Tactile Sensing applied to Robot Manipulation

Me

CMU-RI-TR-23-53

June 22, 2023

figs/cmu_seal.png

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor David Held, *chair*
Professor Zackory Erickson,
PhD Raunaq Bhirangi

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

To my favorite robot.

Abstract

Robotic manipulation of cloth has applications ranging from fabrics manufacturing to handling blankets and laundry. Cloth manipulation is challenging for robots largely due to their high degrees of freedom, complex dynamics, and severe self-occlusions when in folded or crumpled configurations. Prior work on robotic manipulation of cloth relies primarily on vision sensors alone, which may pose challenges for fine-grained manipulation tasks such as grasping a desired number of cloth layers from a stack of cloth. In this paper, we propose to use tactile sensing for cloth manipulation; we attach a tactile sensor (ReSkin) to one of the two fingertips of a Franka robot and train a classifier to determine whether the robot is grasping a specific number of cloth layers. During test-time experiments, the robot uses this classifier as part of its policy to grasp one or two cloth layers using tactile feedback to determine suitable grasping points. Experimental results over 180 physical trials suggest that the proposed method outperforms baselines that do not use tactile feedback and has better generalization to unseen cloth compared to methods that use image classifiers.

Acknowledgments

First and foremost I would like to thank my parents for their incredible financial and moral support throughout my education. It is rare, that one can talk to family members about their research, rarer still when they do not have said background in your research area. In spite of this, my father made it his priority to try to understand my research area, to understand the problems I was struggling with and to give me prescient advice on overcoming them. This is a debt that cannot be repaid, a scale that cannot be balanced, and one I can safely spend the rest of my life working towards.

I would also like to thank Prof David Held, my research advisor, he often gave me as much attention as required, if not more attention than necessary and I never felt that I ever liked direction or guidance throughout my research career here. I would like to thank my committee Prof Zackory Erickson and Raunaq Bhirangi for helping me shape the thesis document to what it is today.

Through-out my time here at Carnegie Mellon University, I have been blessed to received close monitoring from a team of exceptional mentors through whom I could learn how to be more productive, organized and effective with my time. Thomas Weng, my research mentor has been instrumental in helping me learn the best practises required to drive a research project to completion. I grew a lot from his pointed advice and by simply imitating his style of work. Daniel Seita, another research mentor has been an inspiration of productivity, time management and taught me the importance of maintaining a positive attitude in front of looming deadlines. I have also also been fortunate to have some of the smartest and driven peers who motivated me to achieve more than I thought possible. I would like to thank all members of my lab cohort, Fan Yang, Carl Qi, Sarthak Shetty, Edward Li, Mansi Agrawal and many more.

I would also like to thank all the incredible friends I made throughout my masters here and while this is an incomplete list I would like to give a shoutout to Ravi, Soumith, Vineet and many more for their companionship and engaging conversations on technical topics that improved my knowledge.

Funding

We thank LG Electronics, the NSF CAREER grant IIS-2046491, the NSF grant CMMI-2024794, NSF S&AS grant IIS-1849154, and the NSF GRFP (DGE1745016, DGE2140739) for funding. This work was also supported by the CMU Argo AI Center for Autonomous Vehicle Research.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Tactile sensing is one of the most important forms of sensing for humans. Multiple studies such as Westling and Johansson [56],van Erp and van Veen [52], have shown that without tactile sensing humans are incapable of performing simple everyday tasks, such as performing a stable grasp of objects or using tools. However most industrial robots and even research robotic arms do not incorporate any form of tactile sensing. The reasons for this are varied and numerous. A non exhaustive list of these reasons would be:

- Tactile sensors tend to break easily when subjected to excess shear or longitudinal forces.
- Tactile sensors are quite expensive for the small patch of area that they generally cover.
- Tactile sensors barring a few, do not generally provide the resolution capable by the human skin.
- Tactile Sensors cannot replicate the sensing of diverse modalities such as pressure, temperature, electrostatics which human skin is capable of doing.
- Often tactile sensor data is hard to interpret and does not directly measure values we may care about such as force, pressure etc.
- Current tactile sensors are not very stable and we often observe a change in the data distribution over time.

- A general lack of appreciation of the advantages tactile sensors provide to robotics.

We explore these above challenges and strategies to overcome these challenges in detail in this thesis. Particularly we focus on using tactile sensors in the manipulation of deformable objects such as cloths.

A deformable object is an object that does not have a fixed shape, and can easily change shape upon the application of external force. Examples of deformable objects are liquids, clay, elastic objects, fabrics etc;. Manipulating them is quite different from manipulation of rigid objects, as we cannot use notions of rigid body dynamics and kinematics to describe the motion of deformable objects. For example a rigid object can be described in terms of it's 6-dimensional pose and we can use well understood Newton's Laws of Motion to calculate the change in the object's pose over time. For a deformable object such as a cloth, the 6D pose representation is no longer applicable. The ideal representation to describe the dynamics and kinematics of a cloth is an open research problem and you can refer to Long et al. [27] to get a survey of this field. Once we have determined the appropriate representation, (such as say meshes) we also need to determine the appropriate physics model to simulate the cloth. This too is an open research problem with a variety of methods such as Finite Element Analysis [49], Position Based Dynamics [37] etc being used to model cloth dynamics.

Due to the difficulties in cloth representation and dynamics, researchers have tried to apply learning based methods to robot cloth manipulation. This is the approach that we also use in this thesis document, where instead of modelling cloth dynamics in high detail, we instead apply machine learning algorithms that learn a model from large amounts of collected data in order to manipulate cloth.

In robot learning one dominant paradigm is the "sim-to-real" approach. This involves simulation of the task with an accurate physics model and learning a policy in simulation that is then transferred into the real world. Compared to simulation of rigid objects, it is much harder to simulate deformable objects. Progress has been made in simulating deformable objects, particularly with Huang et al. [17], Lin et al. [26], Hu et al. [16] however in our experience, there are still a lot of difficulties in trying to implement the sim-to-real approach for deformable object manipulation. In addition, simulating realistic tactile sensor data is an open research problem as of writing this thesis. Thus in this thesis we focus on data collection in the real world. We develop a system to collect millions of annotated tactile sensor datapoints automatically and use the collected data to train tactile classifier algorithms. Once we have collected the annotated tactile sensor we need to determine an appropriate method of processing this data. In this thesis we experiment with both classical machine learning and deep learning algorithms to process the tactile sensor data. We find specialized deep learning architectures to handily beat classical machine learning algorithms, however naive architectures such as fully connected multi-layer perceptrons perform poorly in processing annotated tactile sensor data. We perform a detailed ablation study on our deep learning network architectures to aid future researchers in choosing an appropriate architecture for processing tactile sensor data.

While not a major focus of this thesis, we use a novel gripper mechanism in manipulation of deformable objects. The gripper mechanism is based on the proposed architure from Mannam et al. [33]. It consists of 2 parallel-linkage based delta robots and we find the additional degrees of freedom afforded by the novel gripper to be useful in manipulation of fine deformable objects. A brief overview of the mechanism and it's advantages is provided background section of this thesis.

All the above collection, analysis and processing of tactile sensor data culminates in developing a robotic system capable of grasping a specific number of layers of cloth from a stack of cloth present on a table. We show significantly improved performance using our system, as opposed to a system based on cameras and vision sensing. This led to a paper publication and a best paper award in a workshop on Deformable Object Manipulation in IROS 2022 conference.

The outline of my thesis is as follows:

- Chapter 2 covers relevant background to understand my thesis. I discuss the various available tactile sensors that are being used today and the different trade-offs involved in using them. I give a short overview of our unique compliant gripper and finally go through the relevant machine learning algorithms for my thesis.
- In Chapter 3 I discuss how I applied the ReSkin sensor to grasp a specific number of layers of cloth from a stack of cloth and how this compares against techniques that use primarily vision based sensing.
- In Chapter 4 I discuss what is still holding tactile sensors back from being widely

used in industry and by researchers. I highlight potential research directions that can be pursued by interested researchers.

Chapter 2

Background

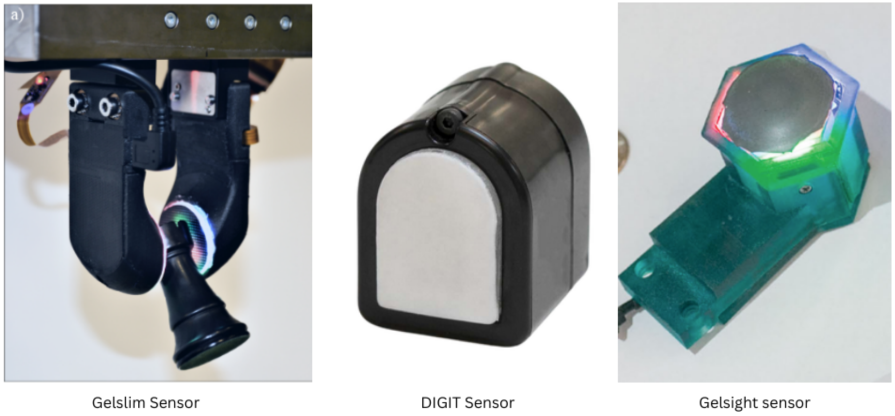
In this chapter we give a brief overview of different Tactile sensors, prior research work done on cloth manipulation, a short overview of the mechanisms used by our gripper system and a small primer on machine learning and deep learning required to understand the results of our thesis.

2.1 Different Types of Tactile Sensors

We first start with a brief review of the available tactile sensors for robotics and the considerations involved in choosing an appropriate sensor for your task:

2.1.1 Optical Gel Based Sensors

This class of sensors uses small cameras to collect images of deformation of a gel patch when in contact with objects. These images are then processed to determine pressure, texture, force and other tactile properties of the object. Examples of such sensors include GelSight [62], GelSlim [6], and DIGIT [22], which have been used by researchers for cloth perception, such as in Yuan et al. [64], Yuan et al. [63] and Luo et al. [28]. The prime advantage of gel sight based sensors are that these sensors are capable of providing much higher resolution data than other types of sensors, making them a good candidate for manipulation tasks that require high fidelity. These sensors are also easy to setup and use, requiring the user to only connect a USB cable to



Gelslim Sensor

DIGIT Sensor

Gelsight sensor

Figure 2.1: Different Optical Gel Based sensors. Gelslim, Digit and Gelsight from left to right.

a computer to read the sensor data with no additional circuitry or installation of firmware. If the system is capable of reading data from a webcam, it can read the images generated by the GelSight tactile sensor. The gel being deformable makes this sensor a good fit for handling soft objects. On the negative side, these sensors tend to be bulky as they need to house a camera inside them. This makes them unsuitable for handling fine objects such as thin cloth. They are also fragile, the gel being quite susceptible to shear force and tearing. It is also not straightforward to parse the data produced by these sensors, and often deep convolutional networks are used to parse the tactile image data as shown in [63] . Images of this sensor are shown in Figure 2.1. Examples of data collected by this sensor is shown in Figure 2.2.

