Unfolding the Potential of Point-Based Correspondences for Cloth Manipulation

Mansi Agarwal

CMU-RI-TR-23-46

06 July 2023

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

Thesis Committee:
Prof. David Held, *Chair*
Prof. Oliver Kroemer
Prof. Shubham Tulsiani
Alex LaGrassa

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

For my beloved grandparents.

Abstract

Robotic cloth manipulation is an active area of research with numerous applications in domestic and industrial environments. However, prior work in this field has limitations that restrict their applicability in real-world scenarios. For instance, these approaches often require subgoals for long-horizon tasks and face challenges in handling unaligned configurations. By "unaligned configurations," we refer to situations where the initial orientation of the cloth surface differs from the goal orientation. To curb these issues, we propose utilizing point-based correspondences, which capture geometric relationships and deformations in cloth surfaces. Point-based correspondences refer to establishing correspondences between points on the cloth surface, allowing us to track and model the cloth's behavior accurately. In this work, we present two automated cloth manipulation solutions that incorporate the use of point-based correspondences. Our focus centers on fundamental cloth manipulation tasks, including folding, smoothing, and alignment. Through extensive experiments and evaluations, we demonstrate the effectiveness of our proposed approaches, which surpass state-of-the-art methods. Comparative experiments against existing techniques highlight the distinct advantages of using point-based correspondences for achieving efficient, robust, and long-horizon cloth manipulation.

Acknowledgments

I am profoundly grateful to my advisor, Prof. David Held, for his guidance, support, and mentorship throughout this research journey. His expertise and encouragement have been instrumental in shaping this thesis and elevating my understanding in the field. I extend my heartfelt thanks to my mentors, Thomas Weng and Daniel Seita, for their continuous support and encouragement. I would like to express my sincere appreciation to my thesis committee members, Prof. Shubham Tulsiani, Prof. Oliver Kroemer, and Alex LaGrassa, for their valuable feedback and constructive criticism that have enriched this work.

I am forever indebted to my family for their immeasurable love and support. I extend my heartfelt thanks to my beloved parents, Dr. Manoj Agarwal and Mrs. Seema Agarwal, for being my unwavering guiding stars. Their constant presence has been the bedrock of my strength and courage. A special place in my heart is reserved for my brother, Ojas M. Agarwal, who has been my comforting refuge during life's storms. Gratitude overflows in my heart for my friends who have filled my life with warmth and joy. I am thankful for Ashwin Misra, who has been a rock in my life, always ready to uplift my spirits. I am also grateful to Mononito Goswami for offering consistent support and encouragement, no matter the challenges. To Muskan Golyan, my dearest best friend, you have been the pillar of my soul, supporting me through every twist and turn, sharing laughter and tears, and making each step of this journey meaningful and unforgettable.

Finally, I wish to extend my heartfelt appreciation to Barbara Fecich for contributing to an enriching and memorable experience during my time at Robotics Institute, Carnegie Mellon University.

Funding

This research was funded by the National Science Foundation under Grant No. IIS-1849154 and Grant No. IIS-2046491. Additionally, the research received support from LG Electronics, the US Air Force, and DARPA (FA8750-18-C-0092).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cloth manipulation is a fascinating and challenging field that revolves around the manipulation and control of cloth materials. It encompasses a wide range of tasks, including folding, unfolding, draping, grasping, and shaping cloth to achieve desired configurations. Cloth materials possess unique characteristics, such as flexibility, deformability, and non-rigidity, which make their manipulation a complex and intricate process.

The focus of this thesis is to explore and advance the field of cloth manipulation by developing innovative techniques and methodologies. The aim is to address the challenges associated with cloth behavior and enhance the effectiveness and efficiency of cloth manipulation tasks. By leveraging cutting-edge technologies in robotics and learning, we strive to automate and optimize cloth manipulation processes.

Cloth manipulation poses several challenges that researchers and practitioners have been actively working to address. These challenges stem from the unique properties of cloth materials, including their flexibility, deformability, and non-rigidity.

Firstly, the high configuration space of cloth materials makes it challenging to precisely control their movements and achieve desired configurations. Cloth can exhibit a wide range of deformations and can undergo complex transformations, making it difficult to predict and control its behavior accurately. This necessitates the development of sophisticated algorithms and strategies to overcome these challenges.

Another significant challenge in cloth manipulation is the issue of self-occlusions. When cloth folds or drapes over itself, certain parts become hidden from view,

resulting in partial or complete occlusions. These occlusions obstruct the visibility of the cloth's underlying structure, making it difficult to estimate its shape and deformations accurately.

Furthermore, cloth materials are subject to non-rigid transformations, meaning they can undergo deformations and shape changes without breaking or losing their structural integrity. This characteristic poses additional difficulties in achieving precise and controlled cloth manipulation. Non-rigid transformations require sophisticated modeling and estimation techniques to accurately capture and predict cloth behavior, taking into account its material properties and the external forces acting upon it.

1.1 Motivation

The motivation behind this research is driven by the limitations of existing manual cloth manipulation techniques. Manual methods for cloth manipulation are time-consuming, labor-intensive, and prone to errors. They often rely on human expertise and dexterity, which can vary and be limited in certain scenarios. Automating cloth manipulation tasks has the potential to overcome these limitations and offer significant advantages.

Automated cloth manipulation can help address labor shortages and improve overall efficiency in various industries. By automating the process, tasks that were previously performed manually can be completed more quickly, allowing for higher production throughput. Moreover, automated systems can achieve higher levels of precision and accuracy, ensuring consistent results and reducing the risk of errors. This is particularly crucial in industries where quality control and precision are paramount, such as manufacturing and textiles.

In addition to improving efficiency, automated cloth manipulation techniques can also enhance safety in the workplace. Manual cloth manipulation tasks often involve handling heavy or bulky cloth materials, which can lead to physical strain or potential injuries. By automating these tasks, the risk of accidents and injuries can be minimized, creating a safer working environment for operators.

The applications of cloth manipulation are diverse and span across various industries. In manufacturing processes, automated cloth manipulation can optimize production lines by efficiently folding, draping, or arranging cloth materials. In

the household laundry domain, automated folding and sorting systems can simplify the process for consumers, saving time and effort. Healthcare and assistive dressing applications can assist individuals with limited mobility in dressing themselves independently, promoting their autonomy and well-being. Textile industries can benefit from automated cloth manipulation in terms of material handling, quality control, and innovative design possibilities. The e-commerce and retail sectors can utilize automated folding systems to improve packaging efficiency, presentation, and customer experience. Moreover, cloth manipulation techniques have vast potential in research and development, enabling advancements in material science, robotics, and interactive textiles.

By addressing the challenges and harnessing the potential of automated cloth manipulation, one can realize significant societal impacts. Increased productivity, reduced labor requirements, improved safety standards, and enhanced overall quality in cloth-related processes are just some of the positive outcomes that can be achieved. Moreover, automation in cloth manipulation can drive innovation and unlock new possibilities for applications in various industries, leading to economic growth and improved living standards.

1.2 Contribution

The objective of this thesis is to propose a holistic approach to cloth manipulation that addresses the challenges associated with cloth behavior. We aim to develop a methodology that is efficient, robust, and fully autonomous, enabling seamless and automated cloth manipulation tasks.

Our goal is to develop algorithms and systems that are accurate and quick in making decisions for cloth manipulation. By focusing on efficiency, we aim to enable real-time or near-real-time cloth manipulation, enhancing the overall productivity and effectiveness of the system.

Robustness is another essential attribute we aim to incorporate into our approach. Cloth manipulation involves dealing with various challenges, such as rotations, where the observation and goal are not aligned or oriented in the same direction. To overcome such obstacles, our methodology should exhibit resilience and adaptability. It should be able to handle variations in cloth behavior and maintain accurate manipulation

even in the presence of uncertainties. By designing a robust approach, we ensure that our system can effectively handle diverse cloth scenarios encountered in real-world applications.

Furthermore, our objective is to develop a fully autonomous cloth manipulation approach. This means that the system should operate with little to no human supervision, minimizing the need for explicit instructions or subgoals. By achieving autonomy, we aim to create a system that can perform complex cloth manipulation tasks independently and make intelligent decisions based on the given input and desired goals. For instance, in long-horizon tasks, our approach should be capable of planning and executing the necessary cloth manipulation steps without relying heavily on human intervention. This level of autonomy not only reduces human effort but also enables the system to handle a broader range of cloth manipulation scenarios efficiently.

To fulfill our objective, we use point-based correspondences, which enable us to capture the intricate geometric relationships and deformations present in cloth surfaces. By harnessing these correspondences, we can address key challenges, such as unaligned configurations and the need for subgoals in long-horizon tasks. In this thesis, we introduce two automated cloth manipulation solutions that employ point-based correspondences. Our primary focus revolves around essential cloth manipulation tasks, including folding, smoothing, and alignment.

To validate the effectiveness of our proposed approaches, we conduct a comprehensive set of experiments and evaluations. The results showcase how our methods outperform state-of-the-art techniques in terms of efficiency, robustness, and long-horizon planning. Through comparative experiments against existing approaches, we highlight the distinct advantages of employing point-based correspondences for achieving efficient and reliable cloth manipulation.

Overall, our research strives to contribute to the development of a holistic and autonomous cloth manipulation approach. By incorporating point-based correspondences and addressing critical challenges, we aim to enhance the efficiency, robustness, and automation capabilities of cloth manipulation systems.

1.3 Organization

The thesis is organized into several chapters. Chapter 1 serves as the introduction, providing an overview of cloth manipulation and the research objectives. Chapter 2 focuses on essential concepts related to cloth manipulation, including point clouds, processing techniques, transformers, correspondence estimation, and reinforcement learning algorithms. In Chapter 3, a literature review is conducted specifically on cloth folding and smoothing techniques. Chapter 4 introduces the first approach, FabricFlowAlignNet, which utilizes correspondences for cloth alignment and manipulation. The methodology, experiments, and findings of FabricFlowAlignNet are described. Additionally, Chapter 5 addresses the limitations of FabricFlowAlignNet and presents a novel approach that combines reinforcement learning and correspondences for long-horizon planning without subgoals. The methodology, experiments, and findings of this approach are discussed. Finally, Chapter 6 concludes the thesis by summarizing the main contributions, reflecting on limitations, and providing recommendations for future research.

Chapter 2

Background

2.1 Cloth Manipulation

Cloth manipulation is a multidisciplinary field that lies at the intersection of robotics, computer vision, and material science. It encompasses a wide range of tasks related to controlling and manipulating deformable cloth objects. Cloth manipulation has significant practical applications in areas such as robotics, virtual garment fitting, virtual reality, and assistive dressing systems. The complexity of cloth manipulation arises from the non-rigid and highly deformable nature of cloth, which requires accurate modeling of its behavior and planning of appropriate actions.

Within cloth manipulation, various subfields have emerged, each focusing on specific aspects of cloth behavior and manipulation. Some of the prominent subfields include cloth folding, cloth smoothing, cloth draping, and assistive dressing. These subfields address different challenges and require distinct approaches and algorithms.

1. **Cloth Folding**: Cloth folding is one of the fundamental tasks in cloth manipulation. It involves transforming a given cloth configuration into a desired folded state. The goal of cloth folding is to find a sequence of actions that deform the cloth from its initial state to a target folded state. There are two primary types of cloth folding scenarios: single-step goals and multi-step goals. In single-step goals, the cloth is transformed directly from the initial state to the final folded state in a single action sequence. The challenge lies in predicting

the appropriate actions that achieve the desired folding outcome. On the other hand, multi-step goals can involve intermediate subgoals between the initial and final states. The problem statement for multi-step goals can be either predicting the actions between each observation and subgoal or directly finding a trajectory that leads from the initial observation to the final folded goal.

2. **Cloth Smoothing**: Cloth smoothing is another crucial task in cloth manipulation that focuses on removing wrinkles and creases from a cloth surface to achieve a smoother appearance. The goal of cloth smoothing is to find a sequence of actions that gradually deform the cloth from its initial state to a target smooth state.

   Similar to cloth folding, the objective is to find a trajectory of actions that leads from the initial cloth observation to the final smooth goal. The actions applied during the trajectory progressively adjust the cloth configuration to eliminate wrinkles and achieve the desired smooth appearance.

3. **Cloth Alignment**: Cloth alignment is often an essential aspect of cloth manipulation tasks. It involves adjusting the cloth configuration to match a specific target alignment, such as aligning the edges or corners of the cloth. The problem of cloth alignment is mathematically similar to cloth folding and smoothing, as it involves finding a trajectory of actions that transforms the initial cloth observation to the final aligned goal configuration.

2.2 Point Clouds and Point Cloud Processing

Point cloud networks have gained significant attention and popularity in recent years due to their ability to effectively process and analyze point cloud data. Point clouds are a powerful representation of 3D data, consisting of a set of points in space that capture the geometry and spatial information of objects or scenes. They have become an essential source of sensory information in various fields, including robotics, computer vision, autonomous driving, and 3D modeling.

The unique challenge in processing point clouds lies in their unordered nature, as the points do not have an inherent ordering or structure. Point cloud networks address this challenge by incorporating permutation invariance into their design. Permutation

invariance ensures that the network's output remains the same regardless of the ordering of the points in the input, making the network robust and invariant to different point cloud representations.

Additionally, point cloud networks need to handle variable-sized point clouds, as the number of points in a point cloud can vary depending on the complexity of the scene or object being captured. These networks are designed to dynamically adapt to the varying number of points and effectively extract features and information from the point cloud data.

Moreover, point cloud networks go beyond per-point processing by incorporating contextual information at different scales. They leverage local and global context to capture both fine-grained details and global relationships within the point cloud. This enables the networks to understand the spatial dependencies and interactions between points, leading to more robust and informative representations.

Numerous architectures have been developed for point cloud networks. Here, we provide a list of some popular networks:

1. **PointNet**: PointNet [12] is a pioneering architecture that revolutionized the field of point cloud processing by introducing the concept of using max pooling to aggregate information from local neighborhoods in point clouds. The network architecture consists of a series of point-wise multi-layer perceptrons (MLPs), followed by a max pooling layer. The point-wise MLPs extract features from individual points in the point cloud, while the max pooling layer aggregates information from local neighborhoods. By using max pooling, PointNet can capture global features and representations from the entire point cloud, enabling it to understand the overall structure and context of the object or scene. PointNet has demonstrated remarkable performance across various point cloud tasks, including object classification, part segmentation, and scene understanding. Its ability to effectively process and learn from unordered point cloud data has made it a foundational architecture in the field.
2. **PointNet++**: PointNet++ [13] is an extension of PointNet that further enhances its capabilities by incorporating hierarchical pooling to capture more global information from point clouds. The network architecture of PointNet++ consists of a series of PointNet modules, where each module uses max pooling

to aggregate information from local neighborhoods in the point cloud. However, unlike PointNet, PointNet++ operates hierarchically, with each module focusing on a different scale of the point cloud. By incorporating hierarchical pooling, PointNet++ can learn features at multiple scales, enabling it to capture the complex and multi-level structure of point clouds. This hierarchical approach enhances its ability to understand fine-grained details as well as global relationships within the point cloud. PointNet++ has demonstrated superior performance compared to PointNet in tasks such as object classification, part segmentation, and scene understanding, highlighting its effectiveness in capturing hierarchical structures in point clouds.

