CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

CrossMark

ShakingBot: Dynamic Manipulation for Bagging

Ningquan Gu[1] , Zhizhong Zhang[1], Ruhan He*[1] and Lianqing Yu[2]

[1]School of Computer Science and Artificial Intelligence, Wuhan Textile University.
[2]School of Mechanical Engineering and Automation, Wuhan Textile University.
*Corresponding author. E-mail: heruhan@wtu.edu.cn.

Abstract

Bag manipulation through robots is complex and challenging due to the deformability of the bag. Based on the dynamic manipulation strategy, we propose a new framework, ShakingBot, for the bagging tasks. ShakingBot utilizes a perception module to identify the key region of the plastic bag from arbitrary initial configurations. According to the segmentation, ShakingBot iteratively executes a novel set of actions, including Bag Adjustment, Dual-arm Shaking, and One-arm Holding, to open the bag. The dynamic action, Dual-arm Shaking, can effectively open the bag without the need to take into account the crumpled configuration. Then, the robot inserts the items and lifts the bag for transport. We perform our method on a dual-arm robot and achieve a success rate of 21/33 for inserting at least one item across various initial bag configurations. In this work, we demonstrate the performance of dynamic shaking action compared to the quasi-static manipulation in the bagging task. We also show that our method generalizes to variations despite the bag's size, pattern, and color. Supplementary material is available at https://github.com/zhangxiaozhier/ShakingBot.

1. Introduction

Plastic bags are ubiquitous in everyday life, including in supermarkets, homes, offices, and restaurants. Bagging is a helpful skill, including opening a bag and inserting objects for efficient transport. Therefore, handling bagging tasks is meaningful and practical. However, the task is challenging for the robot because of the inherent complexity of thin plastic dynamics and self-occlusions in crumpled configurations.

Prior works concerning deformable object manipulation are considerable. Most of the research focuses on manipulating linear objects [1–5] and fabric [6–10]. Among the most recent manipulation of more deformable objects, such as bags and sacks, Chen et al. [11] proposed the AutoBag algorithm to manipulate a robot to open bags, insert items, lift them, and transport them to a target zone. However, their quasi-static action method required a significant number of interactions.

Dynamic manipulation [12] is a common and efficient action in real life, which can dramatically reduce the complex configuration of the object without taking into account the initial complex state. For example, in our daily lives, people will open the crumpled bag by grasping the bag's handles and shaking their arms up and down to get air into the plastic bag without considering the crumpled configurations of the bags. Then, we consider applying the dynamic manipulation strategy to the bagging task with a dual-arm robot.
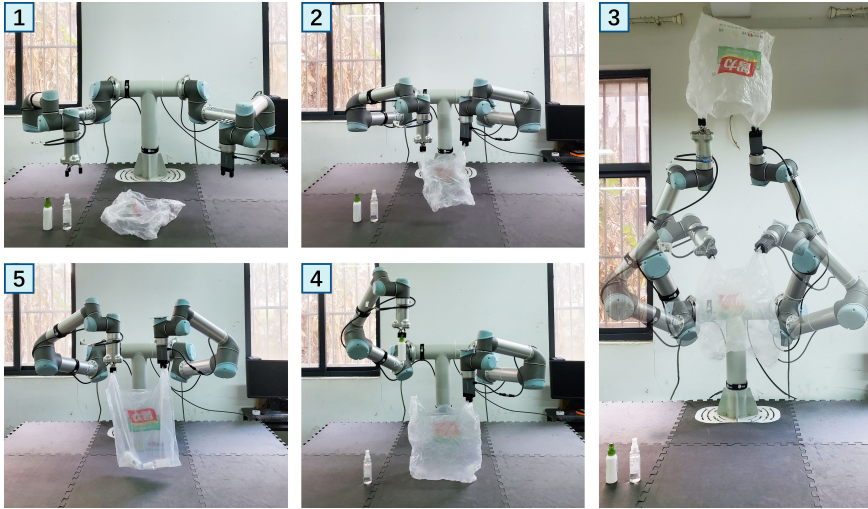
Figure 1. Opening the bag with dynamic actions. *(1) Initial highly unstructured bag and two solid objects. (2) Through region perception, the robot grasps the two handles and adjusts the distance between the two arms. (3) The arms shake the bag at high speed according to the pre-defined trajectory, which makes the air into the bag. (4) One arm holds the opened bag on the workspace. (5) The two arms lift the bag filled with the inserted items .*

Reinforcement Learning and Learning from Demonstration have demonstrated success in cloth manipulation tasks [13–17]. However, they often require substantial data and are frequently trained in simulation. It's more challenging for plastic bags to build dynamic models and collect accurate datasets whether from simulation or real-world environments. Therefore, we utilize the idea of most visual policy learning approaches [11, 18] to guide the manipulation with action primitives.

In this work, we focus on bagging tasks, including opening the bag and inserting items, where the key goal is to maximize the opening area of the bag. An ideal bagging approach should be:

- Efficient: The approach should reach enough opening area with a few actions from arbitrarily crumpled initial configurations to insert items.
- Generalizable: The algorithm should generalize to different colors, patterns, and sizes.

To achieve this goal, we present a new framework, ShakingBot, for manipulating a plastic bag from an unstructured initial state so that the robot can recognize the bag, open it, insert solid items into it, and lift the bag for transport. Given an initial highly unstructured bag and two solid objects, we train a perception module to recognize the key regions of the bag, where we reduce the workload through the colored bags to get the training dataset. After grasping the recognized handles, the robot iteratively executes a novel set of actions, including Bag Adjustment, Dual-arm Shaking, and One-arm Holding, to open the bag. Through dynamic action, we can be very effective in opening the bag. When the bag opening metric exceeds a threshold, ShakingBot proceeds to the inserting item stage. The simplicity of ShakingBot, combined with its superior performance over quasi-static baselines, emphasizes the effectiveness of dynamic manipulation for bagging tasks. See Figure 1 for ShakingBot in action on a dual-arm robotic (consisting of two UR5 robotic arms).

