machines

MDPI

Article

Fabric Flattening with Dual-Arm Manipulator via Hybrid Imitation and Reinforcement Learning

Youchun Ma [1], Fuyuki Tokuda [2,3], Akira Seino [2,3], Akinari Kobayashi [2,3], Mitsuhiro Hayashibe [1,*] and Kazuhiro Kosuge [2,3]

[1] Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan; ma.youchun.q7@dc.tohoku.ac.jp
[2] Center for Transformative Garment Production, Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China; fuyuki.tokuda@transgp.hk (F.T.); akira.seino@transgp.hk (A.S.); akinari.kobayashi@transgp.hk (A.K.); kosuge@hku.hk (K.K.)
[3] JC STEM Laboratory of Robotics for Soft Materials, Department of Electrical and Electronic Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong, China
* Correspondence: mitsuhiro.hayashibe.e6@tohoku.ac.jp

Abstract

Fabric flattening is a critical pre-processing step for automated garment manufacturing. Most existing approaches employ single-arm robotic systems that act at a single contact point. Due to the nonlinear and deformable dynamics of fabric, such systems often require multiple actions to achieve a fully flattened state. This study introduces a dual-arm fabric-flattening method based on a cascaded Proposal–Action network with a hybrid training framework. The PA network is first trained through imitation learning from human demonstrations and is subsequently refined through reinforcement learning with real-world flattening feedback. Experimental results demonstrate that the hybrid training framework substantially improves the overall flattening success rate compared with a policy trained only on human demonstrations. The success rate for a single flattening operation increases from 74% to 94%, while the overall success rate improves from 82% to 100% after two rounds of training. Furthermore, the learned policy, trained exclusively on baseline fabric, generalizes effectively to fabrics with varying thicknesses and stiffnesses. The approach reduces the number of required flattening actions while maintaining a high success rate, thereby enhancing both efficiency and practicality in automated garment manufacturing.

Keywords: fabric flattening; imitation learning; reinforcement learning; dual-arm manipulator; robotic system

check for updates

Academic Editors: Yanhong Peng and Fernando Gomez-Bravo

Received: 7 September 2025
Revised: 30 September 2025
Accepted: 4 October 2025
Published: 6 October 2025

Citation: Ma, Y.; Tokuda, F.; Seino, A.; Kobayashi, A.; Hayashibe, M.; Kosuge, K. Fabric Flattening with Dual-Arm Manipulator via Hybrid Imitation and Reinforcement Learning. *Machines* **2025**, *13*, 923. https://doi.org/10.3390/machines13100923

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In garment manufacturing processes such as sewing, folding, and ironing, it is critical to ensure that the fabrics are sufficiently flattened before each operation [1,2]. A properly flattened surface improves process accuracy, consistency, and efficiency. Despite the importance of this step, fabric flattening is still predominantly performed manually. This is due to the inherent challenges of automating the task using robotic systems, including the fabric's high degrees of freedom, nonlinear and deformable dynamics, and the absence of consistent geometric features [3].

Previous research has mainly focused on single-arm robotic systems for fabric manipulation and flattening. These systems typically rely on learning-based or vision-guided

strategies and have demonstrated success in simplified or controlled scenarios [4,5]. However, single-arm setups are inherently limited, as they can only manipulate one point at a time, reducing control flexibility and often leading to inefficient operation. Existing studies report that such systems generally require four to nine actions to fully flatten a piece of fabric [6,7]. More recent work has investigated dual-arm systems that can manipulate fabric from both sides simultaneously. While these approaches provide better control of surface tension, they still predominantly rely on wrinkle-based actions; thus, multiple iterations are often needed to achieve complete flattening [8,9].

To address these challenges, we propose a dual-arm fabric flattening method based on a cascaded neural network and a hybrid training framework. The proposed network consists of two modules: a Proposal Network that identifies promising manipulation regions, and an Action Network that predicts optimal operation points. The hybrid training framework first applies imitation learning to build an initial policy from human demonstrations for the Proposal Network. Then it refines this policy through joint reinforcement learning based on real-world robotic feedback. This integration enables the system to adaptively optimize its actions with the explicit goal of reducing the number of required operations while maintaining robust performance across diverse fabric conditions.

Our approach provides two main advantages. First, by combining region-level proposals with point-level decision-making, it improves the quality of initial manipulation choices. Second, by incorporating real-world feedback into the learning loop, the policy is directly optimized to reduce the number of required flattening actions. As a result, the robot can perform more effective operations with fewer steps, which is essential for enhancing the efficiency and practicality of fabric flattening.

2. Related Work

In recent years, learning-based methods have become the mainstream approach in deformable object manipulation due to their adaptability and improved performance. Many studies use convolutional neural networks for tasks such as wrinkle detection [10–13] and corner detection [14,15] to identify manipulation points.

Among learning-based techniques, imitation learning (IL) has been widely applied to fabric flattening. These methods train models to imitate expert behavior by predicting grasp points or manipulation strategies from labeled demonstrations [16,17]. IL has been successfully used in various tasks involving cloth flattening [18], folding [19], and bed-making [20]. However, the effectiveness of IL is constrained by the quality and quantity of human demonstrations, which may not reflect optimal strategies. In addition, IL models often lack adaptability to variations in fabric properties, physical dynamics, or sensor noise.

On the other hand, reinforcement learning (RL) provides a data-driven alternative, allowing robots to learn policies through interaction with the environment [21–29]. Recent efforts have shown its potential for cloth flattening [30], cloth unfolding [31,32], and assistive dressing [33]. Nonetheless, RL often suffers from high sample complexity, making it costly to obtain sufficient interaction data in real-world settings.

To mitigate the limitations of both IL and RL, hybrid approaches have been proposed to combine the strengths of the two. For instance, Vecerik et al. introduced Deep Deterministic Policy Gradient from Demonstration [34], which initializes the replay buffer with expert demonstrations to guide exploration in sparse-reward environments. Generative adversarial imitation learning [35] aligns the learned policy with expert behavior using a discriminator. This framework was further extended by Tsurumine and Matsubara [36] through a goal-conditioned variant specifically designed for cloth manipulation.

Additional efforts have focused on improving stability and data efficiency. For example, batch-constrained Q-learning [37] constrains policy updates to remain close to

the distribution of expert data, while deformable manipulation from demonstrations [38] incorporates advantage-weighted exploration to enhance expert trajectories. Although these methods benefit from improved learning efficiency and reduced dependence on large datasets, they typically treat expert demonstrations as static supervision signals that remain fixed throughout training.

In contrast to previous studies that consider human-labeled data as immutable guidance, our method treats human-provided manipulation points as initial priors rather than fixed targets. This allows the policy to be initialized from expert behavior while remaining flexible to improve upon it through feedback from real-world execution. By continuously refining these priors during training, the proposed framework bridges the gap between human intuition and physical performance, enabling more effective and efficient fabric flattening.

3. Materials and Methods

3.1. Robot System Setup

We first present the fabric flattening system, as shown in Figure 1. The details of the robot system are as follows.

