an effective tool for evaluating the efficiency of RL-based approaches in solving shaping tasks [11]. The baseline results it provided demonstrate that RL-based methods cannot solve most shaping tasks in this context efficiently. Based on this work, RoboCraft is proposed to model a particle-based dynamics for elastoplastic objects using Graph Neural Network (GNN) [12]. We use the differential simulator to generate massive interactive data to pre-train the SoftGPT model and obtain the dynamics based on the graph representation of shape changes.

The main contributions of this paper can be summarized as follows:

- **SoftGPT model:** We propose a generative pre-trained Transformer-based model (SoftGPT) trained by massive interaction data for compact representation and dynamics modeling of soft objects.
- **Goal-oriented policy agent:** We provide a goal-oriented policy for each downstream tasks. By alternately performing the policy and SoftGPT, a thinking process is built in the robot's mind to provide rollout for policy learning.

- **Soft object manipulation:** We demonstrate that the pre-trained SoftGPT model can accelerate the learning of downstream policies, makes it efficient to learn task-specific soft object manipulations.

## II. GENERATIVE PRE-TRAINED HETEROGENEOUS GRAPH TRANSFORMER AND GOAL-ORIENTED POLICY AGENT

The present study proposes a pipeline for robot manipulation of soft objects (Fig. 1), comprising two main stages: pre-training and thinking while learning. The pre-training stage involves training a SoftGPT model using a dataset generated by a maximum exploration agent. This SoftGPT model serves as a world model of soft objects, including the prior knowledge about manipulation consequences of them. The thinking stage employs a model-based policy learning mechanism that alternatively predicts the next action and state using both a goal-oriented policy agent and the SoftGPT model. The key idea behind this approach is to provide the robot with a fundamental understanding of soft objects through extensive contact with the soft object environment. This approach allows the robot to leverage prior knowledge when learning specific manipulation skills and avoid exploring meaningless actions. In summary, the proposed pipeline seeks to enhance the robot's cognitive capabilities in manipulating soft objects and enable it to learn new skills more efficiently.

In this section, we first elaborate the structure of SoftGPT model, including dynamic modeling by GPT and the corresponding graph-based shape representation of soft objects (Sec. II-A). Then we show how to construct a goal-oriented policy agent in the latent space of the representation, and describe how it alternates with SoftGPT during policy learning to form the robot's thinking process (Sec. II-B).

### A. Learn contact dynamics model from massive interaction

The core module of SoftGPT is a GPT-based dynamics modeling model, which consists of a heterogeneous graph convolutional neural network-based [13] encoder and stacked decoder-only Transformers. Fig. 2 provides an illustration of its components. SoftGPT is pre-trained in an unsupervised manner, similar to GPT-2 [8]. It is trained to predict the next state of the object in a sequence based on the previous states and actions. However, unlike typical sequence-to-sequence tasks in natural language processing, robot manipulation tasks involve two heterogeneous roles: the manipulator and the object. To accommodate this, an additional processing should be first performed on the training set, that is, to construct a shifted dataset based on the exploration graph dataset represented by the visual model. Specifically, manipulator nodes should be shifted right to form a graph like  $\mathcal{G}(p_{t+1}, s_t, e_t)$ . Since the edges between the manipulator nodes and object nodes are fully-connected and constant, the time annotation of edges follows the change of object nodes. Given a sequence of graph samples in this shifted dataset as input, a heterogeneous graph encoder converts the graphs to latent embedding vectors  $E(p_{t+1}, s_t)$ . These latent vectors are then considered as past experience and inputted to a stacked decoder-only Transformer to predict the embedding of the next state,  $E(\tilde{s}_{t+1})$ . Soft objects deform as a result of the continuous movement of a robot, and the state feature of the manipulator node only captures the current pose. Hence, it is reasonable to replace the LSTM-based Recurrent State-Space Model (RSSM) [14] in some world model works [4][15] with Transformer [16] that can handle longer sequences of historical manipulations.