In summary, the main contributions of this paper are:

1. Based on the dynamic manipulation strategy, we propose a new framework, ShakingBot, for the bagging tasks, which improves the success rate and efficiency compared to the quasi-static strategy.
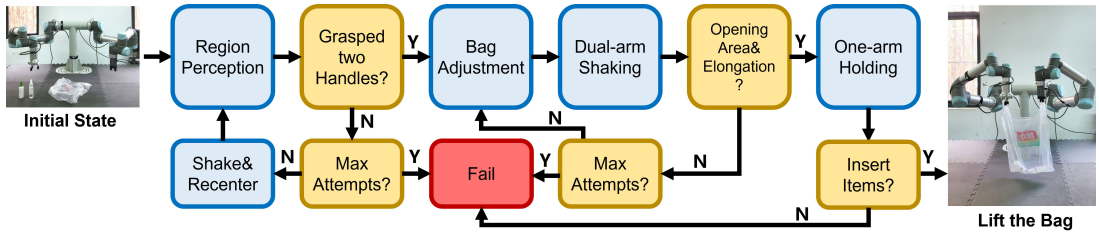
Figure 2. ***Overview of ShakingBot.*** *See the left of the figure. The robot starts with an unstructured bag and two items. As shown in the flow, ShakingBot opens the bag according to the steps shown (see Section 4.4 for details). When the bag opening metric exceeds a certain threshold, ShakingBot proceeds to the item insertion stage. If the robot lifts the bag with all the items inside, the trial is a complete success.*

2. We design a novel set of action primitives for dynamic manipulation, including Bag Adjustment, Dual-arm Shaking, and One-arm Holding, which make it possible to apply dynamic manipulation in bagging tasks.
3. Experiments to demonstrate that ShakingBot generalizes to other bags with different sizes, patterns, and colors, even without training on such variations.

2. Related work

2.1. Deformable object manipulation

Deformable object manipulation remains challenging for robots, primarily due to the complex dynamics and the limitless set of possible configurations. There have been significant prior works in the manipulation of controlling ropes [1, 3, 19], cables [2, 4, 5, 20, 21], smoothing fabric [8–10, 22] and folding cloth [6, 7, 13, 23]. However, these methods cannot be generalized to increased complexities of deformable objects, such as bags and sacks. For example, Weng et al. [6] employed the optimal-flow method to predict the movement of each fabric particle to calculate pick-and-place points. Yet, their method can only handle one-step manipulation and requires knowledge of the object's next state. It is difficult to know the comprehensive information regarding the state of the bag in each step and to predict the movement of each bag particle. Mo et al. [7] proposed a sequential multi-step approach with space-time attention for cloth manipulation. However, their approach focuses on predicting intermediate states during multi-step manipulation with limited configuration, which cannot be applied to arbitrary configurations of a bag. On the other hand, the early research work of bag-shaped objects focused on the mechanical design of robots capable of grasping [24] or unloading [25] sacks. Some studies paid attention to zip bags [26] in constrained setups or supposed that a sturdy, brown bag was already open for item insertion, as with a grocery checkout robot [27]. In recent work, Gao et al. [18] proposed an algorithm for tying the handles of deformable plastic bags. While their approach modeled bags with a set of 19 manually labeled key points, it is challenging to estimate the representation with a bag in highly unstructured configurations. Chen et al. [11] proposed an AutoBag algorithm to open the bag and insert items into the bags, where the bags began empty and in unstructured states. They defined a novel set of quasi-static action primitives for manipulating bags. However, their approach was not validated on bags with different colors and required many actions to open the bag. In contrast, we focus on the bagging tasks, which can generalize to different colors, and we are able to execute fewer actions to finish the task.

2.2. *Dynamic manipulation*

As opposed to quasi-static manipulation, dynamic manipulation [12] uses the acceleration forces produced by robots to manipulate objects. By building up momentum with high-velocity actions, the system can manipulate out-of-contact regions of the deformable object. A great deal of progress has been made in the field of deformable dynamic manipulation. Examples include high-speed cable knotting [28, 29], cable manipulation with fixed endpoints using learning-based approaches [1] or a free endpoint [30, 31]. Recently, Ha and Song proposed the FlingBot, which learned to perform dynamic fling motions with a fixed parameter to smooth garments using Dual-UR5 arms. Their approach was firstly trained in SoftGym simulation [32] and fine-tuned in the real world to get the learned grasp points with two UR5 robots. The results suggested significant efficiency benefits compared to quasi-static pick-and-place actions. However, their research object is fabric, and their action primitives are not suitable for bagging tasks. In terms of bag manipulation, Xu et al. [33] produced high-speed interactions using emitted air in order to open the bag. However, they required more equipment. They used a setup with three UR5 robots to control two grippers and a leaf blower. In contrast, we apply dynamic manipulation to the bagging tasks with less equipment. By executing a novel set of dynamic action primitives with two UR5 robots, we can achieve high performance from crumpled initial bag configurations.

2.3. *Learning for deformable object manipulation*

Concerning learning methods for deformable object manipulation, Learning from Demonstration (LfD) and Reinforcement Learning (RL) have demonstrated success in cloth manipulation tasks [13–16]. For instance, Seita et al. [34] employed deep imitation learning to achieve sequential fabric smoothing, while Canberk et al. [35] utilized reinforcement learning to unfold and smooth cloth. However, it is important to note that these methods often require a substantial amount of data, which can be challenging to collect in real-world environments due to factors such as wear and tear on the robot and limitations in available observation methods. As a result, they are frequently trained in simulators [32, 36, 37]. When it comes to manipulating plastic bags, building dynamics models is challenging due to the complex non-linear dynamics involved. Besides, obtaining accurate datasets for plastic bag manipulation proves particularly difficult, regardless of whether they are obtained from simulations or real-world environments. This challenge is exacerbated by the thinness of the bags and their interactions with the surrounding air. Considering these challenges, we chose a more direct approach for our bagging task [11, 18]. Leveraging visual policy learning approaches to detect the key points of the bag and adopt the method of action primitives to tackle the task. In order to incorporate dynamic manipulation into the bagging task, we design action primitives to overcome the potential challenges that may arise during dual-arm manipulation. Moreover, we conduct ablation experiments to demonstrate the effectiveness of these primitives.