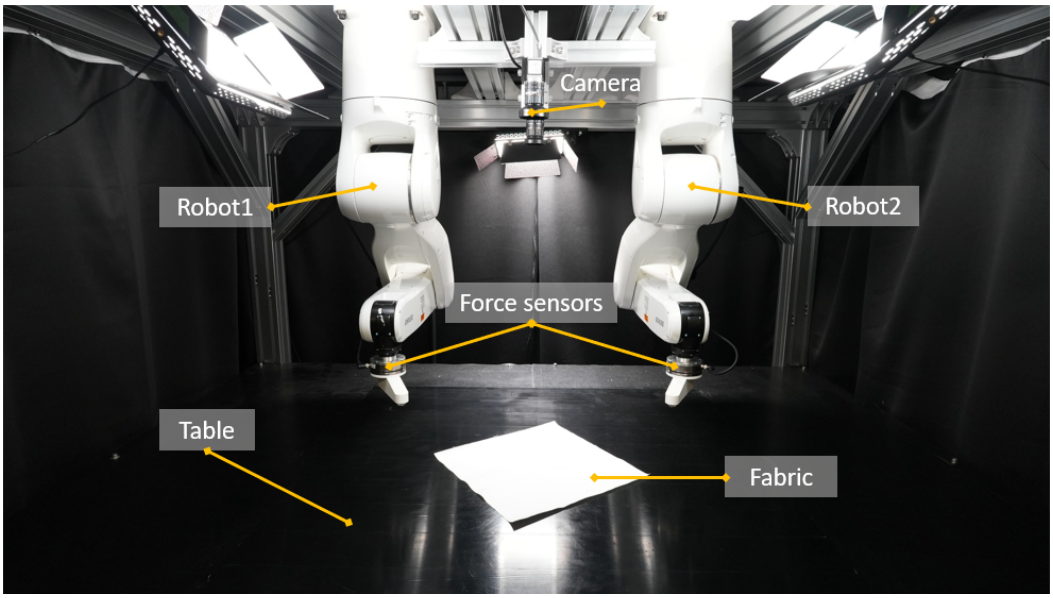


Figure 1. The experimental flattening system consists of a dual-arm manipulator system, force/torque sensors attached to each arm, two end-effectors attached to each manipulator, a camera, and a flat table.

3.1.1. Manipulator

Two 6-DoF robotic manipulators (VS-087, Denso Corporation, Kariya, Aichi, Japan) are suspended from the robot frame. Each manipulator is equipped with a wrist-mounted force/torque (F/T) sensor and an end-effector. The manipulators are controlled by a PC via EtherCAT communication.

3.1.2. F/T Sensor

A 6-axis F/T sensor (Axia80-M8, ATI Industrial Automation, Apex, NC, USA) is mounted on each manipulator to measure both the contact force applied to the end-effector and the tension of the fabric. The F/T sensors are daisy-chained into the EtherCAT network through the controller.

3.1.3. End-Effector

The end-effector for the flattening operation is 3D-printed using PC-ABS (Stratasys, Eden Prairie, MN, USA) as the main body, as shown in Figure 2. Elastic 50A Resin V1 (Formlabs, Somerville, MA, USA) is used as a soft contact pad to secure the surface-to-surface contact between the pad and the table.

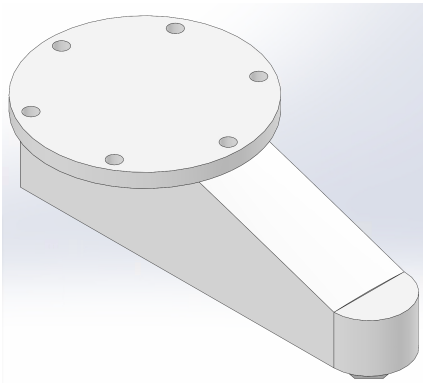
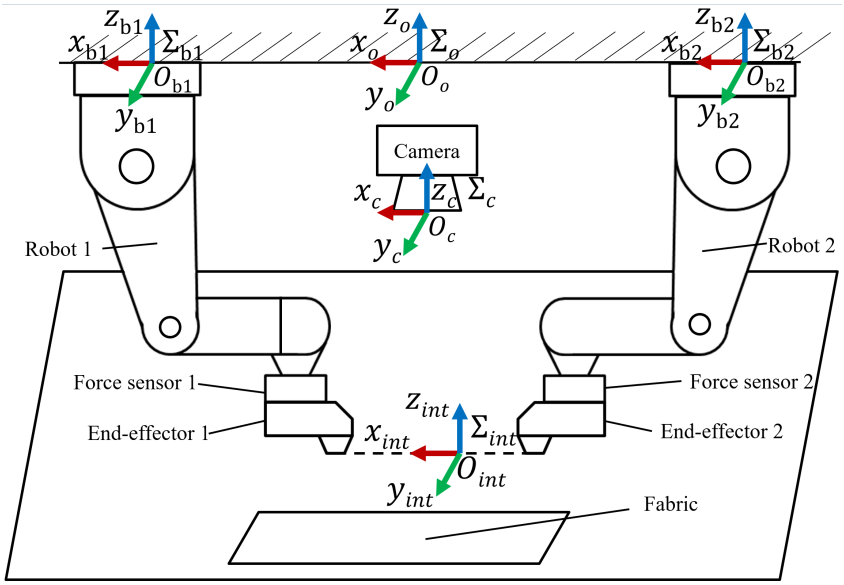


Figure 2. Three-dimensional–CAD model of the end-effector for fabric flattening operation, which is attached to the manipulator.

3.1.4. Vision System

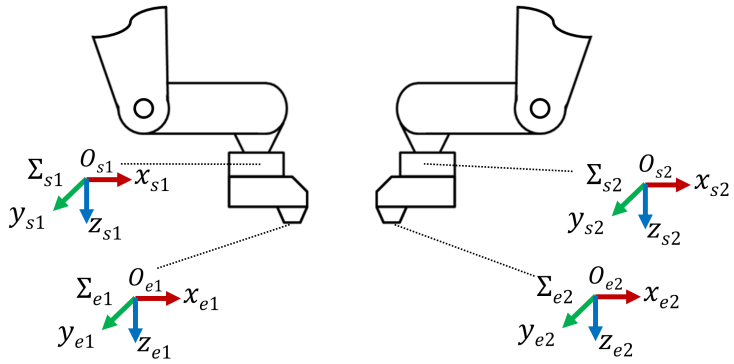
A Basler camera (Basler AG, Ahrensburg, Germany) is mounted above the workspace to capture images of the fabric.

Figure 3a,b show the coordinate systems used for control of the system. Σ_{b1} and Σ_{b2} are defined as the base coordinate frames $O_{b1} - x_{b1}y_{b1}z_{b1}$ and $O_{b2} - x_{b2}y_{b2}z_{b2}$, respectively, which are attached to the base link of each manipulator named Robot 1 and Robot 2. We define Σ_o as a world coordinate frame $O_o - x_oy_oz_o$, which is fixed to the environment at the midpoint between Σ_{b1} and Σ_{b2}. We define Σ_c as a camera coordinate frame $O_c - x_cy_cz_c$, which is attached to the camera. The homogeneous transformation matrices of Σ_{b1}, Σ_{b2}, and Σ_c with respect to Σ_o are calculated from camera–robot calibrations for Robot 1 and Robot 2.



(**a**) Coordinate systems of the dual-arm manipulator and the camera

Figure 3. *Cont.*



(**b**) Coordinate systems related to end effectors

Figure 3. Coordinate systems for fabric flattening experiments.

Σ_{e1} and Σ_{e2} are defined as end-effector coordinate frames $O_{e1} - x_{e1}y_{e1}z_{e1}$ and $O_{e2} - x_{e2}y_{e2}z_{e2}$, which are attached to the tip of each end-effectors of Robot 1 and Robot 2. For each manipulator of the dual-arm robotic system, the position and orientation of its manipulation point for flattening the fabric is defined as the end-effector coordinate system with respect to the world coordinate system. Σ_{int} is defined as the internal force coordinate frame $O_{int} - x_{int}y_{int}z_{int}$, which is attached to the midpoint between Σ_{e1} and Σ_{e2}.

3.2. Imitation Learning Based on Human Demonstration

The whole proposed method for selecting the manipulation points for the fabric flattening is shown in Figure 4.

