Learning-Based Fabric Folding and Box Wrapping

Xiaoman Wang[1], Jie Zhao[1], Xin Jiang[1], and Yun-Hui Liu[1,2]

***Abstract*—Manipulation of deformable objects is an essential task in surgery, the textile industry, and household tasks, such as washing, hanging, and folding clothes. However, current studies on fabric manipulation, have rarely considered the scenes in which rigid objects have to be wrapped by fabrics. This type of operation is widely adopted in the logistic packaging and packaging of surgical instrument baskets. In this study, we propose a method to perform this operation, which can be used in fabric folding or wrapping a box with fabric. Owing to the complex dynamics and configuration spaces of fabric, our method is based on deep imitation learning to estimate pick-place points and the phase of the manipulation process. The dataset was completely generated from an open-source physical simulator. To make the data as realistic as that in actual scenarios, we adopted domain randomization and rendered the texture of fabric and box from the real world to simulation data. This not only helps in transferring the learned policies to a physical robot, but also allows the robot to wrap a box with complex patterns. The experiments demonstrate the efficiency of the developed method for accomplishing complex manipulation tasks. The results also showed that it could be generalized to fabrics with different colors and boxes with different sizes, textures, or geometric shapes.**

***Index Terms*—Deep Learning in Grasping and Manipulation, Grasping**

I. INTRODUCTION

THERE have been some studies on fabric manipulation using a robot, such as surgery (handling gauze [1]), domestic service (ironing [2], folding [3], [4], bed-making [5]) and industry (sewing [6]). However, compared with rigid objects, there are additional challenges in manipulation of deformable objects by a robot considering its complex dynamics and infinite configurations.

Although several prior studies have made significant progress in the manipulation of deformable objects, rigid objects are rarely involved in those tasks. However, in many fields, such as the medical industry, wrapping surgical boxes with fabrics before sterilization is mostly accomplished by humans. This type of work is particularly challenging: it is difficult to separate the box from fabric owing to the complex texture of the box and the mutual occlusion during the manipulation process.

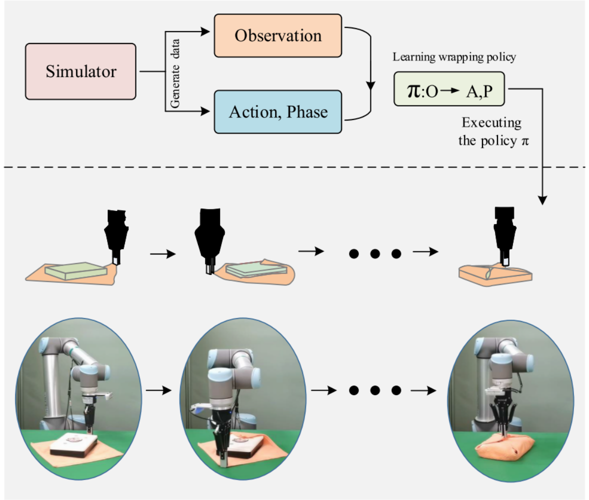

Fig. 1. An overview of using a robot arm to wrap a box. The policy are learned by using a simulated dataset and transferred to a physical robot. Four action sequences are used to realize the task of wrapping a box.

In this study, inspired by Seita *et al.* [7], we address this problem via deep imitation learning. In [7], the pick point and pull vector are used to smooth the fabric. In contrast, the pick-place points and the phases of manipulation were learned in this study, and the phases were used to determine the termination of a task.

A large dataset is necessary for network training. Fortunately, the simulator can provide an effective way to accomplish this task. However, it is difficult for a simulator to fully mimic a real scenario. To transfer the learned policies to a physical robot, we use the same camera domain randomization [8], just like the way used by Seita *et al.* [7]. In the simulation, we adopted the texture of the fabric (similarly considered in [9]) and box from a real environment, which allowed the robot to wrap a box with complex patterns. Figure 1 shows an overview of the wrapping of a box.

In this study, we proposed a method for a robot to learn to manipulate a deformable object mixed with rigid object. We describe the task of folding a piece of fabric or wrapping a box into a target configuration, as shown in Figure 1. The contributions of this study are as follows:

- We proposed an approach for learning folding fabrics or wrapping boxes, which can be used to learn the pick-place point and estimate the phase of the manipulation process. By using domain randomization and texture from a real environment, the learned policies not only can be transferred to a physical robot but also allows the robot to

---

Manuscript received: October, 29, 2021; Revised January, 19, 2022; Accepted February, 21, 2022.

This letter was recommended for publication by Associate Editor Fotios Dimeas and Editor Markus Vincze upon evaluation of the reviewers' comments. This work was supported in part by National Key Research and Development Program of China under Grant 2018YFB1309300, in part by National Natural Science Foundation of China (Grant No. 61873072).

[1] Department of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China (Xiaoman Wang, e-mail: 16b953039@stu.hit.edu.cn, Corresponding author: Xin Jiang, e-mail: x.jiang@ieee.org).

[2] T Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong, China.

Digital Object Identifier (DOI): see top of this page.

wrap a box with complex patterns or various geometric shapes.

- Experiments were conducted to demonstrate that the learned policies are not only effective in performing the tasks of folding fabrics or wrapping boxes on a real robot, but can also be generalized to fabrics or boxes not used in the training.

