Learning to Open Deformable Bags with a Bimanual Robot

Lawrence Chen

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-111
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-111.html

May 16, 2025

Acknowledgement

The work in this thesis is a result of substantial collaborations with Baiyu Shi, Daniel Seita, Roy Lin, and Ayah Ahmad, among others. I thank Professor Ken Goldberg for the guidance. This research was conducted at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab and the CITRIS “People and Robots” (CPAR) Initiative. I gratefully acknowledge the support of Toyota Research Institute, the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. 2146752, and NSF CAREER Award IIS-2046491, all of which helped make this work possible.

Learning to Open Deformable Bags with a Bimanual Robot

by

Lawrence Yunliang Chen

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Ken Goldberg, Chair
Koushil Sreenath

Spring 2025

Learning to Open Deformable Bags with a Bimanual Robot

by Lawrence Yunliang Chen

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

DocuSigned by:
32B42AC2D1E3498...

Professor Ken Goldberg
Research Advisor

5/16/2025

(Date)

* * * * * * *

Signed by:
462CBEA95A8845C...

Professor Koushil Sreenath
Second Reader

5/16/2025

(Date)

Learning to Open Deformable Bags with a Bimanual Robot

Abstract

Learning to Open Deformable Bags with a Bimanual Robot

by

Lawrence Yunliang Chen

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Ken Goldberg, Chair

Deformable bag manipulation is a useful capability for robotic systems with applications such as grocery automation, packaging, recycling, and household assistance. However, bags are uniquely challenging due to their 3D structure, complex and unstable deformations, and difficult visual properties such as translucency and specularity. This thesis investigates the problem of autonomous robotic *bagging*—opening a bag from an unstructured initial state and inserting objects using bimanual manipulation.

The thesis presents two main systems [9, 8]. First, *AutoBag* introduces a self-supervised learning pipeline in which a dual-arm robot learns to detect semantic features of plastic bags, such as handles and rims, by training on UV-labeled data. The system uses these models at test time to manipulate and open plastic bags for object insertion. In experiments, a YuMi robot using AutoBag achieves a success rate of 16/30 insertions across various bag configurations.

Second, *SLIP-Bagging* builds on the insight that opening bags often requires isolating the top layer. This system proposes SLIP (Singulating Layers using Interactive Perception), an algorithm that uses iterative visual feedback to separate the top layer of a bag using standard grippers and RGB cameras. Applied to the bagging task, SLIP-Bagging significantly improves success rates and generality across plastic and fabric bags, achieving 67% to 81% success across varied bag types. Experiments also demonstrate that SLIP generalizes to other tasks such as singulating layers of folded cloth or garments.

Together, these systems demonstrate the viability of general-purpose bimanual robotic bagging using only RGB (or RGBD) perception, parallel-jaw grippers, and data-driven learning. This thesis contributes new algorithms, evaluation metrics, and empirical results toward the broader goal of deformable object manipulation in unstructured environments.

Contents

Contents **i**

List of Figures **iii**

List of Tables **vi**

1 Introduction **1**

2 AutoBag: Learning to Open Plastic Bags and Insert Objects **3**
2.1 Introduction . . . 3
2.2 Related Work . . . 5
2.3 Problem Statement . . . 7
2.4 Method . . . 8
2.5 Physical Experiments . . . 12
2.6 Results and Failure Modes . . . 14

3 Bagging by Learning to Singulate Layers Using Interactive Perception **16**
3.1 Introduction . . . 16
3.2 Related Work . . . 19
3.3 Problem Statement . . . 20
3.4 SLIP: Singulating Layers using Interactive Perception . . . 21
3.5 SLIP-Bagging . . . 23
3.6 Physical Experiments . . . 26

4 Conclusion **30**
4.1 Summary . . . 30
4.2 Limitations and Future Work . . . 30

Bibliography **32**

A AutoBag: Learning to Open Plastic Bags and Insert Objects **40**
A.1 Action Primitives Implementation Details . . . 40
A.2 Details of Grasp Point Selection . . . 40

A.3 Details of Perception Model 41
A.4 Experiment Details 42
A.5 Failure Modes for AutoBag 44
A.6 Failure Modes for Baselines 45

List of Figures

2.1 AutoBag. (1) Initial highly unstructured and deformed bag. (2) After a sequence of manipulation steps, the robot orients the bag upward and opens the bag. (3) The robot inserts 2 items into the bag. (4) The robot lifts the bag filled with the inserted items, so it is ready for transport. 4
2.2 The physical setup with the ABB YuMi. During training, we use 6 UV LED lights and 1 regular LED light. We use a Realsense RGBD camera placed overhead the robot. **Left**: The workspace under regular lighting. **Right**: The workspace and a painted bag under UV lighting. The painted rim and handle regions of the bag look normal under regular lighting but glow under UV lights. The UV lights are only used during training, not during execution time. 6
2.3 **Left**: 5 plastic bags. The first 4 are used to train the perception module (Sec. 2.4). Bags 1 and 5 (test bag) are used in experiments (Sec. 2.5). **Right**: Bag 3 painted with green UV paint on its handles and red UV paint around its rim, under UV lighting. 6
2.4 Overview of AutoBag. The robot starts with an unstructured bag with objects, shown to the left. Given this setup, AutoBag follows the procedure shown in the flow chart to first orient the bag upward and then enlarge the bag opening (see Section 2.4 for details). If the bag opening metric exceeds a threshold, then AutoBag proceeds to the item insertion stage (Section 2.4). A trial is a full success if the robot lifts the bag with all items in it. *Not shown in the figure: In the implementation, at the initial state, the robot will also check whether there is already an initial opening so it does not unnecessarily perform a **Compress** and **Flip** and potentially ruin a good state. 7
2.5 The action primitives in this work. See Section 2.4 for details. For each primitive, we show a small two-frame overview of it in action. The project website contains full videos of all primitives. 8

2.6 Perception Pipeline. The robot collects images by autonomously manipulating a bag into diverse configurations. It extracts segmentation labels of bag handles and rims through color thresholding from the images under UV light. We use this data to train a semantic segmentation network. During execution time, the perception model takes in an RGB image and predicts the segmentation (white: handles; light gray: rim; dark gray: other parts of the bag; black: background). We fit a convex hull to the predicted rim region (indicated by the red boundary), and compute the area and elongation of the convex hull to quantify the bag opening (Section 2.4). We overlay an ellipse with the major and minor axes computed by PCA on the convex hull. 10

3.1 SLIP-Bagging. **Top 2 rows:** (1) Initial unstructured and deformed bag. (2) The robot flattens the bag, and then (3) uses SLIP to grasp the top layer of the bag. The robot rotates the bag by 90° and inserts objects into the bag. (4) The robot lifts the bag filled with the inserted items, so it is ready for transport. **Bottom 2 rows:** (A) Initial configuration of a piece of folded cloth. (B) The robot uses SLIP to grasp the top layer of a folded square cloth. (C) After grasping the top layer, the robot lifts the cloth up. (D) After shaking, the cloth is successfully expanded. 17

3.2 Top: Training bags. Bottom: Test bags. See Section 3.6 for more details and Table 3.1 for results. 20

3.3 Examples of SLIP in action. Each row shows an example of one cyclic, triangular trajectory T in an iteration ("iter" in Algorithm 1) where one gripper moves the bag while the other one pins the bag. **Top row** a third-person view of the robot. **Next three rows**: top-down RGB camera views of one cyclic trajectory for different trials. They show, respectively, a 0-layer, 1-layer, and 2-layer grasp on a plastic bag. We provide zoomed-in versions of images in the third column to see the layers in more detail. See Section 3.4 for details. 22

3.4 SLIP-Bagging Algorithm. (1) The robot starts with an unstructured bag with objects on the side. (2) SLIP-Bagging then flattens the bag, and (3) uses SLIP to grasp the top layer of the bag, followed by (4) insertion and (5) bag lifting. A trial is a full success if the robot lifts the bag with all items in it. 24

3.5 Distribution of the number of layers grasped for different grasp heights for 4 different bags. 25

3.6 Non-bag experiments. Left to right: Folded cloth, dress, hat. The robot pins at the region indicated by the green point and attempts to grasp a single layer at the yellow star region. 28

A.1 Learning curves of the perception model (Section 2.5). **Left**: Training loss. **Right**: IOU on the validation set. 42

A.2 Examples of the prediction of the perception model on the validation dataset. For each of the three columns, the left "GT" are the ground truth labels, and the right "Pred" are the model predictions. Both are overlaid on top of the color images, with green indicating handles and red indicating rim. 43
A.3 Positive and negative example predictions of bag segmentation and illustrations of the metrics (convex hull and elongation) applied to the predicted rim. 44

List of Tables

2.1 Physical experiment results of AutoBag, ablations, and baseline methods using a training bag across 3 different tiers. 14
2.2 Results of AutoBag on the test bag, across the first two tiers. We report similar evaluation metrics as in Table 2.1. 14

3.1 Physical experiment results of SLIP-Bagging compared with baselines. 6 trials were run on each of the 8 bags (Fig. 3.2) for each method. Each trial attempts to insert 6 rubber ducks, and "% Objects Inserted" is the average percentage of ducks inserted and remaining in the bag after bag lifting. **PD**: Perceived-Depth baseline. **HG**: Handle Grasping baseline. **AB**: AutoBag. **SB**: SLIP-Bagging. The "Single-layer Grasp" column for HG refers to grasping the handle associated with the top layer for handbags whose handles overlap (and thus is not applicable to plastic bags and drawstring bags). See Section 3.6 for failure mode categories. 26
3.2 Non-bag experiments. The middle 3 columns show the (multi-class) recall of the video classification model trained on bags and tested on garments without finetuning. The last column shows the success rate of grasping a single layer using SLIP with the classification model. 28

Acknowledgments

I would like to express my sincere gratitude to my advisor, Professor Ken Goldberg, for his invaluable guidance, mentorship, and support throughout the course of these projects. His insights and encouragement have been instrumental to my growth as a researcher.

I am also deeply thankful to my co-authors—Baiyu Shi, Daniel Seita, Roy Lin, and Ayah Ahmad—whose collaboration and contributions made this thesis possible. I extend special thanks to Richard Cheng, Thomas Kollar, David Held, Justin Kerr, Roy Lin, and Kaushik Shivakumar for their thoughtful feedback and suggestions, and to Ryan Burgert for kindly providing the rubber ducks used in our experiments.

This research was conducted at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab and the CITRIS "People and Robots" (CPAR) Initiative. I gratefully acknowledge the support of Toyota Research Institute, the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. 2146752, and NSF CAREER Award IIS-2046491, all of which helped make this work possible.

Chapter 1

Introduction

Opening deformable bags and inserting objects into them is a useful capability for robots in a wide range of applications, including grocery shopping, home cleaning, recycling, and packaging in retail and industrial settings. However, this task is highly challenging for robotic systems due to the physical properties and visual characteristics of typical bags.

Deformable thin objects such as plastic or fabric bags exhibit infinite-dimensional state spaces and nonlinear dynamics. While prior research has explored manipulation of 1D linear deformables such as ropes [79, 67, 38], cables [65], and elastic beams [13, 86], or 2D deformables like fabrics [40, 74, 80, 62], gauze [69], and paper [51, 23], there has been relatively little work on the manipulation of 3D deformables such as bags.

Bags introduce unique challenges: their 3D structure and elastoplastic materials allow for more diverse and unpredictable deformations, and they often include features like handles that deform differently depending on their pose. They are extremely lightweight, and moving one part of the bag often results in the entire bag shifting without a meaningful change in the opening. Additionally, many bags are reflective, translucent, or transparent, making perception particularly difficult. The ideal state for object insertion—a bag standing upright with an open rim—is inherently unstable. An alternative is to insert items while the bag lies flat, but this requires isolating only the top layer of the bag, a difficult task without tactile sensing.

This thesis studies robotic systems for autonomous bagging: opening a deformable bag from an unstructured initial state and placing objects inside. We focus on using bimanual robots with standard parallel-jaw grippers, overhead RGB (or RGBD) perception, and self-supervised learning pipelines. The thesis consists of two parts:

- Chapter 2: *AutoBag*, which focuses on manipulating and opening thin plastic bags using self-supervised learning from UV-labeled data and novel metrics for bag opening.
- Chapter 3: *SLIP-Bagging*, which introduces an interactive perception approach to singulate layers for precise opening of various bag types, including plastic and fabric bags.

Each chapter describes a separate system that contributes toward the larger goal of enabling general-purpose bimanual robotic bagging.

Chapter 2

AutoBag: Learning to Open Plastic Bags and Insert Objects

Thin plastic bags are ubiquitous in retail stores, healthcare, food handling, recycling, homes, and school lunchrooms. They are challenging both for perception (due to specularities and occlusions) and for manipulation (due to the dynamics of their 3D deformable structure). We formulate the task of "bagging:" manipulating common plastic shopping bags with two handles from an unstructured initial state to an open state where at least one solid object can be inserted into the bag and lifted for transport. We propose a self-supervised learning framework where a dual-arm robot learns to recognize the handles and rim of plastic bags using UV-fluorescent markings; at execution time, the robot does not use UV markings or UV light. We propose the AutoBag algorithm, where the robot uses the learned perception model to open a plastic bag through iterative manipulation. We present novel metrics to evaluate the quality of a bag state and new motion primitives for reorienting and opening bags based on visual observations. In physical experiments, a YuMi robot using AutoBag is able to open bags and achieve a success rate of 16/30 for inserting at least one item across a variety of initial bag configurations. Supplementary material is available at `https://sites.google.com/view/autobag`.

2.1 Introduction

In this chapter, we formulate "*bagging*"—manipulating a plastic bag from an unstructured initial state so that a robot can open it, insert solid objects into it, and then lift it for transport. We use overhead RGB images for perception and a bimanual robot. We propose a novel pipeline for bagging that trains a perception model to segment the bag rim and handles through self-supervised data collection. This involves the robot systematically exploring the bag state space by manipulating a bag annotated with ultraviolet (UV) labels [70]. At test time, we deploy the learned segmentation model on bags without UV labels and evaluate the bag opening using novel metrics. We present a novel algorithm, AutoBag, for opening

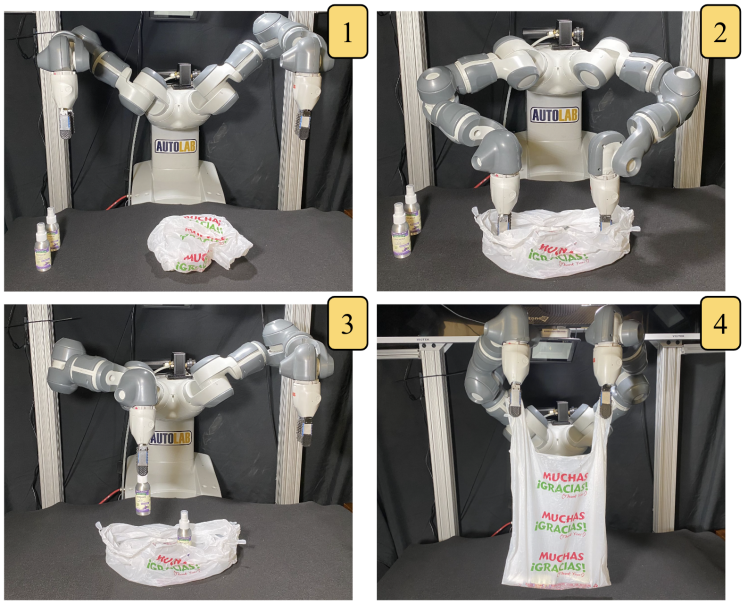

Figure 2.1: AutoBag. (1) Initial highly unstructured and deformed bag. (2) After a sequence of manipulation steps, the robot orients the bag upward and opens the bag. (3) The robot inserts 2 items into the bag. (4) The robot lifts the bag filled with the inserted items, so it is ready for transport.

and inserting items into bags. See Figure 2.1.