3. Problem statement

The bagging task is formulated as follows: First, a bag is placed on a flat surface with random configurations. Second, we need to open the bag. Lastly, we insert n items and lift the bag for transport.

We use the most commonly used plastic bag, the vest plastic bag, as our research object. We define two labels for a bag, "handle" and "rim" (see the right of Figure 3). The "handle" is the point where to grasp by the robot. The rim surrounds the opening area, and its opening orientation is determined by the direction of the outward-pointing normal vector from the plane formed by the opening. The larger the opening area, the easier it is to put items in. In random configurations, the direction and area of the opening are various. It is assumed that the bag's initial state is unstructured: deformed, potentially compressed. The handles and rim of the bag may be in a partial or fully occluded configuration. We need to manipulate the bags with different colors, patterns, and sizes. See the left of Figure 3 for an illustration. We do not consider the case of a brand-new bag without any deformation and wrinkles, the

Figure 3. *Left: Various plastic bags adopted to train and test the region perception module. The bags include different sizes, patterns, and different colors. Right: A bag with red paint on its handles and green paint around its rim. The paint color can be changed into others according to the pattern color of the bag .*

configurations of a brand-new bag are almost known, and the two sides of the bag stick to each other tightly without any space. This is an extremely special case.

We consider a bimanual robot (consisting of two UR5 robotic arms) with grippers and a calibrated camera above a flat surface. Let $\pi_\theta : \mathbb{R}^{W \times H} \to \mathbb{R}^2$ be a function parameterized by θ that maps a depth image $\mathbf{I}_t \in \mathbb{R}^{W \times H}$ at time t to an action $\mathbf{a}_t = \pi_\theta(\mathbf{I}_t)$ for the robot. In order to simplify the question, we constrain the experimental area $\mathbb{R}^2$ to be within reach of the robot. We assume a set of rigid objects O, placed in known poses for grasping. We are interested in learning the policy π such that the robot can open the bag. After opening the bag, we insert the objects into the bag. Lastly, we lift the bag off the table while containing the objects for delivery.

4. Approach

4.1. Method overview

Bagging aims to open the bag from an arbitrarily crumpled initial state and insert the object. Concretely, this amounts to maximizing the opening area of the bag on the workspace to make it easier to put objects in. It is intuitive that dynamic actions can appropriately make use of the airflow through a high-velocity action to open the bag, and then it can achieve high performance on bagging tasks.

Figure 4. Pipeline for our method: *The perception module takes depth images and outputs segmentation masks for the bag handles and rim. The robot grasps the key points and executes dynamic actions. Last, the robot inserts the items and lifts the bag .*

Figure 4 provides the overall pipeline of our method. We propose a learned region perception module to recognize the bag handles and the rim (Section 4.2). We define a novel set of action primitives (Section 4.3), including Bag Adjustment, Dual-arm Shaking, and One-arm Holding, for dynamic manipulation.

We then describe the ShakingBot framework (visualized in Figure 2) to open the plastic bag and insert objects (Section 4.4).

4.2. Region perception module

In this module, we utilize semantic segmentation to identify important regions of the bag, including the grasp points and the opening area. We define the handles of the bag as the grasping points, while the rim of the bag is used to calculate the opening area. The neural network is trained by the depth images of the scene containing the bag. By predicting semantic labels for each pixel point, the network gives the probability that the pixel contains the handles and rim of a bag. Lastly, we can obtain semantic segmentation masks of the bag.

In this paper, we employ DeeplabV3+ [38] as the segmentation algorithm for our region perception module. The rationales behind this choice are demonstrated in Section 5.2.

DeepLabV3+ [38] is an advanced semantic image segmentation model and the latest version in the DeepLab series. It consists of two parts: the Encoder and the Decoder, see Figure 5. The Encoder part consists of a DCNN backbone network and an Atrous Spatial Pyramid Pooling (ASPP) module in a serial structure. The DCNN backbone, which utilizes the ResNet model, is responsible for extracting image features. The ASPP module processes the output of the backbone network using a 1x1 convolution, three 3x3 dilated convolutions, and a global pooling operation. The input to the Encoder is the depth image, and it produces two outputs. One output is directly passed to the Decoder, while the other output is processed through the ASPP module and then concatenated. A 1x1 convolution is applied to compress the features and reduce the number of channels. The Decoder part takes the intermediate output of the DCNN backbone and the output of the ASPP module and transforms them into the same shape. These outputs are then concatenated and further processed through a 3x3 convolution and upsampling operations. The final result is obtained after these operations.
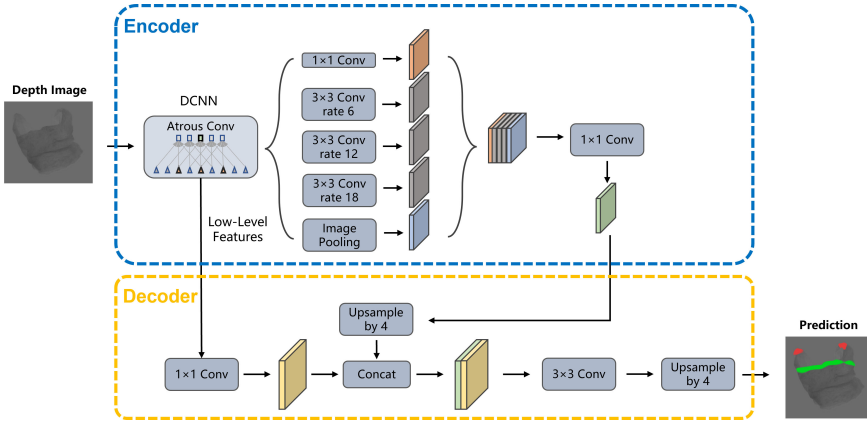

Figure 5. The architecture of Deeplabv3+ .