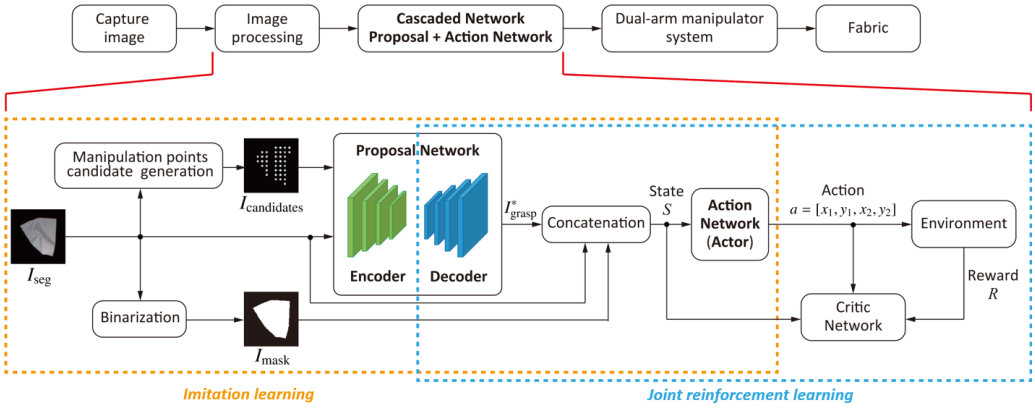


Figure 4. Pipeline for fabric flattening. Imitation learning from human demonstrations is illustrated in the orange dotted box, while joint reinforcement learning with real-world feedback is shown in the blue dotted box. The cascaded network is first trained with human demonstrations and then fine-tuned through joint reinforcement learning to optimize the policy for fabric flattening.

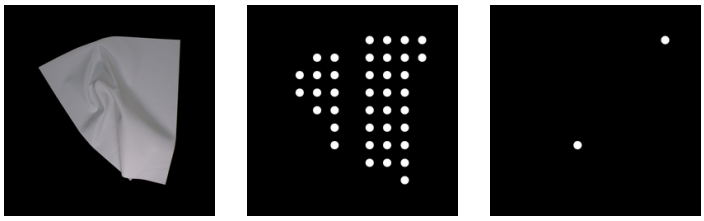
The imitation network is composed of two modules: a Proposal Network (PN) that identifies promising manipulation regions and an Action Network (AN) that predicts the optimal operation points. We adopt a cascaded design rather than a single large model, since a single model struggles to map complex fabric appearances to human-demonstrated points and tends to make reinforcement learning less stable and efficient. The PN reduces the search space by proposing feasible manipulation regions from the input image and candidate map, while the AN selects the final two operation-point coordinates based on these structured cues.

3.2.1. Human Demonstration

We collected a dataset for imitation learning of fabric flattening from human demonstrations. These demonstrations provide the model with ground-truth references to learn

the correspondence between fabric states and operation points. In each demonstration, the human demonstrator selects two manipulation points from the provided options based on their experience with fabric flattening and then flattens the fabric by moving the points in opposite directions along the line connecting them. The candidates for the manipulation points are generated by uniformly discretizing the segmented fabric image.

The collected dataset is shown in Figure 5, where (a) illustrates the segmented fabric area (I_{seg}) with background information removed, and (b) presents the manipulation point candidates ($I_{\text{candidates}}$), which are generated by uniformly discretizing the segmented fabric image I_{seg} and eliminating points along a vertical line passing through the centroid of the segmented fabric pixels (see candidate generation in Figure 4). Figure 5c shows I_{grasp}, which contains two manipulation points for fabric flattening selected by a human demonstrator from the manipulation point candidates $I_{\text{candidates}}$.



(**a**) (**b**) (**c**)

Figure 5. Dataset sample for training the PN. (**a**) segmented fabric image I_{seg}; (**b**) candidates of manipulation points $I_{\text{candidates}}$; (**c**) human-selected manipulation points I_{grasp}.

For each fabric flattening trial, a 350 mm × 350 mm square white fabric piece is randomly placed on the experimental platform in a deformed state. Fabric images with a resolution of 256 × 256 pixels, captured by the camera, are used for training the imitation network. In total, 2431 training samples (I_{seg}, $I_{\text{candidates}}$, I_{grasp}) were collected and used to train the PN. The image coordinates of the selected manipulation points, $((x_1, y_1), (x_2, y_2))$, were directly obtained from I_{grasp} and used to train the AN.

3.2.2. PN Architecture and Training

The PN is based on a dual-branch encoder–decoder architecture, adapted from the standard U-Net [39], which effectively integrates global structural information with localized uniform priors. The network takes two input modalities: the segmented fabric image I_{seg} and the manipulation point candidates $I_{\text{candidates}}$. The schematic diagram of PN is shown in Figure 6.

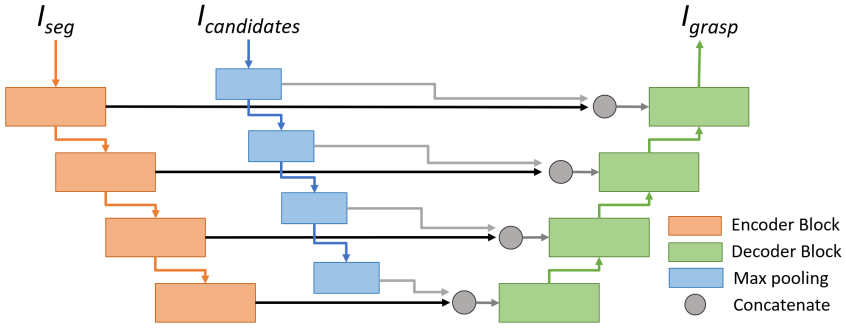


Figure 6. The structure of PN. I_{seg} and $I_{\text{candidates}}$ participate in the network training through different input methods.

The backbone processes the segmented fabric image I_{seg} through a conventional encoder–decoder structure with four hierarchical encoding blocks. Each block contains two 3 × 3 convolutional layers with batch normalization and ReLU activation, followed by a

max pooling layer to downsample the spatial resolution and enhance feature abstraction. The decoder mirrors this design, employing transposed convolutions and skip connections to progressively reconstruct fine-grained spatial details from the abstracted features.

Note that the manipulation point candidates $I_{\text{candidates}}$ are not processed through convolutional layers within the network. Instead, they are successively downsampled using max pooling to match the spatial resolutions of each decoder stage. At every decoder stage, the downsampled candidate features are concatenated with the corresponding decoder features. This fusion explicitly incorporates localized candidate priors, guiding the network's predictions toward regions emphasized by the downsampled candidate features.

To address the significant class imbalance between foreground pixels (manipulation points, pixel value 1) and background pixels (pixel value 0), we adopt a composite loss function that combines Binary Cross-Entropy (BCE) [40] and Dice loss [41]. The BCE term captures pixel-wise differences, while the Dice loss is particularly effective in mitigating the imbalance caused by sparse foreground regions and dense background areas. The overall loss is defined as follows:

$$L_{\text{loss}} = \alpha \cdot \text{Dice Loss} + \beta \cdot \text{BCE Loss} \tag{1}$$

where α and β are weighting factors that balance the two components.

3.2.3. AN Architecture and Training

The AN is introduced to select the manipulation points and output the image coordinates of the two manipulation points. The network takes three inputs: the segmented fabric image I_{seg} and the probability map I^*_{grasp} generated by the PN and I_{mask} obtained by binarizing I_{seg}.

I_{seg} and I^*_{grasp} are concatenated along the channel dimension and passed through four convolutional layers, followed by max pooling as the main encoding branch. As the downsampling using convolutional layers can blur fabric boundaries, I_{mask} is processed through four levels of Haar wavelet downsampling (HWD) [42] to capture multi-scale structural cues, as shown in Figure 7. At each encoding stage, features from the HWD are integrated into the main branch via cross-attention blocks [43].

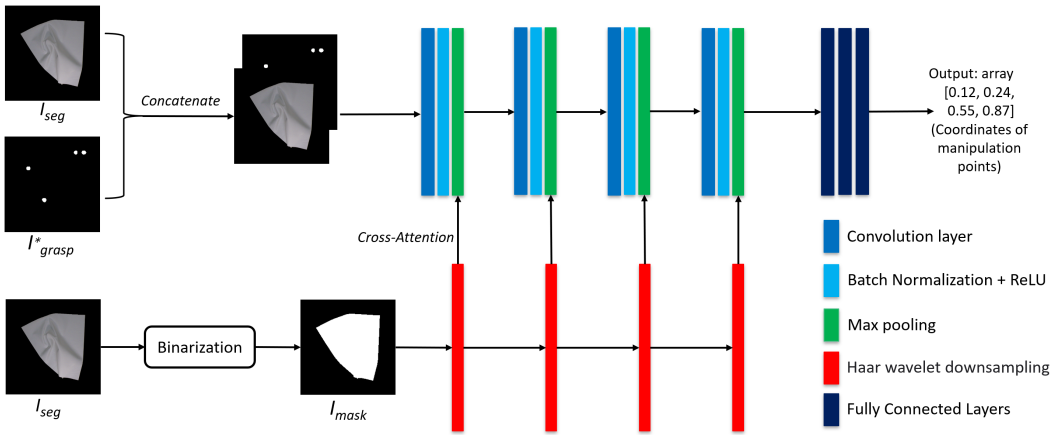


Figure 7. The structure of AN. AN takes two sets of inputs: the output I^*_{grasp} from the PN, and the same original inputs that are also fed to the PN.