This chapter makes 6 contributions:

1. A novel problem formulation for "bagging;"
2. A novel set of primitive actions for manipulating bags including shaking, compressing, flipping, and dilation;
3. A self-supervised data collection process where a robot efficiently explores its state space to manipulate bags into diverse configurations to enable recognizing the handles and rim of bags from UV-fluorescent markings;
4. Two metrics that quantify the opening of the bag based on the convex hull area and elongation;
5. The AutoBag algorithm for bagging;
6. An implemented system with experimental results achieving a success rate of 53.3% for inserting at least 1 object over 30 physical trials.

2.2 Related Work

Deformable Object Manipulation

Deformable object manipulation remains challenging for robots [58, 6, 84]. Typical reasons include the complex dynamics and the infinite set of possible configurations. As a rough categorization, deformable object manipulation can be divided into tasks that involve 1D, 2D, or 3D objects.

Manipulation of 1D deformable objects refers to manipulation of items such as cables [49, 50, 65, 85, 35], ropes [48, 83, 73], and other items which can largely be defined by a single linear component. These are used in tasks such as knotting [47, 26] or untangling [72, 20]. Manipulation of 2D objects refers to items such as clothing and fabrics, as studied in recent work on fabric smoothing [7, 54, 27, 63, 24, 74, 37, 77, 39], which often measuring quality using coverage. A smooth fabric with high coverage may make it easier to later do folding, another canonical task explored in prior work [43, 18, 34, 2]. Assistive dressing [14, 15] encompasses a third set of tasks utilizing 2D deformables.

Manipulation of 3D deformable objects adds another dimension. One type of 3D deformable manipulation involves volumetric 3D objects, including plush toys, sponges, and dough. Prior work has studied manipulation of these items to target configurations [66, 55, 44, 36]. A second type of 3D deformable manipulation refers to objects typically held in containers, as in manipulation of liquids [59] and granular media [60, 10, 45], which may require scooping policies [21]. Other references to 3D deformable manipulation refer to thin surfaces arranged in complex 3D patterns, such as plastic bags, which is the main focus of this work.

Manipulating Deformable Bags

One prior direction in robot manipulation of bags is on the mechanical design of robots suitable for grasping [31] or unloading [32] large sacks. Another direction has provided insights on manipulating knotted bags [28, 29] or bags in highly constrained setups, such as with work on closing ziplock bags [25] or using fully opened, stable paper grocery bags [33]. Recently, DextAIRity [78] used air to efficiently expand bags. They used a setup with three UR5 robots, where two grip the bag and the third manipulated a leaf blower in free space. In this work, we consider a bimanual robot with standard parallel-jaw end-effectors for a highly deformable plastic bag with handles, which the robot has to grasp, open, and then insert items inside for transport.

Seita et al. [61] proposed several deformable object manipulation benchmark tasks in simulation that include a similar problem setup of opening and inserting items into a bag, and then lifting and moving the bag. Transporter Networks [82] are proposed but only evaluated in simulation [11]. Similarly, Weng et al. [75] study modeling and interacting with bags purely in simulation. A large sim-to-real gap in both visual and dynamic properties still needs to be overcome for a policy trained in simulation to transfer to real. Some research

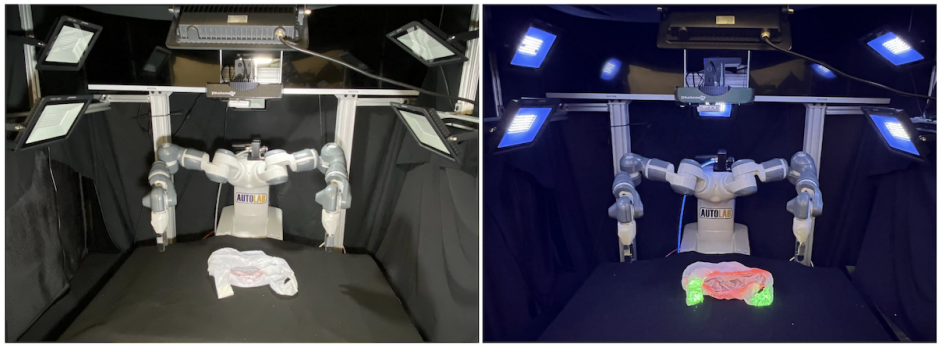

Figure 2.2: The physical setup with the ABB YuMi. During training, we use 6 UV LED lights and 1 regular LED light. We use a Realsense RGBD camera placed overhead the robot. **Left**: The workspace under regular lighting. **Right**: The workspace and a painted bag under UV lighting. The painted rim and handle regions of the bag look normal under regular lighting but glow under UV lights. The UV lights are only used during training, not during execution time.

Figure 2.3: **Left**: 5 plastic bags. The first 4 are used to train the perception module (Sec. 2.4). Bags 1 and 5 (test bag) are used in experiments (Sec. 2.5). **Right**: Bag 3 painted with green UV paint on its handles and red UV paint around its rim, under UV lighting.

with deformable bags [64, 3] assumes that the bag starts with its rim facing upwards and the bag wide open to simplify item insertion and instead focus on the object rearrangement and the bag lifting steps. In this work, we allow bags to start from unstructured configurations and tackle the challenge of orienting and opening the bag.

In recent work, Gao et al. [19] proposed an algorithm for tying the handles of deformable plastic bags. To facilitate tying, they fill in bags beforehand with items which expand the bags and make their handles more likely to be upright and exposed. Instead of tying filled bags, we study a different problem setting of opening and inserting items into bags, where the bags begin empty and in unstructured states.

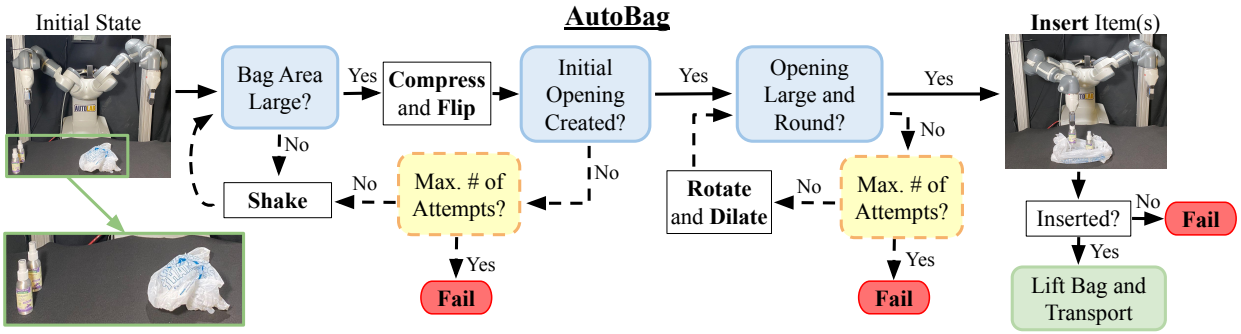

Figure 2.4: Overview of AutoBag. The robot starts with an unstructured bag with objects, shown to the left. Given this setup, AutoBag follows the procedure shown in the flow chart to first orient the bag upward and then enlarge the bag opening (see Section 2.4 for details). If the bag opening metric exceeds a threshold, then AutoBag proceeds to the item insertion stage (Section 2.4). A trial is a full success if the robot lifts the bag with all items in it. *Not shown in the figure: In the implementation, at the initial state, the robot will also check whether there is already an initial opening so it does not unnecessarily perform a **Compress** and **Flip** and potentially ruin a good state.

2.3 Problem Statement

We propose *bagging*: autonomously manipulating a thin plastic bag to open it, insert at least one item, and lift it for transport. We consider the broad class of thin plastic shopping bags commonly used in grocery stores that are made of a single sheet of translucent, reflective, and highly flexible thin plastic material cut with two holes for handles from die-cutting [wik]. We define the bag "rim" as the edge of the bag around its primary opening as if the plastic handles were cut off. See Figure 2.3 for an illustration. The term "opening" in this definition refers to the planar surface enclosed by the rim, through which objects are put inside the bag. The orientation of the opening is the direction of the outward-pointing normal vector from the plane formed by the opening. In many configurations, the opening can have an area of zero (e.g., when the rim is fully folded).

We assume the initial bag state is unstructured: deformed and potentially compressed, and resting stably on the surface, in which the bag rim and handles may be partially or fully occluded. We assume that the two sides of the bag do not stick to each other tightly (which occurs in new bags) and that some opening can be created using actions such as shaking, flinging, or pulling. We assume that the bag can be segmented from the workspace (e.g., by color thresholding).

We consider a bimanual robot with parallel-jaw grippers and an overhead calibrated RGB camera above a flat surface. At each timestep t, the robot receives an RGB image observation $I_t \in \mathbb{R}^{W \times H \times 3}$ of the bag configuration $\mathbf{s}_t$, and executes an action $\mathbf{a}_t$. We assume also a set of small rigid objects $\mathcal{O}$, placed in known poses for grasping and insertion. The objective is to

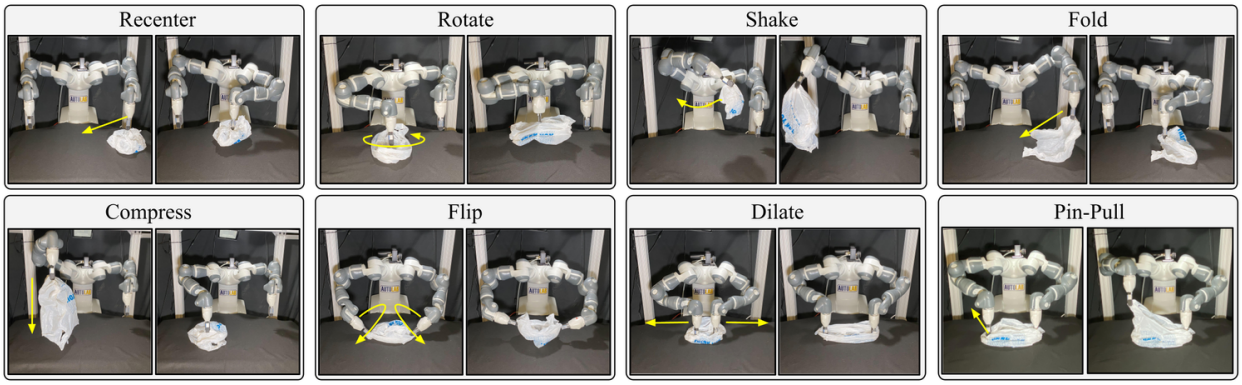

Figure 2.5: The action primitives in this work. See Section 2.4 for details. For each primitive, we show a small two-frame overview of it in action. The project website contains full videos of all primitives.

develop a policy $\pi : I \mapsto \mathbf{a}$ so that the robot opens the bag, places at least one object inside the bag, then lifts the bag off the table while containing the objects for transport.

2.4 Method

We propose a learned perception module to recognize the bag rim and handles that includes a novel self-supervised data collection process for training, during which the robot uses its action primitives. We propose two metrics for quantifying the bag opening. We then describe the AutoBag algorithm (visualized in Figure 2.4) for opening a plastic bag, followed by item insertion.

Self-Supervised Learning for Perception Module

In this work, we propose to represent bags through semantic segmentation. With this representation, we conjecture that the robot may be able to estimate the bag state from complex and deformed bag configurations. The robot perceives 2 key parts of a bag: **bag handles** and **bag rim** (see Figure 2.3). We formulate this as an image segmentation task where, given an RGB image, the output is a per-pixel classification among 4 classes: the 2 aforementioned semantic regions, any remaining area of the bag, and the background.

We use a self-supervised data collection procedure for training a segmentation model using UV-fluorescent markings to avoid time-consuming and expensive human annotation [70]. We place 6 programmable UV LED lights overhead and paint the 2 key parts of bags (handles and rim) with transparent UV-fluorescent paints that brightly reflect 2 different colors under UV light. When the UV lights are turned off, the paints are invisible, and the bag looks normal under regular lighting. When the regular lights are turned off and the UV lights are

turned on, everything is dark except for the regions with UV paints, which glow their unique colors.

With this setup, the robot uses its action primitives (see Section 2.4) to manipulate the bag into different configurations, and by alternating the lighting conditions, the camera collects paired images of the bag in both standard and UV lighting. By extracting the segmentation masks from the image under the UV lights through color thresholding, the system obtains the ground truth segmentation labels corresponding to the image of the bag under regular lighting conditions, which are then used to train the segmentation network. The objective of this process is to manipulate a plastic bag into a diverse set of configurations—both in terms of its volume and its orientation—as the increased data diversity can lead to a higher-quality perception model.

Action Primitives

We consider a set of action primitives $\mathcal{A}$ (see Figure 2.5). Each primitive has a type $m \in \mathcal{M}$ and action-specific parameters ϕ_m: $\mathbf{a} = \langle m, \phi_m \rangle \in \mathcal{A}$. Gripper positions are specified as Cartesian coordinates in pixels, and the grasping height is set to the height of the workspace to ensure successful grasping of the bag. The primitives are:

1. **Recenter** (x, y): Grasp the bag at (x, y) from top down, lift it up, and translate it to the workspace center at $(0, 0)$. This prevents the bag from moving off the workspace.

2. **Rotate** $(x, y, \alpha, \beta, \gamma)$: Grasp the bag at (x, y) from top down, lift it up, rotate the gripper by Euler angles (α, β, γ), and then directly place the gripper down.

3. **Shake** (x, y, k_s, ℓ, f): Grasp the bag at (x, y) from top down, lift it up, then perform k_s shaking motions of amplitude ℓ and frequency f, where the gripper rotates its wrist side by side, followed by a swing action to lay the bag on the table. This action often expands the surface area of a compressed bag.

4. **Fold** (x, y, d): Grasp the bag at (x, y) from top down, lift it off the workstation, move the gripper horizontally outward by distance d and then inward and downward to fold the bag and reduce its top-down visible surface area. This action enables the robot to compress and partially reset the bag state. Combined with other actions that expand the bag, we find that this induces greater diversity of bag configurations during self-supervised data collection. The robot does not use this action when opening the bag at execution time.