II. RELATED WORK

Fabric manipulation has received more attention in recent years owing to the improved performance of simulators, artificial intelligence, cameras, etc. There are related works for folding/unfolding in the field of recognition of folded garments configuration as well as in the field of cloth parameters estimation and simulation. In the following paragraphs, we introduce related works in this field in detail.

Fabric manipulation tasks solved by traditional robotics algorithms include folding a fabric [10], [11] or bringing fabric into a desired configuration [12]. Some studies rely on vision techniques, geometric information or planning to realize manipulation tasks. Maitin-Shepard *et al.* [13] proposed a vision-based grasp point detection algorithm to fold a towel using a dual-arm robot. Van Den Berg *et al.* [10] presented the use of gravity and the geometry of garments to fold cloth. Stria *et al.* [14] introduced a new garment polygonal model to fold a cloth using the vision sensing, segmentation, and planning. Hayashi *et al.* [15] proposed to perform a wrapping-with-fabric task using a planning method. In some studies, physics model or simulator were used to manipulate fabric. Petrík *et al.* [16] used a physics model to design a folding path to fold a garment on a low-friction table using a dual-arm robot. Subsequently, they proposed a method that can automatically estimate the material properties for the garment folding task [17]. Li *et al.* [3] introduced a method to find the optimal trajectory for folding a garment based on an offline simulator.

Recently, some studies have focused on performing fabric manipulation using learning methods. Most of these methods obtain the training datasets with a simulator and then apply the learned policies to manipulate the fabric in the real world. Seita *et al.* [7] learned fabric smoothing in a simulation using dataset aggregation [18]. Hoque *et al.* [19] presented VisuoSpatial Foresight (VSF) to learn multi-step fabric smoothing and folding. Wu *et al.* [9] used model-free visual reinforcement learning to spread a rope and a cloth. Some studies have addressed the problem of fabric manipulation using dynamic actions. Jangir *et al.* [20] tackled a dynamic cloth folding and flattening problem in a simulation using deep reinforcement learning. Ha *et al.* [21] proposed an approach named FlingBot to execute cloth unfolding using dynamic flinging actions. Some studies have also aimed at solving the problem of fabric folding. Matas *et al.* [22] used sim-to-real reinforcement learning algorithms to accomplish fabric folding and hanging tasks. Tanaka *et al.* [23] presented a motion planning method to fold a rectangular cloth into a target shape using the EMD Net. Ganapathi *et al.* [24] addressed fabric manipulation using dense object descriptors [25]. Weng *et al.* [26] proposed the FabricFlowNet (FFN) based on flow to complete the fabric folding task. Lee *et al.* [27] learned fabric folding using real-world robot data only. Seita *et al.* [28] used a goal-conditioned transporter network to learn how to rearrange and manipulate deformable objects in the simulation.

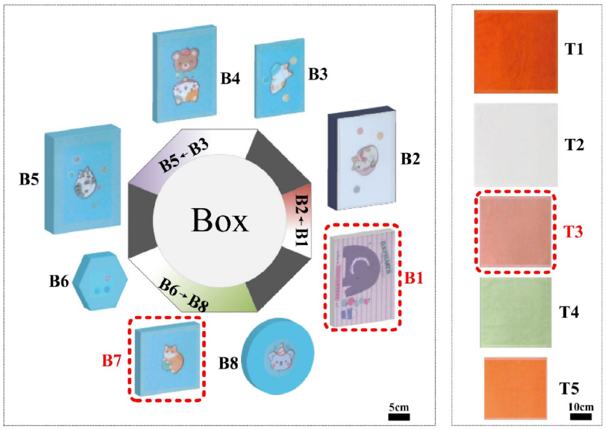

Fig. 2. The fabrics and boxes used in the experiments. The fabric T3 and the boxes B1, B7 are the trained samples, others are the test samples.

Most of the above studies focus on the manipulation of deformable objects when no other rigid objects are mixed together. In contrast, we leverage simulation to learn the manipulation tasks for folding fabric and wrapping the box into a specific shape. In the manipulation process, we evaluate the phase of the manipulation process to determine whether the folding task is completed. By using domain randomization and the texture information of the box from the real environment, the learned policies can be transferred to a physical robot without additional training using real data. This enables us to accomplish the task of wrapping a box with complex patterns.

III. METHOD

A. Problem Statement

In this subsection, we present the formulation of the tasks of folding a fabric or wrapping a box using fabric to a specific shape via learning. Before performing the tasks, the fabric was smoothly and randomly placed on a table. For the convenience, the word *phase* is treated as an instance of folding fabric and wrapping a box using the fabric.