In our region Perception module, we define the loss $\mathcal{L}(o, t)$ to be the mean of the point-wise binary cross-entropy loss for every class k, $k \in K$:

$$\mathcal{L}(o, t) = \frac{1}{K} \sum_{k}^{K} (-\frac{1}{N} \sum_{i} w_i * (t[i] * \log(o[i]) + (1 - t[i]) * \log(1 - o[i]))) \tag{1}$$

Where $o[i]$ is the prediction segmentation at the image location i for the class K, while the $t[i]$ is the ground truth. We add a weight w to deal with the problem of imbalanced distribution between the positive and negative labels.

As we showed in Section 4.4, the perception allows us to grasp the handle and calculate the opening area.

4.3. Action primitive definition

In order to achieve an efficient, generalizable, and flexible bag-opening system, we propose that the system contains two arms that can operate in a dynamic action space. Additionally, we address the potential challenges associated with dual-arm bagging manipulation, such as ensuring sufficient airflow into the bag, preventing bags from sticking together, maximizing the dual-arm robot's range of movement, and preventing the opened bag from collapsing during item insertion. To overcome these challenges, we introduce a novel set of action primitives, including Bag Adjustment, Dual-arm Shaking, and One-arm Holding. These novelly defined action primitives enable dynamic manipulation in bagging tasks. The primitives are as follows.

4.3.1. *Bag Adjustment* $(d, \Delta d, k_s, l, f)$

After grasping the handles, the two grippers maintain the position left and right symmetrical. We decrease the initial distance d between the two grippers by Δd. This can increase the chances of getting more air into the bag during the dynamic shaking action, which is beneficial to enlarge the bag. But, if the distance between the arms is less than the threshold, the action doesn't execute. After the action, the robot performs k_s times swinging movements with length l and frequency f in the horizontal right and left directions. It can separate the two layers of the rim, which is beneficial for preventing them from sticking to each other. (See Figure 6)
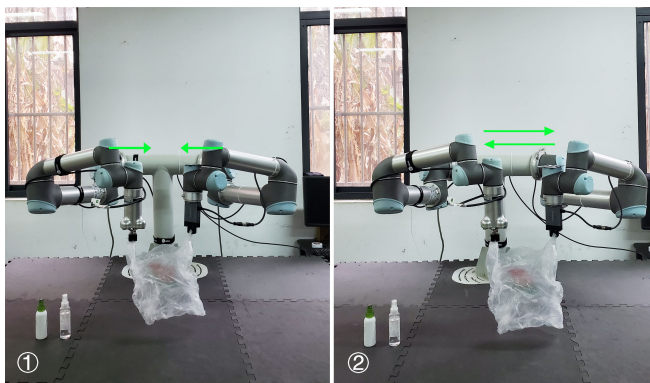
Figure 6. **Bag Adjustment:** ①*The two arms decrease their distance.* ②*Robot swings the bag in the horizontal left and right directions .*

The Bag Adjustment primitive could increase the success rate of subsequent Dual-arm Shaking attempts to enlarge the bag more efficiently.

4.3.2. *Dual-arm Shaking* (H, H', v)

We define the dynamic shaking action as follows: The dual arms shake the bag dramatically from the current position to the height of H at v velocity in the vertical direction, then pull down to the height of H' with the same velocity. The direction of the grippers should change from the downward direction

at the initial position to the $45°$ direction from the horizontal at the highest position. When the grippers return to the height of H', the direction of the grippers returns vertically downward. It can make better use of the bag's acceleration and movement space by changing the direction of the grippers. We set the $H=1.4$m to give the bag plenty of movement distance. It is beneficial for making more air into the bag to enlarge the opening. The H' should be higher than the bag's bottom to prevent the bag from touching the surface. The RGBD camera can detect the height of the bag's bottom. The setting of primitive could make full use of the movement space to open the bag during shaking. (See Figure 7)

Figure 7. Dual-arm Shaking: ① *Preparation for dynamic dual-arm shaking.* ②*Long-distance and high-speed shaking.* ③ *An enlarged bag .*

4.3.3. *One-arm Holding* (h)

The dual arms, which grasp the bag, move down to the height of h, and the bottom of the bag just touches the workspace. The h is determined by the height of the bag's bottom. After that, the right gripper releases the handle and moves away. The workspace surface and the left gripper support the bag together, which can prevent the bag from deforming itself when inserting items. (See Figure 8)
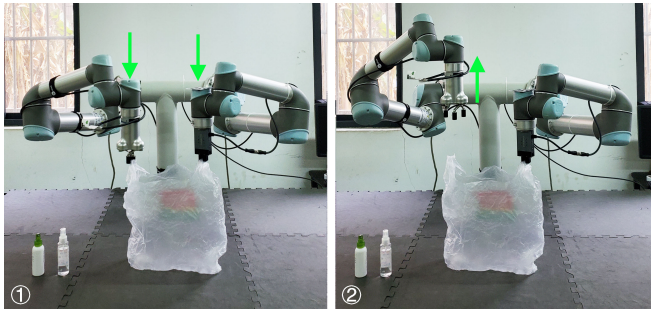
Figure 8. One-arm Holding: ①*Move the bag to touch the surface of workspace.* ② *The gripper releases the right handle while the bag keeps its shape .*

In addition to the above action primitives, we also adopt some primitives from [11], including Shake and Recenter primitives. The difference between our Dual-arm Shaking and their Shake is that our method is a dual-arm action with a significant movement distance in the vertical direction to get air into

the bag. In contrast, their action is to rotate one of the robot's wrists side by side to expand the bag's surface area.