5. **Compress** (x, y, k_c): Grasp the bag at (x, y), lift it up in midair, then press it downwards until contact with the workspace, and repeat for a total of k_c motions. This action changes the side of the bag that is flat after being compressed. Holding the bottom part with the bag opening facing downward and compressing also allow air to inflate the bag opening.

6. **Flip** $(x_l, y_l, x_r, y_r, \alpha)$: Orient both grippers towards each other, each with an angle α with the horizontal plane, grasp opposite ends of the bag at positions (x_l, y_l) and (x_r, y_r).

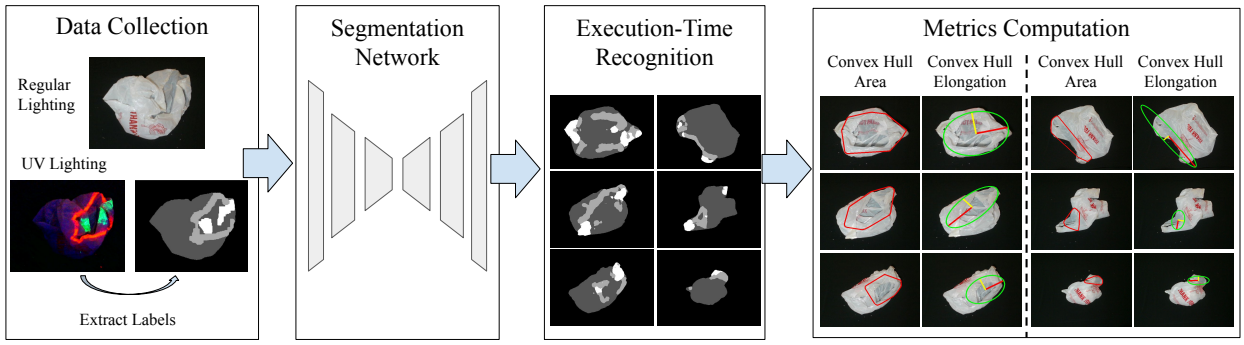

Figure 2.6: Perception Pipeline. The robot collects images by autonomously manipulating a bag into diverse configurations. It extracts segmentation labels of bag handles and rims through color thresholding from the images under UV light. We use this data to train a semantic segmentation network. During execution time, the perception model takes in an RGB image and predicts the segmentation (white: handles; light gray: rim; dark gray: other parts of the bag; black: background). We fit a convex hull to the predicted rim region (indicated by the red boundary), and compute the area and elongation of the convex hull to quantify the bag opening (Section 2.4). We overlay an ellipse with the major and minor axes computed by PCA on the convex hull.

Then, lift both grippers, rotate each gripper by 180 degrees, and then place the bag down. This action tends to change the direction of the bag opening (e.g., from downwards to upwards).

7. **Dilate** $(x_l, y_l, x_r, y_r, \alpha, \theta, d)$: Bring both grippers together at positions (x_l, y_l) and (x_r, y_r) and with them pointing downwards with an angle α with the horizontal plane. Then move both grippers away from each other horizontally along direction θ, each moving by distance d. This action can enlarge a small opening. We use this action during bag opening but not during data collection.

During data collection, the robot uses the following policy to select actions to manipulate the bag into diverse states:

1. When the bag area (as viewed from above) exceeds a threshold, sample these primitives uniformly at random: **Rotate, Shake, Fold, Compress, Flip**;
2. When the bag area is small, sample these primitives uniformly at random: **Rotate, Shake, Compress**;
3. After **Compress**, perform a **Flip**;
4. When the bag is off the center, perform **Recenter**.

Bag Opening Metrics

We propose to use two metrics for quantifying the bag opening from a segmented image. Both use the convex hull of the bag rim, denoted as CH, which is the convex region in the image enclosed by the pixels that the perception model (Section 2.4) identifies as the rim.

1. **Normalized convex hull area** A_{CH}: to approximate the size of the bag opening, we compute its convex hull area in 2D pixel space, and divide that by the maximum convex hull of the bag rim. To compute the maximum convex hull value, a human manually manipulates the bag offline to maximize the opening.

2. **Convex hull elongation** E_{CH}: we approximate the bag opening elongation using the ratio of the PCA major and minor axes of the convex hull in 2D pixel space.

See Figure 2.6 for visualizations. The normalized convex hull area makes the metric bag size-agnostic. In addition, we use convex hull elongation because we observe that for inserting items, a sideways-facing bag with a closed opening is worse than an upward-facing opening which is small but rounded. The normalized convex hull area, however, may give higher values to the former. Consequently, measuring elongation gives extra information about the bag opening.

AutoBag Algorithm

The first part of the AutoBag algorithm uses the perception module to choose actions to open a bag from an unstructured initial state. See Figure 2.4 for an overview. This consists of the following two stages.

Stage 1: If the 2D pixel surface area of the bag is below a threshold $S^{(1)}$, the robot executes a **Shake** to expand the bag, where its grasp points are sampled from the handle region (or anywhere on the bag if handles are not visible). Otherwise, if the area exceeds $S^{(1)}$, the robot grasps the bottom and executes a **Compress**. This flattens the bottom so that when the robot next executes a **Flip**, the bag may stand stably with its opening facing upward. With an updated top-down image, the algorithm uses the perception model to compute the metrics A_{CH} and E_{CH}. When the former is above threshold $A_{CH}^{(1)}$ and the latter is below threshold $E_{CH}^{(1)}$, it is likely that an initial upward-oriented opening exists. If the thresholds are not met, the bag is likely tilted on its side or folded inward, with the opening closed, and the robot resets the state by executing a **Shake** and repeats.

Stage 2: The robot uses **Rotate** and **Dilate** actions to iteratively enlarge the opening. At each iteration, the algorithm queries an overhead image of the bag, estimates its rim positions, and uses PCA to identify the direction of the major and minor axes of the opening. The robot performs a **Rotate** about the z-axis so that the minor axis of the bag aligns with the horizontal axis, and then uses **Dilate** to pull the bag opening farther apart from the opening center along its minor axes. This process repeats until the normalized convex

hull area reaches a threshold $A_{CH}^{(2)}$ and the elongation metric falls below a threshold $E_{CH}^{(2)}$, suggesting that the opening is large and round enough for object insertion.

AutoBag: Object Insertion and Bag Lifting

The final steps of AutoBag involve inserting the objects into the bag and lifting the bag. To lift the bag, the robot grasps the bag at the workstation height to avoid missed grasps, but this can lead to grasping multiple layers, a common challenge in deformable manipulation [71]. We thus propose the **Pin-Pull** $(x_{pin}, y_{pin}, x_{pull}, y_{pull})$ primitive: one gripper goes to position (x_{pin}, y_{pin}), and presses down onto the bag ("pinning"). The other gripper goes to position (x_{pull}, y_{pull}), closes the gripper, and lifts up to a fixed height h or until a torque limit is reached ("pulling"). The purpose of **Pin-Pull** is to stretch the bag after grasping. The additional layer that is accidentally grasped can slip out of the gripper during this process, leading to a higher success rate of grasping a single layer.

Given an overhead image of the bag, the robot estimates the opening by fitting a convex hull on the perceived rim, then divides the space by the number of objects. It grasps each object using known poses and places them in the center of each divided region. Then it identifies the positions of the handles and performs two **Pin-Pull** actions to grasp the left and right handles (or bag boundaries if handles are occluded). Finally, the robot lifts the bag off the table.

2.5 Physical Experiments

During training and experiments, the bags we use are of size 28 −30 cm by 49 −54 cm when laid flat (see Figure 2.3). The flat workspace has dimensions 70 cm by 90 cm.

Self-Supervised Training of Perception Module

For data collection, we use 4 training bags (see Figure 2.3) and collect 500 images for each bag. All images come with automatic labels using the self-supervised procedure. We use a U-Net architecture [57] for the segmentation network, trained with soft DICE loss [46]. We use one NVIDIA Titan Xp GPU, with a batch size of 8, and a learning rate of 5e-4. The trained model achieves a 77% intersection over union (IOU) on the validation set. See the supplement for the learning curve and example predictions.

Experiment Protocol

To evaluate AutoBag, we use two bags, one of which is the smallest bag from training. The other, unseen bag has the same size as the largest training bag but has different patterns (see both in Figure 2.3). Neither has UV paint. The goal is to insert 2 identical 2 oz. spray bottles into each bag. We define a *trial* as an instance of the robot attempting to perform

the full end-to-end procedure: to open a bag, insert the items into it, and then lift the bag (with items) off the surface. We allow for up to $T = 15$ actions (excluding **Recenter**) before the robot must formally lift the bag. If the robot encounters motion planning or kinematic errors during the trial, we reset the robot to the home position and continue the trial. We consider 3 tiers of initial bag configurations:

- Tier 1: The bag starts upward-facing with the rim recognizable but with a small opening. This requires enlarging the bag opening to allow placing objects inside.
- Tier 2: The bag starts at an expanded, slightly wrinkled state lying sideways on the workspace. This requires reorienting the bag upwards and then opening the bag.
- Tier 3: Any other, more complex configuration, such as when the bag is compressed with no visible handles.

At the start of each trial, we initialize the bag by compressing and then adjusting it to make it fall into one of the tiers. A trial is an "$n \geq 1$ success" if the robot can contain at least one bottle in the bag when lifting. A trial is an "$n = 2$ success" when the robot lifts the bag off the surface while containing both bottles. Additionally, we report the number of times the robot successfully opens the bag ("Open Bag") (i.e., proceeds to item insertion), and the number of objects that the robot correctly places in the bag opening before lifting ("#Placed"). Objects may fall out when the robot lifts the bag, meaning that the number of objects contained ("#Contained") is upper-bounded by #Placed.

AutoBag and Baseline Methods

We compare AutoBag to baselines and ablations, which we summarize here and defer details to the supplement.

To evaluate the proposed perception module, we perform ablation studies in Tier 1 and Tier 2 that remove the perception module ("AB-P"). Instead of detecting the rim and using the rim area and elongation as the metrics for determining whether the bag is sufficiently opened, it uses the area and elongation of the bag as an approximation. In addition, for **Dilate**, instead of dilating from the opening center, it uses the bag center. To evaluate the proposed metrics for quantifying the bag opening, we perform ablation studies in Tier 1. "AB-A_{CH}" uses only the convex hull elongation metric, and "AB-E_{CH}" uses only the convex hull area metric.

Additionally, for Tier 2, since the bag opening faces sideways, we design 2 different baselines. "Side Ins." attempts to grasp the top layer of the bag to open it and then insert objects from the side. "Handles" grasps the two handles, performs horizontal and vertical flinging to open the bag, and then releases one gripper to insert the objects while the other gripper still holds the bag.

Train Bag	Method	Open Bag	#Placed	#Contained	$n \geq 1$ Succ.	$n = 2$ Succ.
Tier 1	AB-P	2/6	0.7±1.0	0.7±1.0	2/6	2/6
	AB-A_{CH}	5/6	1.0±0.9	0.5±0.5	3/6	0/6
	AB-E_{CH}	5/6	0.8±1.0	0.7±1.0	2/6	2/6
	AutoBag	**5/6**	**1.6±0.7**	**1.1±0.8**	**4/6**	**3/6**
Tier 2	Side Ins.	1/6	0.3±0.8	0.3±0.8	1/6	1/6
	Handles	2/6	0.2±0.4	0.2±0.4	1/6	0/6
	AB-P	2/6	0.7±1.0	0.5±0.8	2/6	1/6
	AutoBag	**4/6**	**1.3±1.0**	**0.8±1.0**	**4/6**	**2/6**
Tier 3	AutoBag	**1/6**	**0.3±0.8**	**0.3±0.8**	**1/6**	**1/6**

Table 2.1: Physical experiment results of AutoBag, ablations, and baseline methods using a training bag across 3 different tiers.

AutoBag Test Bag	Open Bag	#Placed	#Contained	$n \geq 1$ Succ.	$n = 2$ Succ.
Tier 1	4/6	1.2±1.0	1.0±0.9	4/6	2/6
Tier 2	3/6	0.9±1.0	0.7±0.8	3/6	1/6

Table 2.2: Results of AutoBag on the test bag, across the first two tiers. We report similar evaluation metrics as in Table 2.1.

2.6 Results and Failure Modes

We report results on the training bag in Table 2.1, where we run 6 trials per experiment setting. The results suggest that AutoBag achieves an $n = 2$ success rate of 3/6, 2/6, and 1/6 on the three respective tiers and outperforms baselines and ablations in all metrics. The low success rate of "AB-P" for opening the bag suggests the perception module is important for determining whether a bag is actually opened and for placing the gripper inside the opening for **Dilate**. Using only the opening area metric achieves a good $n \geq 1$ success rate but fails to put both objects in, as the opening is not sufficiently wide. On the other hand, using only the elongation metric ensures the bag opening is round but does not guarantee that the opening is large enough. For sideways insertion, the main challenge is grasping the top layer only to create a sideways opening. We find that it often either misses the grasp or grasps two layers, due to the tight space between the two layers. The main failure reason for the "Handles" baseline is that the bag collapses or the opening closes as soon as one gripper releases the handle to grasp the inserted object, leaving no room for the robot to insert the objects.

On the test bag, we conduct experiments on Tier 1 and Tier 2 only, due to the extra challenge from the test bag perception. The results in Table 2.2 suggest that AutoBag can attain 4/6 and 3/6 partial successes.

We summarize the failure modes of AutoBag:

(A) Number of trials reaches the maximum limit (38%).

(B) Bag gets pushed out of the workspace (3%).

(C) Objects are not placed inside the opening (13%).

(D) Objects falls out when lifting the bag (23%).

(E) Fail to lift the bag successfully (23%).

(A) is often caused by instability of the perception model prediction, especially in complex bag configurations and around decision boundaries, leading to unsuitable/ineffective actions. (B) occurs due to robot arm motion planning. (C) is due to inaccuracy in the rim recognition, especially for the unseen test bag. (D) is a major bottleneck for achieving *full* success, due to non-ideal grasp position for lifting. (E) is caused by slipped grasps. More details, including failure reasons for baselines, are in Appendix A.

Chapter 3

Bagging by Learning to Singulate Layers Using Interactive Perception

Many fabric handling and 2D deformable material tasks in homes and industries require singulating layers of material such as opening a bag or arranging garments for sewing. In contrast to methods requiring specialized sensing or end effectors, we use only visual observations with ordinary parallel jaw grippers. We propose SLIP: Singulating Layers using Interactive Perception, and apply SLIP to the task of autonomous bagging. We develop SLIP-Bagging, a bagging algorithm that manipulates a plastic or fabric bag from an unstructured state and uses SLIP to grasp the top layer of the bag to open it for object insertion. In physical experiments, a YuMi robot achieves a success rate of 67% to 81% across bags of a variety of materials, shapes, and sizes, significantly improving in success rate and generality over prior work. Experiments also suggest that SLIP can be applied to tasks such as singulating layers of folded cloth and garments. Supplementary material is available at `https://sites.google.com/view/slip-bagging/`.