2.1.2 ReSkin Sensor

Recently, Bhirangi et al. [1] proposed the ReSkin, a class of magnetic sensors for tactile sensing. The ReSkin consists of 5 magnetometers capable of measuring the X,Y and Z components of the magnetic flux in the near vicinity of the sensor. It also consists of a ”magnetic skin”, a thin deformable foam sprinkled with magnetic particles. The deformation of the skin causes changes in the readings of the magnetic flux measured by the 5 magnetometers. The changes of these readings can later be parsed to estimate tactile sensing data. One of the major advantages of using the ReSkin is it’s form factor, with a thickness of $< 3mm$, ReSkin are one of thinnest

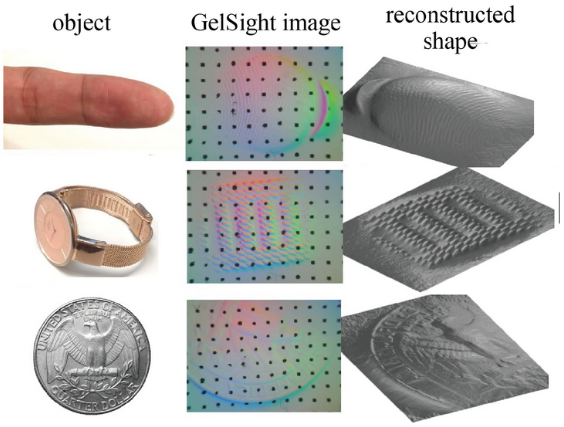


Figure 2.2: Image taken from [62]. The gel is peppered with uniformly distributed black dots. By measuring the displacement of the dots from the image collected by the camera, we can estimate the 3D structure of the object in contact. Note the high resolution of the 3D structure obtained by the Gelsight sensors.

tactile sensors aailable and are quite suitable for manipulating fine objects like thin fabrics. ReSkin is also relatively low cost, and can cover relatively large surface areas compared to Gelsight sensors. The main disadvantage of the Reskin sensor is it's lack of resolution. The ReSkin data at every timestep is a 15 dimensional magnetic flux vector. The sensor tries to solve it's lack of resolution by collecting data at a much higher frequency than other sensors. The frequency of the ReSkin sensor is 400Hz compared to 60Hz of the GelSight sensor. Thus when using the ReSkin sensor, we consider a timeseries of data as opposed to data from a single time-step. ReSkin data is also very hard to interpret. Often ReSkin data can only be interpreted using Machine Learning Algorithms, and there is no simple conversion from magnetic flux to contact location/ force etc. In this thesis we use the ReSkin sensor for cloth manipulation, primarily because of it's superior form factor and resolution being sufficient enough for our task in hand. In general if the task at hand requires only a single contact region, such as grasping then we find the ReSkin data to be adequate. Figure 2.4 shows the reskin sensor and Figure 2.3 shows an example of the data collected by the reskin sensor.

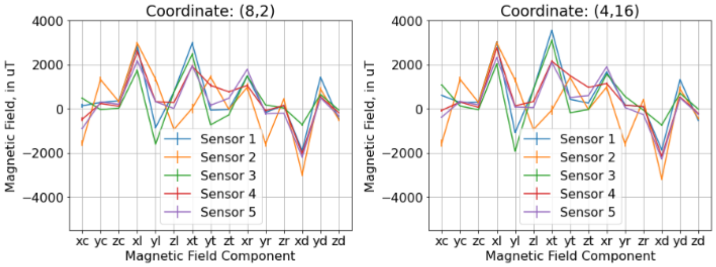


Figure 2.3: Example of the data collected by a ReSkin Sensor

2.1.3 BioTac Sensor

BioTac [48] is a popular tactile sensor used in industry and research. It is a multi-modal tactile sensor capable of measuring forces, vibrations and heat-flow of the elastomeric skin that comes into contact with various different objects. This fingertip shaped sensor consists of a flexible rubber skin, ionically conductive fluidic layer and a rigid core. The sensor makes it measurements with 23 electrodes located on the outer surface of the core. As the fluidic layer changes shape upon contact, the voltages at the electrodes change in a complex manner which can be parsed to determine the surface contact points and the forces involved on the sensor. The main advantage of this sensor is it's multimodal measurements and commercialization. BioTac is robust, accurate and used by many industrial robots in the automotive, cosmetics and apparel industries. The main disadvantage of using this sensor, is that the skin is fragile and often requires reapplication in between trials. It is also quite expensive at around 10000$ per sensor. We choose to not use BioTac sensors because they are expensive, bulky and not easily replacable. The image of the BioTac sensor is present in Fig 2.5.

2.1.4 Other Types of Sensors

There are several other types of tactile sensors that have been developed for various applications. Some examples include:

- Resistive sensors: These sensors use a conductive material that changes resistance when deformed by touch. They are low-cost and simple to implement, but their resolution and sensitivity may not be as high as other types of sensors.

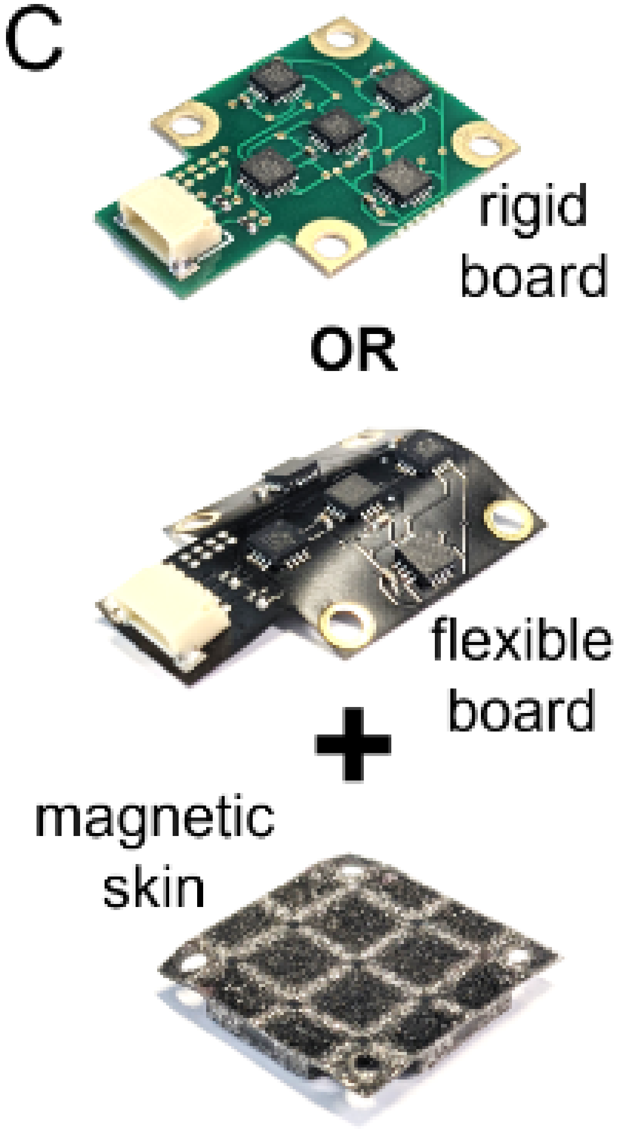


Figure 2.4: Images taken from [1], A cross sectional view of the ReSkin sensor

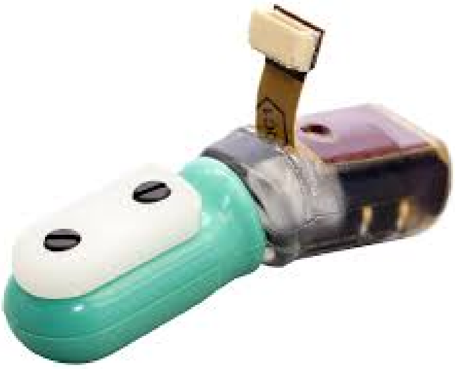


Figure 2.5: Image taken from company website. This image depicts the bioTac sensor.

- Capacitive sensors: These sensors measure changes in capacitance when a material comes into contact with them. They can provide high-resolution data and are commonly used in touchscreens. They however are not deformable but are instead brittle, thus making them not very appropriate for robot arm use cases.
- Piezoelectric sensors: These sensors generate an electric charge when mechanical stress is applied to them. They are sensitive and can provide fast response times, and are used in BioTac. However they are quite expensive and require significant signal conditioning and tuning to be used.
- Force-sensitive resistors: These sensors change resistance based on the force applied to them. They are simple and inexpensive, but their accuracy and resolution is limited. They often only measure force at a specific contact point.

To obtain a more detailed review on all the available tactile sensors for manipulation, please refer to Roberts et al. [41],Yamaguchi and Atkeson [58] and Dahiya et al. [4].

2.1.5 Determining the appropriate sensor for our use case

We choose to compare the sensors on five characteristics as shown in Table 2.1. We compare each sensor along form factor, cost, sensitivity, robustness and resolution. We find that for our task of grasping a specific number of layers of cloth, we do not need high resolution and are fine with medium sensitivity, cost and robustness. We

Name	Thickness (mm)	Sensitivity (N)	Robustness	Resolution (mm)	Cost ($)
GelSight	40	0.05N	Low	**0.03**	800
BioTac	30	**0.01N**	High	2	10000
ReSkin	**3**	0.2N	**High**	1	**6**

Table 2.1: Comparison of the sensors across the five criteria is shown above.

do however need a very small form factor. The ReSkin sensor has the smallest form factor of all the sensors we compared and thus we chose to go ahead with the ReSkin sensor for our task. The higher robustness, and extremely low cost of the the ReSkin sensors are an added bonus.