As mentioned in the introduction, we address the above task via imitation learning [29], [7], that is, learning by demonstration. In this work, we denote the demonstration by D, which is constructed using a set of three-dimensional tuples T_3. Each T_3^i, consists of an observation, an action and a phase, and can be written as $T_3^i = (o_i, a_i, p_i)$, at the *i-th* observation. Thus, the demonstration is denoted by $D = \left\{T_3^i\right\}_{i=1}^N$. Considering the elements of T_3^i, o_i represents the RGB image observation of the manipulated object, a_i denotes the action that the robot arm can perform, and p_i denotes the phase of manipulation process, which indicates the status of the task. Thus, the robot arm can execute sequential actions until the final phase appears or the iteration termination threshold is obtained as in [7]. Then, the problem can be intuitively considered as learning policy from the demonstration, and it can be formulated as a mapping from

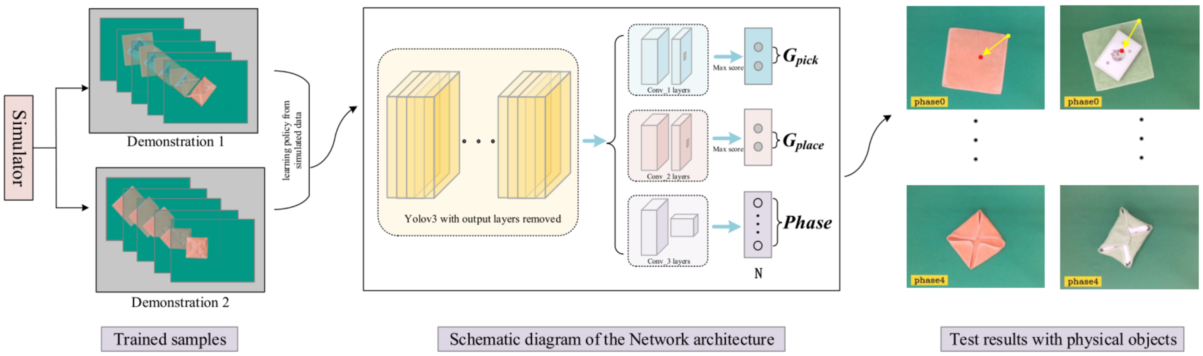

Fig. 3. A complete overview of the approach for learning fabric folding and box wrapping with fabric from demonstration. The left shows example demonstration episodes generated by the simulator. The demonstration mainly includes two tasks, folding fabric and wrapping a box. The middle shows the framework of the extended network. The input to the network is an RGB image and the outputs are the action and phase. The right shows the experimental results based on the learned policies.

O to $\{A,P\}$, denoted by $\pi : \mathrm{O} \rightarrow \{\mathrm{A},\mathrm{P}\}$, in which the capital letters represent the set of its corresponding lowercase letters.

B. Action and Phase

The action consists of two parts: the pick point $G_{pick} = [x_{g1}, y_{g1}]^T$ and the place point $G_{place} = [x_{g2}, y_{g2}]^T$, which are related to the pixel coordinates on the image. Based on the calibration of the camera and robot, the corresponding pick and place position of the end effector of the robot can be obtained. Then, the trajectory of the robot is obtained through Moveit [30]. The action a_t can be denoted as Equation 1.

$$a_t = (G_{pick}, G_{place}) = ([x_{g1}, y_{g1}]^T, [x_{g2}, y_{g2}]^T) \tag{1}$$

The phase represents the step to which the procedure has proceeded. For example, if an entire manipulation process requires $N-1$ sequential steps, it will be divided into N phases (including the initial state) that can be represented by one-hot encoding. In summary, the phase p_t is N dimensional vector which can be denoted as Equation 2.

$$p_t = [q_0, \cdots, q_i, \cdots q_N - 1]^T, t = 0, 1, \cdots, N-1 \tag{2}$$

where $q_i = \begin{cases} 1, & i = t \\ 0, & others \end{cases}$

C. Collision Detection Between Fabric and Box

The fabric simulator we used was based on the open-source code from Seita *et al.* [7]. However, we made several changes to the simulator. First, we added boxes of different shapes to the simulator, such as square, rectangle, cylinder and hexagon, as shown in Figure 2. Second, we also used Blender [31] to render the image, as in [7]. However, we applied textures from the real world to render fabrics and boxes in the simulation. Finally, to make the FOV consistent with the real camera we used, the size $H \times W$ of the simulated image was kept the same as that of the real camera.

When simulating the process of wrapping a box, the fabric may interfere with the box. Therefore, it is necessary to perform collision detection between the box and fabric. Let V be the space configuration of the box. Let $Q = [Q_0, \cdots, Q_i, \cdots Q_n] \subset R^{3\times n}$ be the point set of the fabric. A point Q_i in V, indicates a collision. Then, the point Q_i will pop up to the surface of the box closest to itself.

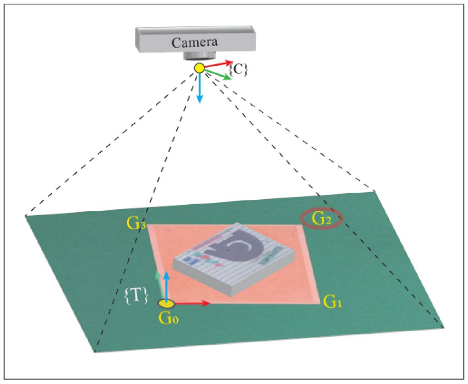

Fig. 4. The schematic diagram of box wrapping in the initial state.

D. Folding and Wrapping

We present the manipulation tasks including fabrics folding and wrapping boxes.

1) *folding task*: In the demonstration of folding task, we consider the fabric corner as G_{pick}. We set the G_{place} near the center $G_{center} = [x_c, y_c]^T$ of the fabric, where $G_{place} = G_{pick} + \lambda(G_{center} - G_{pick})$, and λ is set as 0.8 based on the task. The process repeats for all corners, and the pulling distance is the same. Demonstration 2 of Figure 3 shows an example of folding the fabric into a specified shape.