3. **DGCNN**: DGCNN (Dynamic Graph Convolutional Neural Network) [19] is a graph-based network specifically designed for processing point clouds. Unlike PointNet and PointNet++, DGCNN represents point clouds as graphs, where each point is considered as a node in the graph. The network architecture of DGCNN consists of multiple graph convolution layers, each utilizing a message passing mechanism to propagate information between points in the graph. By employing message passing, DGCNN can capture the relationships and dependencies between points in the point cloud, allowing it to effectively learn the geometric structure and spatial arrangements. This graph-based approach has demonstrated its efficacy in various tasks, including object classification, part segmentation, and object detection. DGCNN's ability to model the relationships between points makes it particularly suitable for tasks that require capturing the local and global connectivity in point clouds.
4. **GATConv**: GATConv (Graph Attention Convolution) [16] is another graph-based network that leverages attention mechanisms to weigh the interactions between points in point clouds. The network architecture of GATConv consists of multiple graph attention layers, each employing an attention mechanism to determine the importance or relevance of the neighboring points for each point in the graph. By utilizing attention mechanisms, GATConv can focus on the most significant neighbors for each point, allowing it to effectively capture the intricate relationships and dependencies within the point cloud. This attention-based approach enhances its ability to understand the complex interactions

and context of points in point clouds. GATConv has demonstrated strong performance in tasks such as object classification, part segmentation, and object detection, showcasing the effectiveness of attention mechanisms in capturing relevant information in point clouds.

These architectures have significantly advanced the field of point cloud processing by providing effective tools for feature extraction, object recognition, segmentation, and other tasks. Their success demonstrates the importance of considering the unique characteristics of point clouds and developing specialized networks to exploit their rich information.

2.3 Attention in Deep Learning: Understanding Transformers

Attention is a powerful mechanism in deep learning that enables neural networks to selectively focus on specific parts of an input sequence. This capability is crucial in various tasks, such as machine translation, where understanding the entire input sentence is essential for generating accurate output sentences. Transformers, a groundbreaking neural network architecture, harness the power of attention to learn long-range dependencies in sequence data.

The implementation of attention in deep learning involves assigning weights to the input features using a softmax function. These weights determine the relevance and importance of different parts of the input sequence, allowing the network to prioritize and attend to the most significant features.

In Transformer [15] architecture, there are two main components: the encoder and the decoder. Both the encoder and decoder consist of multiple layers of self-attention mechanisms and feed-forward neural networks.

The encoder takes an input sequence and processes it to generate a sequence of hidden states. The input sequence is usually tokenized and embedded into continuous vector representations. These embeddings are then passed through multiple encoder layers, where each layer employs a self-attention mechanism to capture the relationships between different words or tokens in the input sequence. The outputs of the encoder are the final hidden states, which represent contextualized representations of

the input tokens.

The decoder takes the hidden states generated by the encoder and uses them to produce the output sequence step by step. At each time step, the decoder attends to the encoder's hidden states and uses self-attention mechanisms to focus on relevant parts of the input sequence. It predicts the next token in the output sequence based on the context it has learned from the encoder's hidden states. During training, the decoder is provided with the target output sequence and is trained to minimize the difference between its predictions and the ground truth.

One key innovation in Transformers is the self-attention mechanism, which allows the model to weigh the importance of different words or tokens in the input sequence when generating the output. This attention mechanism enables Transformers to handle long-range dependencies effectively and process entire sequences in parallel, making them highly efficient compared to traditional sequential models like LSTMs or RNNs.

The transformer architecture has demonstrated remarkable performance in various NLP tasks, such as machine translation, language modeling, text generation, and sentiment analysis, among others. It has also been extended to handle other types of data beyond natural language, demonstrating its broad applicability and effectiveness.

By harnessing the power of attention, transformers have revolutionized the field of deep learning, providing a robust framework for modeling and understanding complex sequential data. The combination of attention mechanisms and transformers has unlocked new possibilities in tasks that require capturing long-range dependencies and leveraging contextual information for accurate predictions and generation.

2.4 Correspondences

Correspondences refer to pairs of points or features that are associated with each other based on their spatial or semantic relationship. In cloth manipulation tasks, correspondence estimation is of particular importance. Cloth is a deformable object that can undergo complex transformations and shape changes. Estimating correspondences between different regions or points on the cloth is essential for understanding its behavior, tracking its motion, and performing various manipulation tasks.

2D Correspondences: One common type of correspondence estimation is 2D

correspondences or optical flow. Optical flow involves estimating the motion vector or displacement for each pixel between two consecutive frames in a video sequence. By establishing correspondences between pixels, optical flow provides valuable information about the movement and flow patterns within consecutive frames.

Optical flow estimation is typically performed using computer vision techniques, with state-of-the-art approaches leveraging convolutional neural networks (CNNs) to estimate flow. These networks are trained to learn the complex patterns and motion characteristics present in the video data.

While optical flow is a powerful tool for estimating 2D correspondences, it does have its limitations. In cases where the scene undergoes rapid motion or there are occlusions, optical flow may struggle to accurately estimate correspondences. These challenges can lead to errors in tracking the cloth's motion or fail to capture subtle deformations.

FFN (FabricFlowNet) [20] is an approach that leverages optical flow to estimate correspondences between an observed cloth image and a goal cloth image, enabling it to generate a sequence of actions for folding the cloth into the desired shape. By employing a convolutional neural network (CNN) to estimate optical flow, FFN captures the motion and displacement of pixels between the two cloth images.

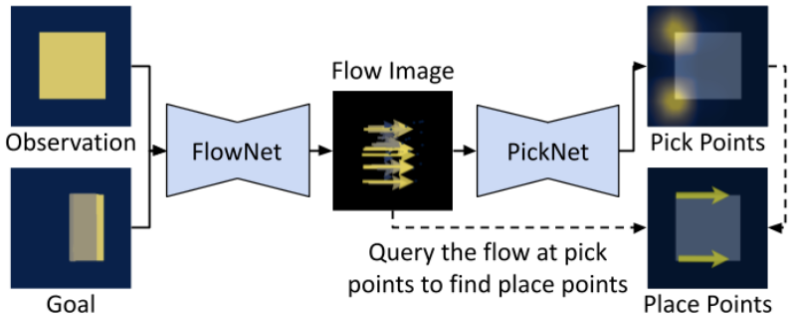


Figure 2.1: FabricFlowNet utilizes 2D correspondences to represent the goal and to predict robot actions.

The estimated correspondences serve as a valuable source of information for predicting the actions needed to transform the observed cloth image into the goal cloth image. By understanding how different regions of the cloth should move and deform, FFN can plan a sequence of actions that facilitate the folding process and achieve the desired shape. The correspondences provide crucial guidance for the

transformation of the cloth's configuration.

Incorporating 2D correspondences into the cloth folding process brings several advantages to FFN. Firstly, it enhances the accuracy and precision of the folding task by leveraging the information obtained from the correspondences. This enables FFN to make more informed decisions regarding the actions needed to align the observed cloth image with the goal cloth image.

Additionally, FFN's utilization of correspondences demonstrates the significance of incorporating contextual information into cloth manipulation techniques. By considering the relationships and correspondences between different regions of the cloth, FFN can better understand the global structure and shape transformations required for successful folding. This utilization of correspondences contributes to the overall success of the cloth manipulation process.

The utilization of correspondences in FFN highlights the importance of correspondence estimation in cloth manipulation tasks. By accurately estimating and leveraging correspondences, cloth folding techniques can achieve higher levels of accuracy, robustness, and efficiency. This provides valuable insights for further research and development in the field of cloth manipulation and inspires the exploration of more advanced approaches that utilize correspondences effectively.

3D Correspondences: 3D correspondences refer to pairs of points that are associated with each other in 3D space. These correspondences are valuable in various applications, such as object tracking and 3D reconstruction as they enable the understanding of the spatial relationships between points and facilitate accurate modeling and analysis of the 3D world. Point-based correspondences are a specific type of 3D correspondences that involve finding pairs of points in two different 3D point clouds that are close to each other. This approach is commonly used in computer vision and robotics tasks, where the goal is to establish correspondences between points in different point cloud data sets. By identifying points that correspond to each other, it becomes possible to perform tasks such as aligning two point clouds, estimating the motion between frames, or reconstructing the 3D shape of an object.

TAX-Pose [11] is a method specifically designed for the task of learning how to place objects in task-specific locations using cross-pose correspondences. The method consists of two main components: a cross-pose correspondence module and a policy learning module. The cross-pose correspondence module in TAX-Pose is responsible

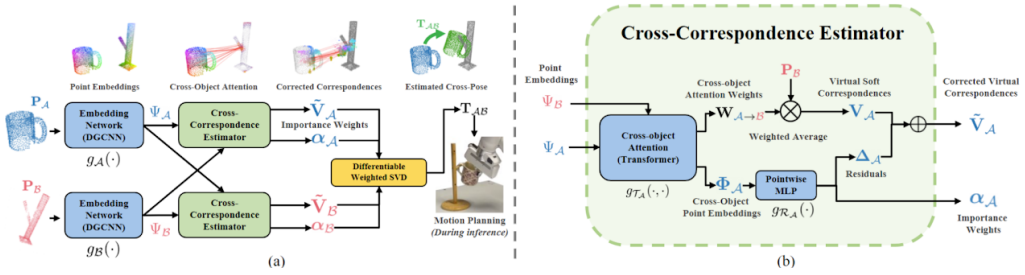


Figure 2.2: TAX-Pose estimates correspondences for predicting rigid transformations between two input point clouds.

for learning to estimate the relative pose between two objects that are in different poses. This module is trained using a dataset that contains pairs of objects in different poses. The input to the module is the point clouds representing the two objects, and the output is the estimation of the relative pose between them. To accomplish this, the module comprises several components.

First, there is a feature extractor that takes the point clouds of the two objects as input and extracts meaningful features from them. These features capture the relevant information needed to establish correspondences between the objects. The feature extractor plays a crucial role in representing the objects' characteristics effectively. Next, a transformer component is employed to estimate the correspondences between the two objects. The transformer leverages the extracted features to establish cross-pose correspondences, identifying the points or regions of the objects that correspond to each other across different poses. These correspondences are essential for understanding the spatial relationships and transformations between the objects.

Finally, a regressor is utilized to predict the relative pose between the objects based on the extracted features and correspondences. The regressor takes the features and correspondences as input and produces an estimation of the relative pose. This information is crucial for determining how the objects should be placed in task-specific locations. Overall, TAX-Pose combines the estimation of cross-pose correspondences with policy learning to enable accurate object placement in task-specific locations. The utilization of 3D correspondences, particularly point-based correspondences, in TAX-Pose demonstrates their significance in manipulation.

Another work which relies on correspondences is point cloud registration where pairs of points from two point clouds are identified as corresponding to the same

physical location. However, traditional Iterative Closest Point (ICP) algorithms [1, 14] encounter challenges in obtaining accurate correspondences, particularly in the presence of noise and partial occlusion. In response to this issue, Deep Closest Point (DCP) [18] introduces a novel approach to address these limitations. DCP leverages deep learning to acquire a discriminative representation of point clouds, capturing the relative pose information effectively. By doing so, it facilitates the more robust identification of correspondences compared to conventional methods. Furthermore, the obtained representation also serves as a valuable initialization step for the subsequent application of ICP, refining the alignment between the point clouds. The effectiveness of DCP is demonstrated through extensive evaluations on various datasets. The results highlight its superiority over traditional ICP techniques in terms of both accuracy and processing speed. This innovative approach presents a promising avenue for improving the robustness and efficiency of correspondence finding in point cloud registration tasks.

2.5 Reinforcement Learning for Manipulation

Reinforcement Learning (RL) is a branch of machine learning that focuses on how an agent can learn to make sequential decisions in an environment to maximize its cumulative rewards. RL is particularly applicable in robotic systems, where the agent interacts with its environment, learns from experience, and adapts its actions to achieve specific goals.

The RL framework involves several key components. First, there is the agent, which is the decision-making entity that learns and takes actions in the environment. The environment represents the external system in which the agent operates. The agent and environment interact in a cyclic manner: the agent takes an action, the environment transitions to a new state, and provides feedback in the form of rewards or penalties. This feedback guides the agent's learning process.

Rewards serve as the primary feedback mechanism in RL. They indicate the desirability of the agent's actions and provide a quantitative measure of the agent's performance. The goal of RL is to find a policy, which is a mapping from states to actions, that maximizes the cumulative rewards over time. The agent's objective is to learn this optimal policy through trial and error, exploration, and exploitation of

its environment.

In the context of cloth manipulation, RL has emerged as a promising approach for controlling robotic systems. Traditional control methods often require complex manual programming or explicit modeling of the cloth's dynamics, which can be challenging due to the inherent complexity and variability of cloth deformations. RL offers a data-driven alternative, where the agent can learn to manipulate cloth by directly interacting with it and receiving feedback from the environment.

One advantage of RL in cloth manipulation is its ability to handle complex and non-linear cloth dynamics without explicitly modeling them. The agent can learn to adapt its actions based on the cloth's responses, allowing for more flexible and robust control. RL algorithms, such as deep reinforcement learning, can leverage neural networks to approximate complex policies and handle high-dimensional inputs, making them well-suited for cloth manipulation tasks.

Furthermore, RL enables the agent to learn from experience and improve its performance over time. By exploring different actions and receiving feedback in the form of rewards, the agent can refine its policy and optimize its actions to achieve desired cloth manipulation objectives. RL algorithms can iteratively update their policies based on observed rewards, leading to adaptive and efficient cloth manipulation strategies.

Actor-critic reinforcement learning: It is a powerful approach that combines policy-based and value-based RL methods. In this framework, an actor network learns a policy that maps states to actions, while a critic network learns a value function that estimates the expected return of a state. By training these two models simultaneously, actor-critic RL can effectively optimize policies in complex environments.

The actor component of the actor-critic architecture is responsible for selecting actions based on the current state. It takes the state as input and outputs the corresponding action to be executed by the agent. The actor is typically implemented as a deep neural network that can handle high-dimensional state spaces and produce continuous or discrete action outputs.

On the other hand, the critic component aims to estimate the value or quality of the chosen actions. It evaluates the value of a state-action pair by estimating the expected return, which is the sum of future rewards that the agent expects to receive. The critic is also implemented as a deep neural network and is trained to minimize

the temporal-difference (TD) error, which measures the discrepancy between the estimated value and the actual observed reward.

The actor and critic networks are trained simultaneously using a process called policy gradient. The policy gradient method computes gradients based on the critic's value estimates to update the actor's policy. By iteratively adjusting the policy based on the feedback from the critic, the actor-critic algorithm can gradually improve its performance in maximizing the expected return.

Actor-critic reinforcement learning has been successfully applied to various manipulation tasks, such as robot arm control, grasping, and object manipulation. Its advantage lies in its ability to combine the strengths of both policy-based and value-based methods. The actor network enables direct policy optimization, allowing for more efficient exploration and exploitation of the state-action space. The critic network provides valuable feedback on the quality of the chosen actions, guiding the actor's learning process.

TD3: Twin Delayed Deep Deterministic Policy Gradient [4] is an algorithm that addresses the instability of the Deep Deterministic Policy Gradient (DDPG) algorithm [7] in environments with high-dimensional action spaces. TD3 introduces three key modifications to enhance stability and learning efficiency: Clipped Double-Q Learning, Delayed Policy Updates, and Target Policy Smoothing.

In Clipped Double-Q Learning, TD3 utilizes two Q-functions instead of one. These Q-functions are updated independently, and the minimum of the two Q-values is used to update the policy. By considering the minimum value, overestimation bias in the Q-function is reduced, mitigating the potential instability arising from inaccurate value estimation.

To further stabilize the learning process, TD3 incorporates Delayed Policy Updates. Unlike DDPG, TD3 updates the policy less frequently compared to the Q-functions. This delayed updating mechanism ensures that the policy changes are more consistent and avoids oscillations that may arise from rapid and frequent policy updates.