During demonstrations, the human demonstration tends to select manipulation points near the fabric boundary, as these locations are more effective for flattening the fabric. However, a standard mean squared error (MSE) loss considers only the Euclidean distance between predicted and ground truth points, without enforcing spatial consistency with

the valid fabric region. As a result, predictions may occasionally fall outside the valid fabric region.

To address this issue, we propose a Projection Loss $\mathcal{L}_{\text{proj}}$, which encourages the predicted points to remain within the fabric region. Specifically, we constrain the predicted manipulation points to lie on the line segment between the two manipulation points of the ground truth. Let $\mathbf{g}_l, \mathbf{g}_r \in \mathbb{R}^2$ denote the ground truth coordinates of the left and right manipulation points, respectively. We first define the segment direction $\mathbf{v}$ from $\mathbf{g}_l$ to $\mathbf{g}_r$ as

$$\mathbf{v} = \frac{\mathbf{g}_r - \mathbf{g}_l}{\|\mathbf{g}_r - \mathbf{g}_l\|} \tag{2}$$

Let $\mathbf{p}_l, \mathbf{p}_r \in \mathbb{R}^2$ denote the manipulation coordinates of the left and right manipulation points. The distances (t_l, t_r) from $\mathbf{g}_l$ to the projected points of $\mathbf{p}_l$ and $\mathbf{p}_r$ on the line connecting $\mathbf{g}_l$ and $\mathbf{g}_r$ can be computed by:

$$t_l = (\mathbf{p}_l - \mathbf{g}_l) \cdot \mathbf{v}, \qquad t_r = (\mathbf{p}_r - \mathbf{g}_l) \cdot \mathbf{v} \tag{3}$$

If $0 \leq t_l \leq 1$, then $\mathbf{p}_l$ is projected onto a point inside the line segment $\mathbf{g}_l\mathbf{g}_r$. Conversely, if $t_l < 0$ or $t_l > 1$, $\mathbf{p}_l$ is projected outside of the segment. Similarly, if $0 \leq t_r \leq 1$, then $\mathbf{p}_r$ is projected onto a point inside the line segment $\mathbf{g}_l\mathbf{g}_r$. Conversely, if $t_r < 0$ or $t_r > 1$, $\mathbf{p}_r$ is projected outside of the segment.

The projection loss function $\mathcal{L}_{\text{proj}}$ can be represented as follows:

$$\mathcal{L}_{\text{proj}} = \mathcal{L}_{\text{proj_order}} + \mathcal{L}_{\text{proj_dist}}, \tag{4}$$

where the projection order loss function $\mathcal{L}_{\text{proj_order}}$ is introduced to penalize any swapping of left and right predictions, and is defined as follows:

$$\mathcal{L}_{\text{proj_order}} = \lambda_{\text{order}}\big(|t_l - 0| + |t_r - 1|\big) \tag{5}$$

Note that λ_{order} is a hyperparameter that controls how strongly we penalize any swapping of left and right predictions.

The projection loss distance $\mathcal{L}_{\text{proj_dist}}$ is defined as follows:

$$\mathcal{L}_{\text{proj_dist}} = \frac{1}{2}\big(\ell_{\text{pd},l} + \ell_{\text{pd},r}\big) \tag{6}$$

where $\ell_{\text{pd},l}$ and $\ell_{\text{pd},r}$ penalize $\mathbf{p}_l$ and $\mathbf{p}_r$ located outside of the line segment $\mathbf{g}_l\mathbf{g}_r$:

$$\ell_{\text{pd},l} = \begin{cases} \lambda_{\text{inside}}|t_l - 0|, & 0 \leq t_l \leq 1, \\ \lambda_{\text{outside}}(-t_l), & t_l < 0, \\ \lambda_{\text{outside}}(t_l - 1), & t_l > 1, \end{cases} \tag{7}$$

$$\ell_{\text{pd},r} = \begin{cases} \lambda_{\text{inside}}|t_r - 1|, & 0 \leq t_r \leq 1, \\ \lambda_{\text{outside}}(-t_r), & t_r < 0, \\ \lambda_{\text{outside}}(t_r - 1), & t_r > 1, \end{cases} \tag{8}$$

Here, λ_{inside} is a relatively small weight that encourages t_l (or t_r) to stay close to 0 when $\mathbf{p}_l$ (or $\mathbf{p}_r$) is within the segment, and λ_{outside} is a larger weight that strongly discourages t_l (or t_r) from moving outside the interval $[0, 1]$.

3.3. Joint Reinforcement Learning

3.3.1. Dataset for Offline Reinforcement Learning

The dataset for joint reinforcement learning contains state S, action a, and reward R. We define the state $S \in \mathbb{R}^{H \times W \times 3}$ consisting of I_{seg}, I^*_{grasp} and I_{mask}, where H and W denote the image height and width, respectively. The fabric flattening policy is a four-dimensional vector $a = [x_1, y_1, x_2, y_2] \in \mathbb{R}^4$, representing the predicted pixel coordinates of the manipulation points in the image plane. (x_1, y_1) corresponds to the left manipulation point and (x_2, y_2) corresponds to the right manipulation point. Given a state S, the AN outputs $a = AN(S)$ as policy.

The output policy $a = [x_1, y_1, x_2, y_2]$, defined in the image coordinate system, is transformed into the 3D space expressed in the world coordinate system Σ_o through camera-robot calibration, assuming that the coordinates $((x_1, y_1), (x_2, y_2))$ are on the working table. Based on the transformation, the end-effectors, defined by the coordinate systems Σ_{e1} and Σ_{e2} attached to Robot 1 and Robot 2, respectively, are commanded to move to their corresponding manipulation points.

Then, the manipulation points are moved in opposite directions using impedance control along the line connecting the manipulation points. This motion is executed using dual-arm impedance control defined in the internal force coordinate frame Σ_{int}. The operation is continued until the internal force reaches a desired target value, determined experimentally to ensure effective fabric flattening while avoiding excessive deformation of the fabric. In our flattening experiments, the internal force threshold was set to 2.5 N, which was determined based on empirical measurements during real-world operations. After a single flattening operation, a fabric image is captured by the camera and used to compute the reward based on the changes in the fabric. A representative cycle is illustrated in Figure 8.



(**a**) Initial pose (**b**) Press at points (**c**) Stretch to force target (**d**) Return to initial pose

Figure 8. A simplified dual-arm flattening cycle.

The reward used for policy update is defined by the increase in fabric area before and after the flattening operation. The fabric area was consistently computed from binary masks obtained through a standardized segmentation pipeline, where images with a pure black background were thresholded and refined with basic morphological operations. The area of the largest connected component was then measured and used as the coverage metric. The reward R is defined as:

$$R = \frac{A_{\text{after}} - A_{\text{before}}}{A_{\text{fully}} - A_{\text{before}}}, \tag{9}$$

where A_{before} denotes the area of the fabric before the flattening operation, A_{after} represents the fabric area after the flattening operation, and A_{fully} is the area of the fully flattened fabric.

During data collection, we construct two types of datasets: an exploitation dataset containing 4300 (S, a, R) tuples, obtained by executing the current Action Network (AN) policy, and an exploration dataset containing 3000 $(S, \tilde{a}, R)$ tuples, obtained by adding Gaussian noise to the AN outputs. Specifically, Gaussian noise sampled from $\mathcal{N}(0, 6)$ was added to the (x, y) coordinates of the predicted operation points to generate the

noisy actions $\tilde{a}$.These noisy actions are then used to enhance the robustness of the Critic pre-training stage.