2) *wrapping tasks*: In the demonstrations of wrapping tasks, considering the difference in folding distance for different objects, we designed two types of boxes: a rectangle and a square. In the initial state, the box was placed on the fabric. The pick point was the fabric corner. A coordinate system $\{T\}$ on the towel was established, as shown in Figure 4. At the start, the upper right corner (G_2) in $\{T\}$ was selected as the first pick point, followed by G_0, G_1, and G_3 sequentially. The place point was determined similarly to the folding task. However, this is related to the position and size of the box. Therefore, we should ensure that there is no relative sliding between the box and the fabric, especially in the first step.

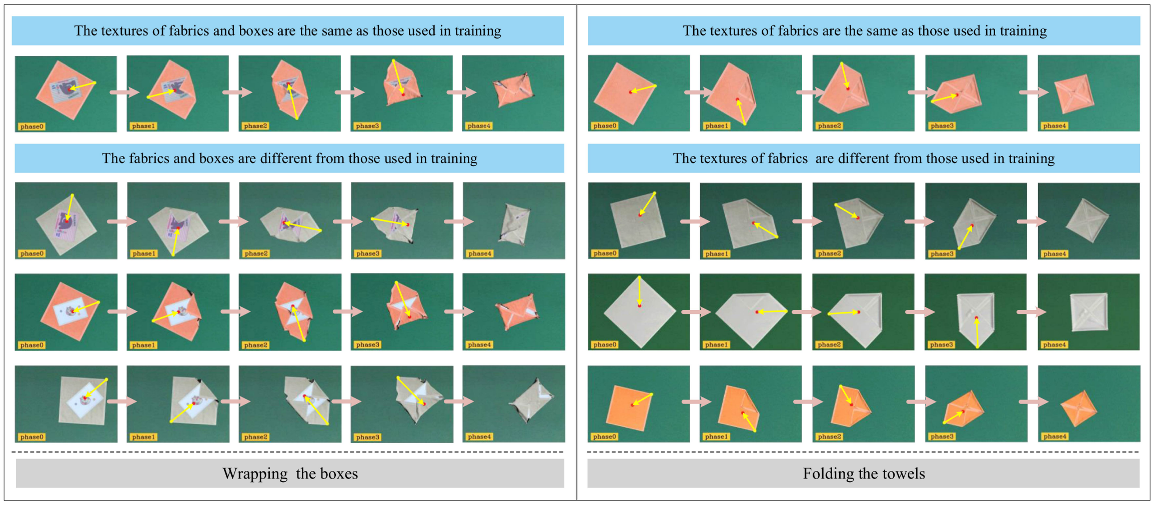

Fig. 5. Results of simulation experiments. We apply the learned policies using a simulator. The experiment shows the test results for folding the fabric and wrapping a box into a specified shape. The first line shows the test results for fabric and box with the same textures as those used in training. The other rows show test results for fabrics and boxes with different textures from those used in training.

TABLE I
EXPERIMENTS RESULTS IN SIMULATION.

Towels	Success of folding fabric				Success of wrapping box (B1)				Success of wrapping box (B2)			
	Phase1	Phase2	Phase3	Phase4	Phase1	Phase2	Phase3	Phase4	Phase1	Phase2	Phase3	Phase4
T2	27/30	27/30	27/30	25/30	30/30	30/30	12/30	11/30	27/30	26/30	23/30	19/30
T3	29/30	29/30	29/30	29/30	30/30	30/30	29/30	29/30	30/30	30/30	30/30	30/30
T4	29/30	29/30	29/30	29/30	30/30	30/30	26/30	24/30	28/30	28/30	28/30	26/30
T5	30/30	30/30	30/30	30/30	×	×	×	×	×	×	×	×

TABLE II
EXPERIMENTS RESULTS IN SIMULATION.

Towels\Box	Success of wrapping boxes with different sizes				Towels\Box	Success of wrapping boxes with different shapes			
	Phase1	Phase2	Phase3	Phase4		Phase1	Phase2	Phase3	Phase4
T1\B3	28/30	28/30	28/30	28/30	T3\B6	30/30	30/30	30/30	28/30
T3\B4	26/30	26/30	24/30	23/30	T3\B7	30/30	30/30	30/30	30/30
T5\B5	25/30	25/30	25/30	25/30	T3\B8	30/30	30/30	30/30	30/30

Demonstration 1 of Figure 3 shows an example of rectangular box wrapping.

E. Data Generation

We demonstrated the manipulation tasks in the simulation: folding fabrics or wrapping boxes to the specific shapes. For the above tasks, we generated 3000 and 6000 (half for the rectangular and square boxes) episodes from the simulator and used them as training images and labels, respectively.

In each episode, the fabric was placed at a random position on the table. To simulate this situation, we changed the pose of the camera and the fabric (position and orientation) in the simulator, using domain randomization. In the demonstration of box wrapping, the variation in the pose of the box is kept pace with the fabric. However, the variety of the camera and fabric positions should not be too large, to avoid the fabric from being beyond the field of view.