Target Policy Smoothing is another enhancement introduced in TD3. It involves adding noise to the target policy during updates. By introducing noise, the policy updates become more robust to variations and inaccuracies in the Q-function, allowing for smoother and more reliable learning.

Algorithm 1 depicts the TD3 algorithm. By iteratively updating the actor and

Algorithm 1 TD3 Algorithm

```
Initialize the actor and critic networks
Initialize the target actor and critic networks
Initialize a replay buffer
for each episode do
    Collect a trajectory of states, actions, rewards, and next states
    Store the trajectory in the replay buffer
    Sample a batch of data from the replay buffer
    Update the critic networks using the sampled data
    Update the actor networks using the sampled data and the target critic
    Update the target actor and critic networks
end for
```

critic networks while incorporating the modifications specific to TD3, the algorithm learns effective policies that maximize cumulative rewards in complex reinforcement learning tasks. The modifications introduced in TD3 enhance stability, reduce overestimation bias, and improve the efficiency of learning compared to the original DDPG algorithm.

HAC-Man: HAC-Man [21] is an approach that extends existing off-policy algorithms, originally designed for continuous action spaces, to handle hybrid action spaces that combine both discrete and continuous components.

For the continuous component, HAC-Man leverages established off-policy algorithms such as TD3. The first step is to train an actor network using the point cloud observations. This actor network generates a vector of continuous action components that represent the desired motion parameters. Additionally, a critic network is utilized to estimate the Q-value based on the observation and the continuous action components. The Q-value provides a measure of the expected return and is used to update the actor network, ensuring that it learns to generate more favorable continuous actions.

In order to predict the discrete component, HAC-Man introduces a discrete set of points within the object point cloud. This discrete component represents the contact location or the specific point to be selected for manipulation. To incorporate this discrete component into the algorithm, the critic network is trained to output a per-point Q-value for each point in the entire point cloud. Each Q-value represents the estimated return when choosing a particular point as the discrete component.

The Q-values serve a dual purpose: they are used to update the actor network to improve the selection of continuous components, and they are also employed to determine the discrete component by selecting the point with the highest Q-value. Furthermore, the actor network is trained to generate per-point continuous vectors for each point in the point cloud. If a specific point is chosen as the discrete component, the corresponding continuous component is used for manipulation, providing the necessary motion parameters for the gripper or manipulator to perform the desired action on the selected point.

By combining these approaches, HAC-Man effectively handles hybrid action spaces, enabling the prediction and coordination of both discrete and continuous components for successful manipulation tasks.

Chapter 3

Literature Review

3.1 Cloth Folding

The field of cloth folding has seen several notable automated policy-based approaches in recent literature. One such approach is FabricFlowNet (FFN) [20] proposed by Weng et al. FFN tackles bimanual cloth folding by estimating flow correspondences between the observed cloth image and the goal cloth image. This method relies on the concept of optical flow to establish correspondences between pixels in different frames of the cloth. However, FFN has a limitation in that it requires strict alignment between the observed cloth and the goal cloth poses in the image. To address this limitation, our first approach extends FFN by proposing a method to align learned 3D correspondences. By establishing spatial relationships between points in the observation and goal configurations, our approach enables precise alignment, resulting in improved folding performance even for unaligned goals compared to FFN. Additionally, FFN relies on the use of subgoals for long horizon cloth folding, introducing complexity and manual intervention, making it less autonomous. In contrast, our second approach, based on reinforcement learning and point-based correspondences, explores and learns the trajectory autonomously without the need for subgoals, enabling long horizon planning.

Another notable policy in the field of cloth folding is FoldsFormer [10], which utilizes a space-time attention mechanism to capture instruction information and perform cloth folding. This method can manipulate cloth even when the observed and

goal cloth configurations are not aligned, due to the space-time attention mechanism. However, similar to FFN, FoldsFormer also requires the use of subgoals, limiting its autonomy. Moreover, FoldsFormer heavily relies on fine-tuning on specific test examples, which leads to poor performance on other examples that it has not been fine-tuned on. In contrast, our RL-based policy, which incorporates point-based correspondences, does not require fine-tuning on specific data and is applicable to both folding and smoothing tasks, making it a general cloth manipulation policy.

Fabric Descriptors [5], introduced by Ganapathi et al., is a method for learning correspondences in fabric manipulation tasks using a dense contrastive loss. This approach focuses on learning correspondences between different fabric configurations. However, once the correspondences are learned, the proposed policy relies on human demonstrations for cloth manipulation. In contrast, our method can learn and estimate correspondences without the need for human demonstrations, demonstrating its capability for autonomous cloth manipulation.

Overall, the literature on cloth folding presents a range of approaches with their respective strengths and limitations. Our proposed methodologies offer advantages such as improved alignment, autonomy, generalization across tasks, and applicability to both folding and smoothing. By incorporating point-based correspondences, our approaches contribute to the advancement of cloth manipulation techniques, providing more effective, autonomous, and robust solutions for cloth folding tasks.

3.2 Cloth Smoothing

In the domain of cloth smoothing, there are several notable approaches that have been proposed. One such approach is the Visual Connectivity Dynamics (VCD) method [9], which estimates a mesh from an input point cloud and predicts cloth dynamics. However, the execution of VCD is relatively slow due to the need for constructing a graph and estimating collisions between edges. In contrast, our proposed method offers a faster alternative for cloth smoothing. Additionally, VCD is limited to performing smoothing and lacks the capability of aligning the cloth to a specific goal configuration. In comparison, our approach is more generic and versatile, as it can handle both smoothing and alignment tasks.

Another approach in the realm of cloth smoothing is Cloth Funnels [2]. This

method utilizes self-supervised rewards to learn both cloth smoothing and alignment. However, their alignment procedure is based on an iterative version of the Procrustes' algorithm, which is primarily designed for aligning rigid objects. Since fabrics are deformable materials, the alignment achieved using Procrustes can be prone to local optima. In contrast, our proposed approach employs the Random Sample Consensus (RANSAC) algorithm for aligning deformed fabrics, which provides an asymptotic and globally optimal alignment. By utilizing RANSAC, our method overcomes the limitations of local optima and achieves more robust alignment results. Furthermore, while Cloth Funnels solely relies on self-supervised learning from random actions, our approach incorporates the use of reinforcement learning (RL) for exploration, enabling more effective and adaptive cloth manipulation.

Overall, the existing literature on cloth smoothing presents various approaches with their respective strengths and limitations. Our proposed method offers advantages such as faster execution, the capability to perform both smoothing and alignment tasks, and the use of RANSAC for globally optimal alignment. By addressing these challenges, our approach contributes to the field of cloth manipulation, providing a more efficient and versatile solution for cloth smoothing and alignment tasks.

Chapter 4

Point-based Correspondence Estimation for Cloth Alignment and Manipulation

4.1 Introduction

A fundamental aspect of successful cloth manipulation is establishing correspondences between the current observation and the goal configuration. These correspondences provide critical spatial associations necessary for planning and executing folding actions. However, while prior methods have proposed to learn correspondences for cloth [5, 20], they do not explicitly use such methods for reasoning about the *alignment* between the observed cloth and the desired configuration. Alignment refers to adjusting the cloth configuration to match a specific target alignment andis a crucial step in cloth manipulation. Prior correspondence-based policies do not handle cases where the cloth and goal are not already aligned [20], or rely on human demonstrations [5].

In this work, we propose **FabricFlowAlignNet** (**FFAN**), an approach that combines the use of correspondences and symmetry-handling techniques to learn a goal-conditioned cloth manipulation policy. Our method leverages correspondences to "virtually" align the observation and goal point clouds, enabling the policy to

determine the appropriate actions to execute on the observation. By incorporating these correspondences and symmetry handling, our approach aims to acquire an understanding of cloth folding strategies and develop a manipulation policy capable of accurately and efficiently folding clothes. This is particularly beneficial in challenging scenarios where the observed cloth is rotated or unaligned with the desired goal configuration.

We evaluate the performance of our method against a state-of-the-art folding approach [20] on a folding task, where the goal and observation poses are not aligned. Our method reasons about symmetries and employs correspondences to deal with unaligned goals, unlike the baseline. The results demonstrate the effectiveness and robustness of our approach in achieving successful cloth folding when the observation and goal configurations are unaligned.

4.2 Methodology

In this section, we describe FabricFlowAlignNet (FFAN), our approach for estimating observation-goal correspondences to align and manipulate cloth.

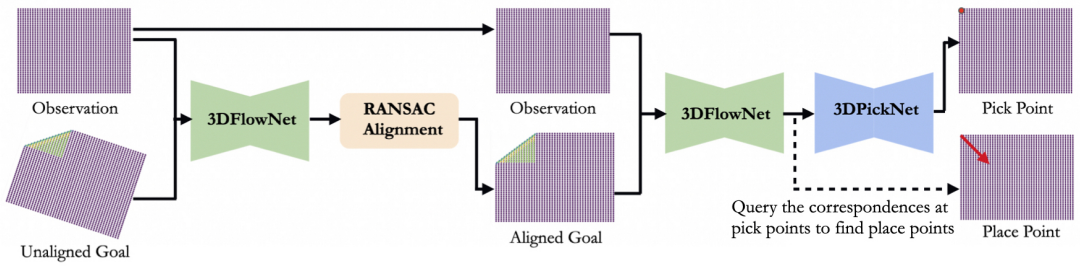


Figure 4.1: Overview of the FFAN pipeline: 1) Estimating correspondences and aligning the observation and goal pair. 2) Estimating correspondences to predict pick points for the action. 3) Querying the correspondences at pick points to identify the place points.

Rather than relying on a single network to estimate the entire cloth manipulation process, we adopt a two-network approach, namely 3DFlowNet and 3DPickNet. The division of tasks between these networks allows for a more specialized and effective approach.

The first network, 3DFlowNet, focuses on estimating correspondences between the current cloth observation and the desired goal configuration. By analyzing the

point cloud data, 3DFlowNet establishes meaningful associations between the two configurations, facilitating subsequent steps in the cloth manipulation process.

On the other hand, the second network, 3DPickNet, utilizes the estimated flow obtained from 3DFlowNet to predict appropriate pick points. These pick points play a crucial role in achieving the desired goal configuration. By leveraging the estimated flow, 3DPickNet determines the optimal locations to initiate cloth folding, enabling precise and effective manipulation.

4.2.1 Learning Correspondences for Point Clouds

We propose a 3D, flow-based correspondence estimator called 3DFlowNet, a component of our overall pipeline. 3DFlowNet takes the observation and goal point clouds c_o and c_g as input, and outputs 3D flow $\hat{f}$. 3DFlowNet is a non-trivial extension of the FlowNet from FabricFlowNet [20], which was limited to image input and 2D flow output. A schematic overview of 3DFlowNet can be found in Fig. 4.2.

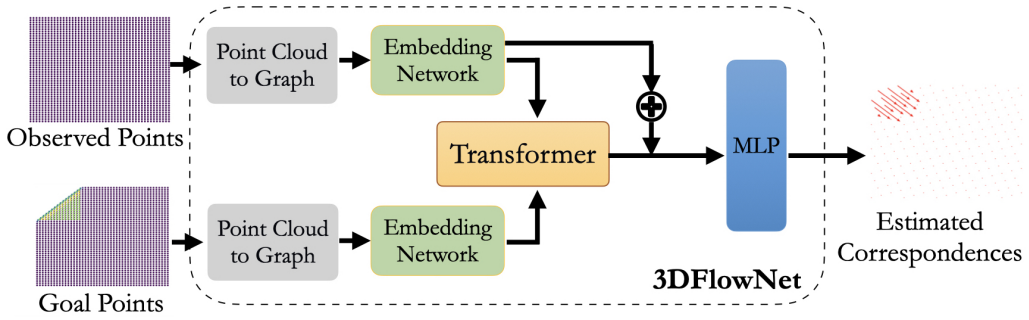


Figure 4.2: 3DFlowNet architecture: Estimates point-based correspondences between observation and goal point clouds.

We first transform the point clouds into a graph, where nodes represent cloth particles and are connected to their neighboring particles on the cloth mesh. This step requires privileged state information from the simulator of the cloth mesh edges, which would not be available in the real world; estimating these edges is an area of future work and could leverage prior methods like VCD [9]. We embed the input graphs by employing a graph neural network H, which outputs embeddings for each node in the graph: $c'_o, c'_g \in \mathbb{R}^{N \times F}$.

We then use a Transformer network [15] denoted as T to perform cross-attention between observation and goal features. Our approach is inspired by prior Transformer-based per-point networks like DCP [17] and TAX-Pose [11]. T takes c'_o and c'_g as input and outputs transformed embeddings $c' \in \mathbb{R}^{N \times F}$. c' The resulting transformer embeddings, c', are then summed with the original observation embeddings c'_o to produce c''_o.

To estimate correspondences between the two configurations, we pass c''_o through MLP layers M to produce estimated correspondences $\hat{f} \in \mathbb{R}^{Nx3}$. These correspondences represent how each cloth particle in N transports to achieve the goal configuration.

To train 3DFlowNet, we use a weighted L2 loss between the estimated and ground truth correspondences. The ground truth correspondences are computed as the difference between the point clouds c_g and c_o. The weighted L2 loss function is defined as:

$$\mathcal{L}_2(\hat{f}, f) = \sum_{i=1}^{N} w_i (f_i - \hat{f}_i)^2 \tag{4.1}$$

where $\hat{f}$ represents the estimated correspondences, f represents the ground truth correspondences, and N is the total number of points in the point cloud. The weights w_i are higher for ground truth pick points.

4.2.2 Iterative Correspondence Estimation

To improve correspondence estimation when the displacement between observed and desired goal configurations is large, we introduce an iterative approach to improve the accuracy of our correspondence estimation.

Our iterative process involves transporting the input point cloud to the positions indicated by the estimated correspondences, and then re-computing the estimation with this intermediate point cloud. Each iteration of this procedure should further refine the estimated correspondence.

In each iteration, we utilize the trained 3DFlowNet model to estimate the correspondence between the intermediate point cloud $\hat{c}_o$ and the target configuration c_g. By integrating the estimated correspondence into the observation, we simulate

Algorithm 2 Iterative Correspondence Estimation

Input: Trained 3DFlowNet, Point Clouds c_o, c_g
Initialize all zeros $\bar{f} \in \mathbb{R}^{N\times 3}$
$\hat{c}_o := c_o$
for k = 1...K **do**
    $\hat{f}$ = 3DFlowNet($\hat{c}_o$, c_g)
    $\bar{f}$ += $\hat{f}$
    $\hat{c}_o$ += $\hat{f}$
end for
return $\bar{f}$

the application of the flow to progressively approach the target configuration. The algorithm for iterative correspondence estimation is summarized in Alg. 2.

4.2.3 RANSAC Alignment for Unaligned Goals

The correspondences estimated by 3DFlowNet indicate how each point in the observed cloth configuration should move to reach the desired configuration. In the case where the observation and goal are aligned with respect to each other, this per-point flow correspondence represents a desired cloth manipulation, which we can use to estimate the action (Sec. 4.2.4). However, in cases where the observation and goal are not aligned, the flow correspondences contain both information about alignment as well as the desired manipulation.