Given that reconstructing accurate heterogeneous graph data with continuous features from hidden vectors is both time-consuming and meaningless, we opted to learn dynamics directly within the hidden space of the graph encoder. To ensure that SoftGPT remains a goal-agnostic pre-trained model, it only generates representations of object subgraphs rather than the same format as inputs, which are heterogeneous graphs. As a result, the heterogeneous graph neural network encoder needs some additional techniques to handle this situation. Specifically, the GNN-based encoder consists of a layer of homogeneous graph neural network solely for object subgraphs, followed by a heterogeneous graph neural network for the entire scene graph, to learn the relationship between manipulator nodes and object subgraphs. Additionally, the next embedding prediction of the object's state is fed between these two graph neural networks and combined with the next action, whether derived from data or generated from the policy agent. The details are illustrated in Fig. 3.

To support the efficient process of Heterogeneous Graph Encoder, a visual model is required to obtain compact and effective representations of soft object shapes. We use Point2Skeleton model [17] to extract three-dimensional graph with a small number of nodes from dense point clouds. The node features are the positions and radii of the extracted skeletal points. This graph is used as a subgraph denoted

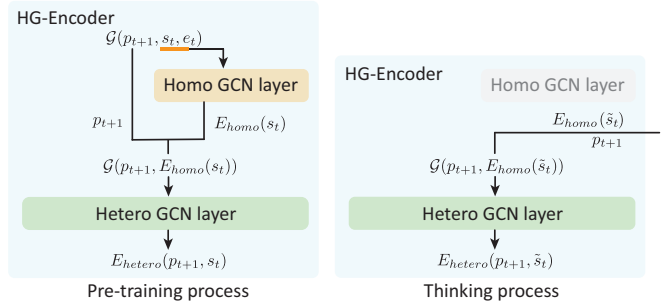


Fig. 3. Heterogeneous Graph Encoder in pre-training and thinking process. Left: The object subgraph in a real interaction is encoded first and a new graph with embedding object features is built to obtain the final graph embedding. Right: Embedding object features and the action are provided by the SoftGPT and policy, the homogeneous layer is skipped.

as  $\mathcal{G}(s_t), t \in \mathbb{R}^+$ , to represent the shape characteristics of the soft object. When combined with the manipulator node, which includes the feature of its pose, denoted as  $\mathcal{G}(p_t)$ , it forms a heterogeneous graph  $\mathcal{G}(p_t, s_t, e_t)$  that depicts the scene. Here,  $e_t$  refers to the edges connecting the nodes in the graph.

### B. Learn goal-oriented policy with goal-agnostic SoftGPT

In the policy learning process, SoftGPT functions as a pre-trained world model with updates that occur infrequently. As depicted in Fig. 2, SoftGPT takes the embedding sequence from real interactions as input and predicts the consequence  $s_{t+1}$  of pose  $p_{t+1}$  at state  $s_t$ . The policy agent has the ability to generate actions based on either real states or predicted states in rollout. The data flow during the policy learning process is represented by the curves in Fig. 2.

The policy agent operates on states in hidden space and includes encoded goal states, predicting the next pose of the manipulator. Specifically, the policy agent uses the actor-critic framework [18] with double Q-network [19], and the training of these two networks employs data from both real interactions and SoftGPT rollout by the thinking process. The modified loss functions are defined as follows:

*Policy*

$$\mathcal{L}_\pi = -\mathbb{E} \left[ \min_{i=1,2} \tilde{Q}_{\phi_i}(\varepsilon_t, \tilde{a}_\theta(\varepsilon_t)) + \eta(\varepsilon_t | E(g)) + \alpha \mathcal{H}(\cdot | \varepsilon_t) \right]$$

*Q functions*

$$\mathcal{L}_Q = \mathbb{E} \left[ Q(\varepsilon_t, a) - \left( r + \eta(\varepsilon_{t+1} | E(g)) + \alpha \mathcal{H}(\cdot | \varepsilon_{t+1}) \right) \right]^2 \\ \gamma (1 - d) \left( \min_{i=1,2} \tilde{Q}_{\phi_{tar,g,i}}(\varepsilon_{t+1}, \tilde{a}_\theta(\varepsilon_{t+1})) \right) \right] \quad (1)$$