4.4. ShakingBot: bagging task

Firstly, the ShakingBot utilizes the perception module to recognize the positions of the handles to grasp them. Secondly, it chooses actions to execute according to the bag opening metrics [11]. Finally, the ShakingBot inserts the items into the bag and lifts it. See Figure 2 for an overview. The ShakingBot consists of the following three steps:

4.4.1. Grasping the handles

We input the depth image to the region perception module and get the predicted handle segmentation. The center of each handle is the grasp point. If the two handles region of the bag can be recognized, the robot grasps the two key points directly. Otherwise, the robot executes a Shake. The Shake action can expand the area of the bag, and its grasp point is obtained from the handle region (if handles are not visible, we select anywhere on the bag as the grasp point). If the bag is not in the center of the camera frame, we execute the Recenter action. After these actions, we execute region perception and grasping action again. If the two handles still can not be grasped, the robot repeatedly executes the above actions. The grasping height is set to the height of the workspace in order to ensure that the handles can be grasped successfully. The grasp position coordinates are specified as Cartesian coordinates in pixels.

4.4.2. Opening the bag

After grasping the two handles of the bag, the two grippers move the bag to a fixed height with a pre-set distance between the two grippers. And the bag is still in a crumpled configuration. The robot applies Bag Adjustment and Dual-arm Shaking primitives to enlarge the bag iteratively. During each iteration, our algorithm observes an overhead image of the bag to estimate its rim position and adopts the two opening metrics, Normalized convex hull area A_{CH} and Convex hull elongation E_{CH} [11], to evaluate the enlarged results. Repeating this process until the normalized convex hull area reaches a threshold A_{CH} value and the elongation metric falls below a threshold E_{CH} value, which means that the opening is large enough for inserting objects.

4.4.3. Inserting and lifting

The robot performs the One-arm Holding action to place the bag and estimates the openings by fitting convex hulls on rims. We divide the opening spaces by the number of objects. Later, the gripper grasps the objects that are placed in known poses, and the robot places them in the center of each divided region. After performing these actions, the robot identifies the position of the released handle and executes the grasping action. In the end, the robot lifts the bag from the table.

5. Experiments

In training and evaluation experiments, the bags we use are of size 25-35cm by 40-53cm when laid flat. The color includes red and white with different patterns. (see Figure 3). The flat workspace has dimensions of 120cm by 180cm.
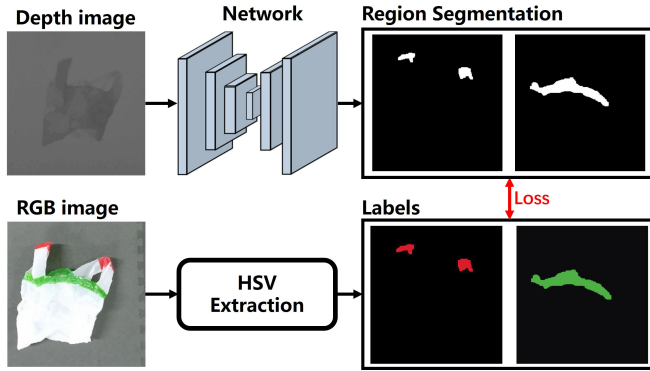
5.1. Data collection and processing for training

In order to train the region perception module, it is necessary to have a dataset that includes labels for the handles and rim. However, labeling these regions in images with crumpled bags is challenging and

would require a significant amount of human annotation effort. To address this challenge, we adopt an approach similar to [39, 40], where different colors of paint are used to mark the objects.

In our work, we use the marker pen to mark the handles and rim of the bag with different colors. It should be noted that the colors cannot be similar to the pattern color of the bag. Moreover, the marker pen is very friendly to the plastic bag because it can be cleaned by alcohol very easily. We utilize the red and green colors to mark the bag. Of course, We can use the blue and orange colors to mark the bag, but we only need to modify the corresponding color parameter of the HSV. Figure 9 depicts the data collection process and how applied for training.

We utilize a Microsoft Kinect V2 camera, which can capture RGB-depth information. The camera is positioned above the workspace, providing a top-down perspective. We collect the data, including RGB pictures and Depth images, from four training bags by taking 2500 images each (resulting in 10,000 total). Our dataset includes bags in various configurations, such as lying on the workspace or being grasped by the robots. We aimed to capture a wide range of volume and orientation configurations to ensure the accuracy and robustness of our perception model.


*Figure 9. **Data collection process and its utilization for training** Handles and rims are labeled in different colors. By color-labeling, the ground truth for the network is provided without the need for expensive human annotations. The network receives a depth image as input, while output is the region segmentation results. We can get the optimized segmentation network by calculating the loss between the results and the labels .*

5.2. Comparison of segmentation network

To optimize the performance of the region perception module, we conduct a comprehensive analysis of various segmentation algorithms. The algorithms compare in this study include U-Net [41], PSPNet [42], SegNet [43], and DeeplabV3+ [38]. The U-Net serves as the perception module in one of the baselines, AutoBag [11], discussed in Section 5.5.

We ensure that all four algorithms are configured with the same parameters and trained on the same dataset. During training, we employ a batch size of 32 and initialize the learning rate to 1e-3. To enhance the diversity of our training data, we apply data augmentation techniques such as random image flipping by 50 percent chance, scaling by 50 percent chance, and rotation by 50 percent chance within the range of [-30 degrees, 30 degrees].

For the purpose of model training, we adopt an 80-20 train/validation split. All training procedures are conducted on a Ubuntu 18.04 machine equipped with 4 NVIDIA GTX 2080 Ti GPUs, a 3.50 GHz Intel i9-9900X CPU, and 128 GB RAM.
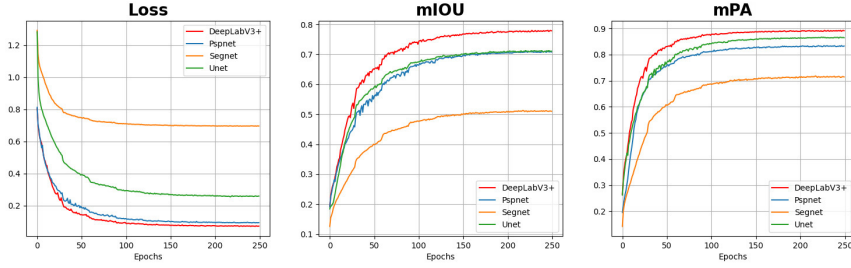
Figure 10. *Comparing the Learning Curves of Four Region Segmentation Models. After 250 epochs, the DeeplabV3+ model achieves remarkable results in our task. It obtains the lowest loss value among all models, with a mean intersection over union (mIOU) of* 78.0%. *Additionally, it demonstrates superior performance in terms of mean Pixel Accuracy (mPA), achieving a score of* 89.4%, *surpassing the other segmentation models in the experiment. These results highlight the performance of DeeplabV3+ compared to the other models .*

The results depicted in Figure 10 clearly demonstrate the superior performance of DeeplabV3+ compared to the other three segmentation algorithms in our task.