2.2 Cloth Manipulation with Robot Learning

Manipulation of deformable objects such as cloth has a long history in robotics; see Yin et al. [60] and Zhu et al. [65] for representative surveys.

2.2.1 Cloth Manipulation Policies

In early research on cloth manipulation, a common strategy was to utilize a bimanual robot to grip cloth in midair to smooth it using gravity. This standardizes the configuration of cloth and exposes its corners, which can then facilitate planning subsequent manipulation tasks such as smoothing and folding [7, 31]. Other researchers have relied on using geometric features of cloth, such as by fitting polygon contours to clothing [36]. While these works showed impressive results, such approaches may be time-consuming or require strong assumptions on cloth configurations. With the rise of deep learning, researchers have recently employed data driven techniques to obtain large amounts of interaction data with cloth to learn manipulation policies using powerful function approximators, often with the help of simulators [24, 50]. These works tend to learn quasi-static pick-and-place policies, which allow the cloth to settle between robot actions [11, 15, 25, 29, 38, 44, 45, 55, 57, 59]. Other researchers have learned continuous servoing policies [34], dynamic policies [13] or have explored learning cloth manipulation from purely real world interaction [23].

In contrast to these works which employ vision-manipulation policies, we focus on tactile sensing for cloth grasping.

2.2.2 Grasping for Cloth Manipulation

Perhaps the most important part of cloth *manipulation* tasks is cloth *grasping*, since a suitable grasp is necessary for subsequent actions such as dragging or lifting. Defining and identifying ideal cloth grasps remains challenging and is the subject of extensive research [2]. Early cloth manipulation research focused on vertically smoothing via gravity. A common such grasping strategy to reliably standardize cloth was to hold it with one gripper while iteratively grasping the lowest hanging corner with the other gripper [3, 19, 20, 31].

Other cloth grasping techniques do not require assuming that the cloth is lifted in midair. For example Ramisa et al. [40] and Sun et al. [47] determine suitable grasping points for cloth on a flat table by detecting wrinkles and edges using depth and classical computer vision techniques. Other applications of cloth manipulation may utilize specialized gripper designs [21] or may simplify the process by assuming that cloth is gripped in advance of the task [18].

Recently, Qian et al. [39] study how to robustly grasp cloth using dense segmentation of images to distinguish between edges and interior creases. Their method involves a self-supervised labeling procedure and a sliding grasp. Nonetheless, robustly grasping cloth remains challenging, particularly when the goal is to generalize to a wide variety of types and configurations of cloth. Prior work has reported that a typical failure cases is grasping the wrong number of cloth layers, particularly when unfolding [34, 44, 55]. Furthermore, many works employ heuristics such as hand-tuning the gripper design and grasp depth [11].

Prior work has also investigated learning to grasp one cloth from a stack using grasping and scooping actions from vision input only [5], as well as designing a robot system to turn a single book page using vision and force sensing [12]. In this work, we consider the novel task of grasping more than one cloth layer, and show the benefits of tactile sensing without requiring vision.

2.3 Specialized Grippers for Fine Manipulation

The human hand is extremely impressive in it's degrees of freedom, compliance and robustness that researchers have tried to match with their grippers over the years. Building grippers with the same properties as the human hand has turned out to be extremely difficult. At the same time simple parallel jaw grippers used extensively in industry is often found inadequate for fine manipulation tasks that we desire of our robots. In this section we give a brief overview of the available grippers and the reasoning behind using the novel Delta Grippers in this work.

2.3.1 Anthropomorphic Dextrous Hands

The simplest solution to creating grippers capable of fine manipulation is to mimic the design of the human hand. There are a number of popular grippers that aim to do that such as the shadow hand and Allegro Hand. These arms are capable of mimicking the degrees of freedom of a human hand but face a number of challenges in usage for fine manipulation tasks such as:

- They are not robust and are prone to breaking very easily upon application of high forces
- They are extremely expensive. Shadown Hand costs around $100,000$\$, Allegro Hand 25000\$ and so.
- The high degrees of freedom creates additional challenges of control due to the large number of possible points of contact.

Due to the above reasons we choose not to go ahead with anthropomorphic dextrous hands for this project.

2.3.2 Delta Gripper

Delta Robots are parallel linkages that are capable of three translational degrees of freedom as described in Merlet [35]. These robots are regularly used in industry to pick and place objects from conveyor belts due to their high speed and accurate placement. The Delta Gripper is a novel gripper made from two delta robots where each finger has 3 degrees of freedom in the translational axis. Thus the delta gripper

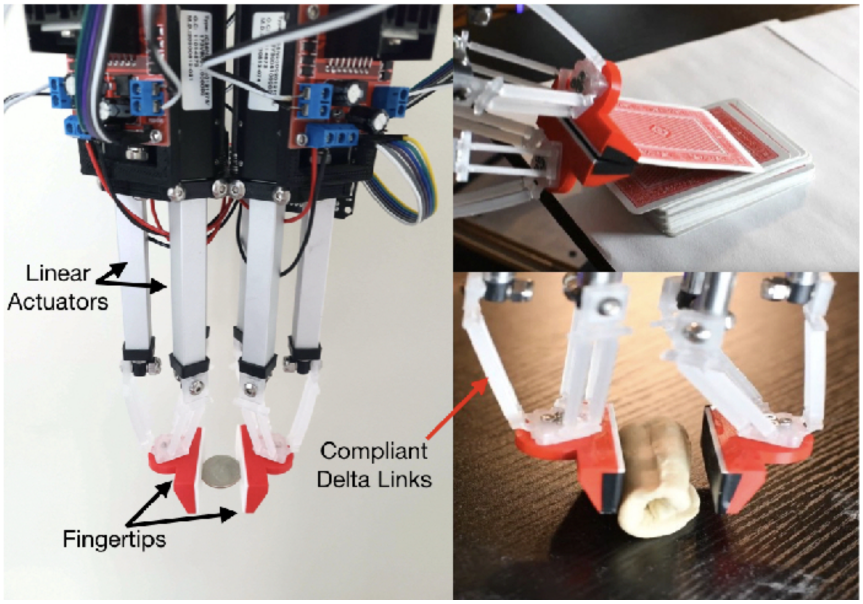


Figure 2.6: Image taken from [33] depicting the specialized gripper used in our setup.

possesses 6 actuators. The specific delta gripper used in this paper uses linear actuators, it is however possible to use rotary actuators though this will cause a larger form factor than using linear actuators. The materials used for the gripper links is TPU, Thermoplastic Polyurethane which is very compliant compared to hard plastic and metals. The joint is created using a thin plastic layer that acts as an hinge to minimize the number of moving parts in a structure. The entire gripper can be built for 300$ as opposed to the $100,000$$ required by the shadow hand and $30,000$$ required by the allegro hand. An arduino is used to interface with the linear actuators and the gripper is setup with it's own power source. An image of this gripper is present in Fig 2.6

2.4 Machine Learning and Deep Learning Primer

2.4.1 Classical Machine Learning

A number of classical machine learning algorithms are used in this thesis. We give a brief description of each algorithm we use in this thesis below. To determine the most appropriate algorithm we calculate statistics such as F1 Score and accuracy in real world robot trials across the different algorithms and datasets.

K-Nearest Neighbours

K-Nearest Neighbours (K-NN) is a non-parametric algorithm that classifies a data point based on how its neighbors are classified. The algorithm works by finding the k closest points to the data point in question and classifying it based on the majority class of its neighbours. It can be applied to both classification and regression problems. The advantages of using this algorithm are its simplicity and lack of assumptions about the underlying data distribution. However, it is computationally intensive, especially for large datasets, sensitive to irrelevant features and the scale of the data, and the choice of k can greatly affect the results.

Logistic Regression

Logistic Regression is a statistical method used for modeling binary outcomes. By using a logistic function, it models the probability that the dependent variable belongs to a particular category. The advantages of using Logistic Regression are its simplicity, interpretability, and efficiency, requiring less computational resources. It is particularly well-suited for binary classification problems. However, it may struggle with non-linear relationships, be prone to overfitting in high-dimensional datasets, and is sensitive to outliers.

Support Vector Machines

Support Vector Machines (SVM) are supervised learning models used for classification and regression. By finding the hyperplane that best divides a dataset into classes, SVM is particularly effective in high-dimensional spaces. The advantages of using SVM are that they are effective in high-dimensional spaces, robust to outliers, and can handle both linear and non-linear relationships between variables. However, SVM requires careful tuning of parameters, is computationally intensive, particularly for large datasets, and may be difficult to interpret.

Random Forest Classifier

Random Forest is an ensemble learning method that consists of numerous decision trees. The more uncorrelated trees in the forest, the more accurate the model. Random Forest is particularly known for its flexibility and can handle a mixture of numerical and categorical features. The advantages of using the Random Forest algorithm are its high predictive accuracy, robustness to overfitting, and its ability to effectively handle both classification and regression tasks. On the other hand, it can be slow in creating predictions once trained, may be complex, require more computational resources, and necessitate careful tuning of different hyperparameters.

2.4.2 Deep Learning

Deep learning is a relatively new paradigm that employs neural networks and large datasets to learn complex function approximators. Compared to classical machine learning algorithms, neural networks have demonstrated superior performance, particularly in handling large-scale data. This thesis bears witness to these results. There are numerous possible neural network architectures, and we describe a few of the architectures that we explore in this thesis.

Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is a class of feedforward artificial neural networks. It consists of at least three layers of nodes: an input layer, hidden layer(s), and an output layer. The advantages of using an MLP include its ability to model non-linear relationships and its adaptability to various kinds of data. However, MLPs can be prone to overfitting, especially with insufficient training data, and they often require careful tuning of hyperparameters, such as the learning rate and the number of hidden layers and nodes.

Convolutional Network

Convolutional Networks (ConvNets or CNNs) are a category of neural networks that have proven highly effective in areas such as image recognition and classification. CNNs apply a convolutional layer that filters inputs to create feature maps, thereby preserving the relationship between pixels by learning image features using small squares of input data. The advantages of using CNNs include their ability to automatically and adaptively learn spatial hierarchies of features. They are particularly powerful for tasks like image classification. However, CNNs can be computationally intensive to train, and their performance is highly dependent on the careful tuning of hyperparameters.