3.3.2. Learning Strategies

After imitation learning, the AN and the decoder of the PN are fine-tuned using fabric flattening results obtained from real-world experiments. We employ reinforcement learning with an Actor–Critic architecture [44], as illustrated in the blue box in Figure 4. The AN acts as the policy network and is paired with a Critic Network to evaluate the actions generated by the AN. During this process, the decoder of the PN is also fine-tuned to further improve the fabric flattening performance.

In order to evaluate the effectiveness of each action, we introduce a Critic Network to estimate the expected reward by the policy from AN, which is formalised as the Q-value, $Q(S, a)$. The Critic Network uses a similar network architecture to the AN. The input to the Critic Network consists of the current state $S = [I_{\text{seg}}, I^*_{\text{grasp}}, I_{\text{mask}}]$ and the policy a. These two inputs are processed in parallel: the state S passes through a series of convolutional and attention layers, while the policy a is embedded through fully connected layers. The resulting features are then merged and further processed through additional fully connected layers to produce a scalar output $Q(S, a)$, which represents the estimated performance of the policy a for state S.

To improve the Critic Network's generalization and ensure stability during reinforcement learning, we first conduct a pre-training phase. While freezing the PN and AN, we pre-train the Critic Network on these noisy samples. After pre-training the Critic Network, the system proceeds to joint reinforcement learning. In this phase, the parameters of the AN and decoder part of PN are unfrozen to allow for fine-tuning. The AN is trained to output the policy for maximizing the Q-values based on the Q-values predicted by the Critic.

4. Results

4.1. Proposal Network

A total of 2431 training samples were collected and used to train the proposal network. The α and β parameters of the loss function shown in Equation (1) are empirically set to 0.3 and 0.7, respectively. The PN is trained using the Adam optimizer with an initial learning rate of 5×10^{-4} for 200 epochs. Figure 9 showcases the results produced by the trained network.

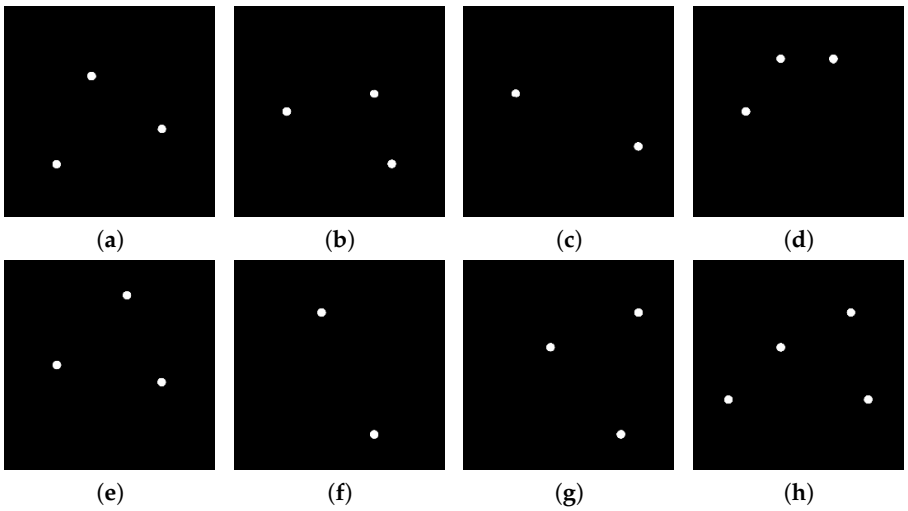


Figure 9. The PN output examples (**a**–**h**) show eight representative outputs. In some cases, such as (**c**,**f**), the network predicts only one operation point per side. In other cases, multiple candidate points are generated on at least one side.

4.2. Pre-Training for Action Network

The dataset used for pre-training the AN is derived from the same samples used to train the PN, without any additional data collection. This pre-training stage aims to initialize a policy that can directly predict the coordinates of two manipulation points based on the PN output and related inputs. To stabilize the initial training process and ensure reliable convergence, the model is first trained for 30 epochs using only MSE loss. This allows the network to learn the coarse positional distribution of the points. Beginning with the 31st epoch, the projection loss $\mathcal{L}_{\text{proj}}$ (4) is introduced to constrain the predicted points to encourage manipulation points to remain inside the fabric region. The weights of the MSE loss and projection loss are empirically set to 0.9 and 0.1, respectively. In addition, the projection loss incorporates a penalty coefficient (λ_{outside}) of 1.5 for points predicted outside the cloth region, a reward coefficient (λ_{inside}) of 1.0, and an order term weight (λ_{order}) of 1.0 to encourage left–right consistency. These values were determined empirically to balance training stability and prediction reliability.

To further verify the role of projection loss, we conducted an ablation study comparing models trained with MSE only and with MSE + $\mathcal{L}_{\text{proj}}$, as shown in Table 1. The robustness of predicted action points was evaluated by checking whether they remained within the valid fabric region. On a test set of 50 images, the model trained with MSE only produced seven cases in which at least one manipulation point fell outside the fabric area, whereas the model with $\mathcal{L}_{\text{proj}}$ reduced this to one case. This result indicates that $\mathcal{L}_{\text{proj}}$ effectively constrains the predicted points within the fabric region during AN pre-training.

Table 1. Ablation study on the effect of projection loss during AN pre-training.

Method	Valid Cases	Fabric Images with Invalid Points
MSE	86% (43/50)	14% (7/50)
MSE + $\mathcal{L}_{\text{proj}}$	98% (49/50)	2% (1/50)

Training is performed using the Adam optimizer with an initial learning rate of 1×10^{-4} for 150 epochs. The output of the trained network on the test dataset is shown in Figure 10.

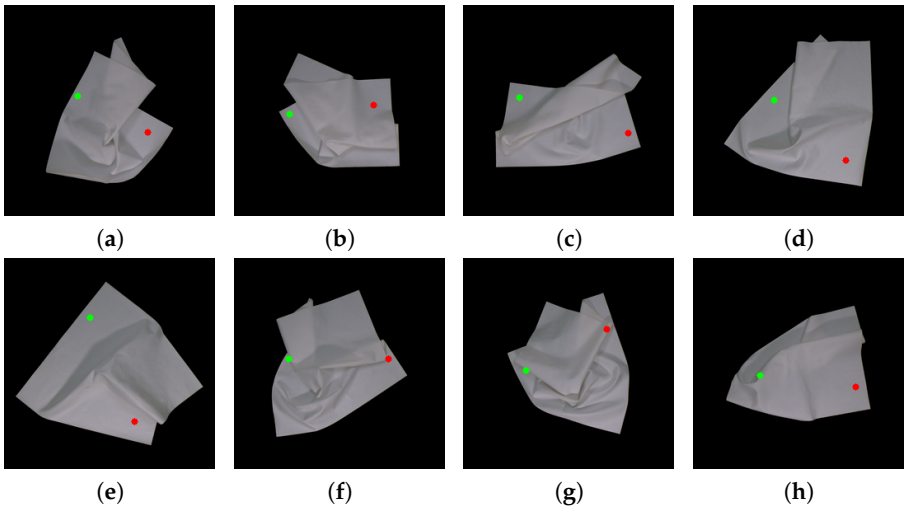


Figure 10. Output examples of the AN, corresponding to the examples in Figure 9. The predicted manipulation points in (**a**–**e**) successfully enable effective fabric flattening, whereas those in (**f**–**h**) are located on wrinkle regions and thus fail to achieve effective flattening.

Figure 10a–h show the resulting images where the predicted manipulation points are marked on I_{seg} according to their coordinates. To clearly distinguish between the

outputs for the two end-effectors, the manipulation point corresponding to the left arm (end-effector 1) is marked in green, while the manipulation point corresponding to the right arm (end-effector 2) is marked in red.

During the evaluation of these outputs, we also observed several challenging cases where flattening was very difficult or even unsuccessful, particularly when the predicted manipulation points fell directly on fabric wrinkles, as shown in Figure 10f–h.