However, the training time for the four networks varies, as shown in Table 1. Among them, SegNet has the shortest training time, followed by U-Net and PSPNet. The limitation of DeeplabV3+ is that it requires the longest training time due to its more complex architecture and a larger number of parameters.

Table 1. *The Training Time of the Four Models.*

Metrics	U-Net	PSP-Net	SegNet	DeeplabV3+
Time (Hour)	34	35	27	43

Considering these comparisons, we choose to employ the DeeplabV3+ algorithm for our region perception module. This decision is primarily based on the accuracy of detection, which is the main metric we prioritize.
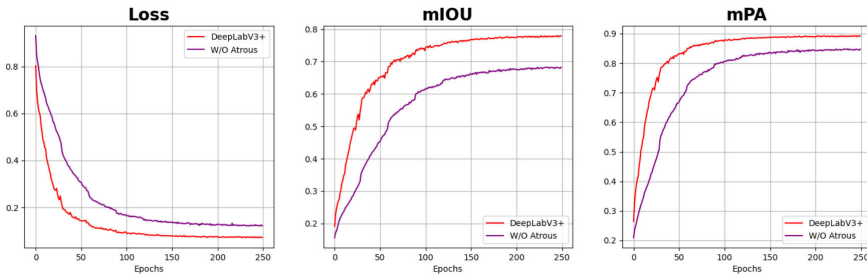
Figure 11. *The Learning Curves of DeeplabV3+ without Atrous Convolution. After 250 epochs, the loss value of the model without atrous convolution has increased compared to the original DeeplabV3+.The mean intersection over union (mIOU) has decreased from 78.0% to 68.2%, while the mean Pixel Accuracy (mPA) decreased from 89.4% to 83.8% .*

In addition, we analyze the reasons why DeeplabV3+ can achieve the best performance in our task. According to the work of Chen et al. [38], DeepLabV3+ utilizes atrous convolution to merge the coarse features from the lower layers with the fine features from the higher layers, thereby enhancing the

segmentation results in terms of edges and details. To further evaluate the effectiveness of the atrous convolution in our task, we conducted an additional experiment that excluded the atrous convolution, as shown in Figure 11.

5.3. *Action primitive parameters setting*

In the experiment, we set the parameters of the action primitive. Table 2 shows the meaning of each parameter and how they can be obtained. It should be noted that the optimal values may vary depending on the particular configuration of the robot and the operating environment.

***Table 2.*** *The Action Primitive Parameters Setting.*

Action Primitive	Parm.	Meaning	Setting
Bag adjustment	d	The initial distance between the two grippers after moving to the dangling	Pre-set according to the size of the bags
	Δd	The amount of change in the distance between the two grippers each time	Pre-set fixed values according to site configuration
	k_s	The times of horizontal swinging movement	Pre-set fixed values according to site configuration
	l	The horizontal length of swinging movement	Pre-set fixed values according to site configuration
	f	The frequency of swinging movement	Pre-set fixed values according to site configuration
Dual-arm shaking	H	The maximum height that the robot arm can reach during the shaking action	Pre-set according to the capability of the robot. In general, the larger the value, the better
	H'	The height of grippers after the robot shaking action	Pre-set fixed value and the value should be higher than the bag's bottom to prevent the bag from touching the surface
	v	The speed of shaking action	Pre-set according to the max capability of the robot. In general, the larger the value, the better
One-arm holding	h	The height at which the dual arms descend until the bottom of the bag comes in contact with the workspace	Real-time control based on detecting the lowest point in the middle of the bag through the RGB-D camera

5.4. *Experiment protocol*

To evaluate ShakingBot, we use four bags, one of which is the middle-size bag from training. The other three bags include an unseen-size bag, a different pattern bag, and a pure red color bag. The goal is to insert 2 identical bottles into each bag(see Figure 1). In our definition of a trial, the robot attempts to perform the entire end-to-end procedure: opening a bag, inserting n items into it, and lifting the bag (with items). We allow up to $T = 15$ actions before the robot formally lifts the bag. The robot will

be reset to its home position if it encounters motion planning or kinematic errors during the trial. We evaluate the ShakingBot with three difficulty tiers of initial bag configurations (see Figure 12):

- Tier 1: The two handles of the bag can be recognized, and the bag has an expanded, slightly wrinkled state lying sideways on the workspace. Besides, the rim has an opening. This requires reorienting the bag upwards.
- Tier 2: The tier is similar to tier 1, but the rim area and the degree of wrinkle. The rim area is smaller than tier 1. This will need more actions to enlarge the bag.
- Tier 3: There are one or two handles hidden. Another, it has a more complex initial configuration. This requires some actions to expand the bag.
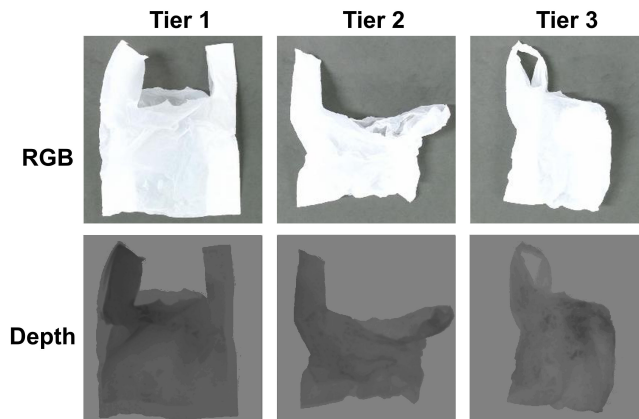
Figure 12. ***Three tiers of initial bag configurations.*** *The first row shows the RGB images, while the second shows the corresponding depth images .*

At the start of each trial, we perform the initialization process, which involves manipulating the bag so that it falls into one of the tiers. The trial is considered successful ("Full Succ."), if the robot lifts the bag off the surface for at least two seconds while containing all items. It is a partial success ("Partial Succ.") if the robot lifts at least one object into the bag during the trial. In the report, we describe the number of times the robot successfully opens the bag ("Open Bag"), the number of objects that the robot correctly places in the bag opening before lifting ("Placed"), the number of actions which the robot executes ("Actions"), and the time the robot costs ("Time").