To address the cases where the goal is not aligned, we first propose estimating the alignment using the flow-based correspondence and RANSAC [3]. The forward pass through 3DFlowNet provides the estimated correspondences. The RANSAC procedure attempts to find an alignment transform with the maximum number of inlier cloth points as follows:

1. Sample three indices (i, j, k) on the cloth.
2. Compute the transformation matrix T between the 3 sampled cloth points (p_i, p_j, p_k) and their estimated correspondences ($p_i + \hat{f}_i$, $p_j + \hat{f}_j$, $p_k + \hat{f}_k$).
3. Compute inliers by transforming all current cloth points p according to T, computing the distance between transformed points and points transported using estimated flow $||Tp - (p + \hat{f})||$, and thresholding the per-point distance

by an epsilon ϵ.

4. Sample m times and choose the transformation matrix T with the maximum number of inliers.

Once the alignment T has been estimated, we virtually align the observation and goal, re-estimate correspondences given the alignment, and determine the manipulation action.

We propose to decouple these two problems: first, we use the flow-based correspondences to virtually align the observation and goal, then we compute the desired manipulation by re-estimating correspondences the aligned configurations.

4.2.4 Estimating the Pick Location for an Action

To predict the pick points necessary for cloth manipulation, we introduce a neural network called 3DPickNet. Similar to FFN [20], our method supports bimanual manipulation and is capable of estimating both pick points p_1 and p_2. The inputs to 3DPickNet are the current observation c_o and the estimated correspondences $\hat{f}$ between c_o and the goal configuration c_g. The architecture of 3DPickNet is depicted in Figure 4.3.

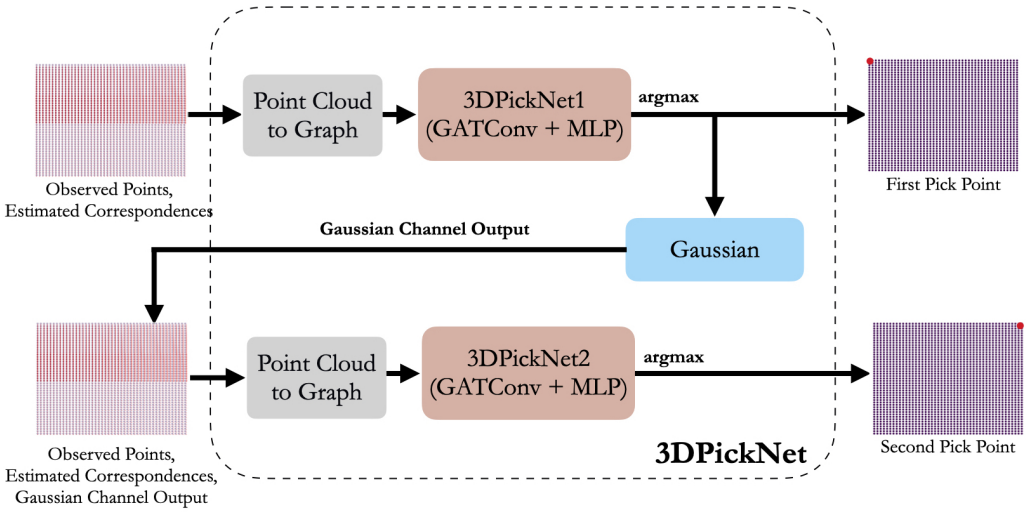


Figure 4.3: 3DPickNet architecture: Predicts pick point quality using estimated correspondences between observation and goal point clouds.

To enable the prediction of the second pick point conditioned on the first pick point, we utilize two separate networks: 3DPickNet1 and 3DPickNet2. In 3DPickNet1, we concatenate c_o and $\hat{f}$ and create a graph representation of the point cloud. Each node in the graph is represented as $[x, y, z, \hat{f}]$. 3DPickNet1 generates a probability value for each node to be selected as the first pick point. The node with the highest probability is identified as p_1.

3DPickNet2 is responsible for predicting the second pick point p_2, taking p_1 into account. In this network, we introduce an additional input channel called $\hat{p_1}$, which represents a 3D Gaussian distribution centered on p_1. This channel assigns higher values to nodes near p_1 and lower values to nodes farther away, to give PickNet2 information about the first pick location when predicting the second pick point p_2.

The expression for $\hat{p_1}$ is given by:

$$\hat{p_1} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}{2\sigma^2}\right) \tag{4.2}$$

Here, x_0, y_0, z_0 are the coordinates of the pick point p_1, and σ controls the spread of the Gaussian distribution. Here, σ is a hyper-parameter and is empirically chosen.

Note that although the output of 3DPickNet1 provides a probability value for each node to be selected as the first pick point, it does not represent a distribution since the sum of the values is not guaranteed to be 1. Hence, we introduce the Gaussian channel centred around the node with the highest probability, p_1.

For training 3DPickNet, we use a weighted binary cross-entropy loss. The loss function compares the predicted probabilities of nodes being pick points with the ground truth labels. The binary cross-entropy loss function is defined as:

$$L(p, y) = \sum_{i=1}^{N} -w_i(y_i \log p_i + (1 - y_i) \log(1 - p_i)) \tag{4.3}$$

where p represents the predicted probabilities, y is the ground truth labels, and N is the total number of nodes. The weights w_i are higher for ground truth pick points.

Once the pick points p_1 and p_2 are predicted using the estimated correspondences $\hat{f}$, the corresponding actions can be executed to achieve the desired goal configuration.

4.2.5 Implementation Details

Dataset

We collect a dataset in SoftGym [8], a deformable object simulator, to train and evaluate our approach. This dataset is the same as the one used in the FabricFlowNet [20], but we extract point clouds from SoftGym to represent the cloth instead of using depth images, and use 3D pick and place points instead of 2D.

The dataset is generated by sampling random actions biased towards grasping the corners of a square towel. Each instance in the training, validation, and test sets consists of a tuple (c_o, c_g, a), where c_o and c_g are point clouds representing the current observation and the desired goal configuration, respectively. The ground truth action a corresponds to the action that achieves the goal configuration. In our approach, the action space is defined as $a = (p_1, p_2, q_1, q_2)$, where p and q represent 3D pick and place points. These pick and place points are selected from a set of indices representing the points in the point cloud.

Figure 4.4 showcases RGB images of two pairs of observation and goal examples from the training set. These images provide a visual representation of the cloth folding task within the dataset.

3DFlowNet

The graph neural network H consists of two Graph Attention layers (GATConv) [16]. The MLP network architecture M consists of two fully-connected layers.

To train all three modules of FlowNet (H, T, M), we perform end-to-end training using gradient descent on the weighted loss function. Due to limited compute, we use a batch size of 1. The Adam optimizer [6] is employed with an initial learning rate (η) of 0.001, ϵ set to 10^{-8}, and β_1 and β_2 values of 0.9 and 0.999, respectively.

3DPickNet

The 3DPickNet architecture consists of three Graph Attention Network layers and two MLP layers for both 3DPickNet1 and 3DPickNet2. At the end of each network, a Sigmoid layer computes the probability of each node being a pick point. 3DPickNet1

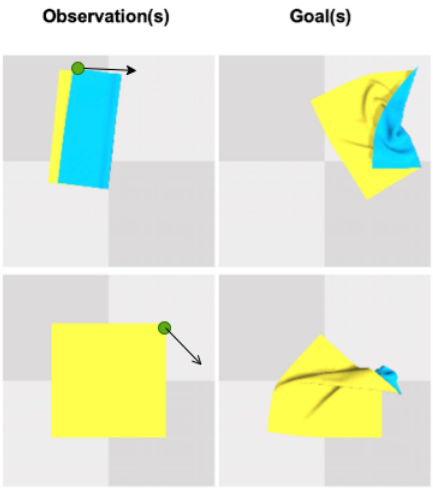


Figure 4.4: Examples of the Cloth Folding Dataset. Images on the left depict the observation, while images on the right showcase the desired goal. The action is indicated by a pick point, representing the location where the cloth should be manipulated. The arrow(s) indicate the action vector(s) to be applied at the pick point.

represents each node with a six-dimensional feature, while 3DPickNet2 utilizes a seven-dimensional feature to accommodate the additional information provided by $\hat{p_1}$.

To train both 3DPickNet1 and 3DPickNet2, we use a batch size of 64. Similar to 3DFlowNet, we utilize the Adam optimizer with an initial learning rate of 0.001, and β_1 and β_2 values of 0.9 and 0.999, respectively.

4.3 Experiments

Our experiments investigate the following questions: (1) How does FFAN compare with FabricFlowNet (FFN) [20] on aligned goals? (2) How does FFAN compare with FFN on unaligned goals? We evaluate the methods in simulation, using the average L2 distance between cloth points in the achieved vs. desired point clouds as our error metric.

4.3.1 Performance on Aligned Goals

We use the same test set as FFN [20] to evaluate performance on aligned goals. This test set consists of 40 single-step goals, where both the observation and the goal positioned at the center of the workspace with the same orientation. For this experiment, since the observation and goal pairs are aligned, we do not use alignment estimation with FFAN.

Table 4.1 presents the performance comparison between our method and FFN. The results demonstrate that our method performs comparably to FFN on aligned goals, with only a marginal difference in average particle distance.

Table 4.1: Folding performance on 40 aligned single-step test goals.

Method	**Average Particle Distance** (mm) ↓
FFN [20]	4.26 ± 2.62
FFAN (Ours)	5.54 ± 1.71

The comparison reveals that our method performs comparably to FFN on aligned goals, with only a marginal difference in average particle distance. Notably, FFAN requires privileged state information to construct meshes from the input point clouds. Furthermore, FFAN employs complete point clouds whereas FFN operates on visible depth images. To address the first assumption, methods like VCD [9] can be used to construct meshes. Our second approach (Chapter 5) eliminates the need for mesh construction and relies solely on visible point clouds, effectively overcoming both these assumptions.

Qualitative Analysis

To gain a more comprehensive understanding of the performance of individual components within the pipeline, we visualize the folding operation alongside the quantitative error metric. Figure 4.5a showcases the achieved goal configurations obtained using our method. Each example is accompanied by its corresponding error metric in meters, displayed above the visualization. The green dot(s) represent the predicted pick points, while the arrow illustrates the estimated flow for the predicted pick points. In contrast, Figure 4.5b illustrates the desired goal configurations that serve as the

target for the folding operation. This visual analysis allows for a closer examination of the performance of each component.

Our method demonstrates excellent performance on the majority of goals, achieving perfect results. However, there are instances where our method encounters challenges, particularly in scenarios involving bimanual manipulation, which entails folding using two robotic arms, as shown in Fig. 4.6.

4.3.2 Performance on Unaligned Goals

We also evaluate the performance on unaligned goals, where the goal cloth configuration is randomly rotated and therefore not aligned with the initial observed configuration. We conducted experiments on three test sets: Easy, Medium, and Hard, where each test set corresponds to a different range of rotations. Easy encompasses angles between -5 and 5 degrees, Medium ranges from -45 to 45 degrees, and Hard covers a complete rotation from 0 to 360 degrees.

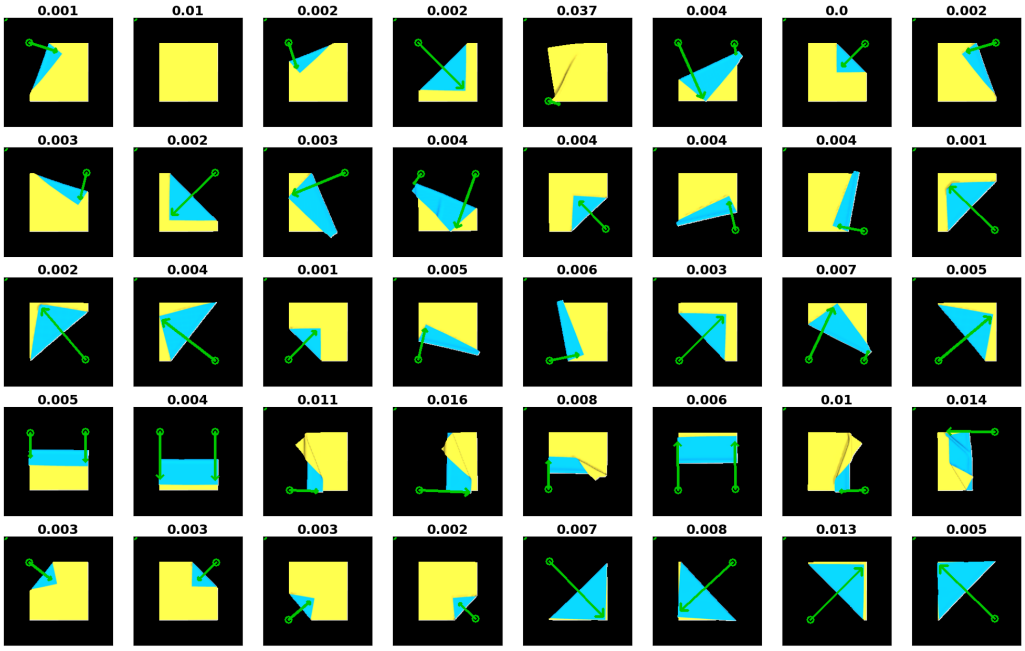
We evaluated the performance of FFAN in two scenarios: using ground truth vs. estimated correspondences for RANSAC alignment. Figure 4.7 presents a comparison of the two methods against FFN across all four test sets: aligned, easy, medium, and hard. From the results, we observe that our method with estimated correspondences outperforms FFN on the Medium and Hard tasks. However, using ground truth correspondences for RANSAC alignment yields even better results across all tasks, surpassing the performance of FFN. This demonstrates the potential for further improvement by improving the correspondence estimation.

By showcasing the comparative performance of our method against FFN and highlighting the benefits of using ground truth correspondences, we emphasize the effectiveness of our approach in handling unaligned goals and the significance of accurate correspondences for achieving superior results.

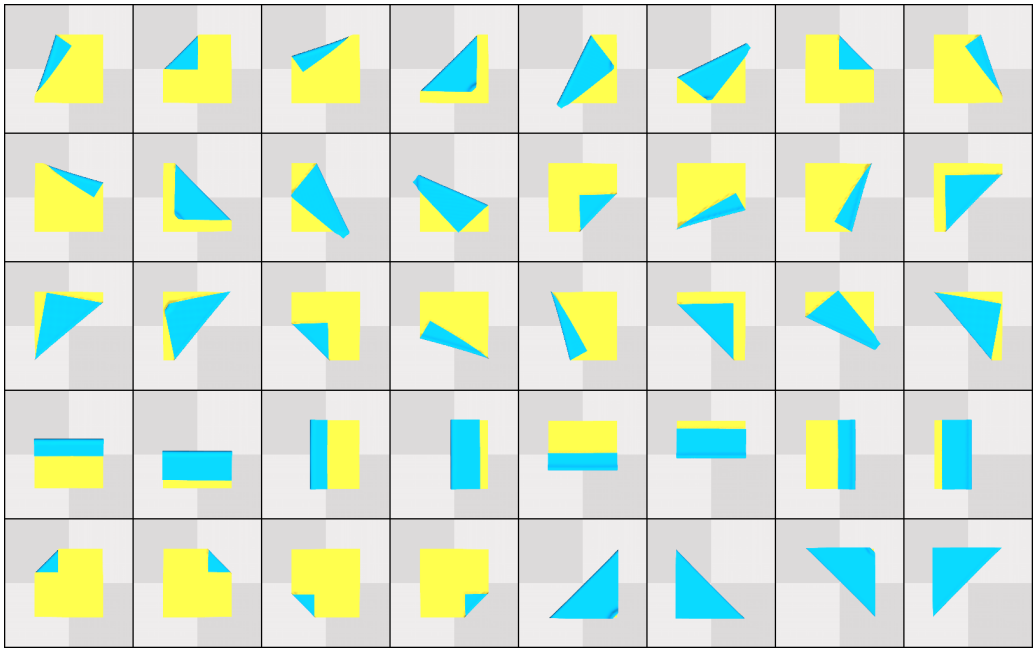
Qualitative Analysis

In addition to quantitative evaluation, we conducted a qualitative analysis to further assess the performance and robustness of our proposed approach, FFAN, in comparison to FFN.