LSTM (Long Short Term Memory)

Long Short Term Memory (LSTM) networks are a type of recurrent neural network (RNN) well-suited to classifying, processing, and predicting time series given lags of unknown size and duration between events. LSTMs are explicitly designed to avoid long-term dependency issues, making them effective for sequence prediction tasks. The advantages of using LSTMs include their ability to capture long-term dependencies in sequence data, handle sequences of various lengths, and work with a large amount of data. On the downside, LSTMs can be computationally expensive to train and may require substantial tuning and experimentation to find the optimal architecture and parameters.

Chapter 3

Applying Tactile Feedback to Cloth Grasping

3.1 Introduction

Cloth manipulation remains an active research area in robotics with significant real world applications, including folding laundry [7, 31], assistive dressing [8, 9, 10, 61], bed-making [43], and manufacturing fabrics [51]. Cloth manipulation is challenging because it is difficult to infer the complete configuration of the cloth from robot observations when the cloth is in a crumpled or folded state, due to the high degrees of freedom and self-occlusions [2, 42].

In light of these challenges, researchers have recently proposed numerous data-driven methods for canonical cloth manipulation tasks such as smoothing [44, 57] and folding [23, 34, 55]. While showing promising results, many prior works focus on top-down grasping of one cloth. Such grasping may be ineffective for manipulation tasks involving multiple cloths, such as picking a desired number of layers of a stack of cloth, because performance is extremely sensitive to the height of the gripper when it grasps. Indeed, a common failure case reported in prior work [11, 55] is picking the wrong number of layers. Yet, manipulating a specific number of cloth layers is common in daily life, such as in folding and unfolding tasks, or handling piles of stacked clothing in stores. How, then, can robots achieve accurate grasping of

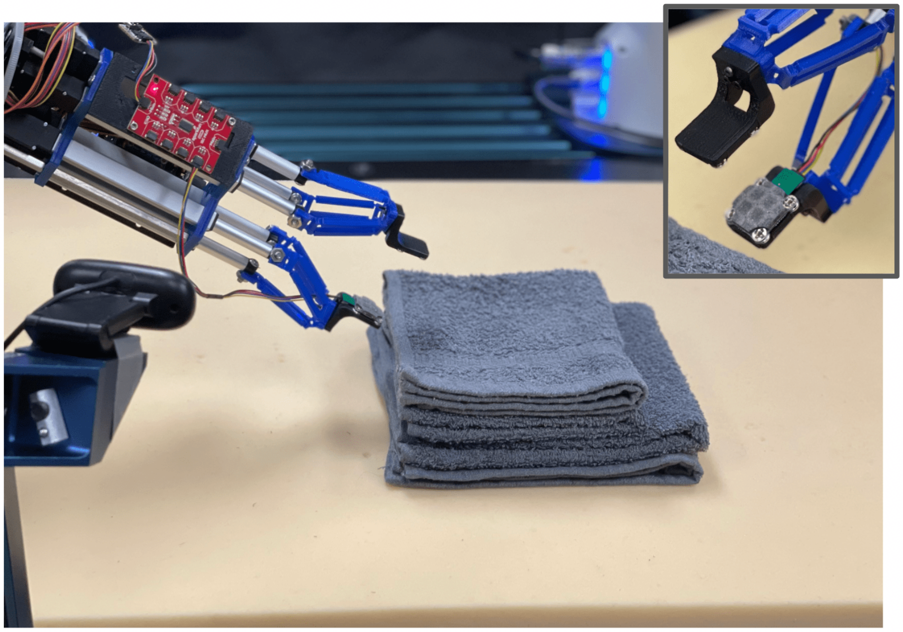


Figure 3.1: We present a tactile-based cloth manipulation system. The robot utilizes a ReSkin [1] sensor attached to the lower one of its two fingertips, which is visualized in more detail in the upper right inset. We train a classifier to distinguish among grasping different numbers of cloth layers from tactile feedback (no images are provided as input). The robot then uses this classifier at test time to determine suitable grasping points for obtaining a desired number of cloth layers.

multiple layers of cloth?

Incorporating tactile sensing is an under-explored direction for deformable object manipulation. While there has been recent work on optical-based tactile sensors such as GelSight [62] and DIGIT [22], these sensors have primarily been applied to cloth *perception* [28, 64] instead of cloth *manipulation*. Recent work on magnetometer-based sensors such as ReSkin [1] have benefits over optical sensors, such as lower-dimensional sensor readings, more direct measurements of normal and shear forces, and a compact form factor. However, research into the applications of *magnetometer-based* sensors for deformable object manipulation is currently limited.

In this chapter, we study the application of magnetometer-based tactile sensing for deformable cloth manipulation. We focus on precisely grasping and lifting layers of stacked cloth; due to the flexibility of cloth and unpredictable crumpling behavior, this task is challenging while being a well-defined manipulation problem. Furthermore,

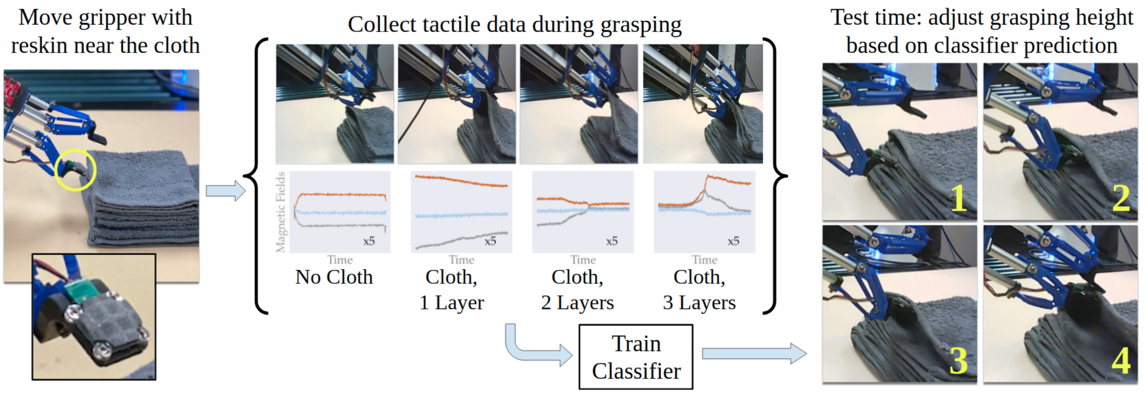

Move gripper with reskin near the cloth
Collect tactile data during grasping
Test time: adjust grasping height based on classifier prediction
Magnetic Fields
Time
x5
No Cloth
Cloth, 1 Layer
Cloth, 2 Layers
Cloth, 3 Layers
Train Classifier
1
2
3
4

Figure 3.2: The proposed tactile-based cloth manipulation pipeline. A 7-DOF Franka robot uses a mini-Delta [32] gripper with two finger tips, the lower one of which has a ReSkin [1] sensor (see yellow circle and zoomed-in inset). Using this gripper, we collect tactile data from the ReSkin by performing grasps of different categories: grasping nothing, or pinching 1, 2, or 3 cloth layers (see Fig. 3.3 for more examples). The graphs above visualize the tactile time series data. At test time, the robot uses the trained tactile-based classifier to grasp a desired number of cloth layers.

precise grasping of layers of cloth is a prerequisite for many downstream manipulation tasks (folding cloth in half twice).

We present a robotic system consisting of a 7-DOF Franka arm, a mini-Delta gripper [32], and a Reskin [1] sensor on the gripper finger to perform precise cloth grasping (see Fig. 3.1). The system uses a tactile classifier as feedback for a grasping policy. We show that simple approaches to both classifying tactile data and incorporating feedback into the policy (as a termination condition) work surprisingly well.

This chapter makes the following contributions:

1. A robot hardware system which incorporates ReSkin tactile sensors for cloth manipulation.
2. A training procedure for developing a classifier based on this hardware to use in a grasping policy.
3. Experiments showing success on the task of grasping a desired number of cloth layers.

3.2 Problem Statement

We study the task of grasping a desired number of layers from a stack of cloths. Given a set of at least 3 cloth layers stacked on each other, the goal is to grasp the top $k \in \{1, 2\}$ cloth layers. For each *trial* (a given instance of the task), we specify a target value for k. We assume a robot has a two-finger gripper where one of the gripper tips is equipped with a *tactile sensor*. We assume each trial begins with the robot's tactile sensor facing a set of edges from a stack of cloth layers, as shown in Fig. 3.1. A trial is a *success* when the robot grasps exactly k cloth layers and is able to lift its gripper upwards by 4 cm while preserving its grasp of the k layers.

3.3 Method

This proposed system for tactile sensing involves designing hardware with tactile data (Sec. 3.3.1), training a classifier to distinguish grasping cloth layers (Sec. 3.3.2), then using this classifier for a grasping policy (Sec. 3.3.3). See Fig. 3.2 for the overall pipeline.

3.3.1 Hardware

The proposed system uses a ReSkin [1] sensor, which comprises of a soft magnetized skin and a circuit board with a 5-magnetometer array (see bottom-left inset of Fig. 3.2). The board sits beneath the skin, and any deformations caused by normal/shear forces are read via distortions in magnetic fields. For each of the 5 magnetometers, 3 magnetic flux values $\langle B_X, B_Y, B_Z \rangle$ are reported, corresponding to flux in the X-, Y-, and Z- magnetometer coordinate axes. Concatenating these values for a single time step t results in a 15-dim vector $\mathbf{B}^{(t)} \in \mathbb{R}^{15}$. ReSkin publishes these values at up to 400 Hz.