5.5. *Approach comparison*

We compare ShakingBot to two baselines and two ablations on three tiers. To evaluate the dynamic shaking action, we compare it against the quasi-static baseline AutoBag [11], which is the state-of-art in bagging tasks. They also used a region perception module, of which architecture is U-Net [41], to recognize the key region of the bag. While they only used the quasi-static action to manipulate the bag, we adopted the dynamic shaking action. In the comparison experiment, we utilize AutoBag with Depth images, similar to our training dataset. To evaluate the region segmentation module, we use an analytical method to detect the handles and rim. The analytical method consists of two parts: Harris [44] is used to detect the handles, and Canny [45] is used to calculate the rim area. Subsequently, we execute the action primitives based on the detection results obtained from the analytical method. We test these baselines for Tier 1, 2, and 3 configurations.

Table 3. *Results of ShakingBot and Baseline Methods.*

Tiers	Method	Open Bag	Placed	Partial Succ.	Full Succ.	Actions	Time (Sec.)
	Analytic&Primitives	2/8	0.4±0.7	2/8	1/8	N/A	N/A
Tier 1	AutoBag	6/8	1.3±0.9	6/8	3/8	8.2±3.9	195.7±38.8
	ShakingBot	**7/8**	**1.6±0.7**	**7/8**	**6/8**	**7.2±1.2**	**178.5±15.7**
	Analytic&Primitives	1/8	0.3±0.7	1/8	1/8	N/A	N/A
Tier 2	AutoBag	4/8	0.9±0.9	4/8	3/8	10.8±2.6	219.0±31.8
	ShakingBot	**6/8**	**1.5±0.9**	**6/8**	**6/8**	**8.8±2.6**	**193.6±29.5**
	Analytic&Primitives	0/8	0.0±0.0	0/8	0/8	N/A	N/A
Tier 3	AutoBag	1/8	0.3±0.7	1/8	1/8	14.4±1.5	258.6±17.2
	ShakingBot	**4/8**	**0.6±0.8**	**3/8**	**2/8**	**12.5±5.1**	**227.1±47.9**

We report results on the middle-size bag from the training bag in Table. 3, where we run 8 trials per experiment setting. ShakingBot can achieve both a partial success rate and a full success rate on the three respective tiers. Compared with the traditional analytic method, the analytical way is insufficient for the bagging task. It often fails to detect the right grasping points and fails to calculate the position to place items, especially in complex initial configurations. Compared with the quasi-static AutoBag, our method can achieve better results with fewer actions and less time, demonstrating dynamic manipulation's efficiency and performance in the bagging tasks.

We additionally analyze the failure modes of ShakingBot. Except for the inaccuracy in the prediction and planning errors, there are other failure modes. For example, when inserting items, the right gripper grasping the items may touch the opened bag, causing the bag to deform, resulting in failure to put the items in the bag.

5.6. Ablations

To evaluate the utility of the Bag Adjustment and One-arm Holding primitives, we perform two ablations. ShakingBot-A, where the robot does not perform Bag Adjustment action after grasping the two handles. ShakingBot-H, where the robot does not perform One-Arm Holding to hold the bag, instead releasing one of the handles in the air without the support of the desk surface. We perform these ablations for Tier 2 configuration.

Table 4. *Results of Ablations.*

| Method	Open Bag	Placed	Partial Succ.	Full Succ.
ShakingBot-A	2/6	0.8±1.1	2/6	2/6
ShakingBot-H	0/6	0.0±0.0	0/6	0/6

The results are shown in Table 4. The ablations underperform the full method, demonstrating that Bag Adjustment (Section 4.3.1) as well as One-Arm Holding (Section 4.3.3) help the bagging tasks.

5.7. Generalization

In this part, we adopt three bags for the experiments, an unseen size bag, a different pattern bag, and a pure red color bag. Table 5 presents that ShakingBot can attain 5/9 partial success and 4/9 full success rate. Our approach can generalize to the bag with different patterns and colors because it only takes depth

Table 5. *Results of ShakingBot on the Bags with Different Sizes, Patterns, and Colors.*

Types	Open Bag	Partial Succ.	Full Succ.
Unseen-size	2/3	2/3	1/3
Unseen-pattern	1/3	1/3	1/3
Red Color	2/3	2/3	2/3
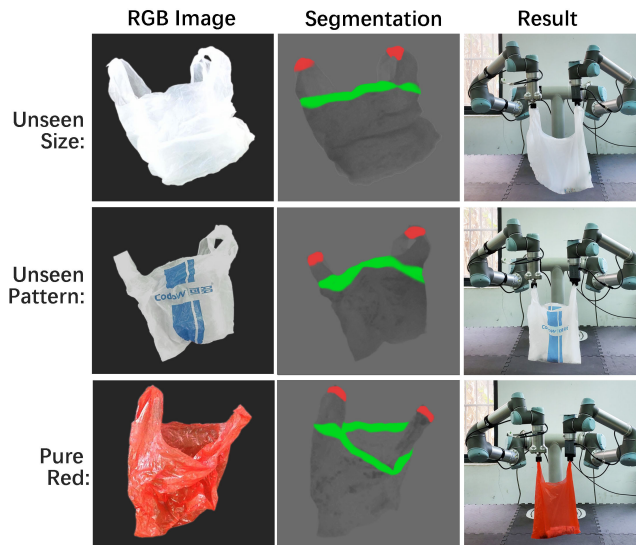
Figure 13. Generalization. *Our network is able to generalize to different sizes, patterns, and colors .*

as input. It can also perform on the bag of different sizes due to its fully convolutional architecture. See Figure 13 for visualizations of these generalization experiments.