Here,  $\varepsilon_t = E(s_t)$ ,  $d$  represents the done signal,  $r$  refers to the real reward, and  $\mathcal{H}$  refers to the entropy of states.  $\eta(\cdot | E(g))$  is a learned goal-oriented reward function in the latent space.  $\tilde{Q}_{\phi_i}(\varepsilon_{t+1}, \tilde{a}_\theta(\varepsilon_{t+1}))$  refers to the modified Q evaluation that incorporates the thinking process:

$$(1 - \lambda) Q_{\phi_i}(\varepsilon_t, \tilde{a}_\theta(\varepsilon_t)) + \lambda Q_{\phi_i}^\lambda(\tilde{\varepsilon}_{t+1}) \quad (2)$$

The above equation is averaged over all  $N \in [t+1, t+H]$  with  $\lambda$  discount, where  $H$  is the thinking horizon, and  $\tilde{\varepsilon}_N$  refers to one of the rollout states generated from the

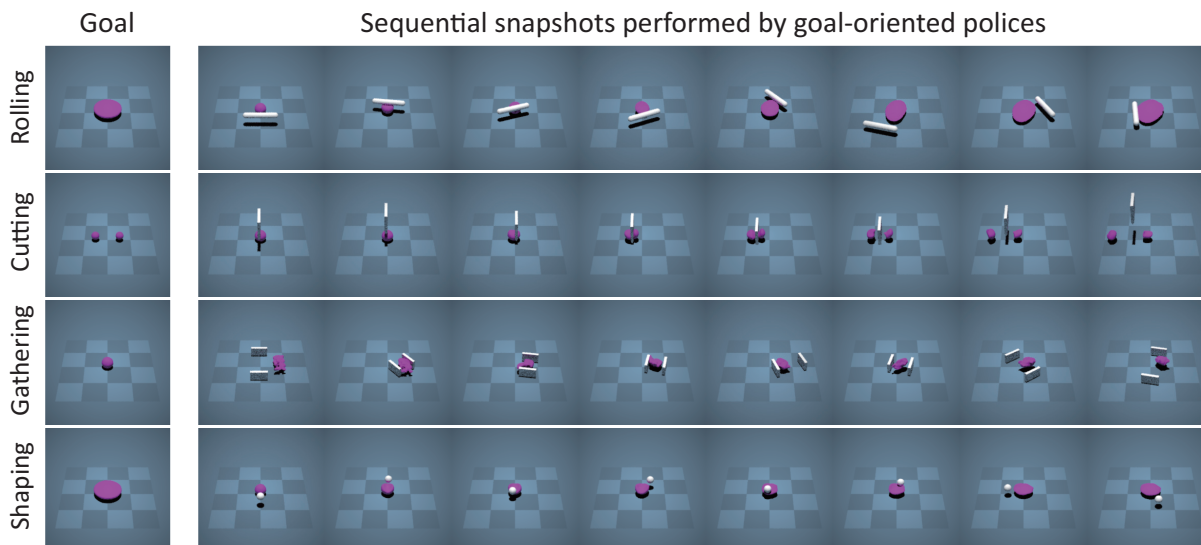


Fig. 4. Sequential snapshots performed by goal-oriented policy for each specific task.

thinking process. In this process, the policy alternates with the SoftGPT model, following the loop:

$$\tilde{\varepsilon}_{t+1} = \text{SoftGPT}(\varepsilon_t, p_{t+1}) \rightarrow p_{t+2} = \pi(\tilde{\varepsilon}_{t+1}) \rightarrow \dots \rightarrow \tilde{\varepsilon}_{t+H} \quad (3)$$

The incorporation of additional rollout from the thinking process enables the policy to consider long-horizon results, thereby improving the efficiency of policy learning through increased interaction in the robot’s cognitive process.