To illustrate the robustness of FFAN against misalignments, we depict the observed



(a) Achieved Configurations by FabricFlowAlignNet



(b) Desired Goal Configurations

Figure 4.5: Qualitative evaluation of FFAN on the aligned testset. The green dot(s) represent the predicted pick points, while the arrow illustrates the estimated flow for the predicted pick points. Each example is accompanied by its corresponding error metric in meters, displayed above the visualization.

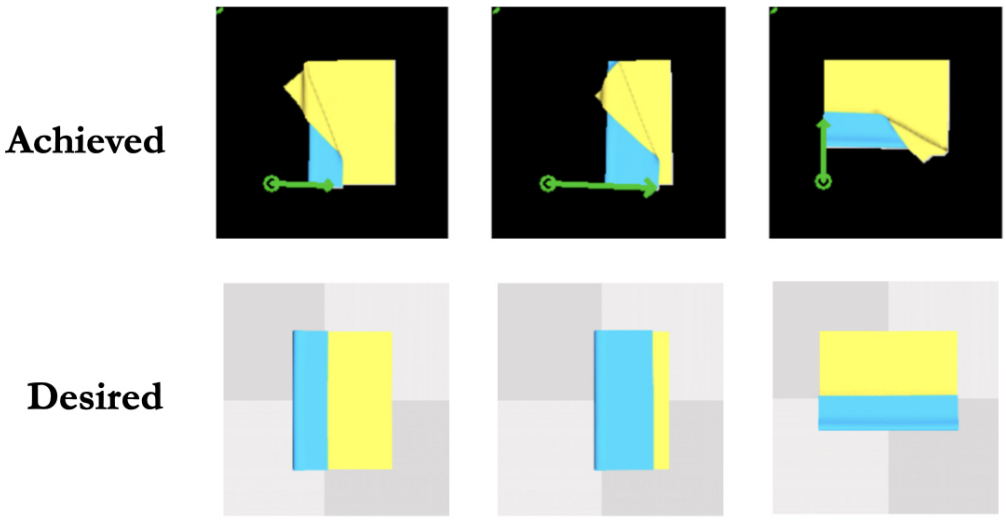


Figure 4.6: Examples showcasing FFAN's prediction of both pick points as same cloth node, resulting in suboptimal cloth manipulation.

cloth configuration, the unaligned goal configuration, and the performance of both FFAN and FFN. Fig 4.8 clearly demonstrates how FFAN outperforms FFN in aligning the cloth configuration with the unaligned goal. This showcases the ability of FFAN to handle misalignments and effectively align the cloth with the desired goal configuration.

Furthermore, we Fig 4.9a showcases the achieved cloth configurations by FFAN on unaligned goals, compared to the unaligned desired goals (Fig 4.9b) from the medium test set. The results from this qualitative analysis look promising, with FFAN successfully generating cloth configurations that closely resemble the desired goals. This demonstrates the efficacy of our approach in achieving accurate and desired cloth configurations, even in challenging scenarios where the initial and desired goals are not aligned.

4.3.3 Ablations

No Iterative Correspondence

In this section, we ablate our approach by removing iterative correspondence estimation (Sec. 4.2.2). Through this experiment, we aim to highlight the importance of

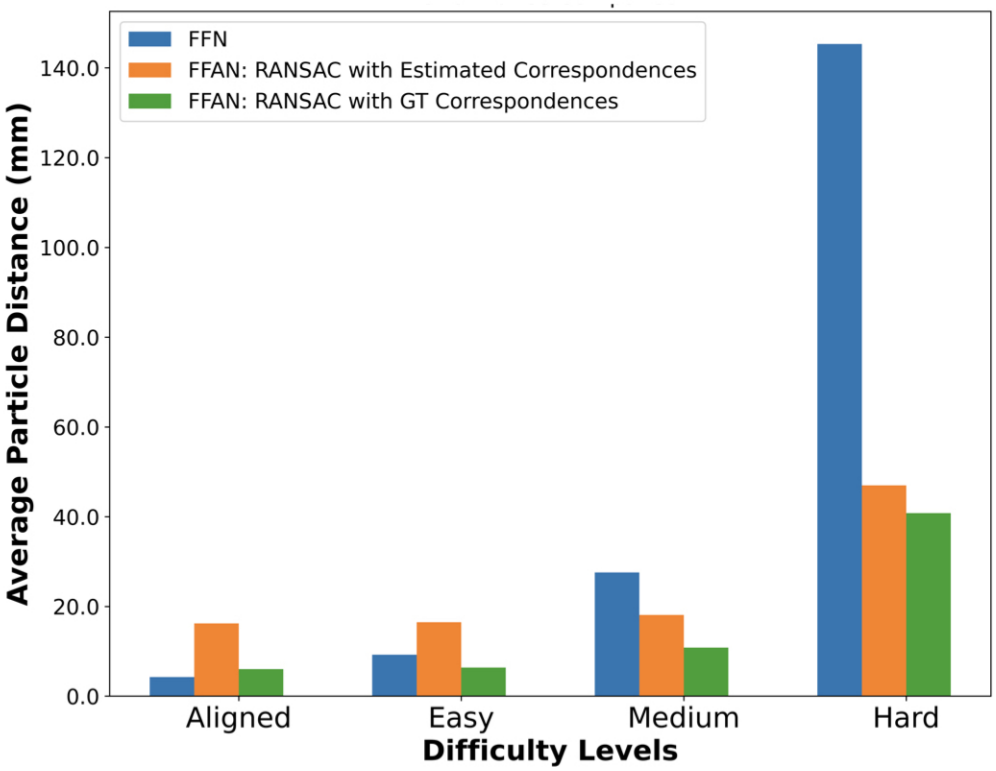


Figure 4.7: Comparison of folding performance on different test sets.

iterative correspondence estimation and its impact on the quality of correspondences, ultimately influencing the success of the folding task.

Table 4.2 shows that average particle distance error is higher when iterative correspondence estimation is removed.

Table 4.2: Effect on folding performance of iterative correspondence estimation.

Method	**Average Particle Distance** (mm) ↓
FFAN w/o Iter. Corresp.	10.591
FFAN w/ Iter. Corresp.	**5.54**

The contrasting results obtained from the two sets of experiments underscore the importance of iterative correspondence estimation and its impact on achieving successful folding performance.

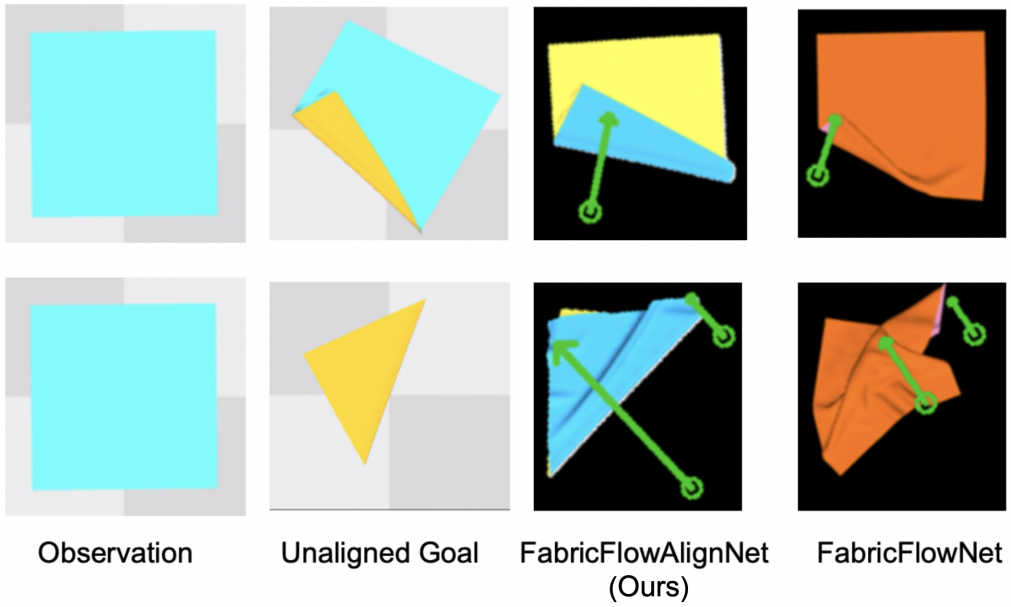


Figure 4.8: Performance of FabricFlowNet (FFAN) vs. FabricFlowNet (FNN) on unaligned goals.

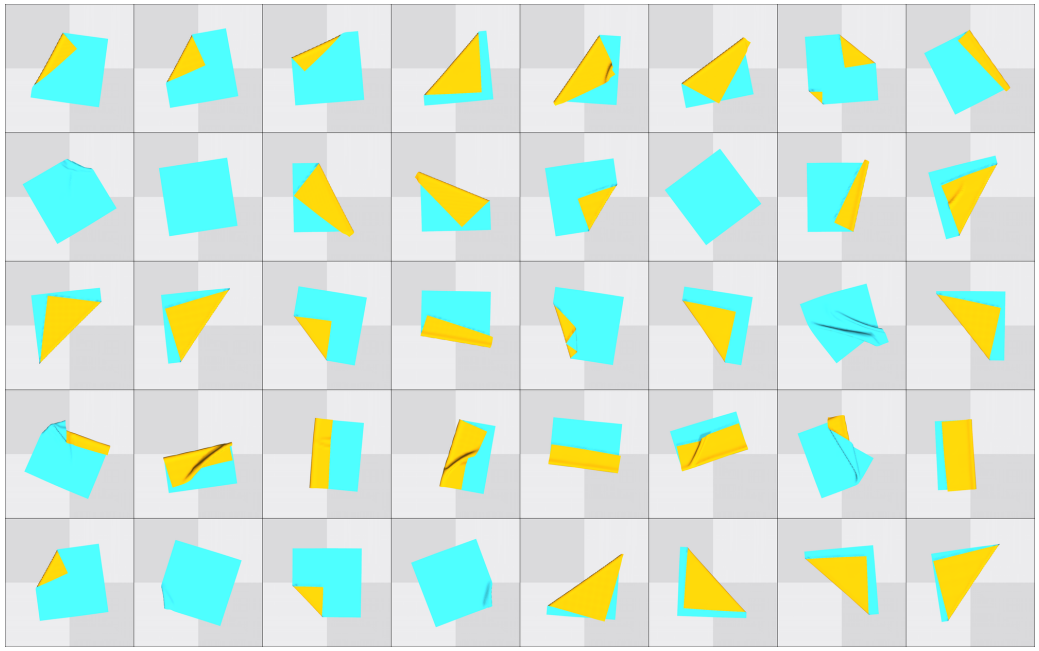
Number of Iterations for Iterative Correspondence

To determine the number of iterations to run for iterative correspondence estimation, we measured performance while increasing the number of iterations on a validation set. We used flow prediction error, an unweighted version of Eq. 4.1, as our performance metric. We evaluated number of iterations ranging from $k = 1$ (run 3DFlowNet once) to 4. Note that we did not retrain 3DFlowNet in an iterative manner.

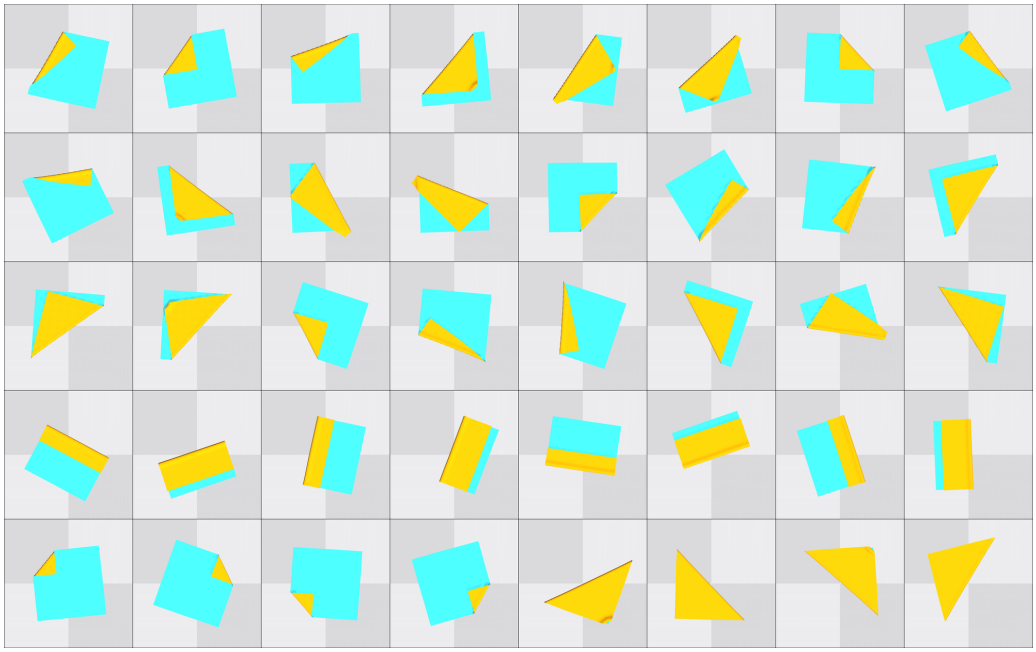
Figure 4.10 shows the flow prediction error as a function of the number of iterations (k) in the iterative flow process. As we increase k from 1 to 3, there is a notable decrease in the flow prediction error; however, beyond $k = 3$, we observed a slight increase in the error. Based on these observations, we empirically determined that the number of iterations for iterative flow correspondence estimation is $k = 3$.

4.4 Conclusion

In this work, we propose **FabricFlowAlignNet** (**FFAN**), a goal-conditioned policy for cloth alignment and folding. Our approach uses flow-based correspondence



(a) Achieved Configurations by FFAN



(b) Unaligned Goal Configurations

Figure 4.9: Qualitative Evaluation of FFAN on Medium testset.

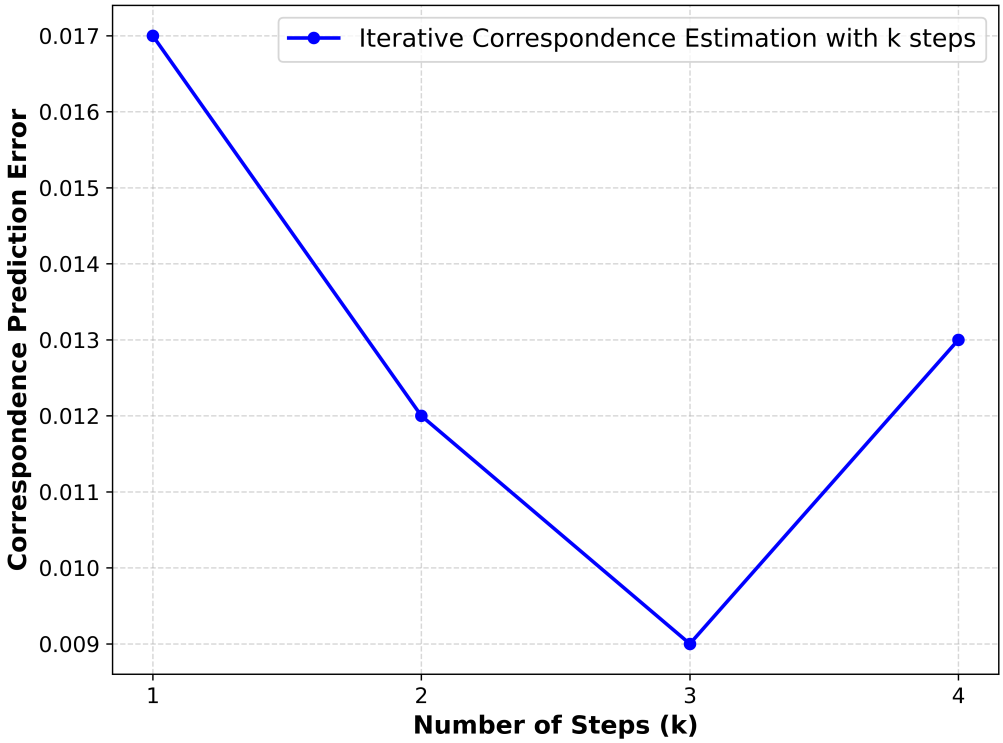


Figure 4.10: Effect of number of iterative correspondence steps on the correspondence prediction error.

estimation to reason about the alignment between the observed cloth and desired goal, before predicting manipulation actions given an estimated alignment.