6. Conclusion and future work

In this paper, we propose a novel framework, ShakingBot, for performing bagging tasks, including physical bag opening and item insertion. To realize the tasks, we design a novel set of action primitives for dynamic manipulation. We demonstrate the effectiveness of our method in various bagging tasks, showing its ability to generalize to different sizes, patterns, and colors. We believe that our method, including our proposed action primitives, can serve as a valuable guide for handling other similar bag-shaped objects or deformable containers. In the future, we plan to expand our research from plastic bags to encompass other deformable objects.

References

[1] H. Zhang, J. Ichnowski, D. Seita, J. Wang, H. Huang, and K. Goldberg, "Robots of the lost arc: Self-supervised learning to dynamically manipulate fixed-endpoint cables," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4560–4567, IEEE, 2021.

[2] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[3] A. Wang, T. Kurutach, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," in *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019* (A. Bicchi, H. Kress-Gazit, and S. Hutchinson, eds.), 2019.

[4] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8282–8289, IEEE, 2022.

[5] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects withenvironmental contacts," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2019.

[6] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *Conference on Robot Learning*, pp. 192–202, PMLR, 2022.

[7] K. Mo, C. Xia, X. Wang, Y. Deng, X. Gao, and B. Liang, "Foldsformer: Learning sequential multi-step cloth manipulation with space-time attention," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 760–767, 2022.

[8] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," in *The International Symposium of Robotics Research*, pp. 36–51, Springer, 2022.

[9] 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 XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020* (M. Toussaint, A. Bicchi, and T. Hermans, eds.), 2020.

[10] X. Lin, Y. Wang, Z. Huang, and D. Held, "Learning visible connectivity dynamics for cloth smoothing," in *Conference on Robot Learning*, pp. 256–266, PMLR, 2022.

[11] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, and K. Goldberg, "Autobag: Learning to open plastic bags and insert objects," *CoRR*, vol. abs/2210.17217, 2022.

[12] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1, pp. 152–159, IEEE, 1993.

[13] J. Hietala, D. Blanco-Mulero, G. Alcan, and V. Kyrki, "Learning visual feedback control for dynamic cloth folding," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1455–1462, IEEE, 2022.

[14] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," in *Robotics: Science and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020* (M. Toussaint, A. Bicchi, and T. Hermans, eds.), 2020.

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

[16] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, "Cloth manipulation using random-forest-based imitation learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2086–2093, 2019.

[17] 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*, pp. 2317–2327, PMLR, 2021.

[18] C. Gao, Z. Li, H. Gao, and F. Chen, "Iterative interactive modeling for knotting plastic bags," in *Conference on Robot Learning*, pp. 571–582, PMLR, 2023.

[19] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2146–2153, IEEE, 2017.

[20] H. Nakagaki, K. Kitagi, T. Ogasawara, and H. Tsukune, "Study of insertion task of a flexible wire into a hole by using visual tracking observed by stereo vision," in *Proceedings of IEEE international conference on robotics and automation*, vol. 4, pp. 3209–3214, IEEE, 1996.

[21] H. Nakagaki, K. Kitagaki, T. Ogasawara, and H. Tsukune, "Study of deformation and insertion tasks of a flexible wire," in *Proceedings of International Conference on Robotics and Automation*, vol. 3, pp. 2397–2402, IEEE, 1997.

[22] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*, pp. 24–33, PMLR, 2022.

[23] N. Gu, R. He, and L. Yu, "Defnet: Deconstructed fabric folding strategy based on latent space roadmap and flow-based policy," *arXiv preprint arXiv:2303.00323*, 2023.

[24] H. Kazerooni and C. Foley, "A robotic mechanism for grasping sacks," *IEEE transactions on automation science and engineering*, vol. 2, no. 2, pp. 111–120, 2005.

[25] A. Kirchheim, M. Burwinkel, and W. Echelmeyer, "Automatic unloading of heavy sacks from containers," in *2008 IEEE International Conference on automation and logistics*, pp. 946–951, IEEE, 2008.

[26] R. Hellman, C. Tekin, M. Schaar, and V. Santos, "Functional contour-following via haptic perception and reinforcement learning," *IEEE Transactions on Haptics*, vol. 11, no. 1, pp. 61–72, 2018.

[27] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, "Grasping with application to an autonomous checkout robot," in *2011 IEEE international conference on robotics and automation*, pp. 2837–2844, IEEE, 2011.

[28] J. E. Hopcroft, J. K. Kearney, and D. B. Krafft, "A case study of flexible object manipulation," *The International Journal of Robotics Research*, vol. 10, no. 1, pp. 41–50, 1991.

[29] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot planning from observation," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3, pp. 3887–3892, IEEE, 2003.

[30] S. Zimmermann, R. Poranne, and S. Coros, "Dynamic manipulation of deformable objects with implicit integration," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4209–4216, 2021.

[31] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *CoRR*, vol. abs/2203.00663, 2022.

[32] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, pp. 432–448, PMLR, 2021.

[33] Z. Xu, C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Dextairity: Deformable manipulation can be a breeze," *arXiv preprint arXiv:2203.01197*, 2022.

[34] 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)*, pp. 9651–9658, IEEE, 2020.

[35] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation," in *International Conference of Robotics and Automation (ICRA)*, 2022.

[36] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016.

[37] 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.

[38] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[39] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making," in *The International Symposium of Robotics Research*, pp. 275–290, Springer, 2019.

[40] J. Qian, T. Weng, L. Zhang, B. Okorn, and D. Held, "Cloth region segmentation for robust grasp selection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9553–9560, IEEE, 2020.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[42] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[43] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[44] C. Harris, M. Stephens, *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.

[45] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.