3.1 Introduction

In this chapter, we focus on the task of autonomous bagging using a novel algorithm for singulating the top layer of a deformable bag through visual feedback and interactive perception.

Many tasks in homes and factories require grasping a single layer of 2D deformable objects. Examples include taking one napkin from a stack of napkins, grasping the top layer of a folded towel to unfold it, grasping a single layer of a T-shirt to insert into a hanger, and grasping a single layer of a bag to hold it open while placing items inside. Humans manipulate such deformable objects with great dexterity using touch and vision. Such tasks are very challenging for robots. Enabling touch sensing may require equipping the robot end effector with compliant grippers or special tactile sensors such as the mini-Delta gripper [42] and the ReSkin sensor [4] used in Tirumala et al. [71] and GelSight [81] used in Sunil et

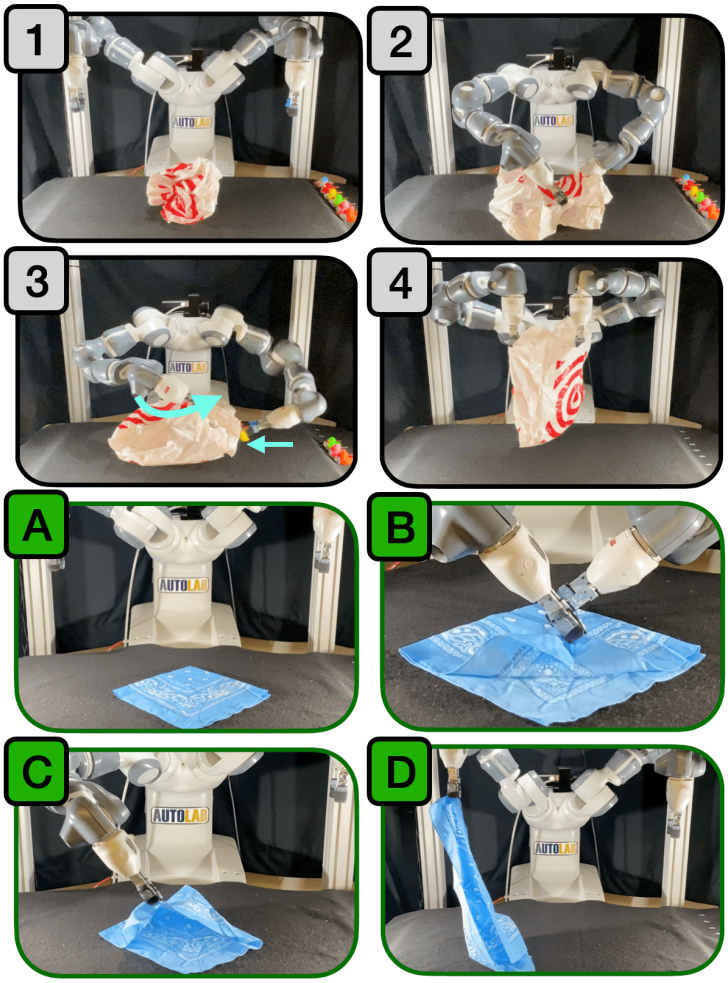

Figure 3.1: SLIP-Bagging. **Top 2 rows:** (1) Initial unstructured and deformed bag. (2) The robot flattens the bag, and then (3) uses SLIP to grasp the top layer of the bag. The robot rotates the bag by 90° and inserts objects into the bag. (4) The robot lifts the bag filled with the inserted items, so it is ready for transport. **Bottom 2 rows:** (A) Initial configuration of a piece of folded cloth. (B) The robot uses SLIP to grasp the top layer of a folded square cloth. (C) After grasping the top layer, the robot lifts the cloth up. (D) After shaking, the cloth is successfully expanded.

al. [68] for cloth manipulation.

In this work, we achieve single-layer grasping using a bimanual robot with ordinary parallel-jaw grippers. We use self-supervised learning to identify where to grasp, and we use interactive perception to determine the number of layers grasped. The robot iteratively adjusts its grasp until it successfully grasps a single layer.

In Chapter 2, we see that there are many ways for bags to deform and create self-occlusions. Many plastic bags are also reflective and translucent, as well as elastoplastic, meaning they have a tendency to restore their shape under small forces. However, an ideal

state for object placement—a bag standing upward with its opening wide open—is also not a naturally stable pose for soft deformable bags, which tend to lie on their side. An alternative approach is to open the bag and insert objects horizontally, but this requires grasping only the top layer to create the bag opening. This is nontrivial because: (1) The depth values from a typical RGBD camera are often severely inaccurate on bags due to reflection and transparency; (2) even if the depth of the top layer is known, moving the gripper to that height to perform the grasp will push the surface downward and cause a missed grasp due to lack of friction on the surface; and (3) grasping from the side is not always kinematically feasible since the two layers can be stuck together with no space in between. As we demonstrate in experiments, a 1 mm change in the gripper height can lead to the difference between a missed (0-layer) grasp, a 1-layer grasp, and a 2-layer grasp of plastic bags, and the heights of successful 1-layer grasps are different each time depending on the shape of the bag (wrinkles and flatness of both the top and bottom layers).

We use interactive perception [5] to recognize how many layers a robot grasps, and to adjust its grasp if needed. While the robot cannot a priori know what grasp height it should go to grasp only one layer of the bag, after a grasp is performed it can tell how many layers it has grasped by perturbing the bag and observing how the bag moves with the gripper. Intuitively, if the bag does not move, it indicates a 0-layer grasp. If the top layer of the bag moves with the gripper while the bottom layer only moves a little and mostly stays on the table, it indicates a 1-layer grasp. If the entire bag moves with the gripper, it suggests a 2-layer grasp.

We propose SLIP: Singulating Layers using Interactive Perception. Using SLIP, we present an algorithm for opening deformable bags, which we call SLIP-Bagging, that is effective across a variety of bag materials, including non-reusable plastic bags such as thin and soft bags made of low-density polyethylene (LDPE) and thicker and stiffer grocery bags made of high-density polyethylene (HDPE), as well as reusable fabric bags such as drawstring backpacks, mesh drawstring bags, and large fabric handbags. Physical experiments suggest that SLIP-Bagging achieves a 5x success rate compared to a prior state-of-the-art method for autonomous bagging [9]. Moreover, we conduct physical experiments to evaluate the applicability of SLIP to singulating layers for a variety of fabrics and garments (see Fig. 3.1).

This chapter makes the following contributions:

1. SLIP, an algorithm to singluate layers of bags using interactive perception, with visual feedback to enable the robot to adapt its grasp height without tactile sensors;

2. SLIP-Bagging, an algorithm that uses SLIP for opening and inserting objects into a deformable bag in an unstructured initial state;

3. A SLIP-Bagging system and physical experiments that achieve a success rate of 67% to 81% across bags of different materials, shapes, and sizes (unseen in training). On thin plastic bags, SLIP-Bagging's success rate is 5x that of the state-of-the-art method designed specifically for thin plastic bags.

4. Physical experiments suggesting the applicability of SLIP to other tasks such as singulating layers of fabrics and dresses.

3.2 Related Work

Deformable Objects and Single-Layer Grasping

There is a rich literature on deformable object manipulation; see [58, 6, 84] for representative surveys. Deformable objects are challenging due to their infinite degrees of freedom, which both induce complex dynamics that are hard to model and control and lead to self-occlusions that makes planning challenging. Among deformable object manipulation, fabric manipulation is one of the most widely-studied areas [7, 27, 56, 63, 24, 74, 37, 77, 39, 2]. These works focus on learning grasp locations that are effective for pick-and-place actions or dynamic actions to achieve smoothing and folding for one piece of fabric.

While some prior work has studied singulating a single sheet or fabric layer from a stack, most use tactile sensing or specialized end effectors. Tirumala et al. [71] use a ReSkin sensor [4] to singulate layers of cloth from tactile feedback. Manabe et al. [41] design a rolling hand mechanism to separate a single sheet from a pile of fabrics. Guo et al. [23] use a XELA uSkin tactile sensor combined with visual inputs to turn a single book page. In this work, we propose to singulate layers with standard end effectors purely from visual feedback. Demura et al. [12] study grasping the top folded towel from a stack using visual feedback with a scooping action, using towels which are each several millimeters thick. In contrast, we study manipulation tasks where layers can be thinner than 1 mm.

Manipulating Deformable Bags

Some early work on bag manipulation studies mechanical design or policies for grasping [31], lifting [29] or unloading [32] large sacks. Prior work also studies bag manipulation in simulation. For example, Seita et al. [61] benchmark several simulation tasks that involve opening a bag and inserting objects into it, and Weng et al. [75] study modeling bags using graph neural networks. Much of the prior work on physical experiments with deformable bags assume a semi-structured bag state, such as pregrasped [25, 30, 78], filled with objects [19], oriented upwards with the bag wide open [64, 3], and focus on a specific task such as packing and arranging objects [3], opening the bag [78], or lifting the bag [64]. In contrast to these works, we study physical bag manipulation where bags start in unstructured states.

Recently, Chen et al. [9] propose the AutoBag algorithm for manipulating a thin plastic bag from an unstructured state. In their setting, the bags can be compressed, deformed, and arbitrarily oriented, and the task is to reorient the bag upward, enlarge the opening, insert objects, and then lift the bag up. However, AutoBag frequently fails when attempting to orient the bag upward. Gu et al. [22] improve AutoBag by using dynamic shaking actions and performing item insertion with one gripper grasping the bag handle in midair. In contrast,

Figure 3.2: Top: Training bags. Bottom: Test bags. See Section 3.6 for more details and Table 3.1 for results.

we flatten the bag, singulate the top layer to open it, and insert objects sideways, which results in much higher success rates. Moreover, while AutoBag is designed specifically for opening thin plastic bags, we show evidence that AutoBag is effective on other bag materials and shapes.

Interactive Perception

Interactive perception combines perception and action, enabling a robot to reduce uncertainty through active physical exploration and interaction with objects [5]. Interactive perception has a variety of applications, ranging from manipulation to object segmentation, to grasp planning [53]. Recent work has used interactive perception to better understand properties of geometrically challenging objects for robotic manipulation. For example, [17, 52] study how to interact with articulated objects to discover their geometric information. For deformable object manipulation, Shivakumar et al. [72] use interactive perception to autonomously untangle cables, and Willimon et al. [76] use interactive perception to detect and classify clothing. In this work, we use interactive perception to identify and singulate individual layers of bags to improve robotic bag manipulation.

3.3 Problem Statement

We study the autonomous bagging task. As defined in prior work [9], the task consists of manipulating and opening a deformable bag from an unstructured state, inserting n items, and then lifting it for transport. Unlike prior work [9] which only considers thin plastic

bags made of LDPE, we additionally test heavy-duty grocery plastic bags made of HDPE, drawstring backpacks and mesh bags, and reusable fabric handbags (as shown in Figure 3.2).

We consider a bimanual robot with two standard parallel jaw grippers, operating in an (x, y, z) cartesian coordinate frame with a flat manipulation surface parallel to the $xy-$plane and a calibrated overhead RGBD camera. At each time step, the robot uses the RGBD input $I \in \mathbb{R}^{W \times H \times 4}$ of the bag to select and execute a parameterized open-loop action primitive $\mathbf{a}$ according to a policy $\pi : I \mapsto \mathbf{a}$. We assume the initial bag state is unstructured and resting stably on the workspace, that is, it may be deformed and compressed, but not tied or zipped. We assume a set of rigid objects $\mathcal{O}$, placed in known poses for grasping. We use the following metrics for the task: (1) the success rate of grasping a single layer, (2) the percentage of objects successfully inserted into the bag, and (3) the percentage of objects contained in the bag after the robot lifts the bag.

We make the following assumptions: (1) the two sides of the bag do not stick to each other tightly (which occurs in brand-new plastic bags), (2) the bag can be segmented from the workspace via color thresholding, and (3) the size of the bag when fully flattened, denoted A_{max}, is known a priori.

3.4 SLIP: Singulating Layers using Interactive Perception

In this section, we describe SLIP in the context of grasping a single layer of a deformable bag, but the algorithm may apply to other fabric materials (see Section 3.6). As discussed in Section 3.1, SLIP is motivated by how, after the robot has performed a grasp, it cannot easily determine how many layers it grasped from visual inputs of a static scene, as the top layer is occluding the layers underneath. However, by moving the gripper and observing how the bag is moved with the top layer, the robot can infer how many layers it has grasped. Formally, SLIP requires 3 components: a cyclic trajectory T of the robot gripper, a video classification model M, and an iterative height adjustment algorithm.

The trajectory T needs to satisfy two properties: (1) The movement should reveal enough information for the robot camera to infer how many layers are grasped, and (2) the trajectory should be cyclic, so the bag roughly recovers its original state after executing T, allowing the robot to retry the grasp at the same location but with a different height. For (1), the main consideration is occlusion, as the robot gripper and wrist occlude the grasp point and its nearby region if we use a top-down grasp with an overhead camera. Thus, we tilt the gripper at an angle $\theta = 50^\circ$ so the grasp point is visible in the camera. For (2), we use a triangular trajectory, where the robot gripper first moves backward, then upward, and finally forward and downward back to the original position. To prevent the deformable object from translating as a whole, we use the robot's second gripper to pin the other side. See Figure 3.3 for a visualization. In our implementation, the trajectory takes about 5 secs.