4.3. Joint Reinforcement Learning

The exploration dataset is used to pre-train the Critic Network, enabling it to learn a generalizable reward model. The exploitation dataset is then used for joint reinforcement learning of the AN and the decoder part of the PN, fine-tuning both modules with real-world feedback. During this process, both the AN and the PN decoder are updated. A flattening operation is considered successful when the relative coverage area exceeds 95%, a threshold determined experimentally to ensure sufficient fabric flattening. Coverage below this level typically leaves visible wrinkles. In addition, prior studies on fabric flattening [6] have also adopted 95% as the evaluation criterion, allowing for consistency and comparability with existing work.

To further improve the success rate, we conducted a second round of joint reinforcement learning using an additional 1500 exploration samples to update the value estimator and 3000 exploitation samples to update the policy and PN decoder. Furthermore, we analyzed the training curves of both rounds of joint reinforcement learning, as shown in Figure 11a,b, which illustrate the training curves for the first and second rounds, respectively.

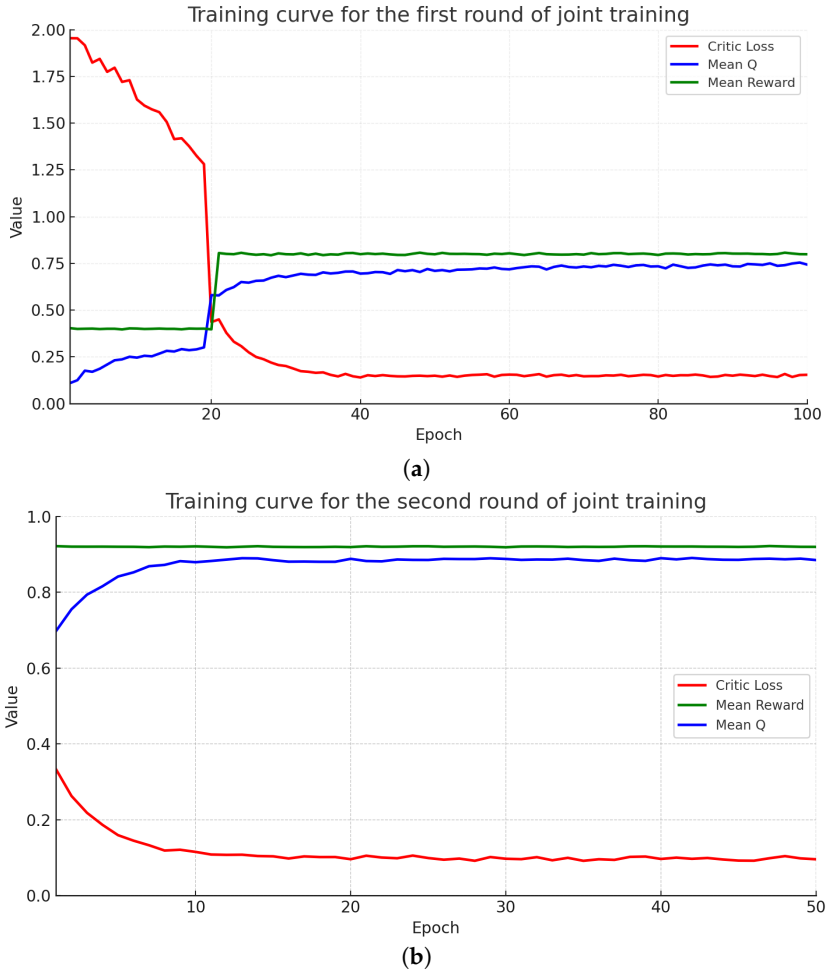


Figure 11. Training curves of the joint training process: (**a**) first-round results; (**b**) second-round results. Each curve shows the evolution of Critic Loss, Mean Q, and Mean Reward.

During the first round, the Critic Network was first pre-trained using exploration data. In this stage, the Critic loss decreased gradually from a relatively high value, while the Q-values slowly increased toward the mean reward. Once pretraining ended and joint optimization of the Critic, Actor, and PN decoder began, the curves exhibited a clear discontinuity: the Critic loss dropped sharply and the Q-values jumped upward, reflecting the sudden effect of coupling policy learning with value estimation. After this transition, both metrics converged more smoothly, with the Q-values progressively approaching the mean reward and the Critic loss stabilizing at a low level.

In the second round, the Critic was initialized directly from the first round, so no pretraining phase was required. The Q-values start close to those achieved in the first round and further increase until stabilizing near the new mean reward, while the Critic loss continues to converge to a lower plateau. These results indicate that the second round primarily refines the already learned policy and improves stability.

We compared the following three training strategies: a network trained by human demonstrations, a network trained with one round of joint reinforcement learning, and a network trained with two rounds of joint reinforcement learning, each evaluated over 50 trials. The results are summarized in Table 2. We also compared another dual-arm flattening strategy [9] and two single-arm strategies [6,7]. The dual-arm strategy is based on wrinkle detection, where the largest wrinkle is flattened at each step until the fabric is fully flattened. We conducted 50 trials under exactly the same experimental conditions, and the results are also reported in Table 2.

Table 2. Flattening success rate comparison.

Policy	Final Success Rate	One-Time Success Rate	Average Operation Count
Trained by human demonstration	82% (41/50)	74% (37/50)	–
One-round joint reinforcement learning	94% (47/50)	90% (45/50)	–
Two-round joint reinforcement learning	100% (50/50)	94% (47/50)	1–3
Wrinkle detection with dual-arm [9]	84% (42/50)	30% (15/50)	3–9
Wrinkle detection with single-arm [7]	–	–	7–9
Deep imitation learning [6]	–	–	4–8

A comparison of the network output trained only by the human demonstration (left images in each pair) and the one trained by joint reinforcement learning (right images in each pair) is shown in Figure 12.

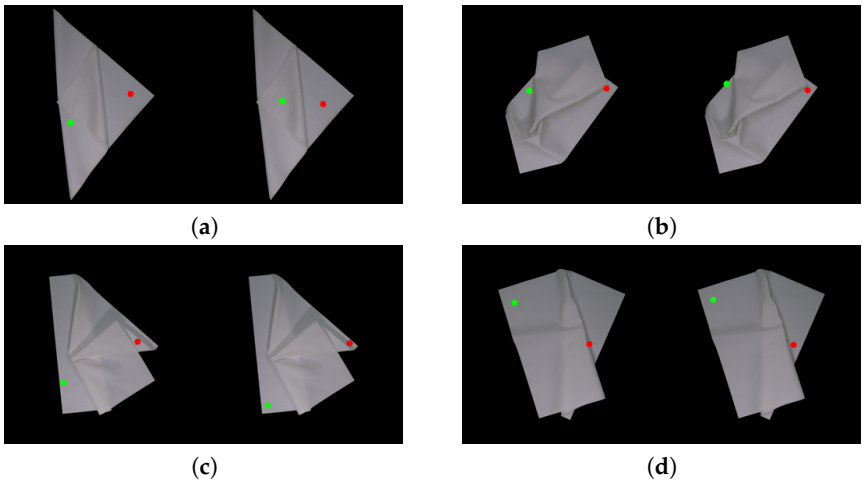


Figure 12. Comparison of the AN outputs shown in (**a**–**d**), where each pair presents the results of the initial policy trained only with human-demonstrated data (left) and the policy after joint reinforcement learning (right).

Compared to the initial policy, the overall success rate increased from 82% to 94%. Additionally, the probability of achieving successful flattening in a single operation also improved from 74% to 90%. After the second round of joint reinforcement learning, performance improved further, and the flattening success rate reached 100%. Since the flattening success rate reached a high level and there was no significant improvement compared to the first round of joint reinforcement learning, this paper adopts the flattening policy obtained after two rounds of joint reinforcement learning as the flattening policy.

Figure 13 shows the flattening performance after the second round of joint reinforcement learning. For each image pair, the left image shows the initial state of the fabric with the selected manipulation points overlaid. The bottom-left corner displays the percentage of fabric coverage compared to the fully flattened fabric coverage. The right image shows the fabric's state after a single operation, with the updated coverage percentage displayed in the same location.