We attach ReSkin to a finger on a mini-Delta gripper [32]. We use the mini-Delta largely due to its length and form factor, since it facilitates grasping a layered stack of cloth folds by approaching it from the side, instead of top-down. The mini-Delta has 3 DOFs for each finger, and is compliant due to the 3D-printed soft links (blue component in Fig. 3.1). The gripper and attached sensor are mounted on a 7-DOF

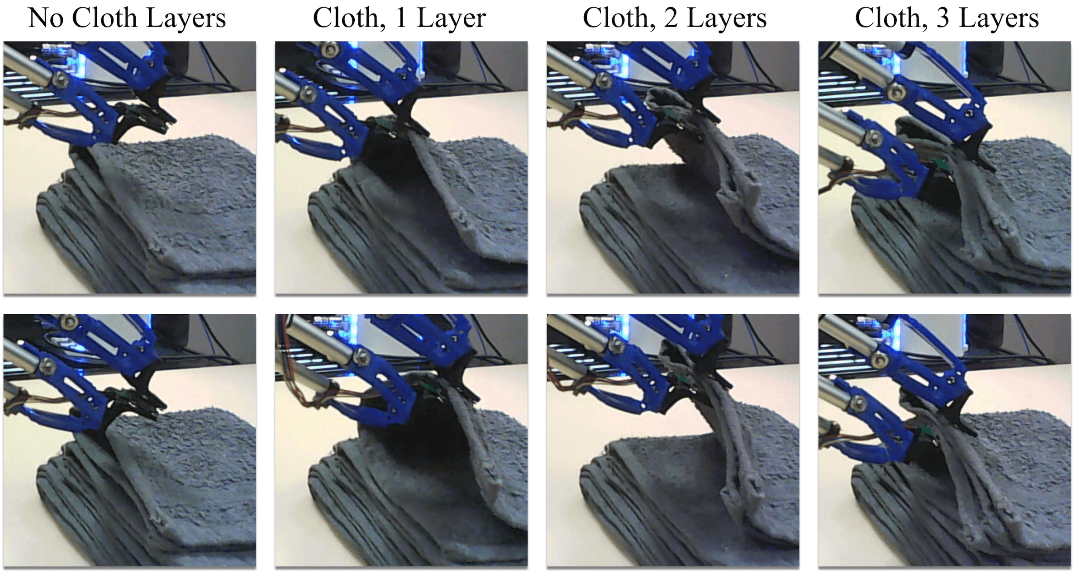


Figure 3.3: Examples of collecting data for tactile-based classification, with the ReSkin attached to the bottom gripper finger tip. From left to right, we show two examples each of collecting data with (1) contact, but without cloth, (2) 1 cloth layer, (3) 2 cloth layers, and (4) 3 cloth layers. The classifier only takes as input the data collected from the ReSkin sensor $\mathbf{B}^{(t)}$ at any give time step. The images above are collected with a webcam and are used both to visualize the tactile data collection, and also are the RGB inputs to the image classifiers that we train as baselines for comparison. See Sec. 3.3 and Sec. 3.4 for further details.

Franka robot.

3.3.2 Grasp Classifier Training

We train a classifier to predict the number of cloth layers grasped to use as part of the grasp policy (Sec. 3.3.3). The classifier takes as input a tactile reading from a single time step $\mathbf{B}^{(t)}$. While analyzing sensory data across a time series seems natural for the tactile modality, we find that predictions based on point estimates are surprisingly effective, as we later show in Sec. 3.5. We do not take proprioceptive data as input, as this modality is not currently available with the mini-Delta gripper: the compliant links can bend from their commanded position given sufficiently high external force, and estimating proprioception for these types of compliant links is an area of active research.

The classifier uses the tactile readings to predict how the gripper is interacting with the cloth, among 4 classes: (1) pinching with no cloth between the fingers, (2) pinching 1 cloth layer, (3) pinching 2 cloth layers, and (4) pinching 3 cloth layers. We limit the number of cloth layers under consideration to 3 to make classification tractable, while also allowing feedback-based policies to recover if they overshoot when grasping two layers. We leave classifying an arbitrary number of layers to future work.

We collect training data in the real world for the classifier due to the lack of a suitable simulator.[1] We define a single "training episode" as the process of getting a set of tactile data from one grasp. First, a human stacks several layers of cloth with edges facing the gripper. The height at which the robot approaches the cloth is uniformly sampled per attempt within a $\pm 2\,\text{mm}$ range to collect a variety of grasps. The robot then approaches the cloth, closes its fingertips to grasp firmly, records ReSkin data during the grasp, then releases. Each training episode lasts roughly 5 seconds and produces approximately 350 sensor readings of 15 values each (3 per magnetometer). We visually inspect videos from the recorded data to determine the number of grasped cloth layers, and we label all points from a training episode with the same label, speeding up annotation time and effort. See Fig. 3.3 for example visualizations of training episodes for all classes.

We then use this collected data to train a classifier to distinguish the numbers of layers grasped from the tactile readings. We experimented with various types of classifiers, including k-Nearest Neighbor (kNN), SVM, Logistic Regression, and Random Forests, and we found the performance to be fairly similar across classifiers. For simplicity, we use a k-Nearest Neighbor (kNN) classifier with $k = 10$ neighbors.

3.3.3 Proposed Grasp Policy

Next we describe how we use the above trained classifier to enable the robot to grasp the desired number of layers. We divide the robot trajectory into three parts. First, the gripper moves vertically down by a distance d_{vert}, then horizontally towards the cloth stack by a distance d_{slide}, then lifts up by a distance d_{lift}, then closes its gripper

[1] While there has been progress in developing high-fidelity simulators for tactile sensors [46, 54] and for deformables [24, 30], simulating both is challenging and to our knowledge has not yet been shown.

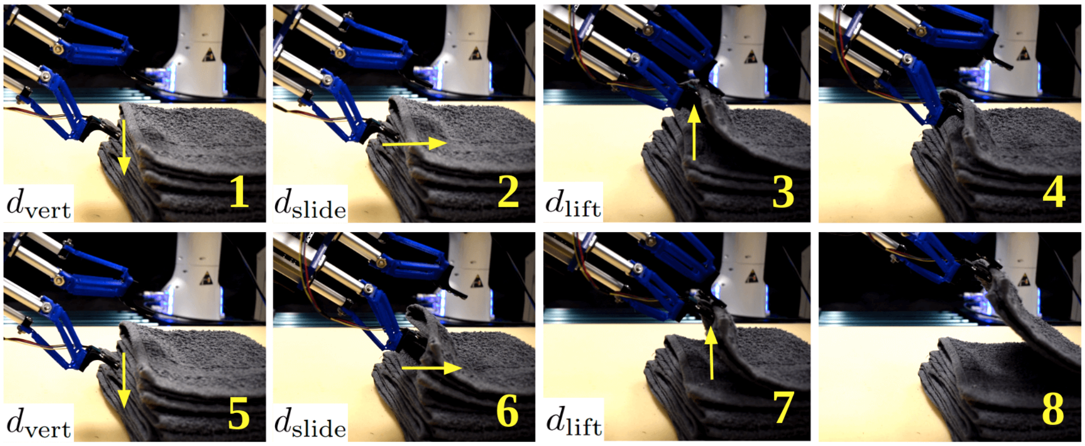


Figure 3.4: The proposed grasp policy parameterization (described in Sec. 3.3.3), visualized with a frame-by-frame overview of an example trial from the experiments. Each row, consisting of four frames, shows one action. The first part of an action (shown in frames 1 and 5) adjusts the initial gripper height by d_{vert}, possibly from prior tactile feedback. The second part of an action (shown in frames 2 and 6) moves towards the cloth stack by some distance d_{slide}. Then, the third part (frames 3 and 7) lifts upwards by d_{lift} and closes the grippers. At this point, the robot queries the classifier and may decide to release and re-attempt the grasp (frames 4 and 5) or the robot concludes that it has grasped the correct number of layers and further lifts the cloth to end the trial (frame 8).

tips (see Fig. 3.4 for a visualization). At this point, we record tactile data and classify the number of layers that are grasped. If the predicted number of grasped layers (according to the classifier) matches the target number of grasped layers, it lifts the gripper further by 4 cm to indicate the end of the trial; otherwise, it resets the gripper back to the starting position and tries again (see below for details). The values of d_{slide} and d_{lift} are tuned and fixed ahead of time by a human operator, while d_{vert} is determined by the policy, as explained below.

The grasping policy uses the output of the grasp classifier (Sec. 3.3.2) to determine the vertical distance that the gripper lowers before grasping, d_{vert}. For a target number of layers k to grasp, the robot begins at some height with the grippers open, moves towards the cloth stack, and attempts a grasp. If the grasp classifier determines that it has not grasped the correct number of layers, then the robot releases, moves back, and adjusts the gripper height (d_{vert}). If the classifier predicts that it has grasped too many layers, d_{vert} is decremented by a small value to decrease the grasp height; if it has grasped too few, d_{vert} is incremented by a small value. The policy continues until either the classifier determines that it has grasped the desired number of layers and ends the trial, or until the maximum number of grasp attempts is reached.

During each grasp attempt on the physical system, the classifier starts predicting the class once the gripper closes, and stops predicting after the robot lifts by d_{vert}. This results in a set of about 160 separate predictions. We use the mode of all the predictions as the final prediction to determine whether to raise or lower the grasp height.

3.4 Physical Experiments

We evaluate the methods using the physical system described in Sec. 3.3.1. The experiments are designed to answer the following questions:

- Can magnetometer-based tactile sensing with ReSkin sensors provide sufficient information about grasping a target number of cloth layers?
- What are the benefits of the proposed method that uses tactile-based feedback to adjust the gripper height?
- Can a classifier trained on tactile feedback generalize to different cloths?

3.4.1 Experiment Protocol

We train our tactile classifier on a gray cloth; we then evaluate our system on the gray training cloth and on two other unseen cloths to measure the generalization of our method to new cloths (see Fig. 3.6). We use the same training data from the gray cloth for all of the tactile-based method variations described in Sec. 3.4.2. The tactile data collection results in a total of 18,838 such $\mathbf{B}^{(t)}$ readings. We train a classifier on 95 % of the training episodes (to allow for a small validation set). We normalize the tactile data so that each of the 15 features has mean 0 and variance 1 in the training set.

We perform two sets of experiments, in which we set the desired number of cloth layers to grasp as one layer and two layers, respectively. Each *trial* begins with a human arranging folded cloths on the workspace with edges exposed and facing the robot gripper. The number of folded cloths is the same across trials, but variations in the depth of the layers up to 1.5 mm can occur due to slight differences in the initial cloth configuration. We initialize the robot's gripper at an angle (30°) which increases the likelihood that a horizontal motion can slide the robot finger tips in between layers of cloth.