### III. EXPERIMENTS

#### A. Setup

**Tasks:** In order to demonstrate that the proposed pre-trained model can promote efficient learning with downstream tasks, we design four tasks with different tools and initial states:

- *Rolling* Train an agent to roll and spread a ball-shaped dough into a specific cake state using a rolling pin.
- *Cutting* Train an agent to separate a ball-shaped dough into several specific parts using a flat knife.
- *Gathering* Train an agent to use two flat-shaped tools to aggregate multiple scattered dough into a specific shape.
- *Shaping* Train an agent to use a sphere-shaped rolling ball to shape the dough into a specific cake state.

The action dimension of rolling pin, knife and rolling ball are 3, while the action dimension of dual flats is 7.

**Simulation:** The simulation environment is built on top of PlasticineLab [11], with a particle-based dough and tools listed in the tasks. We take the position of the particles that make up the dough from the simulator and pass their surface particles into the network as simulated point cloud information.

**Evaluation metric:** In the simulator, the complete state of particles is accessible, allowing us to evaluate the current state of the system in comparison to the target state using various metrics. Precisely, we assess the Signed Distance Field (SDF), density, and Intersection over Union (IoU) between the two states, forming the reward function together.

TABLE I  
THE COMPOSITION OF EXPLORATION DATASET

Shapes	Rolling Pin	Knife	Dual Flats	Rolling Ball
Ball	11398	8973	-	8083
Two balls	-	-	2980	-
Cuboid	7937	5072	-	8923
Random	-	-	4000	-

<sup>†</sup> The numbers refer to the amount of interactive data with specific tools and different initial shapes.

#### B. Exploration dataset generation

The generation of an exploration dataset for pre-trained models can be approached through two methods. One involves collecting human motion information using a motion capture system like Xsens suit and capturing point clouds of soft objects using multi-view cameras. However, this approach poses several engineering challenges, such as occlusion between objects and ensuring the quality of point clouds. Given that this work focuses on exploring the impact of pre-trained models on downstream soft object manipulation skill learning, we opted to use a RL-based exploration agent to generate interactive data in differential simulators. To facilitate full interaction with the soft object, we consider the initial state of the object as the target and frame the loss as a reward to incentivize the agent to shape the object from the initial state as much as possible. The initial state was randomly generated, incorporating both aggregated and scattered patterns to prepare for subsequent gatherings and cutting tasks. During the interaction, we record the sequence of point clouds of the object and the pose of the manipulator for pre-training both the graph representation model and the dynamics model. We screened out data that came into contact with objects, and the composition of the exploration dataset is listed in Table I.

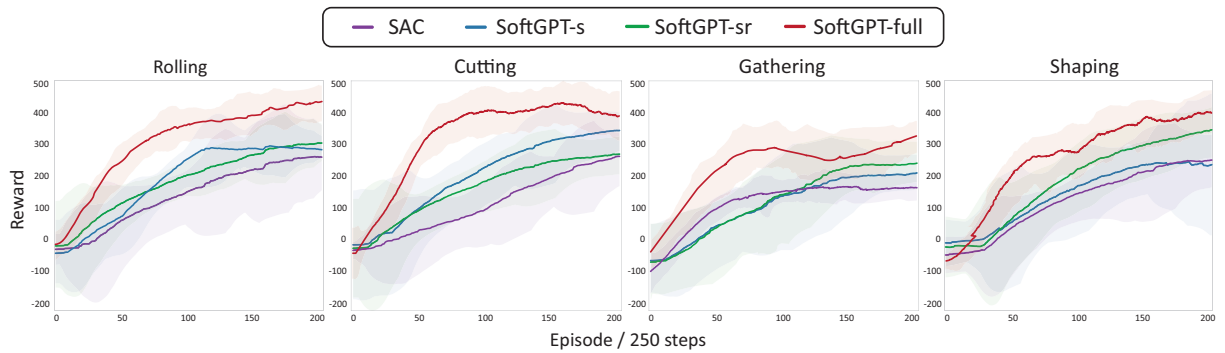


Fig. 5. The comparison of downstream task learning efficiency between using pre-trained SoftGPT and baseline methods. Baseline methods contain original Soft Actor-Critic policy (SAC), Soft Actor-Critic with SoftGPT from scratch (SoftGPT-s) and Soft Actor-Critic with SoftGPT from scratch and goal-oriented reward function  $\eta$  (SoftGPT-sr). All methods share the same visual model as state representation and Heterogeneous Graph Encoder.