To make the visual effect close to that of real-world tablecloth, we collected tablecloth textures in actual as the background color of the fabric in the simulation. In this task, we adopted a towel, whose edge and surface colors were different. To perform the tasks of wrapping boxes with complex patterns, we also added real boxes textures to the simulated boxes. In the process of collecting data, only the fabric T3 and the boxes B1, B7 were used, as shown in Figure 2. The above textures were rendered to the image using the open-source software Blender [31]. To better transfer the learned policies to a real robot, we added noise to the simulation image and changed the brightness of the image, similar to the method used by Seita *et al.* [7].

F. Policy Learning

The robot obtains an RGB image from the camera as observation o_t. We extend the YOLOv3 [32] framework to learn

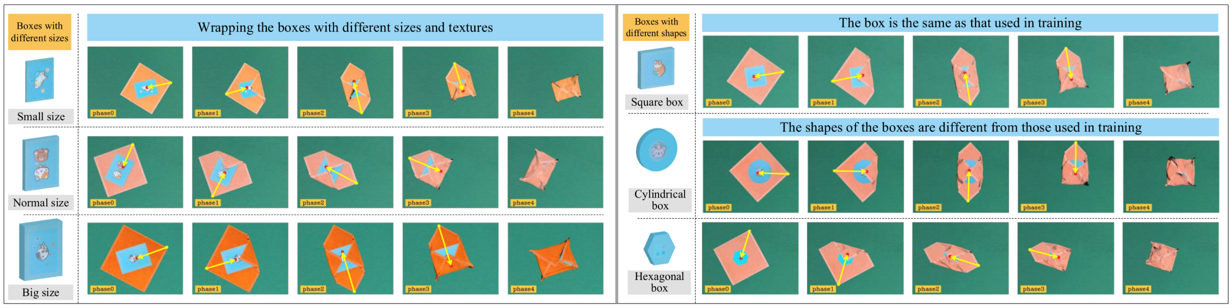

Fig. 6. Results of the simulation experiments for wrapping boxes with different sizes and shapes. The left panel shows the wrapping of the boxes with three sizes. The right panel shows the wrapping of the boxes with three different shapes.

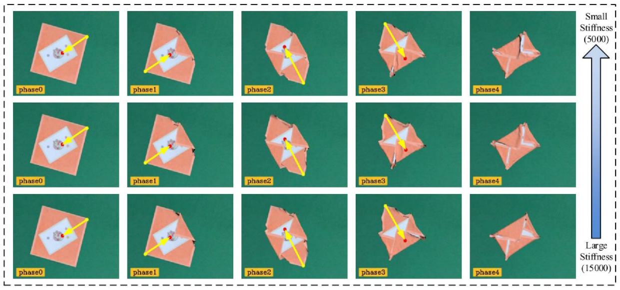

Fig. 7. The simulation experiment results of wrapping box with different stiffnesses.

the manipulation policies: $(\pi : O \rightarrow A, P)$, which maps from the observation o_t to the action a_t and phase p_t. The extended network is based on YOLOv3 [32], which is followed by three network branches: conv_1, conv_2, and conv_3 layers, as shown in Figure 3.

The input of the network is an RGB image with dimensions of $(96 \times 96 \times 3)$. The input image is resized from the simulation image $(480 \times 640 \times 3)$ which is the same as the size of the physical camera. The output of the network includes three grids, with dimensions of $S \times S \times 3$ (position and score), $S \times S \times 3$ (position and score), and $1 \times 1 \times N$. The grid with the highest score corresponds to the pick or place point, and N represents the phase. The training dataset we used was the data collected from the simulator [7]. The policies are trained on the datasets of folding towels, wrapping rectangle boxes, and wrapping square boxes, respectively. We adopted pre-trained weights for the part of YOLOv3 [32] used in the extend network. During the training, we used the mean square error loss for action prediction and the binary cross-entropy loss for phase prediction.

IV. EXPERIMENTS

In this section, we describe the manipulation tasks executed on five types of square fabrics and eight types of boxes with different textures, sizes, and geometric shapes in simulation and physical world. We tested the generalization ability of the learned folding policy or wrapping policies for the towels with different colors, and for the boxes with different textures, sizes, and geometric shapes.

A. Simulation Experiments

In the simulation experiments, we used fabrics with three colors: orange, green, and white. The boxes with different shapes, sizes, and textures are adopted, as shown in Figure 2. The geometric shapes includes rectangle, square, cylinder, and hexagon. In the simulator, each action can be broken into a motion sequence: grasp the pick point, pull up, pull towards the place point, and drop, as in [7]. To make the data different from the training sample, we adjusted some simulation parameters, such as damping and thickness of the fabric. We used domain randomization to set the initial state for the manipulation tasks. Therefore, the fabric can be located at a random position on the table, and the fabric could not exceed the camera field of view.