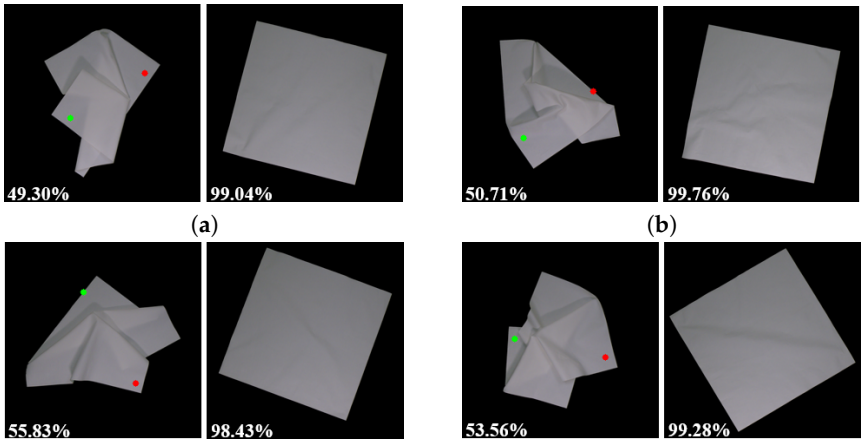


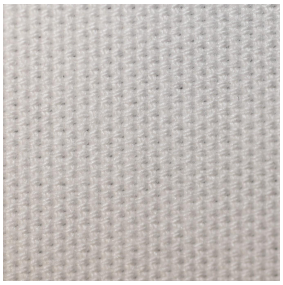
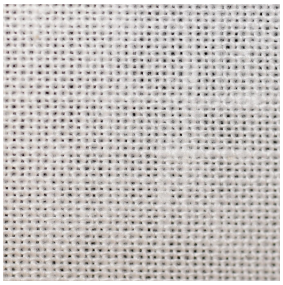
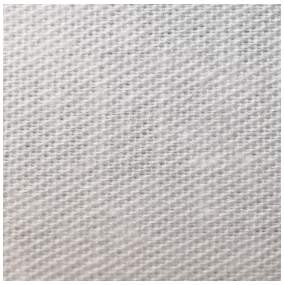
Figure 13. Examples of the flattening results. (**a**–**d**) illustrate the outcomes under different initial fabric configurations using the network trained after two rounds of joint reinforcement learning. For each pair, the left image shows the fabric state before the flattening operation, and the right image shows the fabric state after flattening. The relative fabric area is indicated in the lower-left corner of each image.

4.4. Flattening Evaluation on Different Types of Fabric

After achieving successful flattening on a specific type of fabric, we evaluated the generalization ability of the flattening policy by applying it to different types of fabrics. Without collecting new data or retraining the model, we applied the final policy directly to flatten the different fabric types. Using the original fabric as the baseline, we introduced three additional fabric types and evaluated performance across different material properties through flattening.

The fabrics are categorized into four types based on their stiffness and thickness. The fabric used for training is defined as Type I and is characterized by its thickness and stiffness. To evaluate generalization, three additional fabric types are introduced for testing without retraining: Type II (thick and soft), Type III (thin and stiff), and Type IV (thin and soft). The specimen and parameters of the four fabric types are shown in Table 3. Twenty tests are conducted on each type of fabric until successful flattening is achieved. Figure 14 presents the average flattening performance for each fabric type. The results indicate that different fabrics can be flattened without re-training the network.

Table 3. Physical properties of four fabric types.

Properties	**Type I**	**Type II**	**Type III**	**Type IV**
Specimen				
Material	Cotton	Cotton	Cotton	Linen
ρ (kg/m^2)	0.26	0.24	0.08	0.13
T (mm)	0.28	0.27	0.14	0.18
G (N/mm)	0.39	0.64	0.47	0.68
E (MPa)	17.31	6.60	22.80	14.96

ρ: area density; T: thickness; G: shearing rigidity; E: Young's modulus.

For Type II fabric (thick and soft), 18 out of 20 test cases achieved over 95% area coverage after a single flattening operation. Despite being softer than the baseline fabric, Type II still demonstrates high flattening efficiency under the final policy. In contrast, Type III (thin and stiff) and Type IV (thin and soft) achieved one-shot success rates of 75% (15/20) and 70% (14/20), respectively, and typically required more operations to reach a fully flattened state. On average, Type III required approximately three operations and Type IV required four operations, although both types were eventually flattened successfully after repeated actions.

While the proposed method shows strong performance in fabric flattening across different conditions, it also has certain limitations. The current policy is trained on a specific fabric type and assumes a bilateral structure when selecting manipulation points, which may restrict generalization to more complex or asymmetric cases. In addition, reinforcement learning requires a moderate number of real-world trials, which poses a challenge to scalability. Future work will explore domain-adaptive strategies and lightweight policy architectures to improve generalization and deployment efficiency. In addition, incorporating physical fabric models or real-time online learning may further enhance robustness and facilitate the extension of fabric manipulation to a wider variety of fabric shapes, and enable downstream tasks such as sewing and folding that require a properly flattened fabric surface.

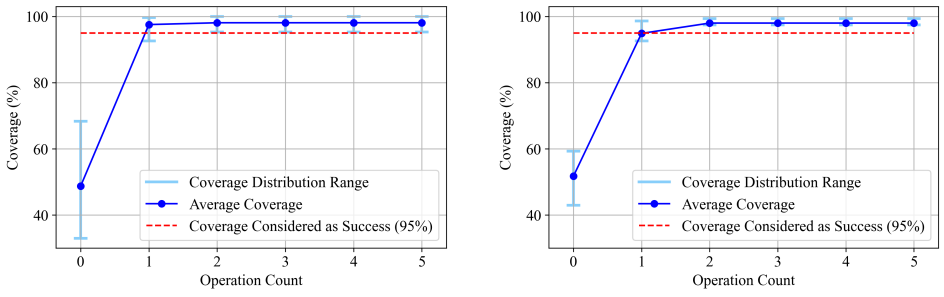


Figure 14. *Cont.*

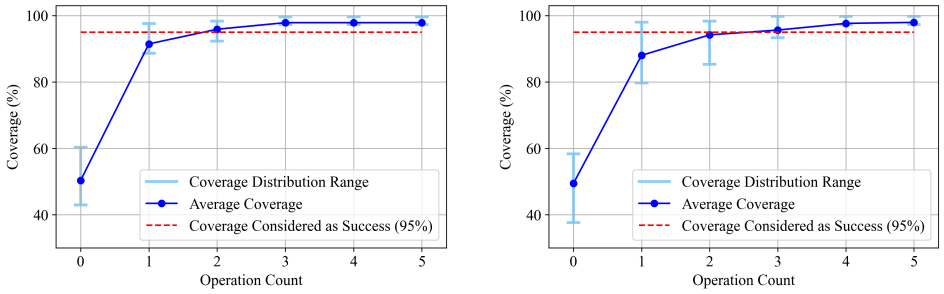


Figure 14. Flattening performance for each different fabric type. (**a**) Type I (baseline, avg of 50 trials); (**b**) Type II (avg of 20 trials); (**c**) Type III (avg of 20 trials); (**d**) Type IV (avg of 20 trials).

5. Conclusions

This paper presented a hybrid training framework for fabric flattening with a dual-arm manipulator, where human demonstrations serve as an adaptive prior. A cascaded Proposal Network and Action Network are first trained on demonstrations and then refined through joint reinforcement learning with real-world feedback. Experiments show that our method improves the overall flattening success rate from 82% to 100% after two training rounds, with single-operation success rising from 74% to 94%. The final policy, trained only on baseline fabric, generalizes effectively to fabrics of varying thickness and stiffness, surpassing single-arm methods in both efficiency and success. Future work will investigate domain-adaptive strategies, lightweight policies, and online learning to enhance generalization, robustness, and applicability to downstream tasks such as sewing and folding. Additionally, we plan to relax the bilateral symmetry assumptions by selecting two manipulation points from the global fabric region instead of explicitly dividing the fabric into left and right halves. This will allow us to explore more flexible flattening strategies.