Each experiment set consists of comparing several grasping methods (see Sec. 3.4.2). When running experiments, we randomly sample the method to run in the given trial *after* the cloths have been set, to reduce potential human bias in the data initialization. The robot employs the selected method to grasp the appropriate number of cloth layers. The robot is allowed up to $T = 10$ actions per trial, though it can terminate earlier if the classifier estimates that it has grasped the appropriate number of layers. Upon termination, the robot lifts the gripper by 4 cm and a human measures this as a success if the correct number of layers are still grasped. All other cases result in the trial as a failure.

We categorize failures into two types, *prediction* and *grasping* failures. Prediction failures are a result of mis-predictions by the trained classifier, where it either: (1) incorrectly predicts that the robot has grasped the desired number of layers and terminates the trial prematurely, or (2) the classifier incorrectly predicts that the robot has grasped the wrong number of layers, causing unnecessary regrasps and leading to the robot reaching the max number of attempts for the trial. Grasping failures are due to either failing to grasp the desired number of layers at the last time step in a trial, or failing to robustly grasp the cloth, such that cloth layers slip out of the robot's control when lifting (see Fig. 3.5).

3.4.2 Methods and Baselines

We evaluate the following methods for grasping 1 and 2 cloth layers:

1. **Fixed-Open-Loop**: Initialize the gripper at a fixed height, manually tuned for grasping 1 or 2 cloth layers: $d_{\text{vert}}^{(1)}$ and $d_{\text{vert}}^{(2)}$ respectively. This method terminates after a single trial as it has no access to feedback.

2. **Random-Tactile**: Randomly try different gripper heights within the range $\left[d_{\text{vert}}^{(2)} - 2\,\text{mm}, d_{\text{vert}}^{(1)} + 2\,\text{mm}\right]$ until the tactile classifier determines that the correct number of layers have been grasped.

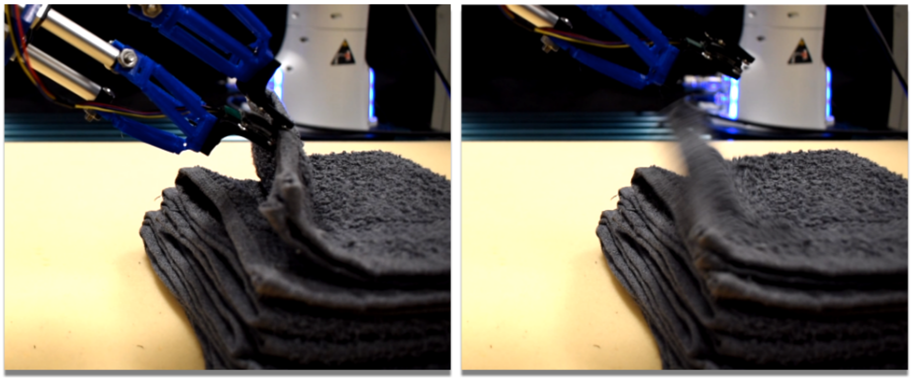


Figure 3.5: An example grasping failure case of the task. Due to an insufficiently robust grasp when lifting (left), the layers may slip out of the robot's control during the lifting portion (right).

3. **Random-Image**: Same as Random-Tactile, but uses an image classifier (instead of a tactile classifier) to determine when the correct number of layers has been grasped. The image classifier is an 18-layer ResNet [14] pre-trained on ImageNet and finetuned on the images (see Figure 3.3) from the same training episodes used to train the tactile classifier.
4. **Feedback-Image**: Same as Feedback-Tactile (our method, below) except with the image classifier.
5. **(Ours) Feedback-Tactile**: Initialize the gripper height to $d_{\text{vert}}^{(1)} + 2\,\text{mm}$; use the grasp policy described in Sec. 3.3.3 to adjust the height per grasp ($\pm 2\,\text{mm}$) based on the tactile classifier predictions.

3.5 Results

We first present results from training a classifier on ReSkin data followed by physical experiment results in which we run 10 trials for each method and condition.

3.5.1 The Tactile Classifier

To better understand the kNN performance, we perform 100 folds of cross-validation and average the validation performance. Each entire training episode is assigned to either the training or validation set.

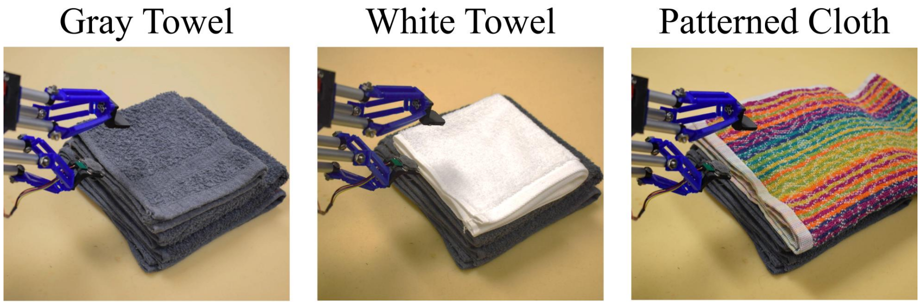


Figure 3.6: The cloths we use for experiments. We use the gray towel (left) for training, and test on all 3 cloths for evaluation. The white towel and patterned cloth test generalizing to novel cloths. The cloths have thicknesses between 3-5 mm and variation in surface texture and stiffness.

Class \ Prediction	0	1	2	3
0 (0 Layers)	1.000	0.000	0.000	0.000
1 (1 Layer)	0.000	0.999	0.000	0.001
2 (2 Layers)	0.030	0.003	0.866	0.100
3 (3 Layers)	0.128	0.256	0.138	0.478

Table 3.1: The average normalized confusion matrix from the cross-validation training results for the k-nearest neighbor classifier we use for tactile-based experiments.

Table 3.1 demonstrates the average normalized confusion matrix obtained from these 100 cross-validation runs, and also reports the average per-class accuracy. We also computed the average balanced accuracy metric [53] to consider the data imbalance and obtain $\mathbf{0.84 \pm 0.06}$. Inspecting the confusion matrix, we find that the tactile classifier can classify classes 0 (pinching with no cloth between the fingers) and 1 (pinching 1 cloth layer) with extremely high effectiveness. Results for classes 2 and 3 suggest that identifying 2 and 3 cloth layers is more challenging.

3.5.2 Grasping 1 Cloth Layer

In the first set of physical experiments, we report the success and failures of methods on grasping and lifting the top layer of cloth from a stack. See Table 3.2 for results. Our method, Feedback-Tactile, succeeds at grasping one layer of cloth in all 10 trials, whereas all competing ablations have lower success rates. Methods with the tactile classifier outperform those using the image classifier, with most failures attributed to mis-prediction rather than poor grasping.

Cloth Type	Method	Success Rate ↑	Prediction Failure	Grasp Failure	Attempts ↓
Gray Towel (Train)	Fixed-Open-Loop	6/10	-	4/10	1 (fixed)
	Random-Image	5/10	5/10	0/10	1.8±0.7
	Random-Tactile	6/10	3/10	1/10	4.8±2.8
	Feedback-Image	8/10	2/10	0/10	2.3±0.8
	Feedback-Tactile	**10/10**	0/10	0/10	3.1±1.0
White Towel (Generalization)	Feedback-Image	3/10	5/10	2/10	1.6±0.5
	Feedback-Tactile	**8/10**	0/10	2/10	2.3±0.8
Patterned Cloth (Generalization)	Feedback-Image	2/10	8/10	0/10	5.1±4.3
	Feedback-Tactile	**7/10**	2/10	1/10	4.6±3.2

Table 3.2: Results for the first set of physical experiments described in Sec. 3.5.2 with grasping at 1 cloth layer. We run all methods for 10 trials each and report the success rate, the failure types (grasping and prediction), and the average number of grasp attempts per trial.

The fixed-height open loop method (Fixed-Open-Loop) poorly handles variations in the initial cloth configuration. There can be up to 1.5 mm variation in the height of the cloth stack based on how they are placed at the start of the trial, which can lead to failures in the open loop grasping method. Both random grasping approaches, Random-Image (5/10) and Random-Tactile (6/10) have lower success rates compared to using feedback-based height adjustment with Feedback-Image (8/10) and Feedback-Tactile (10/10).

For testing generalization, Feedback-Tactile significantly outperforms Feedback-Image on the white towel and patterned cloth. Feedback-Tactile obtains 8/10 and 7/10 success rates for the white towel and patterned cloth, respectively, while Feedback-Image only succeeds in 3/10 and 2/10 trials.

We have analyzed the failure types of each method in Table 3.2. Grasping failures are rare for most methods on 1-layer grasping; grasping failures can occur if the robot does not robustly grip the cloth, and cloth slips out of the grasp when the robot lifts it (see Fig. 3.5). Our method (Feedback-Tactile), also has few prediction failures when generalizing to unseen cloths compared to Feedback-Image.

Cloth Type	Method	Success Rate ↑	Prediction Failure	Grasp Failure	Attempts ↓
Gray Towel (Train)	Fixed-Open-Loop	7/10	-	3/10	1 (fixed)
	Random-Image	6/10	1/10	3/10	5.3±3.0
	Random-Tactile	4/10	4/10	2/10	6.0±3.0
	Feedback-Image	**9/10**	0/10	1/10	4.7±0.9
	Feedback-Tactile	7/10	1/10	2/10	6.4±2.6
White Towel (Generalization)	Feedback-Image	0/10	8/10	2/10	9.2±1.8
	Feedback-Tactile	**4/10**	2/10	4/10	5.0±3.4
Patterned Cloth (Generalization)	Feedback-Image	0/10	10/10	0/10	10.0±0.0
	Feedback-Tactile	**1/10**	3/10	6/10	6.4±3.6

Table 3.3: Experimental results for grasping at the top 2 cloth layers as described in Sec. 3.5.3. Besides the change of 1 to 2 layers, the results are formatted in the same way as in Table 3.2.

3.5.3 Grasping 2 Cloth Layers

In the next set of experiments, we evaluate grasping and lifting the top two layers of cloth. The results in Table 3.3 suggest that the methods achieve success rates similar to 1-layer grasping (Table 3.2) for the gray towel, but performance is lower on the unseen cloths. While Feedback-Image performs slightly better than Feedback-Tactile on the gray towel, Feedback-Tactile performs slightly better on unseen cloths.