We carried out ten sets of experiments with various combinations of fabric color and box height. Each set summarized the results of 30 trials. The results of folding and wrapping using boxes B1, B2 in the simulation environment are shown in Table I, where a successful wrapping should meets two conditions: (1) the sequence of predict phase is correct, (2) the box should be packed well like the demonstration. Figure 5 shows examples of folding the fabrics and wrapping the boxes with different heights and textures in the simulation. In addition, we conducted three sets of experiments for wrapping different sizes of boxes. The test results are shown in Table II and Figure 6. The yellow and red dots represent the predicted pick and place positions, respectively. Phase 0 indicates the initial state of the task. Phase $1 \sim 3$ show the manipulation process. Phase 4 represents the final state of the task. The experimental results show that the learned policies can execute sequential manipulation tasks. Moreover, the folding policy has a good generalization ability for towels with different colors. For the wrapping task, the results for the white fabrics were not good. The reason may be that there is a big difference in the distribution of test data and training data. However, the wrapping policy trained only on rectangular boxes of one size, can generalize to boxes with different sizes and textures.

Furthermore, we also considered changing the shapes of the boxes in the simulation experiments. We tested two shapes: a cylinder and a hexagon. We conducted 30 trials for each shape. The test results are presented in Table II and Figure 6. These experiments indicated that the policy trained only on square boxes, can generalize to the boxes with central symmetry. In addition to the above, we also consider the effect of different

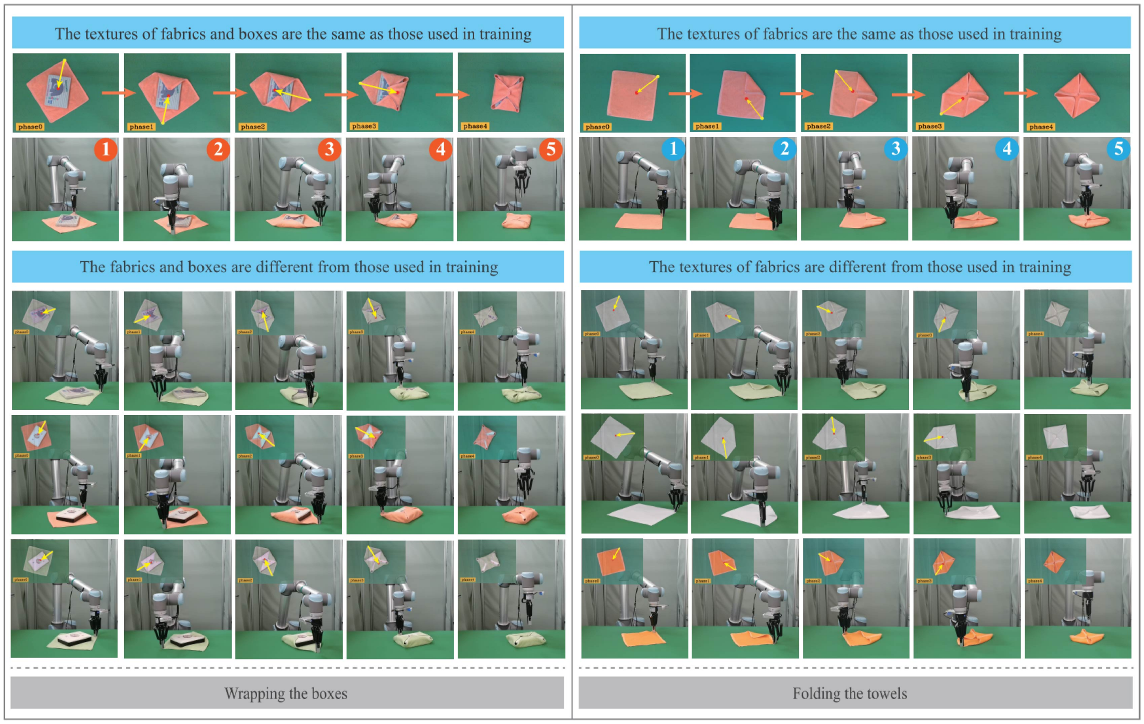

Fig. 8. Results of physical experiments. The left shows the wrapping task using fabrics T3, T4 and boxes B1, B2. The right shows the folding task using fabrics with different colors. The first two rows show the test results for fabrics and boxes with the same textures as those used in training. The other rows represent test results for fabrics and boxes different from those used in training.

TABLE III
EXPERIMENTS RESULTS IN REAL WORLD.

Towels	Success of folding fabric				Success of wrapping box (B1)				Success of wrapping box (B2)			
	Phase1	Phase2	Phase3	Phase4	Phase1	Phase2	Phase3	Phase4	Phase1	Phase2	Phase3	Phase4
T2	9/9	9/9	9/9	7/9	9/9	9/9	0/9	0/9	5/9	3/9	0/9	0/9
T3	9/9	8/9	8/9	8/9	9/9	9/9	9/9	8/9	9/9	9/9	9/9	9/9
T4	9/9	8/9	8/9	8/9	9/9	9/9	9/9	9/9	9/9	9/9	8/9	8/9
T5	9/9	9/9	9/9	9/9	×	×	×	×	×	×	×	×

TABLE IV
EXPERIMENTS RESULTS IN REAL WORLD.

Towels\Box	Success of wrapping boxes with different sizes				Towels\Box	Success of wrapping boxes with different shapes			
	Phase1	Phase2	Phase3	Phase4		Phase1	Phase2	Phase3	Phase4
T1\B3	9/9	9/9	8/9	8/9	T3\B6	9/9	9/9	9/9	9/9
T3\B4	9/9	9/9	9/9	7/9	T3\B7	9/9	9/9	9/9	9/9
T5\B5	9/9	8/9	8/9	8/9	T3\B8	9/9	9/9	9/9	9/9

stiffnesses of fabric for wrapping the same box, as shown in Figure 7. The experiments show that, on the premise of completing the wrapping task, the change of stiffness within a certain range has little effect on wrapping.