While the robot executes the trajectory T, the camera takes an RGB video stream of the

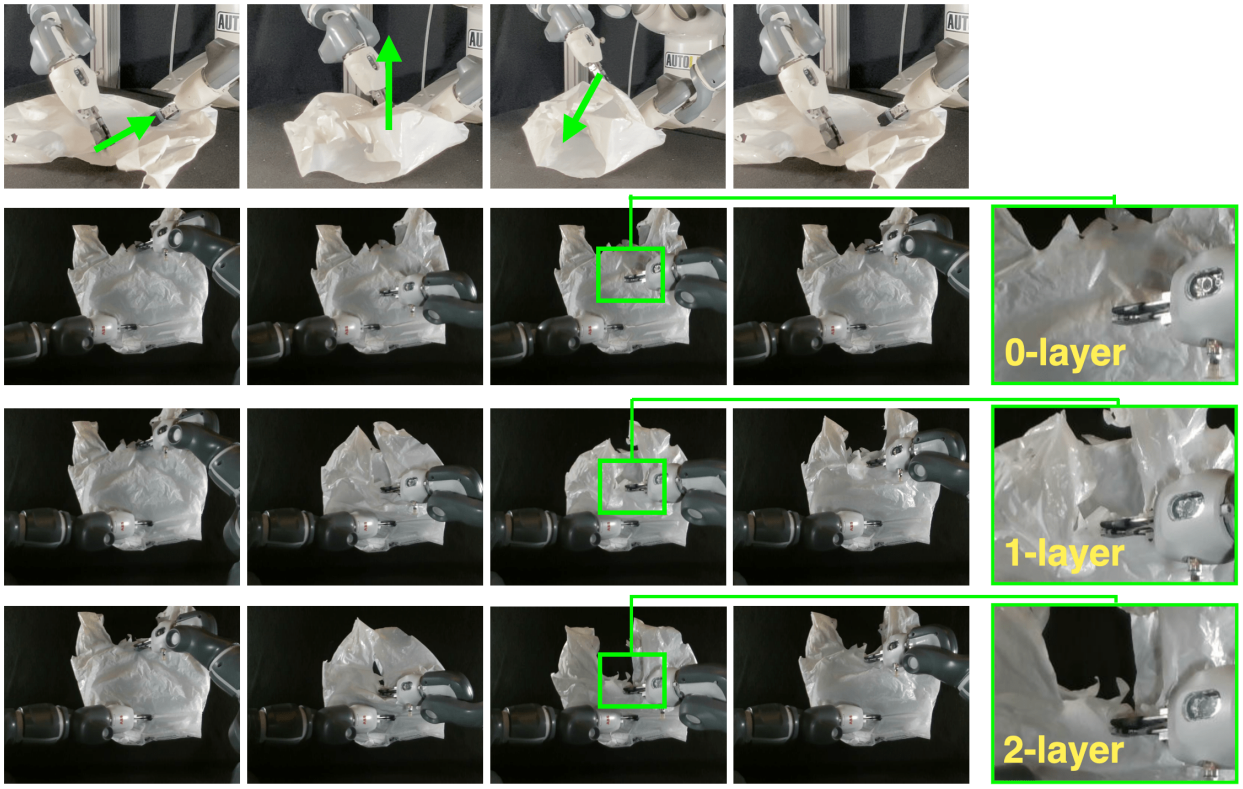

Figure 3.3: Examples of SLIP in action. Each row shows an example of one cyclic, triangular trajectory T in an iteration ("iter" in Algorithm 1) where one gripper moves the bag while the other one pins the bag. **Top row** a third-person view of the robot. **Next three rows**: top-down RGB camera views of one cyclic trajectory for different trials. They show, respectively, a 0-layer, 1-layer, and 2-layer grasp on a plastic bag. We provide zoomed-in versions of images in the third column to see the layers in more detail. See Section 3.4 for details.

bag. A video classification model M takes the video and classifies the grasp into 3 categories: 0 layer, 1 layer, and 2 layers. We use a SlowFast network with a ResNet-50 backbone [16], which takes in 32 images of size 224×224 sampled with a uniform interval from the video stream. Fig. 3.3 illustrates the visual differences among different layers grasped on a plastic bag.

Given the model classification, SLIP adjusts the gripper height and retries the grasp if it does not successfully grasp a single layer. We choose to use a fixed height adjustment each time which is similar to the strategy in [71], with height deltas Δh_- and Δh_+. One could also choose to let the adjustment height decay over time or use the bisection method, but we empirically find the numbers of trials these approaches take are similar while a fixed height adjustment is more robust to a long sequence of iterations than a decaying step size, and more robust to minor bag state changes between grasps and model classification errors

Algorithm 1 SLIP: Singulating Layers using Interactive Perception

Require: RGBD camera, closed trajectory T of the gripper, video classification model M, initial grasp height h_0, iterative height adjustment $\Delta h_+, \Delta h_-$, maximum number of iterations TrialMax, IterMax

Ensure: Single-layer grasp success

Grasp height $h = h_0$, trial $= 0$
while trial $<$ TrialMax **do**
    Sample a grasp location (x, y)
    iter $= 0$
    while iter $<$ IterMax **do**
        Grasp at location (x, y, h)
        Execute trajectory T and record a video stream V from the camera during the motion
        Number of layers grasped $n = M.predict(V)$
        if $n = 0$ **then** $h \leftarrow h - \Delta h_-$
        else if $n = 1$ **then** return True
        else if $n \geq 2$ **then** $h \leftarrow h + \Delta h_+$
        end if
        Open gripper
        iter $\leftarrow$ iter $+1$
    end while
    trial $\leftarrow$ trial $+1$
end while
return False

than bisection, since once bisection enters into a wrong interval, it may not succeed. The pseudocode of SLIP is provided in Algorithm 1.

3.5 SLIP-Bagging

Learned Perception Module

Similar to Chapter 2, we train a perception module to recognize the bag rim and handles. We represent bags through semantic segmentation, where each RGB image is classified per pixel into: bag handle, bag rim, remaining area of the bag, and the background. To collect training data, we use self-supervision with UV-fluorescent markings [70]. We place 6 UV LED lights around the workspace and paint the handles and rim of the bags with UV-fluorescent markers. During data collection, we take image pairs of the bag under regular lighting and under UV lighting. When the UV lights are turned on, the painted regions glow unique colors that can be extracted in the image through color thresholding, allowing us to get ground truth segmentation labels corresponding to the bag image under regular

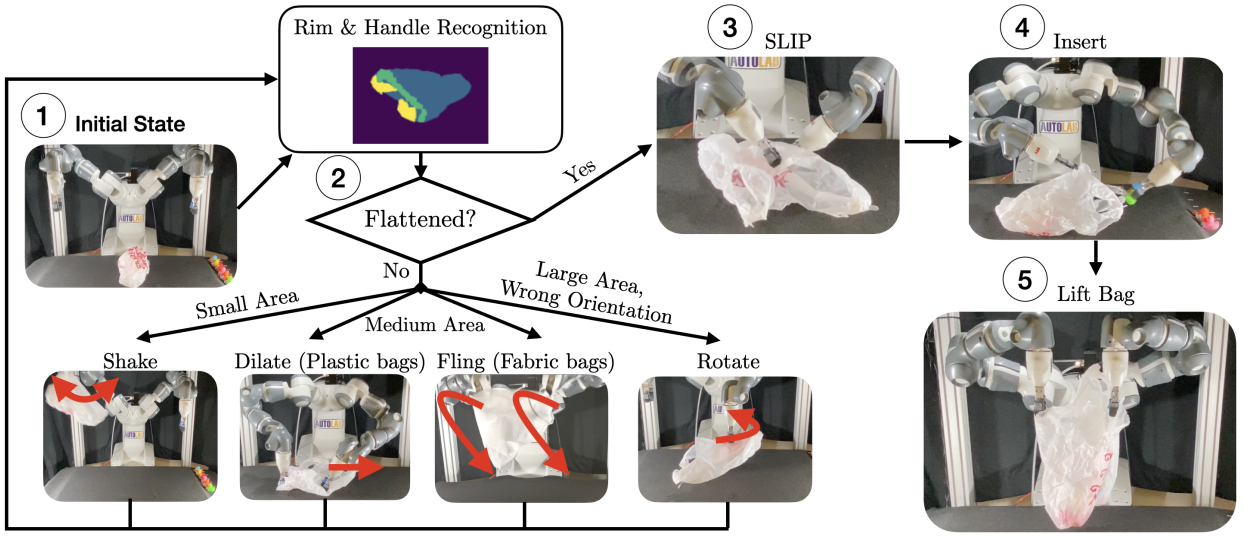

Figure 3.4: SLIP-Bagging Algorithm. (1) The robot starts with an unstructured bag with objects on the side. (2) SLIP-Bagging then flattens the bag, and (3) uses SLIP to grasp the top layer of the bag, followed by (4) insertion and (5) bag lifting. A trial is a full success if the robot lifts the bag with all items in it.

lighting. The robot then performs a random action to disturb the bag into another state, and repeats the process. This allows us to obtain a large dataset with diverse bag configurations efficiently without the need of human annotations. See the project website and [9] and [70] for details and examples.

Action Primitives

Following Section 2.4, SLIP-Bagging uses 4 manipulation primitives to flatten a bag from an unstructured state:

1. **Shake**: Grasp a corner of the bag, lift it up, shake it a predefined number of times, followed by a swing action to lay the bag on the table. This action uses gravity and inertia to loosen the bag.

2. **Rotate**: For small plastic bags, the robot uses one hand to rotate them. It grasps the center of the bag and rotate a desired angle. For large fabric bags, the robot uses two hands since one hand is not effective due to underactuation. It grasps the left and right sides of the bag and pushes one hand forward while pulling the other hand backward to rotate the bag.

3. **Dilate**: First, the left gripper pins the left side of the bag, while the right gripper presses the bag and moves from the center to the right. Then, the right gripper pins the right

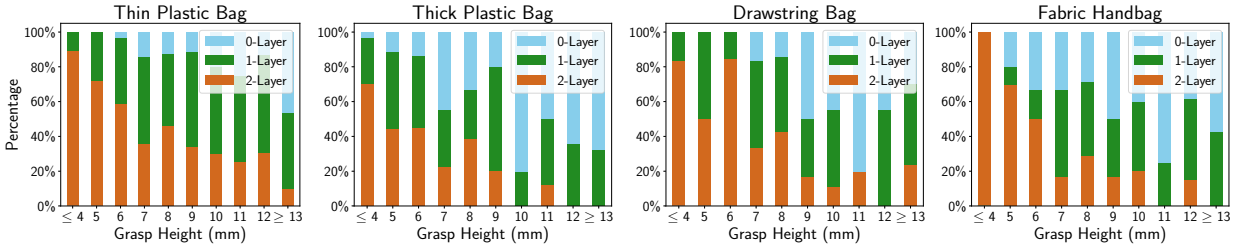

Figure 3.5: Distribution of the number of layers grasped for different grasp heights for 4 different bags.

side of the bag, while the left gripper presses the bag and moves from the center to the left. This action flattens the bag.

4. **Fling**: Inspired by prior work that uses a fling action to smooth garments [24, 7, 2], we design a fling primitive to smooth fabric bags. Given two pick points, the robot lifts the bag above the surface with two hands and flings the bag forward and then backward while putting it down. The fling velocity is set to the robot's maximum speed.

In addition, the robot recenters the bag as needed. For **Dilate**, the robot always first rotates the bag so that the bag's shorter axis aligns with the robot's horizontal axis, as **Dilate** can help expand that axis using friction. See the website for videos.

SLIP-Bagging

The SLIP-Bagging algorithm consists of 5 steps: (1) flatten the bag, (2) grasp the top layer of the bag near the bag opening using SLIP, (3) rotate the bag sideways, (4) use the other gripper to insert objects, and (5) lift the bag. See Figure 3.4 for an overview.

In the first step, the robot iteratively checks the bag area and orientation at each time step to determine whether the bag reaches a "flattened" state ready for single-layer grasping. SLIP-Bagging requires 3 threshold hyperparameters, p_{small}, p_{large}, and α. For a bag with an area A_{max} when fully flattened, if the current bag area is below $p_{\text{small}}A_{max}$, the robot performs the **Shake** primitive. If the current bag area is between $p_{\text{small}}A_{max}$ and $p_{\text{large}}A_{max}$, the robot uses **Dilate** to expand a plastic bag and **Fling** to expand a fabric bag. If the current bag area is greater than $p_{\text{large}}A_{max}$, the robot checks whether the bag opening is pointing forward. If the angle between the bag's opening and the robot's forward axis is greater than α, the robot **Rotates** the bag; otherwise, the bag is considered successfully "flattened." We set $\alpha = \pi/8$ and please see project website for choosing p_{small} and p_{large}.

Next, the robot uses SLIP to grasp the top layer of the bag. One hand pins the bottom of the bag while the other hand grasps a point near the center of the rim. We set the initial height of the grasp $h_0 = \max(h_{PD}, h_{min})$, where h_{PD} is the perceived depth of the grasp point on the bag surface measured by the RGBD camera, and h_{min} is a threshold to prevent

Category	Bag	Open/Flatten		Single-layer Grasp			Full Success				% Objects Inserted				SB Failure Modes
		AB	SB	PD	HG	SB	PD	HG	AB	SB	PD	HG	AB	SB	
Thin Plastic	Train	3/6	**6/6**	1/6	-	**6/6**	1/6	0/6	1/6	**5/6**	17%	0%	39%	**94%**	(D)
	Test	3/6	**6/6**	1/6	-	**6/6**	1/6	0/6	1/6	**4/6**	17%	0%	36%	**75%**	(D) × 2
Thick Plastic	Train	3/6	**5/6**	1/6	-	**5/5***	1/6	0/6	1/6	**3/6**	17%	0%	36%	**56%**	(A) (C) (D)
	Test	2/6	**5/6**	0/6	-	**4/5***	0/6	0/6	2/6	**4/6**	0%	0%	33%	**67%**	(A) (B)
Drawswtring	Train	0/6	**6/6**	0/6	-	**5/6**	0/6	-	0/6	**5/6**	0%	-	0%	**83%**	(B)
	Test	0/6	**6/6**	1/6	-	**5/6**	0/6	-	0/6	**4/6**	0%	-	0%	**67%**	(B) (C)
Reusable Handbag	Train	0/6	**5/6**	0/6	5/6	**3/5***	0/6	2/6	0/6	**3/6**	0%	36%	0%	**50%**	(A) (B) × 2
	Test	0/6	**6/6**	1/6	4/6	**6/6**	1/6	3/6	0/6	**4/6**	17%	50%	0%	**81%**	(C) (D)

*Denominator is the number of successful flattened trials that proceed to the SLIP stage.

Table 3.1: Physical experiment results of SLIP-Bagging compared with baselines. 6 trials were run on each of the 8 bags (Fig. 3.2) for each method. Each trial attempts to insert 6 rubber ducks, and "% Objects Inserted" is the average percentage of ducks inserted and remaining in the bag after bag lifting. **PD**: Perceived-Depth baseline. **HG**: Handle Grasping baseline. **AB**: AutoBag. **SB**: SLIP-Bagging. The "Single-layer Grasp" column for HG refers to grasping the handle associated with the top layer for handbags whose handles overlap (and thus is not applicable to plastic bags and drawstring bags). See Section 3.6 for failure mode categories.

the robot to grasp too deep in the case of erroneous depth measurements such as for mesh bags that have holes. We set $\Delta h_- = 1$ mm and $\Delta h_+ = 3$ mm.

After successfully grasping the top layer of the bag, the robot rotates the bag by 90° while the other arm grasps the inserted items and performs sideways insertion. Finally, the other arm goes inside the bag, grasps it, and lifts the bag together with the arm that has been holding the bag.

3.6 Physical Experiments

For experiments, we use a bimanual ABB YuMi robot with an overhead RealSense D435 camera. The workspace has dimensions 60×90 cm^2, and the bags we use range from 32×32 cm^2 to 55×55 cm^2.

Implementation Details

To train the perception module to recognize the rim and handles of bags, we collect a total of 7,500 images across 15 bags in 4 categories, with about 500 images for each bag, using the self-supervised process described in Sec. 3.5. The 4 categories are: non-reusable thin plastic bags made of LDPE, non-reusable thick plastic bags made of HDPE, drawstring bags including backpacks and mesh bags, and reusable fabric handbags. For the segmentation

network, we use a U-Net architecture [57] trained with soft DICE loss [46]. We use one NVIDIA A100 GPU, with a batch size of 32, an initial learning rate of 5e-4, and a weight decay factor of 1e-5. The trained model achieves a 70% intersection over union (IOU) on the validation set.