4.4.1 Key Insights

Through extensive experimentation and analysis, we have observed that FFAN performs on par with FFN when it comes to handling aligned goals, where there is no rotation between the observation and goal. This indicates that our approach is capable of achieving comparable results to the state-of-the-art method in scenarios where the initial configuration and desired goal are already aligned.

However, where FFAN truly shines is in its robustness to rotations between the observation and goal. Unlike FFN, which experiences a significant degradation in performance as the level of rotation increases, our method demonstrates superior performance across different rotation levels.

This stark contrast in performance highlights the importance of correspondences between the observation and goal. By leveraging estimated correspondences and employing the RANSAC alignment technique, we are able to effectively align the observation and goal, thereby enabling more accurate folding operations.

Our results demonstrate that accurate correspondences play a crucial role in achieving successful alignment and folding. By improving the estimation of correspondences, our method excels in scenarios where rotations are present, outperforming FFN on medium and hard tasks. This emphasizes the significance of robust correspondences in achieving superior performance and showcasing the importance of accurate estimation in addressing alignment challenges.

In summary, FFAN not only performs at par with FFN on aligned goals but also exhibits superior robustness to rotations, providing a more reliable and effective solution. By underscoring the importance of correspondences in achieving accurate alignment and successful folding, our approach offers valuable insights and advancements in the field of goal-driven manipulation tasks.

4.4.2 Limitations and Challenges

While our method exhibits strong performance in cloth folding tasks, it is crucial to acknowledge the limitations and challenges associated with its approach. One notable limitation is the current requirement for input meshes, which may not be readily available in real-world cloth manipulation scenarios. In practical applications, estimating the mesh representation of the cloth becomes an additional step that introduces complexity and potential inaccuracies.

Furthermore, our method, like other state-of-the-art approaches such as FFN, relies on predefined sub-goals to guide the folding process. This reliance on explicit sub-goals can be restrictive, as it assumes prior knowledge of the desired cloth configurations or folding patterns. In real-world scenarios, it may not always be feasible to have access to such predefined sub-goals or easily define them. This limitation can impede the adaptability of the method to handle novel cloth types, unseen initial configurations, or variations in folding requirements.

To address these limitations, it is essential to explore alternative approaches that do not depend on explicit sub-goals. By developing methods that can operate

without predefined sub-goals, we can enhance the generalization capability of cloth manipulation systems. Such approaches have the potential to adapt to a wider range of cloth types, folding scenarios, and configurations, enabling more versatile and flexible cloth manipulation.

Additionally, reducing reliance on human experts for providing subgoals can significantly improve the efficiency and cost-effectiveness of cloth manipulation systems. The process of manually defining subgoals can be time-consuming and expensive, especially when dealing with complex or large-scale cloth manipulation tasks. By developing fully autonomous systems that operate without human supervision, we can alleviate the burden of human intervention and enable more efficient cloth manipulation processes.

In particular, when considering tasks like smoothing, where achieving a desired flattened configuration may not have explicit sub-goals, an alternative approach becomes even more crucial. The ability to handle such tasks without predefined sub-goals allows for greater flexibility and applicability in real-world scenarios.

Addressing the limitations associated with predefined sub-goals and striving towards fully autonomous cloth manipulation systems will contribute to the development of more robust and adaptable approaches.

Chapter 5

Point-based Correspondences for Long-Horizon Cloth Manipulation

5.1 Introduction

In the previous work, we discussed the limitations of existing methods for cloth manipulation, particularly their reliance on predefined sub-goals to guide the folding process. This dependency on explicit sub-goals restricts the adaptability of these methods, as it assumes prior knowledge of desired cloth configurations or folding patterns. However, in real-world scenarios, access to such predefined sub-goals may not always be feasible or easily definable, hindering the versatility of cloth manipulation systems.

Addressing these limitations and exploring alternative approaches that do not rely on explicit sub-goals is crucial for enhancing the generalization capability of cloth manipulation systems. By developing methods that can operate without predefined sub-goals, we can expand the range of cloth types, folding scenarios, and configurations that can be effectively handled. This, in turn, enables more versatile and flexible cloth manipulation, empowering these systems to adapt to novel situations and requirements.

One significant advantage of eliminating the reliance on explicit subgoals is the elimination of the reliance on expensive human experts. In the current paradigm,

human experts are often required to provide the sub-goals necessary for guiding the folding process. This manual definition of sub-goals can be time-consuming, labor-intensive, and expensive, particularly when dealing with complex or large-scale cloth manipulation tasks. Developing fully autonomous systems that operate without human supervision reduces the burden of human intervention, making cloth manipulation processes more efficient and cost-effective.

Moreover, addressing the limitations associated with predefined sub-goals is particularly important for tackling longer horizons and complex cloth manipulation tasks. Tasks such as smoothing, where achieving a desired flattened configuration may not have explicit sub-goals, become more challenging under the traditional paradigm. By developing alternative approaches that do not depend on explicit sub-goals, we can empower cloth manipulation systems to handle such tasks effectively, offering greater flexibility and applicability in real-world scenarios.

In this work, we propose a policy **HAC-Cloth** that leverages correspondences to learn a general cloth manipulation policy capable of performing folding or smoothing tasks without relying on sub-goals. We employ reinforcement learning to autonomously learn the optimal trajectory for achieving the desired goal configuration. In our method, correspondences not only serve as goal specifications but also provide insightful information to guide the policy effectively.

To evaluate the effectiveness of HAC-Cloth, we conduct comprehensive experiments comparing its performance against state-of-the-art folding [20] and smoothing method [9]. The evaluations are conducted on long-horizon cloth manipulation tasks, where the cloth had to be folded or smoothed without the use of explicit sub-goals. By removing the reliance on sub-goals, HAC-Cloth demonstrated its capability to handle complex cloth manipulation scenarios, showcasing its flexibility and adaptability.

The results demonstrate the effectiveness of HAC-Cloth in achieving autonomous cloth manipulation. HAC-Cloth outperforms the state-of-the-art folding method in terms of accuracy, demonstrating its superiority in achieving precise and efficient cloth folding without the need for predefined sub-goals. Additionally, HAC-Cloth showcased remarkable performance when compared to the state-of-the-art smoothing method, providing further evidence of its effectiveness in autonomous cloth manipulation tasks.

5.2 Methodology

In this section, we present the methodology employed for achieving autonomous cloth manipulation. We begin by describing the action space utilized in our approach, HAC-Cloth, which defines the available actions for the agent during the cloth manipulation process. Next, we introduce the concept of hybrid actor critic maps, which serve as a guiding mechanism for the agent's decision-making process. An essential aspect of our methodology is the utilization of point-based correspondence estimation. The subsequent sections will delve into the details of each component of our proposed approach, **HAC-Cloth**.

5.2.1 Action Representation

In HAC-Cloth, the action space is composed of two elements: a pick point and a 3D action vector. The pick point is selected from a discrete set, while the action vector represents a continuous movement applied to the chosen pick point.

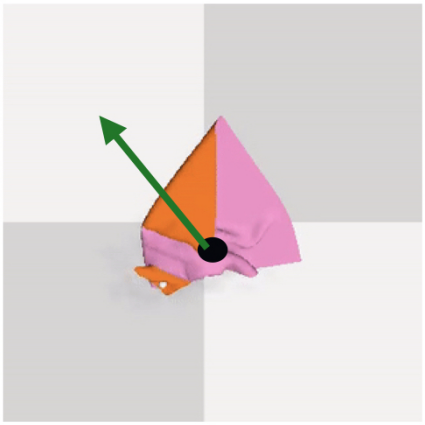


Figure 5.1: Illustration of the action space. The black dot indicates the pick point, while the green arrow represents the 3D action vector applied to the pick point. The pick point is selected from a discrete set of points, while the action vector exists in a continuous space.

To define the pick point, we utilize a set of coordinates represented as $x_{pick} = \{x_i | i = 1, ..., N\}$, where N corresponds to the number of points and each $x_i \in \mathbb{R}^3$ corresponds to a location in the observed point cloud. By selecting a pick point

from this set, the agent can determine an appropriate location on the cloth for manipulation.

In contrast, the action vector (a_m) operates in a continuous action space. It describes the relative movement of the gripper from the contact position and is represented in 3D coordinates ($a_m \in \mathbb{R}^3$). This continuous action vector enables precise control over the manipulation process, allowing the gripper to perform fine-grained movements.

In summary, our action space encompasses the selection of a pick point from a discrete set of points and the specification of a continuous 3D action vector for guiding the gripper's movement. By combining these components, our approach empowers the agent to determine where to pick on the cloth and how to maneuver, ensuring effective and controlled interaction during the cloth manipulation process.

5.2.2 Hybrid Actor-Critic Maps for Cloth Manipulation (HAC-Cloth)

Our approach, HAC-Cloth, adopts the TD3-style policy [21] and utilizes a hybrid action space and an actor-critic architecture. The hybrid action space consists of discrete and continuous components, where the discrete component involves selecting a pick point location (x_{pick}) within the object point cloud, and the continuous component represents the 3D action vector (a_m) for manipulation. A schematic overview can be found in Fig. 5.2.

We first estimate correspondences between the observed point cloud and the desired goal point cloud using the CorrNet model (details in section 5.2.3). Both the actor and critic networks take the observed point cloud and correspondences to the goal point cloud as input. The critic network estimates per-point Q-values across the entire point cloud, while the actor network predicts per-point 3D action vectors. To process the input, for both actor and critic networks, we employ a segmentation-style feature extractor that considers the visible points of the point cloud and the correspondences to the goal points.

The reward is based on the negative average flow between the achieved and goal configurations. It incentivizes the agent to achieve the desired deformable object goal. During training, we employ the Bellman update to train the policy, ensuring

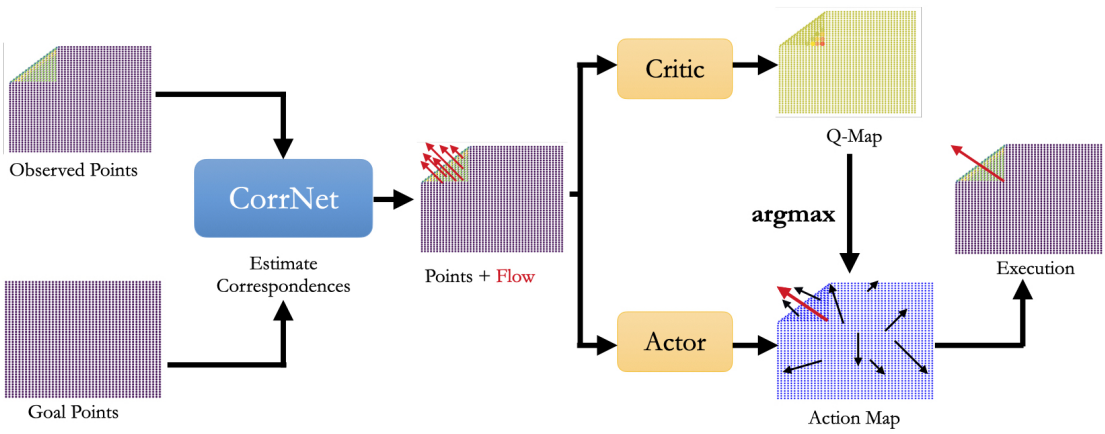


Figure 5.2: Overview of the HAC-Cloth policy. The goal is represented by estimated correspondences between observation and goal point clouds. Both actor and critic networks take the observation point cloud and estimated correspondences as inputs. The critic network predicts a per-point Q-map, which is visualized as a heatmap over the observed point cloud, indicating the Q-values associated with each point. The actor network generates an action map that represents potential actions on the observed point cloud. Based on the highest estimated Q-value, the appropriate pick point and corresponding action vector are selected for execution.

that the agent learns to maximize the expected cumulative reward over time. By iteratively updating the actor and critic networks using the TD3 algorithm with modified update rules for the hybrid policy, HAC-Cloth gradually learns effective manipulation strategies for deformable objects.

During inference, we select the contact location (x_{pick}) by choosing the point with the highest Q-value, indicating its potential for successful manipulation. The corresponding 3D action vector (a_m) is then executed to manipulate the deformable object accordingly.

5.2.3 Point-Based Correspondence Estimation

We train a correspondence estimator **CorrNet** to estimated the point-based correspondences between the object point cloud (c_o) and the goal point cloud (c_g). While in simulated environments we have access to ground truth correspondences ($f = c_g - c_o$), real-world scenarios require us to estimate these correspondences. CorrNet addresses this challenge by predicting the 3D flow between corresponding points based on visible

observed points and goal points.

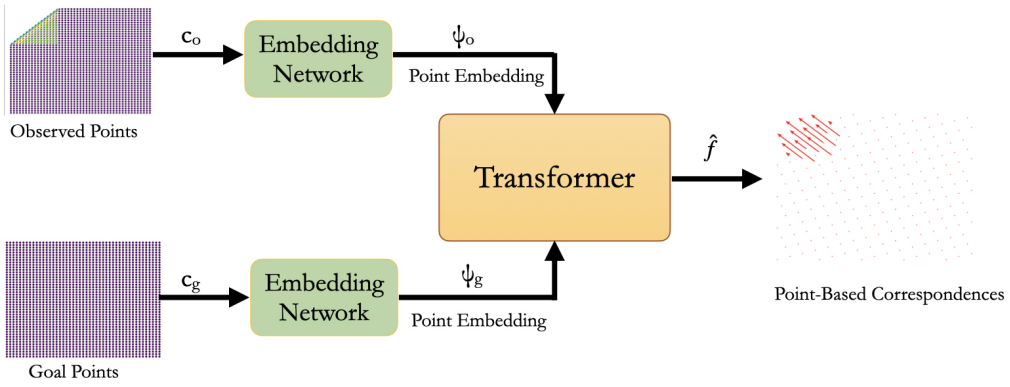


Figure 5.3: Overview of the CorrNet architecture. CorrNet utilizes two embedding networks to learn features from the input observation and goal point clouds, which are then processed by a transformer module to generate per-point-based correspondences.

The architecture of CorrNet is illustrated in Fig. 5.3. To train the correspondence estimator, we leverage the TAX-Pose architecture [11], which incorporates an encoder to process the observed point cloud and goal points and generate dense point-wise embeddings (ψ). Separate encoders are used to handle the zero-centered point clouds.

To integrate information from both point clouds, we employ a cross-object attention module [15]. This module capture the relationships and dependencies between the dense feature sets from the observed point cloud and the goal points, resulting in point-based correspondences.