Table 3.3 shows that both prediction and grasping failures lead to errors for our method (Feedback-Tactile), though grasping failures are more common (accounting for 2/3 of our total failures). The higher incidence of grasp failures by our method in this experiment suggests that 2-layer grasping is more difficult than 1-layer grasping. Fig. 3.7 highlights some challenges with grasping two layers; for example, we observe that failures tend to occur due to crumpling the fabric when attempting to grasp 2 layers. Furthermore, the top layer of cloth can push downwards on the layer below it, which reduces the gap between the second and third layers; this reduced gap can make it difficult to grasp 2 layers. These observations and results suggest that further innovation on grasp policies may be necessary to improve 2-layer grasp performance on unseen cloths.

Architecture	Accuracy on Test Set
Vanilla LSTM	0.33
Bi-Directional LSTM	0.94
LSTM + CNN	**0.98**
MLP	0.24
1D CNN	0.72
GRU	0.55
GRU + CNN	0.96

Table 3.4: Variation of accuracy with different deep learning architectures.

3.5.4 Deep Learning based Tactile Classifiers

Here we evaluate Deep Learning based classifiers for the collected dataset. Note, we do not evaluate the deep learning classifiers on the robot trials, thus we cannot do a comparison between success rates using classical machine learning classifiers and deep learning based classifiers. However we can compare the accuracies of the classical classifiers compared to the deep learning based classifiers on the dataset collected to train the classifiers. We perform this comparison in this section.

We test out Multi-Layer Perceptron networks, Convolution Networks and Multi Layer LSTM networks as shown in table 3.4. Convolution Networks have the best results among them but none of them beat classical machine learning based approaches. We use 1 Dimensional convolution, across the time axis. Then we tested out Convolution networks as the first layer, whose output is sent into an LSTM network. Effectively the convolution network finds useful features over time, and the LSTM network using these features makes predictions every timestep. This approach turns out to be successful. We reach a very high accuracy of 98%, completely beating out all the classical machine learning approaches on test accuracy with this network architecture as shown in table 3.4. We find that the LSTM's can be replaced with GRU's with no significant loss in performance. In table 3.5 we vary the number of hidden layers of the CNN and check the accuracy on the test set. We find 4 CNN layers followed by 2 LSTM layers to be optimal in accuracy. In table 3.6, we vary the dimensions of the hidden layer with accuracy on the test set and find a dimension of 256 to be optimal in accuracy.

LSTM+CNN Depth Layers	Accuracy
1	0.88
2	0.96
4	**0.96**
8	0.94

Table 3.5: Variation of accuracy with number of layers in network architecture. The number of LSTM layers is fixed at 2 as we see deterioration of performance with too many layers. The number of CNN layers are varied as shown above.

LSTM+CNN Hidden Dimension	Accuracy
256	**0.96**
512	0.95
128	0.92

Table 3.6: Variation of accuracy with the hidden layers dimension. The input is 5 dimensional, output is 1 dimensional and the hidden dimension of the CNN and the LSTM is shown above.

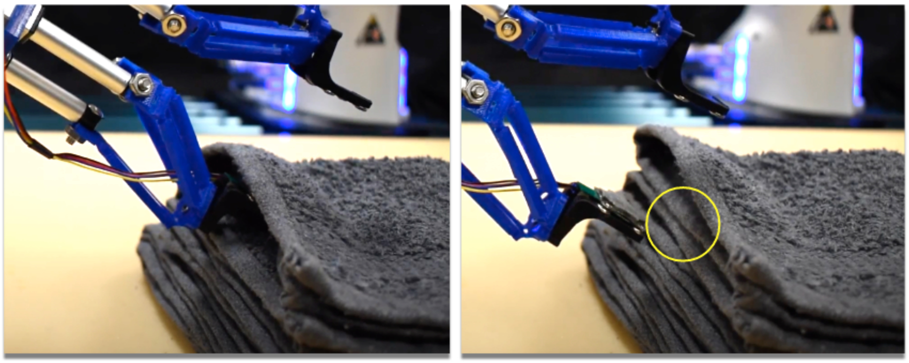


Figure 3.7: A qualitative example of how the task is challenging, particularly with grasping two layers. Because of the horizontal motion of the gripper, layers of cloth can be pushed apart (left), creating air pockets between the top and second layer after the action has finished (right). This gap makes it easier to grasp the top layer but harder to grasp the top *two* layers, due to a smaller gap between the second and third layers (see overlaid yellow circle).

Chapter 4

Conclusions and Future Work

In this thesis, we present a robotic system that uses a magnetometer-based tactile sensor for precisely grasping the desired layers of cloth given a stack of cloth. Our broad approach can be categorized with 3 major steps:

- Collect a large amount of annotated Tactile Sensor Data.
- Train a classifier given the annotated dataset.
- Use the classifier in control algorithms.

. Thus in this work we train a classifier on 1 million+ data points that we collect and label automatically. We use a plethora of classical machine learning algorithms such as K Nearest Neighbours, Support Vector Machines and deep learning based algorithms such as LSTM's and CNN's on learning the dataset. Finally we apply the learnt classifier on a simple test time control algorithm in order to grasp the desired number of layers of cloth. The test time control algorithm chooses a random height and performs a grasp. Then it queries the classifier to find out the number of layers of cloth grasped. If it grasped a lesser number of layers than the desired number, we increase the height, else we decrease the height and so on. This simple method proves to be quite capable in grasping 1 and 2 layers of cloth given a stack of cloth. Crucially we compare our results to a vision baseline that instead of using a tactile based classifier, uses a vision based classifier. The vision based classifier achieves similar train and test accuracies on the dataset but fails to perform in real robot trials. We see that the vision baseline struggles mainly in generalizing to cloths of different textures, colours and shapes, something the tactile sensor is capable of adapting to. The vision baseline is also susceptible to changes in lighting and viewpoint that reduce it's performance relative to the tactile baseline.

Moving forward, we think there are a large variety of fine manipulation tasks that can benefit from using tactile sensing as opposed to vision based sensing. One simple example is riffing through pages of a notebook. This task seems unapproachable using vision based sensing but might be fairly easy to do with a good tactile sensor and dataset. Apart from this, we think there are big benefits available in accurately simulating tactile sensors so that we can collect large amounts of annotated data in simulation as opposed to in the real world. In simulation, it is very easy to collect annotated data and thus we can broaden the range of possible fine manipulation tasks that our robot can do with a good deformable object tactile sensing simulator. Finally, we think there are possible improvements to be made to the tactile sensors themselves. Juxtaposing with the human skin, our sensors fall short in resolution, sensitivity and many other criteria. We think creating a human skin like tactile sensor is one of the big impactful and hard problems that robotics researchers can tackle.

Appendix A

Sensor Data Details

A.1 ReSkin Data Details

In Section 3.3.2, we describe the real-world procedure for collecting a training dataset of ReSkin sensor readings. Each training episode lasts for about 5 seconds to complete the approach, grasp, and release cycle. The grasp stage lasts for 1 second, resulting in approximately 350 sensor readings of 15 values each (3 per magnetometer). We filter out training episodes where no cloth was grasped. Combining all episodes into a dataset results in a total of 18,838 ReSkin tactile sensor readings, each of which is 15-D. The readings in the dataset are separated into training and validation sets, grouped by episode to prevent data leakage. Thus, all time steps in one episode are either all in training or all in validation. We train classifiers taking individual readings as input and found that this input outperformed our baselines on the cloth singulation task; classification on time-series inputs are a promising area of future work.

A.2 Image Classifier Details

In Section 3.4.2, we introduce the **Random-Image** and **Feedback-Image** baseline methods. These use images instead of tactile data, but the objective is the same: to predict the number of cloth layers grasped. When collecting tactile-based data, we simultaneously collect image data, so the data collection time for image data is the same as the time that was spent on collecting the Reskin data. We mount a webcam approximately 30 cm away from the starting position of the ReSkin, which queries images of the robot when it is collecting data. See Fig. 3.3 for examples of what the RGB images look like.

The image classifier is an 18-layer ResNet [14]. We use an off-the-shelf ResNet-18 from PyTorch which has been pre-trained on ImageNet, and change the last layer to output 4 dimensions instead of 1000. This results in a total of 11,178,564 trainable parameters. The RGB input images are first cropped to 360 × 360 such that the ReSkin sensor is located roughly in the middle, then images are further resized to 224 × 224 before being passed as input to the ResNet.

There is about 7-8X more tactile data compared to image-based data because the ReSkin can query data much faster than the webcam. This may be an inherent advantage of the ReSkin sensor over most commercial webcams. To strengthen the image-based baseline, we use data augmentation (which the tactile classifiers do not use). In training, we employ random crops by selecting random 224 × 224 crops within the 360 × 360 image. We also use random horizontal flips. At test time, we only use center crops for consistency. We train for 30 epochs, use a batch size of 64, and optimize using Adam with learning rate 0.001.

Bibliography

[1] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. ReSkin: versatile, replaceable, lasting tactile skins. In *Conference on Robot Learning (CoRL)*, 2021. (document), 2.1.2, 2.4, 3.1, 3.2, 3.3.1

[2] Júlia Borràs, Guillem Alenyà, and Carme Torras. A grasping-centered analysis for cloth manipulation. *IEEE Transactions on Robotics*, 36(3):924–936, 2020. doi: 10.1109/TRO.2020.2986921. 2.2.2, 3.1

[3] Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James F O'Brien, and Pieter Abbeel. Bringing Clothing Into Desired Configurations with Limited Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 2.2.2

[4] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009. 2.1.4

[5] Satonori Demura, Kazuki Sano, Wataru Nakajima, Kotaro Nagahama, Keisuke Takeshita, and Kimitoshi Yamazaki. Picking up One of the Folded and Stacked Towels by a Single Arm Robot. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018. 2.2.2

[6] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. GelSlim: A High-Resolution, Compact, Robust, and Calibrated Tactile-sensing Finger. In *IEEE Robotics and Automation Letters (RA-L)*, 2018. 2.1.1

[7] Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, and Sotiris Malassiotis. Autonomous Active Recognition and Unfolding of Clothes Using Random Decision Forests and Probabilistic Planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 2.2.1, 3.1

[8] Zackory Erickson, Henry Clever, Greg Turk, C. Karen Liu, and Charles Kemp. Deep Haptic Model Predictive Control for Robot-Assisted Dressing. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 3.1