To train the video classification model for SLIP, we collect 800 video examples on 4 bags (one in each category), with 200 each. During data collection, for each sample, we manually set the bag into a roughly flat state, and then specify a grasp point as well as a grasp height. We randomize the grasp height so that the number of 0-layer, 1-layer, and 2-layer examples are balanced in the dataset. As described in Sec. 3.4, we use a SlowFast network architecture [16] with cross-entropy loss. We train the model on an A100 GPU with a batch size of 32, a learning rate of 5e-4, and an Adam optimizer. The trained model achieves a 90% accuracy on the validation set.

Bagging Experiments Setup

We evaluate SLIP-Bagging on 8 bags, shown in Figure 3.2. We use 6 rubber ducks of dimension $6 \times 7 \times 5$ cm^3 as the objects for insertion. For each trial, we randomly initialize the bag state by taking the bag, compressing and deforming it with our hands, dropping it onto the workspace, and letting the bag settle into a stable state. We allow for up to 30 actions (excluding recentering) for the robot to flatten the bag and up to 15 grasp iterations during SLIP. If the robot encounters motion planning or kinematic errors during the trial, we reset the robot to the home position and continue the trial.

We compare SLIP-Bagging to 3 baselines:

1. Perceived-Depth (PD): This method ablates the SLIP algorithm in SLIP-Bagging. Instead, the robot directly grasps at h_{PD}, the perceived depth of the grasp point on the bag surface measured by a depth camera.

2. Handle Grasping (HG): This method selects a handle to grasp instead of the bag rim. After grasping the handle and lifting the bag up, the robot performs sideways insertion underneath the grasping hand, similar to SLIP-Bagging.

3. AutoBag (AB) [9]: AutoBag uses "Compress" and "Flip" primitives to orient the bag upward and open the bag directly. It places the objects from the top onto the bag opening and then lifts the bag up instead of inserting the objects sideways.

For Perceived-Depth and Handle Grasping, we evaluate from an already-flattened bag state since the procedure of flattening the bag is the same as that in SLIP-Bagging. For SLIP-Bagging, we record the success rate of flattening the bag and grasping a single layer (which opens the bag from the side). For AutoBag, we measure the success rate of opening the bag upward. For each bag, we conduct 6 trials.

Figure 3.6: Non-bag experiments. Left to right: Folded cloth, dress, hat. The robot pins at the region indicated by the green point and attempts to grasp a single layer at the yellow star region.

Objects	0-Shot Recall			SLIP Success Rate
	0-layer	1-layer	2-layer	
Folded Cloth	100%	100%	62%	4/6
Dress	100%	75%	75%	4/6
Hat	100%	83%	25%	5/6

Table 3.2: Non-bag experiments. The middle 3 columns show the (multi-class) recall of the video classification model trained on bags and tested on garments without finetuning. The last column shows the success rate of grasping a single layer using SLIP with the classification model.

Bagging Experiments Results

Figure 3.5 shows the distribution of the number of layers grasped at various grasp heights (measured from the surface height) for each of the 4 training bags in the training data. As expected, as the grasp height decreases, it is less likely to grasp 0 layers and more likely to grasp 2 layers. However, while some grasp heights are more likely than others to grasp a single layer for each bag, there is no single grasp height that always works, as the success of single-layer grasping depends highly on the specific configuration of the bags.

Results in Table 3.1 demonstrate that SLIP-Bagging achieves a higher success rate than all baselines for all bags. In most trials, SLIP-Bagging takes fewer than 15 actions to flatten the bag and SLIP successfully singulates a layer within 4 grasp iterations. Among the baselines, Perceived-Depth has a low success rate of grasping a single layer. This is because, for the mesh bag, the perceived depth is often too deep due to holes, resulting in 2-layer grasps, while for other bags, the perceived depth is often not deep enough and leads to 0-layer grasps. For the Handle Grasping baseline, it is not applicable to drawstring bags since they do not have handles, and while it achieves a high success rate on handbags, it is not effective for plastic bags. The handles of plastic bags are on the two sides, so grasping and lifting them does not help create an opening like with a handbag. Its failures on handbags are mainly due to mistakenly grasping the handle associated with the bottom

layer or accidentally grasping both handles. For AutoBag, which is designed for thin plastic bags, is not effective on drawstring bags and fabric handbags. This is because its key action for opening the bag, "Compress," uses air to inflate the bag during a downward motion, and so only works for bags with lightweight material and without holes.

We observe 4 failure modes of SLIP-Bagging:

(A) Failure to flatten the bag with the correct orientation.

(B) Failure to successfully grasp a single layer of the bag.

(C) Bag slips out of the gripper after grasping a single layer.

(D) Robot hand hits the bag handles during insertion and does not put the objects inside the bag.

Failure (A) occurs when the perception module fails to recognize the rim and handle regions, and thus rotates the bag in the wrong orientation. Failure (B) usually occurs when the robot accidentally grasps both layers and manipulates the bag into a configuration which prevents the robot from grasping a single layer in future trajectories. For bags with stiff plastic and fabric materials, failure (C) can occur when the single-layer grasp is not firm enough and the bag slips out. Failure (D) is often due to the bag not being rotated completely sideways, which means the insertion hand can fail to enter the bag. Additionally, the robot hand may sometimes hit the handles or other parts of the bag, pushing the bag away or causing it to rotate, leading to insertion failure.

Single-Layer Grasping on Fabrics

We test SLIP on other materials to evaluate its applicability to general single-layer grasping tasks. We consider 3 deformable objects: a blue piece of cloth folded twice into a square (Fig. 3.1), a white dress, and a red hat (Fig. 3.6). The task goal is to grasp their top layer only. We apply our video classification model to these objects without any finetuning.

Table 3.2 shows the 0-shot multi-class recall metrics for the classification model as well as the success rate of achieving a single-layer grasp. In each case, the model predicts accurately on a 0-layer grasp and 1-layer grasp, but less accurately on a 2-layer grasp, for which there are greater visual differences across objects. A failure mode associated with grasping a folded cloth is that the cloth has 4 layers. Grasping 1, 2, and 3 layers look visually similar, so the model would mistaken those 2- and 3-layer grasps as a 1-layer grasp. While the model accuracy is lower than that of bags the model is trained on, the SLIP success rate is fairly high. This is because if the robot starts from a grasp higher than the surface and gradually decreases its height, it suffices for the model to accurately recognize a 1-layer grasp.

Out of the 5 failed trials, 3 are due to model prediction errors, such as misclassifying a 1-layer (or 2-layer) grasp and adjusting the grasp height in the wrong direction. The other 2 failures occur from the object slipping out of the robot's 1-layer grasp during its gripper movement.

Chapter 4

Conclusion

4.1 Summary

This thesis studies the problem of robotic bagging—manipulating deformable plastic bags from unstructured initial states to enable object insertion and transport.

In Chapter 2, we introduced a novel formulation of the bagging task and proposed the AutoBag algorithm, which leverages self-supervised UV-labeled data collection to train perception models for segmenting bag features. Physical experiments demonstrate that a YuMi robot using AutoBag can open a plastic bag and achieve a success rate of 16 out of 30 trials for inserting at least one item across a variety of initial bag configurations.

In Chapter 3, we presented SLIP-Bagging, a system that uses interactive perception to singulate the top layer of a bag. This approach improves upon AutoBag with significantly higher success rates for bag opening, item insertion, and lifting. Furthermore, we demonstrated that SLIP can generalize beyond plastic bags to tasks such as singulating layers of folded cloth and garments.

4.2 Limitations and Future Work

While the proposed systems represent a step forward in robotic manipulation of 3D deformable objects, several limitations remain. A primary concern is the success rate of the systems. Detailed analysis of failure cases for AutoBag is provided in the Appendix. Notably, AutoBag performs best on plastic bags with specific materials, shapes, and sizes, and its generalization is limited.

Although SLIP-Bagging addresses some of these limitations, both approaches are affected by the inherent fragility of multi-stage pipelines. As the number of stages increases, so does the risk of failure propagation, leading to compounding error and reduced overall performance. In addition, the current systems are slow—end-to-end execution of the bagging process can take more than 10 minutes, which limits practical deployment in real-world settings.

Future work can improve upon these systems in multiple directions. Key areas include reducing execution time to enable real-time or near-real-time operation, extending the methods to related tasks such as packing, folding, or wrapping, and addressing difficult cases where the two sides of a plastic bag adhere to each other due to static cling or material deformation.

We hope this thesis provides a useful perspective on the challenges and opportunities in robotic manipulation of deformable 3D objects and inspires future research on autonomous systems that interact with everyday materials in unstructured environments.

Bibliography

[wik] Shearing (manufacturing) — Wikipedia, the free encyclopedia. [Online; accessed 15-September-2022].

[2] Avigal, Y., Berscheid, L., Asfour, T., Kröger, T., and Goldberg, K. (2022). SpeedFolding: Learning Efficient Bimanual Folding of Garments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[3] Bahety, A., Jain, S., Ha, H., Hager, N., Burchfiel, B., Cousineau, E., Feng, S., and Song, S. (2022). Bag all you need: Learning a generalizable bagging strategy for heterogeneous objects. *arXiv preprint arXiv:2210.09997*.

[4] Bhirangi, R., Hellebrekers, T., Majidi, C., and Gupta, A. (2022). ReSkin: versatile, replaceable, lasting tactile skins. In *Conf. on Robot Learning (CoRL)*.

[5] Bohg, J., Hausman, K., Sankaran, B., Brock, O., Kragic, D., Schaal, S., and Sukhatme, G. S. (2017). Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291.

[6] Borras, J., Alenya, G., and Torras, C. (2019). A Grasping-centered Analysis for Cloth Manipulation. *arXiv preprint arXiv:1906.08202*.

[7] Chen, L. Y., Huang, H., Novoseller, E., Seita, D., Ichnowski, J., Laskey, M., Cheng, R., Kollar, T., and Goldberg, K. (2022). Efficiently Learning Single-Arm Fling Motions to Smooth Garments. In *Int. S. Robotics Research (ISRR)*.

[8] Chen, L. Y., Shi, B., Lin, R., Seita, D., Ahmad, A., Cheng, R., Kollar, T., Held, D., and Goldberg, K. (2023a). Bagging by learning to singulate layers using interactive perception. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3176–3183. IEEE.

[9] Chen, L. Y., Shi, B., Seita, D., Cheng, R., Kollar, T., Held, D., and Goldberg, K. (2023b). AutoBag: Learning to Open Plastic Bags and Insert Objects. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[10] Clarke, S., Rhodes, T., Atkeson, C., and Kroemer, O. (2018). Learning Audio Feedback for Estimating Amount and Flow of Granular Material. In *Conf. on Robot Learning (CoRL)*.

[11] Coumans, E. and Bai, Y. (2021). PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. `http://pybullet.org`.

[12] Demura, S., Sano, K., Nakajima, W., Nagahama, K., Takeshita, K., and Yamazaki, K. (2018). Picking up one of the folded and stacked towels by a single arm robot. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1551–1556.

[13] Duenser, S., Bern, J. M., Poranne, R., and Coros, S. (2018). Interactive Robotic Manipulation of Elastic Objects. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[14] Erickson, Z., Collier, M., Kapusta, A., and Kemp, C. (2018). Tracking Human Pose During Robot-Assisted Dressing using Single-Axis Capacitive Proximity Sensing. In *IEEE Robotics and Automation Letters*.

[15] Erickson, Z., Gangaram, V., Kapusta, A., Liu, C. K., and Kemp, C. C. (2020). Assistive Gym: A Physics Simulation Framework for Assistive Robotics. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[16] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

[17] Gadre, S. Y., Ehsani, K., and Song, S. (2021). Act the Part: Learning Interaction Strategies for Articulated Object Part Discovery. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*.

[18] Ganapathi, A., Sundaresan, P., Thananjeyan, B., Balakrishna, A., Seita, D., Grannen, J., Hwang, M., Hoque, R., Gonzalez, J., Jamali, N., Yamane, K., Iba, S., and Goldberg, K. (2021). Learning Dense Visual Correspondences in Simulation to Smooth and Fold Real Fabrics. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[19] Gao, C., Li, Z., Gao, H., and Chen, F. (2022). Iterative Interactive Modeling for Knotting Plastic Bags. In *Conf. on Robot Learning (CoRL)*.

[20] Grannen, J., Sundaresan, P., Thananjeyan, B., Ichnowski, J., Balakrishna, A., Hwang, M., Viswanath, V., Laskey, M., Gonzalez, J. E., and Goldberg, K. (2020). Untangling Dense Knots by Learning Task-Relevant Keypoints. In *Conf. on Robot Learning (CoRL)*.

[21] Grannen, J., Wu, Y., Belkhale, S., and Sadigh, D. (2022). Learning Bimanual Scooping Policies for Food Acquisition. In *Conf. on Robot Learning (CoRL)*.

[22] Gu, N., Zhang, Z., He, R., and Yu, L. (2023). ShakingBot: Dynamic Manipulation for Bagging. *arXiv preprint arXiv:2304.04558*.

[23] Guo, Y., Jiang, X., and Liu, Y. (2021). Deformation Control of a Deformable Object Based on Visual and Tactile Feedback. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[24] Ha, H. and Song, S. (2021). FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding. In *Conf. on Robot Learning (CoRL)*.

[25] Hellman, R. B., Tekin, C., van der Schaar, M., and Santos, V. J. (2018). Functional Contour-following via Haptic Perception and Reinforcement Learning. In *IEEE Transactions on Haptics*.

[26] Hopcroft, J., Kearney, J., and Krafft, D. (1991). A Case Study of Flexible Object Manipulation. In *Int. Journal of Robotics Research (IJRR)*.

[27] Hoque, R., Seita, D., Balakrishna, A., Ganapathi, A., Tanwani, A., Jamali, N., Yamane, K., Iba, S., and Goldberg, K. (2020). VisuoSpatial Foresight for Multi-Step, Multi-Task Fabric Manipulation. In *Proc. Robotics: Science and Systems (RSS)*.

[28] Howard, A. and Bekey, G. (1996). Prototype System for Automated Sorting and Removal of Bags of Hazardous Waste. In *SPIE, Intelligent Robots and Computer Vision*.

[29] Howard, A. and Bekey, G. (2000). Intelligent Learning for Deformable Object Manipulation. In *Autonomous Robtoics*.

[30] Ichiwara, H., Ito, H., Yamamoto, K., Mori, H., and Ogata, T. (2022). Contact-rich manipulation of a flexible object based on deep predictive learning using vision and tactility. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[31] Kazerooni, H. and Foley, C. (2005). A Robot Mechanism for Grapsing Sacks. In *IEEE Trans. Automation Science and Engineering*.

[32] Kirchheim, A., Burwinkel, M., and Echelmeyer, W. (2008). Automatic Unloading of Heavy Sacks From Containers. In *IEEE International Conference on Automation and Logistics*.