During the training of CorrNet, we employ the L2 loss between the estimated correspondences and the ground truth correspondences to facilitate learning.

5.2.4 Implementation Details

Dataset

To train both the CorrNet and the RL policy, we collect a dataset using the SoftGym framework [8]. The dataset includes point clouds and visibility masks extracted from SoftGym. The dataset is generated by sampling random actions biased towards grasping the edges and corners of a square towel.

RL Policy

For the policy network, we utilize a segmentation style network based on the PointNet2 architecture [13]. The visible observation and goal point clouds are downsampled to 400 points to reduce computational complexity. We set the success threshold to 1e-10, which means that if the average particle distance falls below this threshold, the policy considers the rollout successful and stops.

CorrNet

The embedding network module in CorrNet is based on the DGCNN architecture [19]. This module performs the task of generating point-wise embeddings for the observed point cloud and the goal points. The point-wise MLP, which predicts residual correspondences, consists of 4 layers.

5.3 Experiments

Our research endeavors to answer several key questions related to autonomous cloth manipulation. By addressing these questions, we aim to evaluate the effectiveness and applicability of HAC-Cloth compared to existing state-of-the-art methods. The main research questions we seek to answer are as follows:

1. How does HAC-Cloth compare with FabricFlowNet (FFN) [20] in achieving single-step and multi-step folding goals, without the need for predefined subgoals? 2. How does HAC-Cloth perform in comparison to the VCD [9] method on different smoothing test sets? 3. How does the input provided to HAC-Cloth impact its performance?

5.3.1 Cloth Folding

We evaluated and compared the performance of HAC-Cloth in cloth folding using an error metric based on the average L2 distance between the cloth points in the achieved point cloud and the desired point cloud. For training HAC-Cloth, we used a task horizon of 2, where the ground truth flow was provided as input. During

evaluation, we extended the horizon to 5, allowing the policy to correct its actions and iteratively approach the desired goal configuration.

Performance on Single-step Goals

We compared HAC-Cloth in simulation using a test set consisting of 40 single-step goals. The test set is the same as the one used in the evaluation of FabricFlowNet (FFN) [20].

Table 5.1 presents the performance comparison between HAC-Cloth, FFAN, and FFN in terms of average particle distance.

Table 5.1: Folding performance comparison on single-step test goals.

Method	**Average Particle Distance** (mm) ↓
FFN [20]	4.26
FFAN	5.54
HAC-Cloth [Ours]	5.87

The experimental results demonstrate that our method, HAC-Cloth, achieves comparable performance to both FFAN and FFN on single-step folding goals, with only a marginal difference in average particle distance. However, it is crucial to emphasize the differing assumptions made by each method. While FFAN utilizes complete point clouds (including occluded regions of the cloth) and requires privileged state information for graph construction, HAC-Cloth overcomes these assumptions by working solely with visible point clouds, as well as eliminating the need for mesh construction.

For the reported results in this section, we utilized ground truth correspondences as input for HAC-Cloth. In a subsequent section 5.3.3, we will delve into the performance of HAC-Cloth when using estimated correspondences, highlighting the significance of accurate correspondences in achieving precise cloth manipulation.

Performance on Multi-step Goals without Subgoals

Our main focus is to develop an autonomous system that can operate with limited human supervision. In line with this objective, we tested HAC-Cloth on folding cloth without the use of explicit subgoals. We compared HAC-Cloth to the performance of

FabricFlowNet (FFN), which relies on predefined subgoals for cloth folding. The test set used for evaluation consisted of 6 multi-step goals, where each goal was 2-4 steps away from the initial cloth configuration.

We conducted two scenarios to compare against FFN. In the first scenario, FFN was provided with subgoals, while in the second scenario, FFN was not provided with any subgoals. HAC-Cloth was run with a horizon of 5 for all the goals and was not provided with intermediate subgoals.

Table 5.2 presents the performance comparison between HAC-Cloth and FFN in terms of average particle distance. The results demonstrate that HAC-Cloth outperforms FFN when no subgoals are provided. Additionally, HAC-Cloth achieves comparable performance to FFN when subgoals are provided to FFN, showcasing the autonomous ability of HAC-Cloth to fold clothes without the need for explicit subgoals.

Table 5.2: Folding performance comparison on multi-Step test goals without subgoals.

Method	**Average Particle Distance** (mm) ↓
FFN (with subgoals)	25.04
FFN (without subgoals)	37.97
HAC-Cloth (without subgoals)	**25.74**

Figure 5.4 visually presents the desired multi-step goal configurations that serve as the target for the cloth folding operation. In our evaluation, we only provide the final goal configuration to HAC-Cloth without explicitly specifying intermediate subgoals. The achieved goal configurations obtained using HAC-Cloth are illustrated in Figure 5.5.

Upon analyzing the results, we observe that HAC-Cloth exhibits the ability to find a trajectory to reach the final goal configuration for most of the examples, even if it fails to reach the optimal subgoals. However, there are two notable cases where HAC-Cloth fails to achieve the desired goal: when the cloth is folded in a double square and a double triangle configuration (last two examples in the figures).

One plausible reason for these failures is the absence of occluded points. During training and evaluation, HAC-Cloth operates solely on the visible parts of the cloth point cloud. Consequently, the policy has no information about how the cloth is folded underneath the visible surface. As a result, it struggles to accurately fold

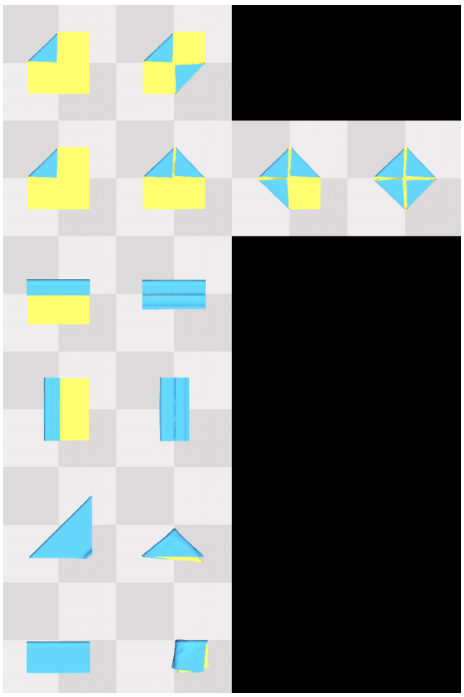


Figure 5.4: FFN [20] multi-step goals test-set. Each row presents an example with corresponding subgoals. The test set has been constructed using two robotic arms in simulation.

the cloth in complex configurations that require knowledge of the underlying folded structure.

The results highlight the autonomous capability of HAC-Cloth to fold clothes without relying on explicit subgoals. This demonstrates the effectiveness of our method in performing cloth folding tasks with limited human intervention.

5.3.2 Cloth Smoothing

HAC-Cloth demonstrates a general capability to achieve cloth smoothing in addition to cloth folding. To train the smoothing policy, we utilized the same dataset as for cloth folding, but with flipped observation-goal pairs. Specifically, we trained the policy for a horizon of 5 steps to allow for more extensive smoothing actions and iterations towards the desired goal configuration.

To evaluate its performance, we employ three distinct test sets: the VCD test set, the Flipped test set, and the Alignment+Smoothing test set.

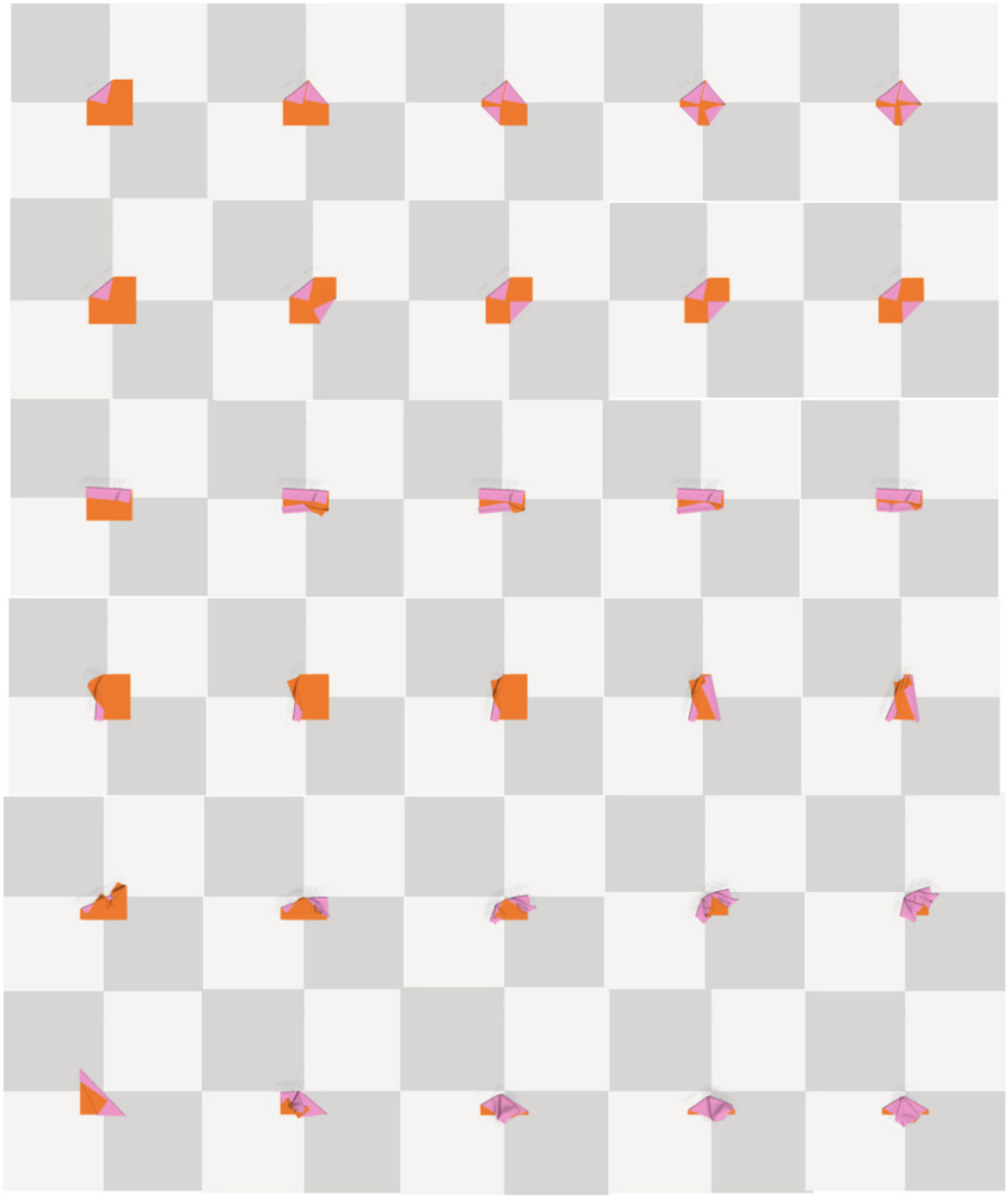


Figure 5.5: HAC-Cloth performance on multi-step goals. Each row displays the intermediate states in the trajectory for a multi-step goal.

The VCD test set comprises highly crumpled states, where the initial cloth

configurations are generated by simulating the process of picking up and dropping the cloth onto the table. This test set is the same as the one provided by VCD [9]. On the other hand, the Flipped test set consists of folds located underneath the cloth. These folds are created by folding and flipping the cloth. Lastly, the Alignment+Smoothing test set contains states with small folds randomly distributed in the workspace. This test set is designed by taking random actions biased towards the edges and corners of the cloth in various directions. The initial states in this test set are typically not aligned, allowing us to evaluate the performance of HAC-Cloth on both alignment and smoothing tasks.

Figure 5.6 presents a selection of examples from each test set.

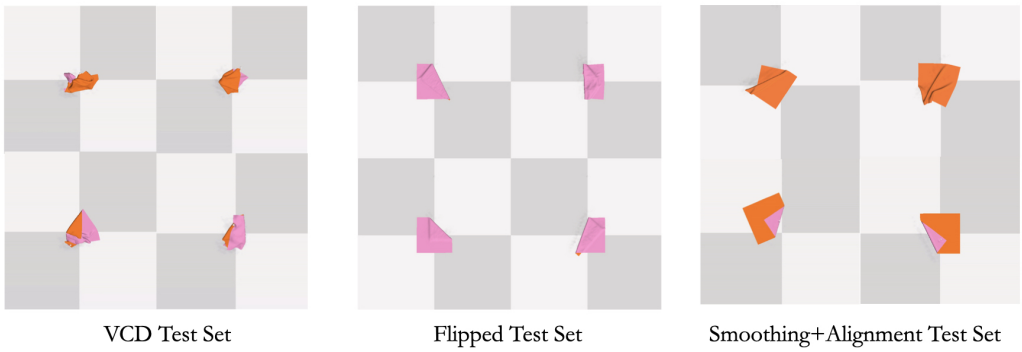


Figure 5.6: Examples from the three smoothing testsets.

To evaluate and compare the performance of HAC-Cloth, we employ Normalized Improvement (NI), which aligns with the evaluation metric used in VCD.

Normalized Improvement (NI) quantifies the increase in the covered area, taking into account the maximum potential improvement. It is calculated by subtracting the initial covered area (s_o) from the achieved covered area (s) and dividing it by the difference between the maximum achievable covered area (s_{max}) and the initial covered area. Mathematically, NI is computed as $NI = \frac{(s-s_o)}{(s_{max}-s_o)}$.

Comparative Analysis with VCD

We conducted a comparative analysis between HAC-Cloth and VCD to evaluate their respective performances in cloth smoothing. HAC-Cloth was trained using a horizon of 5, but we present results for two different horizons: 5 and 10. It is important to

note that VCD is a planning technique that is not specifically trained for a particular horizon. Therefore, to ensure a fair comparison, we evaluate both HAC-Cloth and VCD on the same set of horizons.

Figure 5.7 illustrates the comparison of HAC-Cloth against VCD using the normalized improvement metric across all three test sets for the two different horizons.

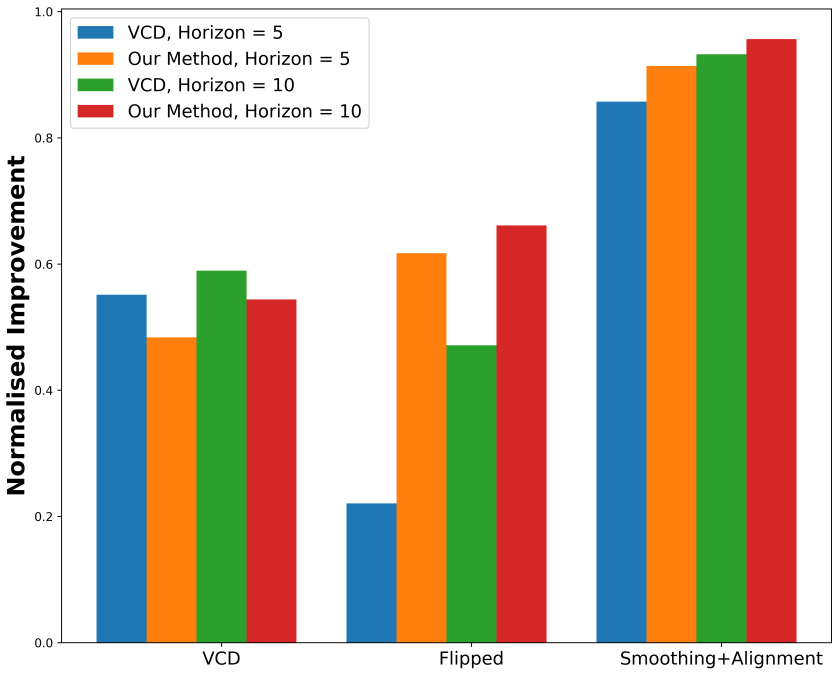


Figure 5.7: Normalized improvement of VCD [9] and HAC-Cloth on three test sets for two different horizons.