### C. Implementation details

**Maximum exploration agent:** This agent is implemented directly based on the typical soft actor-critic algorithm shared by Rofunc [20].

**Point2Skeleton visual model:** We draw on the ideas of the Point2Skeleton approach [17] and utilize the Medial Axis Transformation (MAT) technique [21] to extract the skeleton points. The representation model takes the point cloud of soft objects as input and learns a geometric transformation to extract skeletal points and their radii. To achieve this, the PointNet++ approach [22] is utilized as the encoder to obtain the sampled input points and their contextual features. The convex combination of input points is then used to generate the skeletal points. To ensure the reliability of the resulting representations, a set of reliable links is generated based on topology prior, where a node has a link to its closest node rather than its  $k$ -farthest nodes. As we do not require mesh information, the further link prediction model in the original work is omitted for the sake of efficiency. For additional details on the Point2Skeleton approach, please refer to [17]. The shell of particles is extracted by alpha shape algorithm [23], and used as point cloud data input of the visual model. The model is set to generate 30 skeleton points.

**Heterogeneous graph encoder:** The encoder is composed of one homogeneous graph convolutional network layer and one heterogeneous graph convolutional network layer, both of which have a hidden state with a dimension of 32.

**Stacked Transformer:** The stacked decoder-only Transformer is based on the bare GPT-2 model shared by [24], which outputs raw hidden-states without any specific head on top. It has 12 hidden layers with a hidden dimension of 32, and 4 attention heads for each attention layer.

**Goal-oriented policy:** The policy is implemented based on SAC with several modifications, with a hidden dimension of 256 and learning rates of 0.0003. In order to handle the input of graph data, both actor and critic have a preprocessing model composed of a heterogeneous graph neural network, which encodes graph data into latent space vectors. The output of policy is the next pose of tools, which means a trajectory planning between poses is necessary. We adopt Linear Quadratic Tracking (LQT) shared by [20] to plan a 50-point trajectory between each two poses, and it will be

smoothed according to the previous series of points.

The actor  $\pi$ , critic  $Q_{\phi_i}$ , reward model  $\eta$  and SoftGPT model are trained using the Adam optimizer. Among them, the actor, critic and latent reward model are updated per 250 steps. SoftGPT model is updated per 500 steps.

### D. Goal-oriented policies for downstream tasks

For each specific task, a goal-oriented policy is trained with the help of the SoftGPT model. We use these learned policies to evaluate in the test environment of the same task, and the results are shown in Fig. 4. It can be observed that all four policy agents manipulate the dough towards their target shapes. The effect of rolling task is the best, which is largely related to the difficulty of the task. On the contrary, the effect of the other three tasks is poor, especially the gathering task. This might be due to the fact that the action dimension of dual flats is higher, which makes the manipulation more difficult. Moreover, the exploration agent is more inclined to break the dough when collecting interaction data, which is not conducive to learning the aggregation task.

### E. Learning efficiency analysis of downstream tasks

**Baselines:** We conducted an ablation study by comparing our entire proposed method to three baselines.

- *Soft Actor-Critic (SAC):* a SAC-based policy.
- *Soft Actor-Critic with SoftGPT from scratch (SoftGPT-s):* a SAC-based policy with a learned reward function.
- *Goal-oriented Soft Actor-Critic with SoftGPT from scratch (SoftGPT-sr):* a SAC-based policy with a learned reward function and a thinking process supported by a non-pretrained SoftGPT that learns from scratch alongside the policy.

All baselines were trained using the Point2Skeleton visual model and took the heterogeneous graph representation as input. Additionally, all SAC-based policies have graph preprocessing models in their actor and critic models.