[9] Zackory Erickson, Maggie Collier, Ariel Kapusta, and Charles Kemp. Track-

ing Human Pose During Robot-Assisted Dressing using Single-Axis Capacitive Proximity Sensing. In *IEEE Robotics and Automation Letters (RA-L)*, 2018. 3.1

[10] Zackory Erickson, Vamsee Gangaram, Ariel Kapusta, C. Karen Liu, and Charles C. Kemp. Assistive Gym: A Physics Simulation Framework for Assistive Robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 3.1

[11] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minho Hwang, Ryan Hoque, Joseph Gonzalez, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Learning Dense Visual Correspondences in Simulation to Smooth and Fold Real Fabrics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 2.2.1, 2.2.2, 3.1

[12] Yuhao Guo, Xin Jiang, and Yunhui Liu. Deformation Control of a Deformable Object Based on Visual and Tactile Feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 2.2.2

[13] Huy Ha and Shuran Song. FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding. In *Conference on Robot Learning (CoRL)*, 2021. 2.2.1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, A.2

[15] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. VisuoSpatial Foresight for Multi-Step, Multi-Task Fabric Manipulation. In *Robotics: Science and Systems (RSS)*, 2020. 2.2.1

[16] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019. 1

[17] Zhiao Huang, Yuanming Hu, Tao Du, Siyuan Zhou, Hao Su, Joshua B Tenenbaum, and Chuang Gan. Plasticinelab: A soft-body manipulation benchmark with differentiable physics. *arXiv preprint arXiv:2104.03311*, 2021. 1

[18] Biao Jia, Zhe Hu, Jia Pan, and Dinesh Manocha. Manipulating Highly Deformable Materials Using a Visual Feedback Dictionary. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2.2.2

[19] Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. Clothes State Recognition Using 3D Observed Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009. 2.2.2

[20] Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. A Method For Handling a Specific Part of Clothing by Dual Arms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009. 2.2.2

[21] Panagiotis N. Koustoumpardis, Kostas X. Nastos, and Nikos A. Aspragathos. Underactuated 3-finger Robotic Gripper for Grasping Fabrics. In *International Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2014. 2.2.2

[22] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. DIGIT: A Novel Design for a Low-Cost Compact High-Resolution Tactile Sensor with Application to In-Hand Manipulation. In *IEEE Robotics and Automation Letters (RA-L)*, 2020. 2.1.1, 3.1

[23] Robert Lee, Daniel Ward, Akansel Cosgun, Vibhavari Dasagi, Peter Corke, and Jurgen Leitner. Learning Arbitrary-Goal Fabric Folding with One Hour of Real Robot Experience. In *Conference on Robot Learning (CoRL)*, 2020. 2.2.1, 3.1

[24] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation. In *Conference on Robot Learning (CoRL)*, 2020. 2.2.1, 1

[25] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning Visible Connectivity Dynamics for Cloth Smoothing. In *Conference on Robot Learning (CoRL)*, 2021. 2.2.1

[26] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 432–448. PMLR, 2021. 1

[27] James Long, Katherine Burns, and Jingzhou Yang. Cloth modeling and simulation: a literature survey. In *Digital Human Modeling: Third International Conference, ICDHM 2011, Held as Part of HCI International 2011, Orlando, FL, USA July 9-14, 2011. Proceedings 3*, pages 312–320. Springer, 2011. 1

[28] Shan Luo, Wenzhen Yuan, Edward Adelson, Anthony G Cohn, and Raul Fuentes. Vitac: Feature Sharing Between Vision and Tactile Sensing for Cloth Texture Recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2.1.1, 3.1

[29] Xiao Ma, David Hsu, and Wee Sun Lee. Learning Latent Graph Dynamics for Deformable Object Manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 2.2.1

[30] Miles Macklin, Matthias Muller, Nuttapong Chentanez, and Tae-Yong Kim. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.*, 33(4), July 2014. 1

[31] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth Grasp Point Detection Based on Multiple-View Geometric Cues with Application to Robotic Towel Folding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010. 2.2.1, 2.2.2, 3.1

[32] Pragna Mannam, Avi Rudich, Kevin Zhang, Manuela Veloso, Oliver Kroemer, and F. Zeynep Temel. A Low-Cost Compliant Gripper Using Cooperative Mini-Delta Robots for Dexterous Manipulation. In *Robotics: Science and Systems (RSS)*, 2021. (document), 3.2, 3.1, 3.3.1

[33] Pragna Mannam, Avi Rudich, Kevin L Zhang, Manuela Veloso, Oliver Kroemer, and Z Temel. A low-cost compliant gripper using cooperative mini-delta robots for dexterous manipulation. In *Robotics science and systems*, 2021. (document), 1, 2.6

[34] Jan Matas, Stephen James, and Andrew J. Davison. Sim-to-Real Reinforcement Learning for Deformable Object Manipulation. *Conference on Robot Learning (CoRL)*, 2018. 2.2.1, 2.2.2, 3.1

[35] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2006. 2.3.2

[36] Stephen Miller, Jur van den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A Geometric Approach to Robotic Laundry Folding. In *International Journal of Robotics Research (IJRR)*, 2012. 2.2.1

[37] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 1

[38] Kavya Puthuveetil, Charles C. Kemp, and Zackory Erickson. Bodies Uncovered: Learning to Manipulate Real Blankets Around People via Physics Simulations. In *IEEE Robotics and Automation Letters (RA-L)*, 2022. 2.2.1

[39] Jianing Qian, Thomas Weng, Luxin Zhang, Brian Okorn, and David Held. Cloth Region Segmentation for Robust Grasp Selection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 2.2.2

[40] Arnau Ramisa, Guillem Alenya, Francesc Moreno-Noguer, and Carme Torras. Using Depth and Appearance Features for Informed Robot Grasping of Highly Wrinkled Clothes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 2.2.2

[41] Peter Roberts, Mason Zadan, and Carmel Majidi. Soft tactile sensing skins for robotics. *Current Robotics Reports*, 2:343–354, 2021. 2.1.4

[42] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic Manipulation and Sensing of Deformable Objects in Domestic

and Industrial Applications: a Survey. In *International Journal of Robotics Research (IJRR)*, 2018. 3.1

[43] Daniel Seita, Nawid Jamali, Michael Laskey, Ron Berenstein, Ajay Kumar Tanwani, Prakash Baskaran, Soshi Iba, John Canny, and Ken Goldberg. Deep Transfer Learning of Pick Points on Fabric for Robot Bed-Making. In *International Symposium on Robotics Research (ISRR)*, 2019. 3.1

[44] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, Katsu Yamane, Soshi Iba, John Canny, and Ken Goldberg. Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 2.2.1, 2.2.2, 3.1

[45] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 2.2.1

[46] Zilin Si and Wenzhen Yuan. Taxim: An example-based Simulation Model for GelSight Tactile Sensors. In *IEEE Robotics and Automation Letters (RA-L)*, 2022. 1

[47] Li Sun, Gerardo Aragon-Camarasa, Simon Rogers, and J. Paul Siebert. Accurate Garment Surface Analysis using an Active Stereo Robot Head with Application to Dual-Arm Flattening. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 2.2.2

[48] Balakumar Sundaralingam, Alexander Lambert, Ankur Handa, Byron Boots, Tucker Hermans, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Robust Learning of Tactile Force Estimation through Robot Interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 2.1.3

[49] ST Tan, TN Wong, YF Zhao, and WJ Chen. A constrained finite element method for modeling cloth deformation. *The Visual Computer*, 15:90–99, 1999. 1

[50] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A Physics Engine for Model-Based Control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012. 2.2.1

[51] Eric Torgerson and Fanget Paul. Vision Guided Robotic Fabric Manipulation for Apparel Manufacturing. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1987. 3.1

[52] Jan B.F. van Erp and Hendrik A.H.C. van Veen. Touch down: The effect of artificial touch cues on orientation in microgravity. *Neuroscience Letters*,

404(1):78–82, 2006. ISSN 0304-3940. doi: https://doi.org/10.1016/j.neulet.2006.05.060. URL https://www.sciencedirect.com/science/article/pii/S0304394006005490. 1

[53] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2. 3.5.1

[54] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. TACTO: A Fast, Flexible, and Open-source Simulator for High-Resolution Vision-based Tactile Sensors. In *IEEE Robotics and Automation Letters (RA-L)*, 2022. 1

[55] Thomas Weng, Sujay Bajracharya, Yufei Wang, Khush Agrawal, and David Held. FabricFlowNet: Bimanual Cloth Manipulation with a Flow-based Policy. In *Conference on Robot Learning (CoRL)*, 2021. 2.2.1, 2.2.2, 3.1

[56] Göran Westling and Roland Johansson. Factors influencing the force control during precision grip. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, 53:277–84, 02 1984. doi: 10.1007/BF00238156. 1

[57] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to Manipulate Deformable Objects without Demonstrations. In *Robotics: Science and Systems (RSS)*, 2020. 2.2.1, 3.1

[58] Akihiko Yamaguchi and Christopher G Atkeson. Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision? *Advanced Robotics*, 33(14):661–673, 2019. 2.1.4

[59] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning Predictive Representations for Deformable Objects Using Contrastive Estimation. In *Conference on Robot Learning (CoRL)*, 2020. 2.2.1

[60] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, Learning, Perception, and Control Methods for Deformable Object Manipulation. *Science Robotics*, 6 (54), 2021. 2.2

[61] Wenhao Yu, Ariel Kapusta, Jie Tan, Charles C. Kemp, Greg Turk, and C. Karen Liu. Haptic Simulation for Robot-Assisted Dressing. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 3.1

[62] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. GelSight: High-Resolution

Robot Tactile Sensors for Estimating Geometry and Force. *Sensors*, 17(12), 2017. URL https://www.mdpi.com/1424-8220/17/12/2762. (document), 2.1.1, 2.2, 3.1

[63] Wenzhen Yuan, Shaoxiong Wang, Siyuan Dong, and Edward Adelson. Connecting Look and Feel: Associating the Visual and Tactile Properties of Physical Materials. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2.1.1

[64] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward Adelson. Active Clothing Material Perception using Tactile Sensing and Deep Learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2.1.1, 3.1

[65] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dimtry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, et al. Challenges and outlook in robotic manipulation of deformable objects. *IEEE Robotics and Automation Magazine*, 2021. 2.2