When considering the VCD test set, HAC-Cloth and VCD exhibit similar performance, with VCD slightly outperforming HAC-Cloth. On the flipped test set and our test set, HAC-Cloth demonstrates significantly better and comparable performance to VCD, respectively. Additionally, we observe that performance improves as we increase the horizon, indicating that longer horizons allow HAC-Cloth to better address the complexities of cloth smoothing tasks and achieve more substantial improvements in the covered area. Furthermore, HAC-Cloth demonstrates a significant advantage by addressing both cloth smoothing and alignment tasks simultaneously.

We also evaluated the time required for predicting and executing actions for each method. Table 5.3 indicates that HAC-Cloth is significantly faster than VCD,

Table 5.3: Total time taken per test episode by VCD [9] and HAC-Cloth for smoothing.

Method	**Execution Time (seconds) ↓**
VCD	216.93
HAC-Cloth [Ours]	**4.01**

showcasing its efficiency in terms of computational time as well.

5.3.3 Impact of Input to the Policy

To investigate the impact of input variations on the performance of HAC-Cloth, we considered four different scenarios: ground truth correspondences, estimated correspondences, goal points, and no correspondences. Each scenario represents a different way of providing input to the policy for both folding and smoothing tasks.

1. Ground Truth Correspondences: In this scenario, we used ground truth point correspondences obtained from the simulator. These correspondences were computed as the difference between the goal point cloud and the observed point cloud. By utilizing ground truth correspondences, we aimed to assess the performance of HAC-Cloth when provided with accurate correspondence information.

2. Estimated Correspondences: In this scenario, we employed the correspondences predicted by our model from the observed point cloud and the goal points. These correspondences were obtained through our correspondence estimation module. By utilizing estimated correspondences, we aimed to analyze the performance of HAC-Cloth in relation to the accuracy of the correspondence predictions.

3. Goal Points: Instead of using correspondences as a goal specification, we directly employed goal points that represent the desired configuration of the cloth. By utilizing goal points as input, we aimed to assess the effectiveness of HAC-Cloth in achieving the desired cloth configuration without explicit correspondences.

4. No Correspondences: In this scenario, we omitted providing correspondences or goal points as input to the policy. Instead, we assumed a fixed goal of a flat towel placed at the center of the workspace. This setup allowed us to evaluate the policy's ability to implicitly learn the desired task without explicit correspondence information.

For the folding task, we considered three scenarios: ground truth correspondences,

goal points, and estimated correspondences. These policies were trained with a horizon of 1 but evaluated on a horizon of 5.

Table 5.4: Ablation: We show the impact of input variations on HAC-Cloth's folding performance.

Input	**Single-Step Goals**	**Multi-Step Goals (No Subgoals)**
Ground Truth Flow	**0.007**	**0.028**
Goal Points	0.033	0.050
Estimated Flow	0.045	0.058

The results, as presented in Table 5.4, demonstrate the influence of the input variations on the average particle distance error metric. Notably, the policy utilizing ground truth flow outperformed the other policies by a significant margin. For single step goals, the performance of the policy with ground truth flow was approximately five times better compared to the policy trained on goal points alone. This suggests that correspondences not only serve as a goal specification but also provide valuable information for effective cloth manipulation.Furthermore, we observed a drop in performance when utilizing estimated correspondences, indicating that there is room for improvement in the correspondence estimation module of our approach. Enhancing the accuracy of correspondence predictions could potentially lead to improved performance in folding tasks.

For the smoothing task, we investigated the ground truth correspondences, goal points, estimated correspondences, and no correspondences. All these policies were trained and evaluated with a horizon of 5, and the goal for smoothing remained constant, i.e., a flat square cloth in the center of the workspace.

Table 5.5: Ablation: We show the impact of input variations on HAC-Cloth's smoothing performance.

Input	**Normalised Improvement**
Ground Truth Flow	**0.952**
Goal Points	0.902
Estimated Flow	0.776
No Correspondences	0.899

The results, as presented in Table 5.5, demonstrate the performance of each policy on the Smoothing+Alignment test set. Consistent with the findings in the folding

task, the policy trained on ground truth correspondences outperformed all other input variations for smoothing as well. This contradicts the null hypothesis that the goal specification should not significantly impact the performance of a smoothing policy where the goal is constant, highlighting the fact that correspondences provide valuable information beyond just goal specification. When comparing the scenario of no correspondences and goal points, there was no significant difference in performance. This further suggests that correspondences provide additional information that aids in achieving the desired cloth configurations effectively. Similar to the folding task, the worst performance was observed when employing estimated correspondences.

These findings highlight the importance of providing accurate correspondences as input to the policy for achieving optimal folding results. Ground truth correspondences significantly contribute to the policy's ability to manipulate the cloth effectively, while estimated correspondences still require further refinement to match the performance achieved with ground truth correspondences.

5.3.4 Correspondence Estimation Performance

In Section 5.3.3, we observed a drop in performance when using estimated correspondences compared to ground truth correspondences. In this section, we focus on evaluating the performance of the CorrNet model specifically for the task of correspondence estimation. By isolating this module, we gain a better understanding of its strengths, limitations, and areas that require attention.

To assess the performance of CorrNet, we trained several variants of the model, focusing on two key aspects: the use of full point clouds versus visible point clouds, and the estimation of correspondences for single-step goals versus multi-step goals.

In the first experiment, we compared the performance of correspondence estimation using full point clouds versus visible point clouds. For the latter experiment, we used the visible observations while keeping the complete goal points. For the first experiment, we computed the ground truth flow as the difference between the complete goal and complete observation, while for the latter experiment using visible point clouds, we computed correspondences for each visible point in the observation with respect to the complete goal.

To facilitate batch training and accommodate variable-sized point cloud pairs

in the visible point cloud experiment, we performed upsampling or downsampling to a fixed value, N. In our experiments, we explored N values of 2000 and 400. We report the average L2 loss between estimated correspondences and ground truth correspondences on the respective validation sets.

Table 5.6: Correspondence estimation performance on complete and visible point clouds.

Input	**Mean Squared Error ↓**
Complete Point Clouds	9e-6
Visible Point Clouds (N = 2000)	1e-5
Visible Point Clouds (N = 400)	1e-5

Table 5.6 showcases the results of the experiments on complete versus visible point clouds. We observed that there was no drop in performance when using visible point clouds compared to complete point clouds, regardless of whether upsampling or downsampling was applied. This finding suggests that the visibility information in the observations provides sufficient cues for accurate correspondence estimation, even when only a subset of the complete point cloud is considered.

In the second experiment, we evaluated the performance of correspondence estimation for different goal configurations: single-step goals, a mix of 50% single-step goals and 50% crumpled goals, and crumpled-to-crumpled goals. These tasks progressively increased in difficulty. We utilized visible observations and complete goal points from the previous experiment, with N set to 400.

Table 5.7: Correspondence estimation performance on single-step and multi-step goals.

Goals	**Mean Squared Error ↓**
Single Step	1e-5
50% Single-Step and 50% Crumpled	1e-4
Crumpled	3e-4

The results, presented in Table 5.7, indicated that HAC-Cloth achieved near-perfect performance on single-step goals. However, as the difficulty level increased, as reflected in the transition to a mix of single-step and crumpled goals, and then to crumpled-to-crumpled goals, the correspondence error metric increased. The worst performance

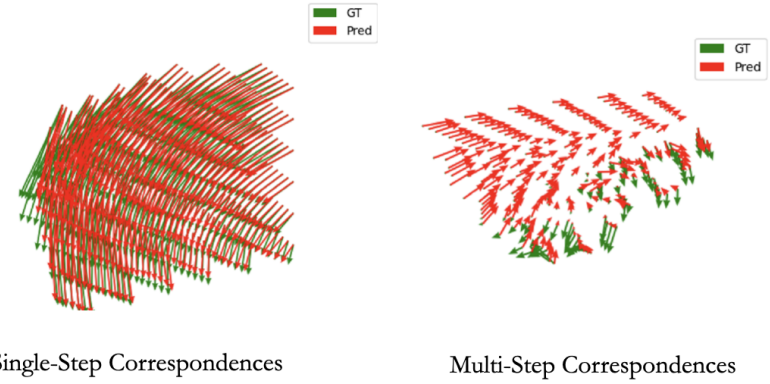


Figure 5.8: Performance of CorrNet on single-step vs multi-step goals.

was observed for the crumpled-to-crumpled goals, highlighting the inherent challenge in estimating correspondences for highly complex and disordered cloth configurations. Fig. 5.8 showcases CorrNet's accurate correspondence estimation for single-step goals, while revealing its difficulty in achieving the same level of performance for multi-step goals.

These experiments provide valuable insights into the performance of our correspondence estimation module. The findings suggest that utilizing visible point clouds, along with appropriate upsampling or downsampling techniques, can yield accurate correspondence estimation without significant degradation in performance. Additionally, the difficulty of the goal configuration plays a crucial role in correspondence estimation, with more complex configurations posing greater challenges.

5.4 Conclusion

In this work, we present a reinforcement learning-based policy for autonomous long-horizon cloth manipulation tasks, leveraging flow-based correspondence estimation. Our approach, HAC-Cloth, enables the policy to operate without the need for explicit subgoals, providing a more autonomous and flexible framework.

5.4.1 Key Insights

Through our investigations, we have gained key insights into the capabilities and performance of HAC-Cloth. Firstly, HAC-Cloth demonstrates the ability to perform long-horizon planning, successfully accomplishing both folding and smoothing tasks with extended horizons. This showcases the effectiveness of HAC-Cloth in addressing complex cloth manipulation scenarios and highlights its potential for real-world applications. Secondly, we find that training the policy using ground truth correspondences leads to near-perfect performance without the need for explicit subgoals. Furthermore, HAC-Cloth surpasses the state-of-the-art performance on the folding task. Additionally, HAC-Cloth excels in alignment with smoothing, showcasing its versatility and effectiveness across different cloth manipulation tasks. Moreover, our analysis reveals that providing correspondences as input to the policy yields better results compared to other input variations. Ground truth correspondences, in particular, exhibit superior performance, highlighting the significance of accurate correspondence predictions. This insight emphasizes the potential for further refinement in the correspondence estimation module of our approach.

5.4.2 Limitations and Challenges

One major limitation stems from the reliance on ground truth correspondences during training. In practical applications, we do not have access to ground truth correspondences, making it necessary to bridge the gap between estimated correspondences and ground truth correspondences.

To deploy HAC-Cloth in real-world settings, we need to enhance the accuracy and robustness of the correspondence estimation module. Currently, the performance drops when estimated correspondences are utilized, indicating the need for improvements in the correspondence prediction process. This presents a significant challenge in developing reliable and accurate correspondence estimation algorithms that can handle complex and varied cloth configurations encountered in real-world environments.

Addressing this limitation requires the exploration of advanced techniques and methodologies for correspondence estimation. This may involve exploring alternative data-driven approaches to improve the accuracy and generalizability of correspondence predictions. Developing a correspondence estimation module that can handle diverse

cloth configurations and adapt to dynamic environments will be crucial for the successful deployment of HAC-Cloth in real-world robotic systems.

Chapter 6

Conclusion

In this thesis, we have presented a holistic approach to cloth manipulation that addresses the challenges associated with cloth behavior. Our methodology focuses on efficiency, robustness, and autonomy, aiming to enable automated cloth manipulation tasks.

Through our contributions, we have advanced the field of cloth manipulation and achieved significant improvements over existing techniques. We have shown the benefit of employing point-based in two key areas: aligning cloth and performing long-horizon cloth manipulation tasks without human supervision.

FabricFlowAlignNet, our first method, utilized point-based correspondences to facilitate the alignment of cloth and enhance the policy's robustness. By leveraging the geometric relationships and deformations captured by these correspondences, our approach achieved superior performance in aligning cloth configurations. The use of point-based correspondences offered a valuable means of accurately manipulating cloth surfaces, overcoming challenges such as rotations and unalignments. Our results demonstrated the effectiveness of this approach and achieved better accuracy over existing methods.

Building upon the advantages of point-based correspondences, our second method, HAC-Cloth, focused on developing an RL policy for long-horizon cloth manipulation tasks. By incorporating point-based correspondences into the RL framework, we successfully performed both smoothing and folding operations on multi-step goals without the need for explicit subgoals. The autonomy of HAC-Cloth allowed for

intelligent decision-making and efficient planning, reducing human intervention and enabling the system to handle diverse cloth manipulation scenarios. HAC-Cloth outperformed state-of-the-art methods in terms of performance and speed, showcasing its effectiveness and efficiency in achieving accurate and reliable cloth manipulation.

The utilization of point-based correspondences can be further explored in various cloth manipulation tasks, such as folding complex shapes or executing intricate draping maneuvers. By leveraging the geometric information captured by these correspondences, we can enhance the accuracy and efficiency of cloth manipulation in these challenging scenarios.

In conclusion, our work has provided compelling evidence regarding the potential and significance of point-based correspondences in the field of cloth manipulation. Our contributions not only advance the current understanding but also pave the way for future research and development in automated cloth manipulation systems.

Bibliography

[1] PJ Besl and ND Mckay. A method for registration of 3-d shapes, ieee trans. *P flattern Anal. and M ac h ine I ntell*, 1(4):23, 1992. 2.4

[2] Alper Canberk, Cheng Chi, Huy Ha, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. *arXiv preprint arXiv:2210.09347*, 2022. 3.2

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4.2.3

[4] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018. 2.5

[5] Aditya Ganapathi, Priya Sundaresan, Brijen Thananjeyan, Ashwin Balakrishna, Daniel Seita, Jennifer Grannen, Minho Hwang, Ryan Hoque, Joseph E Gonzalez, Nawid Jamali, et al. Learning dense visual correspondences in simulation to smooth and fold real fabrics. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11515–11522. IEEE, 2021. 3.1, 4.1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4.2.5

[7] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 2.5

[8] 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. 4.2.5, 5.2.4

[9] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022. (document), 3.2, 4.2.1, 4.3.1, 5.1, 5.3, 5.3.2, 5.7, 5.3

[10] Kai Mo, Chongkun Xia, Xueqian Wang, Yuhong Deng, Xuehai Gao, and Bin Liang. Foldsformer: Learning sequential multi-step cloth manipulation with space-time attention. *IEEE Robotics and Automation Letters*, 8(2):760–767, 2022. 3.1

[11] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. In *Conference on Robot Learning*, pages 1783–1792. PMLR, 2023. 2.4, 4.2.1, 5.2.3

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[13] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 5.2.4

[14] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009. 2.4

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2.3, 4.2.1, 5.2.3

[16] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 4, 4.2.5

[17] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 4.2.1

[18] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 2.4

[19] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 3, 5.2.4

[20] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022. (document), 2.4, 3.1, 4.1, 4.2.1, 4.2.4, 4.2.5, 4.3, 4.3.1, 4.1, 5.1, 5.3, 5.3.1, 5.1, 5.4

[21] Wenxuan Zhou, Bowen Jiang, Fan Yang, Chris Paxton, and David Held. Learning hybrid actor-critic maps for 6d non-prehensile manipulation. *arXiv preprint*

arXiv:2305.03942, 2023. 2.5, 5.2.2