The comparative performance of the proposed method and its baselines is presented in Fig. 5. Notably, the entire proposed method leveraging a pre-trained SoftGPT model (*SoftGPT-full*) exhibits superior efficiency in learning downstream tasks. Its stable performance basically appears around 50-100 episodes, which is two times faster than others.

Compared to the baselines, the proposed method achieves higher rewards in all the tested tasks. In particular, the proposed method shows significant improvement in the task of rolling and cutting dough toward a target state, indicating its potential in learning manipulation skills for soft objects. These results demonstrate the effectiveness of using a pre-trained world model for soft object manipulation tasks and suggest that the proposed method has the potential to support skill learning from a few human demonstrations.

#### IV. DISCUSSION

This work still has some limitations. First, this paper focuses on validating the role of a GPT-based pre-trained model for the robot skill learning of downstream tasks in soft object manipulation. We will introduce data augmentation method [25] to the dataset so that it can support learning new manipulation skills directly from human demonstrations. And we will verify it on real robots by using meta-learning in the graph representation [26], finding the sim2real effect of SoftGPT that is trained with massive simulated interaction data. Second, we can extend the feature of manipulator nodes to add haptics or force information so that the robot can manipulate soft objects with different physical properties based on the relationship between feedback and deformation in the pre-trained model. Finally, this work only draws on the large-scale model and generative concepts of GPT-2. We plan to introduce concepts such as human prompt and RLHF in the future so that humans can guide the manipulation of robots in real-time.

#### V. CONCLUSION

This paper focuses on how robots understand, represent, and manipulate soft objects. To support efficient skill learning of soft object manipulations, we propose that robots should have prior knowledge of soft object dynamics and representation. Thus, we introduce the idea of GPT-2 and design a generative pre-trained model with a three-dimensional graph representation for soft objects. With the help of this pre-trained model, the policy of each downstream task can be obtained easily and efficiently, making it possible for humans to teach robots similar skills through several demonstrations.

#### 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] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2017.
- [3] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-d deformable object manipulation using deep neural networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, 2019.
- [4] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel, "Daydreamer: World models for physical robot learning," *arXiv preprint arXiv:2206.14176*, 2022.
- [5] 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.
- [6] 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.
- [7] J. Liu, J. Guo, Z. Meng, and J. Xue, "Revolt: Relational reasoning and voronoi local graph planning for target-driven navigation," *arXiv preprint arXiv:2301.02382*, 2023.
- [8] 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.
- [9] S. Vemprala, R. Bonatti, A. Buckner, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res.*, vol. 2, p. 20, 2023.
- [10] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "DiffTaichi: Differentiable programming for physical simulation," in *International Conference on Learning Representations (ICLR)*, 2020.
- [11] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan, "Plasticinlab: A soft-body manipulation benchmark with differentiable physics," in *International Conference on Learning Representations (ICLR)*, 2020.
- [12] 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.
- [13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [14] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*, pp. 2555–2565, PMLR, 2019.
- [15] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [17] C. Lin, C. Li, Y. Liu, N. Chen, Y.-K. Choi, and W. Wang, "Point2skeleton: Learning skeletal representations from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4277–4286, 2021.
- [18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.
- [19] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.
- [20] J. Liu, C. Li, D. Delehelle, Z. Li, and F. Chen, "Rofunc: The full process python package for robot learning from demonstration and robot manipulation," June 2023.
- [21] H. Blum, "A transformation for extracting new descriptions of shape," *Models for the perception of speech and visual form*, pp. 362–380, 1967.
- [22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
- [23] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on information theory*, vol. 29, no. 4, pp. 551–559, 1983.
- [24] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020.
- [25] J. Liu, H. Sim, C. Li, and F. Chen, "Birp: Learning robot generalized bimanual coordination using relative parameterization method on human demonstration," *arXiv preprint arXiv:2307.05933*, 2023.
- [26] Z. Wang, L. Cao, W. Lin, M. Jiang, and K. C. Tan, "Robust graph meta-learning via manifold calibration with proxy subgraphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023.