B. Physical Robot Experiments

In real-world experiments, we used a ur10 robotic arm equipped with a Robotiq 2-fingered gripper. We collected the RGB image from an Intel RealSense Depth Camera D435i mounted at the end of the arm. Based on the RealSense Camera, we mapped the pixel positions to 3D scenes. The fabric was placed in a random position on the table. To avoid loss of generality, we randomly selected a point K on the table and made the center of the fabric coincide with it. Five different towels and eight different boxes were adopted in the physical experiments. The surface of the boxes was very smooth. To prevent the fabric from sliding off the surface of the box during the manipulation, we pasted anti-skid glue on the side of the box to increase the friction between the box and fabric. The real textures of the fabrics and boxes are shown in Figure 2.

Similar to the simulation experiment, we performed sixteen

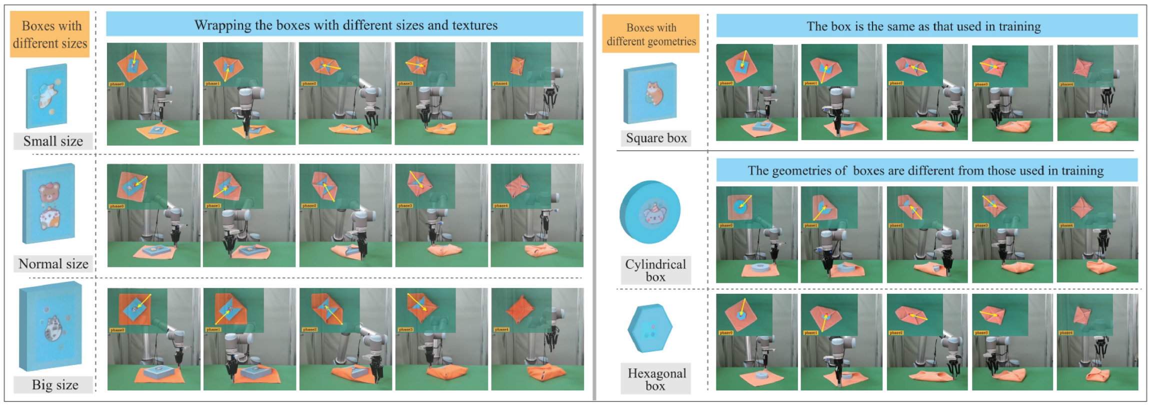

Fig. 9. Results of the physical experiments for wrapping the boxes with different sizes or shapes. The left shows different sizes of boxes and the right shows different shapes of boxes.

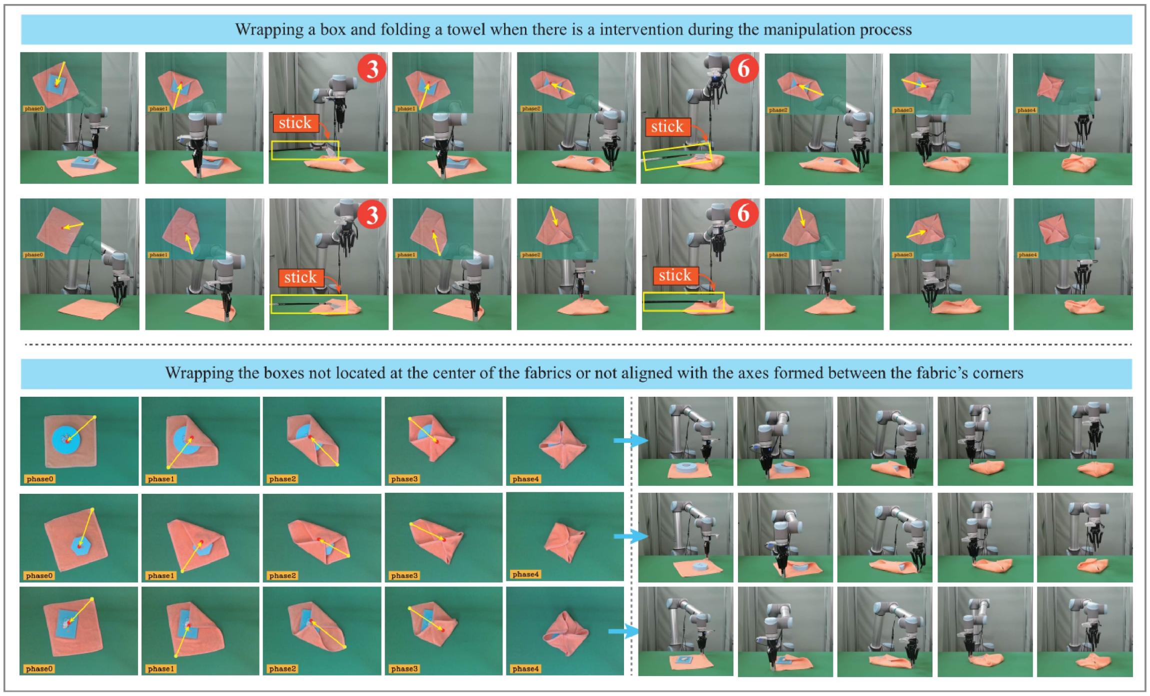

Fig. 10. Results of the physical experiments for wrapping boxes. The first two rows show the test results for wrapping a box when there is a intervention during the manipulation process. The other three rows represent the test results for wrapping the box not located at the center of the fabric.