[33] Klingbeil, E., Rao, D., Carpenter, B., Ganapathi, V., Ng, A. Y., and Khatib, O. (2011). Grasping With Application to an Autonomous Checkout Robot. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[34] Lee, R., Ward, D., Cosgun, A., Dasagi, V., Corke, P., and Leitner, J. (2020). Learning Arbitrary-Goal Fabric Folding with One Hour of Real Robot Experience. In *Conf. on Robot Learning (CoRL)*.

[35] Lim, V., Huang, H., Chen, L. Y., Wang, J., Ichnowski, J., Seita, D., Laskey, M., and Goldberg, K. (2022). Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[36] Lin, X., Qi, C., Zhang, Y., Li, Y., Huang, Z., Katerina Fragkiadaki, C. G., and Held, D. (2022). Planning with Spatial-Temporal Abstraction from Point Clouds for Deformable Object Manipulation. In *Conf. on Robot Learning (CoRL)*.

[37] Lin, X., Wang, Y., Huang, Z., and Held, D. (2021). Learning Visible Connectivity Dynamics for Cloth Smoothing. In *Conf. on Robot Learning (CoRL)*.

[38] Lui, W. H. and Saxena, A. (2013). Tangled: Learning to Untangle Ropes with RGB-D Perception. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[39] Ma, X., Hsu, D., and Lee, W. S. (2022). Learning Latent Graph Dynamics for Deformable Object Manipulation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[40] Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth Grasp Point Detection Based on Multiple-View Geometric Cues with Application to Robotic Towel Folding. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[41] Manabe, K., Tong, X., and Aiyama, Y. (2021). Single sheet separation method from piled fabrics using roller hand mechanism. In *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pages 359–362.

[42] Mannam, P., Rudich, A., Zhang, K. L., Veloso, M., Kroemer, O., and Temel, Z. (2021). A low-cost compliant gripper using cooperative mini-delta robots for dexterous manipulation. In *Proc. Robotics: Science and Systems (RSS)*.

[43] Matas, J., James, S., and Davison, A. J. (2018). Sim-to-Real Reinforcement Learning for Deformable Object Manipulation. *Conf. on Robot Learning (CoRL)*.

[44] Matl, C. and Bajcsy, R. (2021). Deformable Elasto-Plastic Object Shaping using an Elastic Hand and Model-Based Reinforcement Learning. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[45] Matl, C., Narang, Y., Bajcsy, R., Ramos, F., and Fox, D. (2020). Inferring the Material Properties of Granular Media for Robotic Tasks. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[46] Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *4th international conference on 3D vision (3DV)*. IEEE.

[47] Morita, T., Takamatsu, J., Ogawara, K., Kimura, H., and Ikeuchi, K. (2003). Knot Planning from Observation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[48] Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., and Levine, S. (2017). Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[49] Nakagaki, H., Kitagi, K., Ogasawara, T., and Tsukune, H. (1996). Study of Insertion Task of a Flexible Wire Into a Hole by Using Visual Tracking Observed by Stereo Vision. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[50] Nakagaki, H., Kitagi, K., Ogasawara, T., and Tsukune, H. (1997). Study of Deformation and Insertion Tasks of Flexible Wire. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[51] Namiki, A. and Yokosawa, S. (2015). Robotic Origami Folding with Dynamic Movement Primitives. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[52] Nie, N., Gadre, S. Y., Ehsani, K., and Song, S. (2022). Structure from Action: Learning Interactions for Articulated Object 3D Structure Discovery. *arxiv preprint arXiv:2207.08997*.

[53] Novkovic, T., Pautrat, R., Furrer, F., Breyer, M., Siegwart, R., and Nieto, J. (2020). Object Finding in Cluttered Scenes Using Interactive Perception. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[54] Puthuveetil, K., Kemp, C. C., and Erickson, Z. (2022). Bodies Uncovered: Learning to Manipulate Real Blankets Around People via Physics Simulations. In *IEEE Robotics and Automation Letters*.

[55] Qi, C., Lin, X., and Held, D. (2022). Learning Closed-Loop Dough Manipulation Using a Differentiable Reset Module. In *IEEE Robotics and Automation Letters*.

[56] Qian, J., Weng, T., Zhang, L., Okorn, B., and Held, D. (2020). Cloth Region Segmentation for Robust Grasp Selection. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[57] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[58] Sanchez, J., Corrales, J.-A., Bouzgarrou, B.-C., and Mezouar, Y. (2018). Robotic Manipulation and Sensing of Deformable Objects in Domestic and Industrial Applications: a Survey. In *Int. Journal of Robotics Research (IJRR)*.

[59] Schenck, C. and Fox, D. (2017). Visual Closed-Loop Control for Pouring Liquids. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[60] Schenck, C., Tompson, J., Fox, D., and Levine, S. (2017). Learning Robotic Manipulation of Granular Media. In *Conf. on Robot Learning (CoRL)*.

[61] Seita, D., Florence, P., Tompson, J., Coumans, E., Sindhwani, V., Goldberg, K., and Zeng, A. (2021a). Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[62] Seita, D., Ganapathi, A., Hoque, R., Hwang, M., Cen, E., Tanwani, A., Balakrishna, A., Thananjeyan, B., Ichnowski, J., Jamali, N., Yamane, K., Iba, S., Canny, J., and Goldberg, K. (2020). Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[63] Seita, D., Jamali, N., Laskey, M., Berenstein, R., Tanwani, A. K., Baskaran, P., Iba, S., Canny, J., and Goldberg, K. (2019). Deep Transfer Learning of Pick Points on Fabric for Robot Bed-Making. In *Int. S. Robotics Research (ISRR)*.

[64] Seita, D., Kerr, J., Canny, J., and Goldberg, K. (2021b). Initial Results on Grasping and Lifting Physical Deformable Bags with a Bimanual Robot. In *IROS Workshop on Robotic Manipulation of Deformable Objects in Real-world Applications*.

[65] She, Y., Dong, S., Wang, S., Sunil, N., Rodriguez, A., and Adelson, E. (2020). Cable Manipulation with a Tactile-Reactive Gripper. In *Proc. Robotics: Science and Systems (RSS)*.

[66] Shen, B., Jiang, Z., Choy, C., Guibas, L. J., Savarese, S., Anandkumar, A., and Zhu, Y. (2022). ACID: Action-Conditional Implicit Visual Dynamics for Deformable Object Manipulation. In *Proc. Robotics: Science and Systems (RSS)*.

[67] Sundaresan, P., Grannen, J., Thananjeyan, B., Balakrishna, A., Laskey, M., Stone, K., Gonzalez, J. E., and Goldberg, K. (2020). Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[68] Sunil, N., Wang, S., She, Y., Adelson, E., and Garcia, A. R. (2022). Visuotactile affordances for cloth manipulation with local control. In *Conf. on Robot Learning (CoRL)*.

[69] Thananjeyan, B., Garg, A., Krishnan, S., Chen, C., Miller, L., and Goldberg, K. (2017). Multilateral Surgical Pattern Cutting in 2D Orthotropic Gauze with Deep Reinforcement Learning Policies for Tensioning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[70] Thananjeyan, B., Kerr, J., Huang, H., Gonzalez, J. E., and Goldberg, K. (2022). All You Need is LUV: Unsupervised Collection of Labeled Images using Invisible UV Fluorescent Indicators. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[71] Tirumala, S., Weng, T., Seita, D., Kroemer, O., Temel, Z., and Held, D. (2022). Learning to Singulate Layers of Cloth using Tactile Feedback. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[72] Viswanath, V., Shivakumar, K., Kerr, J., Thananjeyan, B., Novoseller, E., Ichnowski, J., Escontrela, A., Laskey, M., Gonzalez, J. E., and Goldberg, K. (2022). Autonomously Untangling Long Cables. In *Proc. Robotics: Science and Systems (RSS)*.

[73] Wang, A., Kurutach, T., Liu, K., Abbeel, P., and Tamar, A. (2019). Learning Robotic Manipulation through Visual Planning and Acting. In *Proc. Robotics: Science and Systems (RSS)*.

[74] Weng, T., Bajracharya, S., Wang, Y., Agrawal, K., and Held, D. (2021a). FabricFlowNet: Bimanual Cloth Manipulation with a Flow-based Policy. In *Conf. on Robot Learning (CoRL)*.

[75] Weng, Z., Paus, F., Varava, A., Yin, H., Asfour, T., and Kragic, D. (2021b). Graph-based Task-specific Prediction Models for Interactions between Deformable and Rigid Objects. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[76] Willimon, B., Birchfield, S., and Walker, I. (2011). Classification of clothing using interactive perception. In *icra*, pages 1862–1868.

[77] Wu, Y., Yan, W., Kurutach, T., Pinto, L., and Abbeel, P. (2020). Learning to Manipulate Deformable Objects without Demonstrations. In *Proc. Robotics: Science and Systems (RSS)*.

[78] Xu, Z., Chi, C., Burchfiel, B., Cousineau, E., Feng, S., and Song, S. (2022). DextAIRity: Deformable Manipulation Can be a Breeze. In *Proc. Robotics: Science and Systems (RSS)*.

[79] Yan, M., Zhu, Y., Jin, N., and Bohg, J. (2020a). Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects. In *IEEE Robotics and Automation Letters*.

[80] Yan, W., Vangipuram, A., Abbeel, P., and Pinto, L. (2020b). Learning Predictive Representations for Deformable Objects Using Contrastive Estimation. In *Conf. on Robot Learning (CoRL)*.

[81] Yuan, W., Dong, S., and Adelson, E. H. (2017). Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762.

[82] Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., Armstrong, T., Krasin, I., Duong, D., Sindhwani, V., and Lee, J. (2020). Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Conf. on Robot Learning (CoRL)*.

[83] Zhang, H., Ichnowski, J., Seita, D., Wang, J., Huang, H., and Goldberg, K. (2021). Robots of the Lost Arc: Self-Supervised Learning to Dynamically Manipulate Fixed-Endpoint Ropes and Cables. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*.

[84] Zhu, J., Cherubini, A., Dune, C., Navarro-Alarcon, D., Alambeigi, F., Berenson, D., Ficuciello, F., Harada, K., Kober, J., Li, X., Pan, J., Yuan, W., and Gienger, M. (2021). Challenges and Outlook in Robotic Manipulation of Deformable Objects. *arXiv preprint arXiv:2105.01767*.

[85] Zhu, J., Navarro, B., Passama, R., Fraisse, P., Crosnier, A., and Cherubini, A. (2019). Robotic Manipulation Planning for Shaping Deformable Linear Objects with Environmental Contacts. In *IEEE Robotics and Automation Letters*.

[86] Zimmermann, S., Poranne, R., and Coros, S. (2021). Dynamic Manipulation of Deformable Objects with Implicit Integration. In *IEEE Robotics and Automation Letters*.

Appendix A

AutoBag: Learning to Open Plastic Bags and Insert Objects

A.1 Action Primitives Implementation Details

Here, we describe the implementation details for the action primitives.

- For **Shake**, we set number of repetitions $k_s = 3$, amplitude $l = 0.7\pi$ radian (40.1°), and frequency $f = 0.4$ Hz.
- For **Fold**, we set $d = 28$ cm.
- For **Compress**, we set $k_c = 4$, with a pause of $0.9s$ after every repetition to let the bag settle. The robot grasps the bag with an angle $\alpha = \pi/7$ radian (25.7°) with the horizontal plane.
- For **Flip**, we set the gripper grasping angle $\alpha = 45°$.
- For **Dilate**, we restrict the dilation angle $\theta = 0°$ and set distance $d = 12$ cm. Gripper stops when it has moved by d or the torque sensor has reached threshold of $0.05\text{N}\,\text{m}$.

A.2 Details of Grasp Point Selection

During data collection, the grasp points for each primitive are sampled as follows:

- **Rotate**: Uniformly on the bag.
- **Shake**: Uniformly from the boundary of the bag.
- **Fold**: Uniformly from the boundary of the bag.
- **Compress**: Uniformly from the bag.

- **Flip**: The left and right endpoints of the bag segmentation along a horizontal line.
- **Recenter**: Center of the bag segmentation.

During the execution of AutoBag, the grasp points for each primitive are sampled as follows:

Stage 1:

- **Shake**: Grasp the center of the handle if at least one handle is visible. Otherwise, sample uniformly from the boundary of the bag.
- **Compress**: The robot grasps the bag with an angle $\alpha = \pi/7$ radian (25.7°) with the horizontal plane. The grasp point is the center of the bottom region of the bag, where the bottom is inferred from the rim of the bag using a heuristic: Fit a rectangle to the bag, and find the two corners that are farther from the rim. The center bottom of the bag is then approximated by finding the midpoint of those two farther corners and shrink towards the bag center until the midpoint lies on the bag.
- **Flip**: Same as during data collection.

Stage 2:

- **Rotate**: Center of the bag segmentation.
- **Dilate**: The two grippers are oriented towards each other with angle $\alpha = \pi/3$ radian (60°) with the horizontal plane and positioned around the center of the bag opening with offset by 0.02 cm each to avoid collision. The bag opening is approximated using the convex hull of the rim prediction. θ is set to be 0°, and $d = 10$ cm. Gripper stops when it has moved by d or the torque sensor has reached threshold of 0.02N m.

Bag Lifting:

- **Pin-Pull**: The two gripper positions (x_{pin}, y_{pin}) and (x_{pull}, y_{pull}) are the center of the two predicted handles if the handles are visible and the left and right endpoints of the bag segmentation if the handles are not visible.

A.3 Details of Perception Model

We collect 2,000 paired regular-UV images in total across 4 training bags. Collecting each regular-UV image pair takes 30-40 seconds. To obtain the ground truth segmentation, we use color thresholding from the UV images and dilate the rim and handle labels to connect regions where the UV paint does not glow strongly and filter noises. We use an 80-20 train/validation split for model training. We use one NVIDIA Titan Xp GPU, with a batch size of 8, a learning rate of 5e-4, and a weight decay factor of 1e-6. Figure A.1 shows the learning curve, including the training loss and validation intersection over union (IOU).

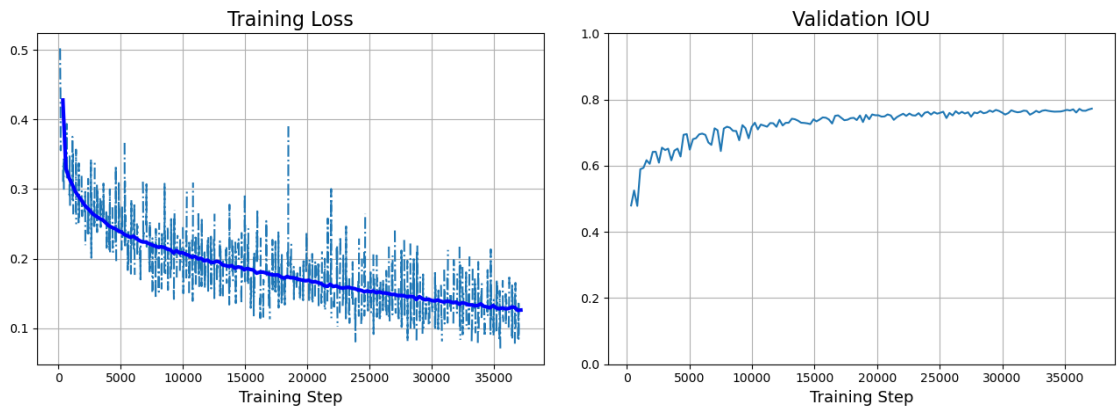

Figure A.1: Learning curves of the perception model (Section 2.5). **Left**: Training loss. **Right**: IOU on the validation set.