sets of physical experiments. In the initial stage, the fabric was first placed in the pre-set point K. Then, the fabric was rotated clockwise and counterclockwise for four times around the vertical direction of point K, with an interval angle of 10°. Then, we reported the nine trials as a one-set experiment result. The success of folding the fabric and wrapping a rectangular box with different heights and textures in a real-world setting is shown in Table III. The test results of wrapping rectangular boxes of different sizes are shown in Table IV and Figure 9. In addition, we also verified the generalization ability to the boxes with different geometric shapes. The test results are presented in Tables IV and Figure 9. As indicated in simulation experiment, the learned folding policy has a good generalization ability for fabric with different colors, as shown in Figure 8. The wrapping policies also achieved the similar performances and can be generalized to the wrapping of boxes with different sizes or geometric shapes. The experiments indicate that our method can be directly transferred to a physical robot without training on real data.

In addition to the above tasks, we also considered the situations: there is an intervention during the folding and wrapping process, the boxes are not located at the center of the fabric, or the box is not aligned with the axes formed between the fabric corners. The experimental results are presented in Figures 10. The results show that our method can still recognize the current folding or wrapping phase and predict the action.

V. DISCUSSION

The learned policies sometimes cannot correctly predict the manipulation phase. This will lead to the early termination of the task before it is completed. In addition, the policies sometimes predict unexpected manipulation actions, which can lead to a manipulation state that is different from the training dataset. Owing to the above limitations, this method still relies on human visual information to determine whether the manipulation task is successful. One possible reason is that the properties of the fabric and the box in the simulation are not the same as those in the real world. This may cause the contact state between the fabric and the box to be different from that in the simulation. In future work, we will try to optimize the physical performance of the simulator to make it as close as possible to the real environment. In addition, future work may explore wrapping boxes with irregular geometric shapes.

VI. CONCLUSIONS

We proposed a method for learning fabric folding and wrapping of a box with a simulation dataset. Based on the domain randomization and the texture information from the real word, the learned policies can be transferred to a physical robot without training on real data. The results of the experiment indicate that the policies are effective and practical for folding and wrapping tasks both in the simulation and in the physical robot. When the policies were tested on textures of fabrics and the sizes or geometric shapes of boxes that were not used in training, that showed good generalization ability.

REFERENCES

[1] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2371–2378.

[2] Y. Li, X. Hu, D. Xu, Y. Yue, E. Grinspun, and P. K. Allen, "Multi-sensor surface analysis for robotic ironing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5670–5676.

[3] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6000–6006.

[4] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[5] D. Seita, N. Jamali, M. Laskey, R. Berenstein, A. K. Tanwani, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making," in *2019 International Symposium on Robotics Research (ISRR)*, 2019.

[6] J. Schrimpf and L. E. Wetterwald, "Experiments towards automated sewing with a multi-robot system," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 5258–5263.

[7] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[9] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," in *16th Robotics: Science and Systems, RSS*, 2020.

[10] J. Van Den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," in *Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 409–424.

[11] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[12] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Motion planning for dynamic folding of a cloth with two high-speed robot hands and two high-speed sliders," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5486–5491.

[13] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2308–2315.

[14] J. Stria, D. Průša, V. Hlaváč, L. Wagner, V. Petrík, P. Krsek, and V. Smutnỳ, "Garment perception and its folding using a dual-arm robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 61–67.

[15] N. Hayashi, T. Suehiro, and S. Kudoh, "Planning method for a wrapping-with-fabric task using regrasping," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1285–1290.

[16] V. Petrík, V. Smutnỳ, P. Krsek, and V. Hlaváč, "Physics-based model of a rectangular garment for robotic folding," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 951–956.

[17] V. Petrík, J. Cmíral, V. Smutnỳ, P. Krsek, and V. Hlaváč, "Automatic material properties estimation for the physics-based robotic garment folding," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7449–7454.

[18] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[19] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," in *Robotics: Science and Systems (RSS)*, 2020.

[20] R. Jangir, G. Alenyà, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4630–4636.

[21] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *5th Annual Conference on Robot Learning*, 2021.

[22] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.

[23] D. Tanaka, S. Arnold, and K. Yamazaki, "Emd net: An encode–manipulate–decode network for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.

[24] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali *et al.*, "Learning dense visual correspondences in simulation to smooth and fold real fabrics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 515–11 522.

[25] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 373–385.

[26] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *5th Annual Conference on Robot Learning*, 2021.

[27] R. Lee, D. Ward, V. Dasagi, A. Cosgun, J. Leitner, and P. Corke, "Learning arbitrary-goal fabric folding with one hour of real robot experience," in *Conference on Robot Learning*. PMLR, 2021, pp. 2317–2327.

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

[29] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[30] "Moveit," https://moveit.ros.org/.

[31] B. O. Community, *Blender - A 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[32] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.