Figure A.3 illustrates some positive and negative example predictions of bag segmentation and the metrics (convex hull and elongation) applied to the predicted rim. We can see that negative examples are mainly due to the wrong predicted rim mask, including both type 1 and type 2 errors. This can lead to the estimated opening through the convex hull being either too large (including other parts of the bag) or too small (when only part of the true rim is recognized). When the predicted rim is noisy and segmented all over the bag, this can also lead to an inaccurate elongation estimate.

A.4 Experiment Details

For **AutoBag**, we define $S^{(1)}$ to be the bag size in 2D pixel space divided by the maximum bag size obtained from a flat bag and we set $S^{(1)} = 0.55$, $A_{CH}^{(1)} = 0.15$, $E_{CH}^{(1)} = 4.5$, $A_{CH}^{(2)} = 0.45$, $E_{CH}^{(2)} = 2.88$.

For the "Side Ins." baseline, we apply the **Pin-Pull** primitive to the opening of the bag. In particular, we choose the pin position (x_{pin}, y_{pin}) as the center of the rim farther from the bag center, and the pull position (x_{pull}, y_{pull}) as the midpoint between the center of the rim closer to the bag center and the bag center. The intention is to pin the bottom layer of the bag while pulling the top layer of the bag in order to separate the two layers of the bag and create a sideways-facing opening. The robot then attempts to insert the object from the side, regardless of whether the two layers have been truly separated or not, since the overhead camera cannot tell this.

For the "Handles" baseline, the robot first grasps the two handles, one in each gripper. It then lifts the bag in midair and performs two sequences of dynamic actions. It first shakes the bag in the horizontal left-right direction 2 times. This has the effect of separating the two layers of the rim to prevent them from sticking to each other. The robot then flings the

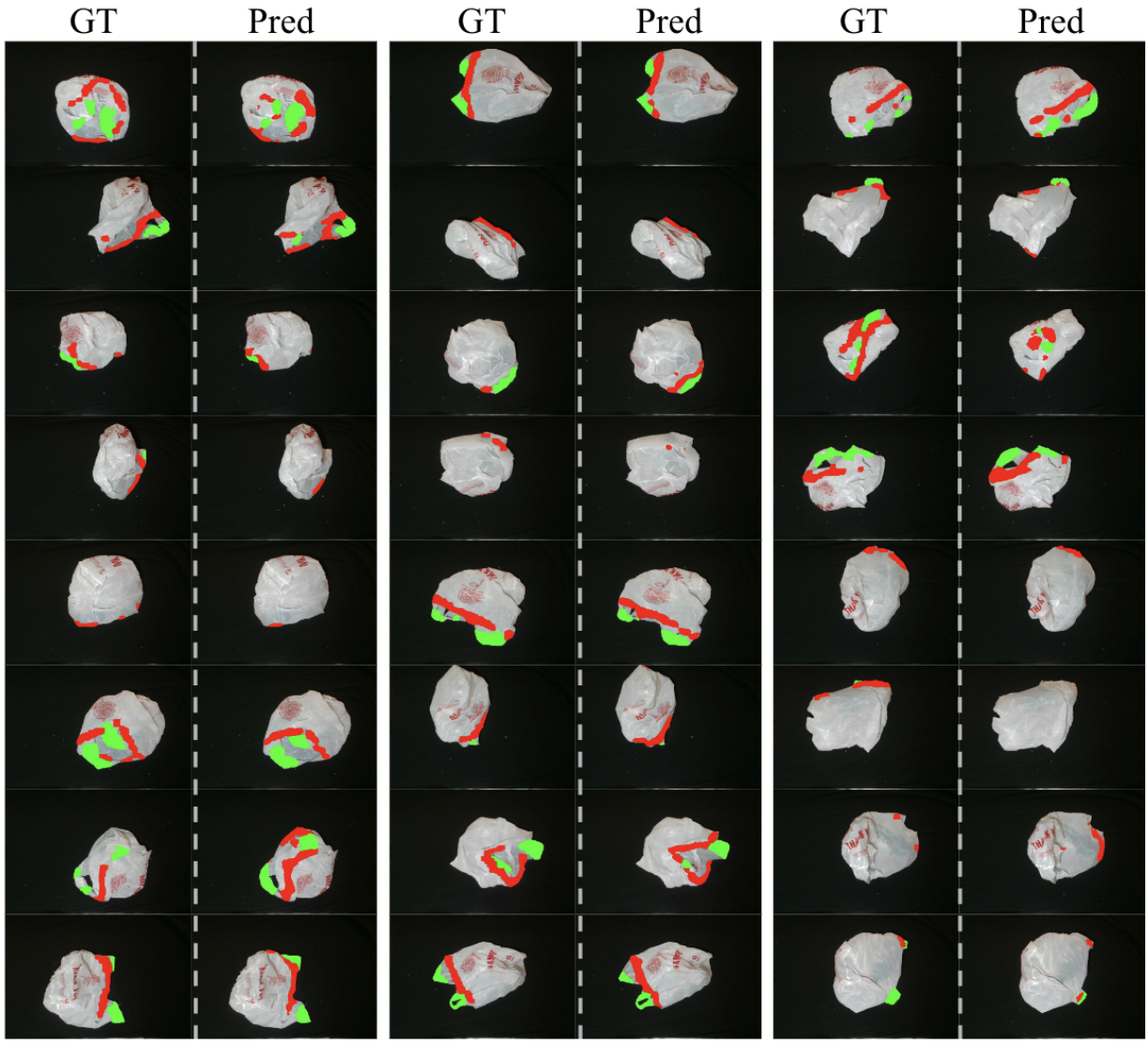

Figure A.2: Examples of the prediction of the perception model on the validation dataset. For each of the three columns, the left "GT" are the ground truth labels, and the right "Pred" are the model predictions. Both are overlaid on top of the color images, with green indicating handles and red indicating rim.

bag vertically 3 times. This action allows the air to come into the bag opening and inflate the bag. Then, the robot's right gripper releases the handle to grasp the object. Finally, the robot inserts the object at the center of the estimated bag opening while the other gripper still holds the bag.

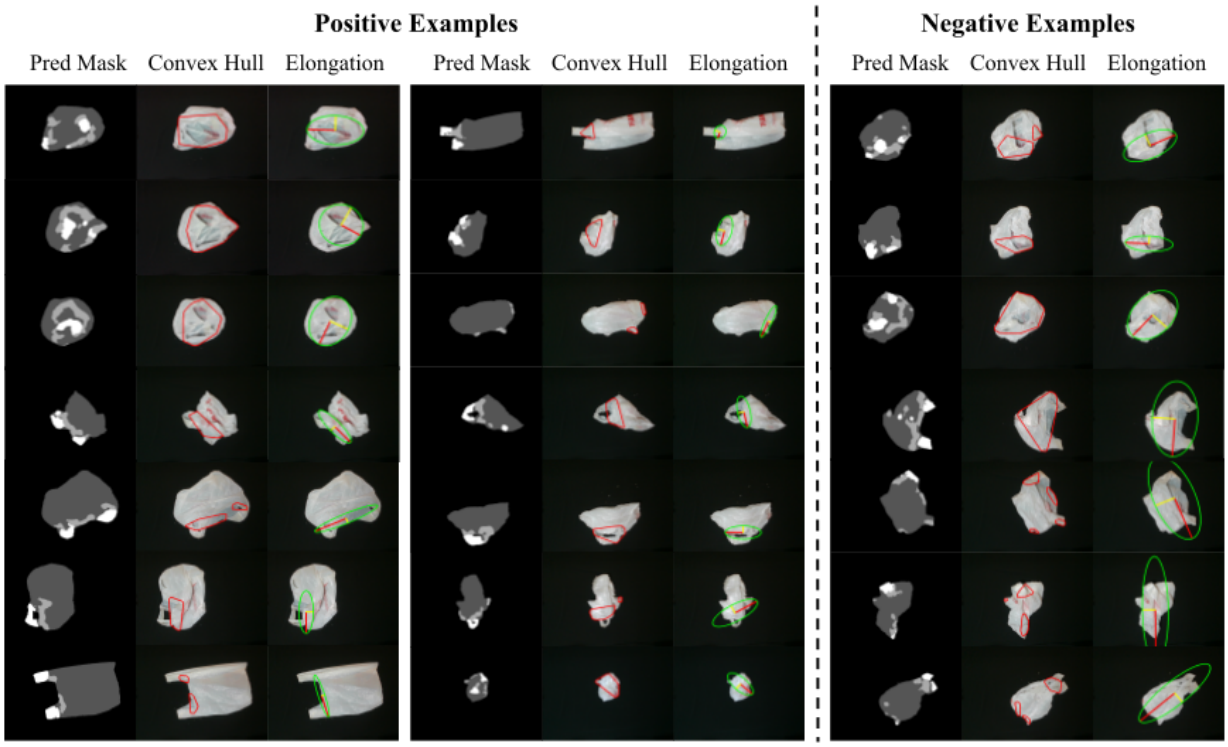

Figure A.3: Positive and negative example predictions of bag segmentation and illustrations of the metrics (convex hull and elongation) applied to the predicted rim.

A.5 Failure Modes for AutoBag

Here, we describe the details of the failure cases for AutoBag and baselines.

For AutoBag, we divide the failure modes into 5 categories.

(A) is caused by both manipulation and perception challenges during the long sequences of manipulation steps. In particular, during manipulation, the gripper may miss the grasp if the grasp region is too flat on the surface to have enough friction, or the bag may slip out of the gripper during dynamic actions such as **Shake** and **Compress**. Additionally, the perception module is not always robust and may sometimes only recognize part of the rim. This leads to the convex hull approximation of the opening underestimating the true opening and causes the robot to perform more Stage 1 and Stage 2 actions than necessary. For example, after **Compress** and **Flip**, even when the bag has a large opening, the robot may fail to recognize the entire rim region, leading to the metrics not meeting the thresholds for Stage 2 and causing the robot to repeat the Stage 1 actions. Another example of unnecessary actions can happen during **Rotate** and **Dilate**, where the perception module only recognizes part of the opening, causing the robot to keep rotating and dilating the bag. While this conservatism is not fatal on its own, the increased action steps lead to a larger chance of

failure rate due to imperfect manipulation, as the opening can easily close again during manipulation. For example, during **Dilate**, the friction of the bag with the two grippers may not be the same, causing the bag to slide along with one gripper while the other gripper slips. This moves the bag off the center of the workspace. Also, **Dilate** is effective only when the two grippers start inside or close to the opening. When the perception module fails to recognize the entire opening, the estimated center of the opening will be inaccurate, causing one gripper to start inside the opening and the other gripper to start far from the opening. When sliding in this asymmetric configuration, the grippers not only fail to enlarge the opening but also tend to compress and realign the separating rim of the bag, making it more difficult for future **Dilate** actions to enlarge the bag.

(B) is caused by the motion planning of the robot. On one hand, YuMi has a limited workspace and collision-free path planning of the two arms is not always successful, especially around singularities. On the other hand, it is difficult to incorporate the bag as an obstacle for motion planning due to inaccurate depth perception and protruding loop handles. As such, the gripper may occasionally hit the bag before and after each manipulation step by accident, pushing the bag out of the workspace.

(C) is caused by inaccuracy in the rim prediction. When the perception module's prediction of the rim has either Type 1 or Type 2 errors, the estimated convex hull will deviate from the true opening. In such cases, some objects may not be placed inside the true opening.

(D) is most often caused by non-ideal grasp positions when lifting the bag. In particular, the bag handles are often occluded by the inserted objects or hidden inside the opening after **Dilate**. In such cases, the robot simply chooses the left and right endpoints of the bag boundary. However, these grasp points may be near the bottom of the bag and lifting at those positions will make the bag upside down, so objects already inside the bag will fall out again. As the objects can all be contained inside the bag only when both grasp points are near the handles or the rim of the bag, poorly chosen grasp points of at least one gripper will prevent full success.

(E) is caused by slipped grasps when the bag contains the inserted objects and has nontrivial weights. This is more due to the mechanical limitations of the YuMi robot, it has a limited (500 g) payload, not-so-strong gripping force, and poorly designed jaw grippers that will slightly tilt and open at the end when a large force is applied at the top of the metal clips to close them.

On the test bag, the perception module is the main bottleneck since it has never seen the bag before, leading to many cases where it only recognizes part but not all of the rim. As such, (A) is the dominant reason for failure.

A.6 Failure Modes for Baselines

Next, we describe the failure modes of the baselines.

AB-P: Without using the perception module, the **Dilate** action often fails to position both grippers inside the opening. As mentioned earlier, when one gripper is inside the

opening while the other is outside, the **Dilate** actions can inadvertently close the bag opening due to asymmetric forces. In addition, without the rim area and elongation estimation, the robot cannot distinguish between a large bag and a large open bag, leading to inappropriate insertion actions even when the bag is not open.

AB-A_{CH}: Without using the size of the opening as a criterion for readiness to insert objects, the robot often prematurely attempts to insert objects when the opening is small but round. As a result, it is able to successfully insert 1 bottle half of the time but fails to insert both bottles in all trials.

AB-E_{CH}: Without using the roundness of the opening as a criterion, the robot would sometimes stop dilation even though the opening is very narrow. In those cases, the robot will fail to insert the objects inside the opening or lift the bag up with the objects contained since there is very little room for error.

Side Ins.: We observe that, unless there is already a large separation between the two layers of the bag (distance-wise) so that the pinning gripper can firmly and precisely press only the bottom layer, pulling the top layer of the bag almost always grasps the two layers together simultaneously and fails to create a sideways opening. As the layer of the plastic bag is quite thin, the pinning hand needs to exert a strong force towards the bottom layer against the table but not touch the upper layer, a stringent initial condition that is almost never satisfied with normal sideways-facing bags.

Handles: Failure cases can be categorized into two main reasons: (1) The fling actions fail to open the bag, and (2) the opening closes and the bag folds on itself after one gripper releases the handle. In particular, (1) is the main reason. For (1), we observe that the fling actions can only open the bag when both grippers grasp one layer of the handles (insert into the ring regions of handles). When a gripper grasps both layers of the handle, it effectively clamps the two layers of the bag together, making it difficult for the fling action to inflate the opening. For (2), we observe that even if flinging successfully inflates the bag, the opening closes as soon as one gripper releases the handle to grasp the inserted object. This is because the handle is heavy and when released, it collapses onto the bag, causing the entire bag to fold itself and the opening to close. As such, there is little room for the robot to insert the objects. Inserting the objects along the angle of the tilted opening could be helpful, but this may require more than an overhead camera to make the insertion direction very precise.