Author Contributions: Conceptualization, Y.M., F.T., A.S., and A.K.; methodology, Y.M.; software, Y.M.; validation, Y.M., F.T., A.S., and A.K.; formal analysis, Y.M.; investigation, Y.M.; resources, Y.M.; data curation, Y.M.; writing—original draft preparation, Y.M.; writing—review and editing, Y.M., F.T., A.S., and A.K.; visualization, Y.M.; supervision, M.H. and K.K.; project administration, M.H. and K.K.; funding acquisition, M.H. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by MEXT Scholarship, Japan and in part by the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust and in part by the Innovation and Technology Commission of the HKSAR Government through the InnoHK initiative.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Code and test data can be found at https://github.com/NCGjss12138/Fabric-Flattening-Hybrid-Learning, accessed on 3 October 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IL	Imitation learning
RL	Reinforcement learning
PN	Proposal network
AN	Action network
BCE	Binary cross-entropy

HWD	Haar wavelet down-sampling
MSE	Mean squared error
SD	standard deviation

References

1. Tang, K.; Tokuda, F.; Seino, A.; Kobayashi, A.; Tien, N.C.; Kosuge, K. Time-Scaling Modeling and Control of Robotic Sewing System. *IEEE/ASME Trans. Mechatronics* **2024**, *29*, 3166–3174. [CrossRef]
2. Tang, K.; Huang, X.; Seino, A.; Tokuda, F.; Kobayashi, A.; Tien, N.C.; Kosuge, K. Fixture-Free Automated Sewing System Using Dual-Arm Manipulator and High-Speed Fabric Edge Detection. *IEEE Robot. Autom. Lett.* **2025**, *10*, 8962–8969. [CrossRef]
3. Zhu, J.; Cherubini, A.; Dune, C.; Navarro-Alarcon, D.; Alambeigi, F.; Berenson, D.; Ficuciello, F.; Harada, K.; Kober, J.; Li, X.; et al. Challenges and Outlook in Robotic Manipulation of Deformable Objects. *IEEE Robot. Autom. Mag.* **2022**, *29*, 67–77. [CrossRef]
4. Seita, D.; Florence, P.; Tompson, J.; Coumans, E.; Sindhwani, V.; Goldberg, K.; Zeng, A. Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 4568–4575. [CrossRef]
5. Shehawy, H.; Pareyson, D.; Caruso, V.; Zanchettin, A.M.; Rocco, P. Flattening Clothes with a Single-Arm Robot Based on Reinforcement Learning. In Proceedings of the Intelligent Autonomous Systems 17 (IAS 2022), Zagreb, Croatia, 13–16 June 2022; Petrovic, I.; Menegatti, E.; Marković, I., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 577, pp. 580–595. [CrossRef]
6. Seita, D.; Ganapathi, A.; Hoque, R.; Hwang, M.; Cen, E.; Tanwani, A.K.; Balakrishna, A.; Thananjeyan, B.; Ichnowski, J.; Jamali, N.; et al. Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–30 October 2020; pp. 9651–9658. [CrossRef]
7. Qiu, Y.; Zhu, J.; Della Santina, C.; Gienger, M.; Kober, J. Robotic fabric flattening with wrinkle direction detection. In Proceedings of the International Symposium on Experimental Robotics, Chiang Mai, Thailand, 26–30 November 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 339–350.
8. Sun, L.; Aragon-Camarasa, G.; Rogers, S.; Siebert, J.P. Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 185–192.
9. Ma, Y.; Tokuda, F.; Seino, A.; Kobayashi, A.; Hayashibe, M.; Jin, B.; Kosuge, K. Partial Fabric Flattening for Seam Line Region Using Dual-Arm Manipulation. In Proceedings of the 2025 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 3–6 August 2025; pp. 1173–1178.
10. Islam, S.; Owen, C.; Mukherjee, R.; Woodring, I. Wrinkle Detection and Cloth Flattening through Deep Learning and Image Analysis as Assistive Technologies for Sewing. In Proceedings of the 17th International Conference on Pervasive Technologies Related to Assistive Environments, Crete, Greece, 26–28 June 2024; pp. 233–242.
11. Li, C.; Fu, T.; Li, F.; Song, R. Design and Implementation of Fabric Wrinkle Detection System Based on YOLOv5 Algorithm. *Cobot* **2024**, *3*, 5. [CrossRef]
12. Yamazaki, K.; Inaba, M. Clothing classification using image features derived from clothing fabrics, wrinkles and cloth overlaps. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–8 November 2013; pp. 2710–2717.
13. Yang, H. 3D Clothing Wrinkle Modeling Method Based on Gaussian Curvature and Deep Learning. In Proceedings of the 2024 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 12–14 April 2024; pp. 136–140.
14. Lee, R.; Abou-Chakra, J.; Zhang, F.; Corke, P. Learning Fabric Manipulation in the Real World with Human Videos. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 3124–3130. [CrossRef]
15. Doumanoglou, A.; Kargakos, A.; Kim, T.K.; Malassiotis, S. Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 987–993.
16. Xue, H.; Li, Y.; Xu, W.; Li, H.; Zheng, D.; Lu, C. Unifolding: Towards sample-efficient, scalable, and generalizable robotic garment folding. *arXiv* **2023**, arXiv:2311.01267.
17. Ganapathi, A.; Sundaresan, P.; Thananjeyan, B.; Balakrishna, A.; Seita, D.; Grannen, J.; Hwang, M.; Hoque, R.; Gonzalez, J.E.; Jamali, N.; et al. Learning Dense Visual Correspondences in Simulation to Smooth and Fold Real Fabrics. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11515–11522. [CrossRef]

18. Kant, N.; Aryal, A.; Ranganathan, R.; Mukherjee, R.; Owen, C. Modeling Human Strategy for Flattening Wrinkled Cloth Using Neural Networks. In Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kuching, Malaysia, 6–10 October 2024; pp. 673–678. [CrossRef]
19. Weng, T.; Bajracharya, S.M.; Wang, Y.; Agrawal, K.; Held, D. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In Proceedings of the Conference on Robot Learning, PMLR, Auckland, New Zealand, 14–18 December 2022; pp. 192–202.
20. Seita, D.; Jamali, N.; Laskey, M.; Tanwani, A.K.; Berenstein, R.; Baskaran, P.; Iba, S.; Canny, J.; Goldberg, K. Deep Transfer Learning of Pick Points on Fabric for Robot Bed-Making. In Proceedings of the Robotics Research, Geneva, Switzerland, 25–30 September 2022; Asfour, T., Yoshida, E., Park, J., Christensen, H., Khatib, O., Eds.; Springer: Cham, Switzerland, 2022; pp. 275–290.
21. Avigal, Y.; Berscheid, L.; Asfour, T.; Kröger, T.; Goldberg, K. SpeedFolding: Learning Efficient Bimanual Folding of Garments. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 1–8. [CrossRef]
22. Hietala, J.; Blanco–Mulero, D.; Alcan, G.; Kyrki, V. Learning Visual Feedback Control for Dynamic Cloth Folding. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 1455–1462. [CrossRef]
23. Puthuveetil, K.; Kemp, C.C.; Erickson, Z. Bodies Uncovered: Learning to Manipulate Real Blankets Around People via Physics Simulations. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1984–1991. [CrossRef]
24. Antonova, R.; Shi, P.; Yin, H.; Weng, Z.; Jensfelt, D.K. Dynamic environments with deformable objects. In Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual (Online), 6–14 December 2021.
25. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement Learning in Robotics: A Survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [CrossRef]
26. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-End Training of Deep Visuomotor Policies. *J. Mach. Learn. Res.* **2016**, *17*, 1–40.
27. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
28. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
29. Zeng, A.; Song, S.; Yu, K.; Donlon, E.; Hogan, F.; Bauza, M.; Rodriguez, A. Learning Synergies between Pushing and Grasping with Self-supervised Deep Reinforcement Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 747–754.
30. Shehawy, H.; Pareyson, D.; Caruso, V.; De Bernardi, S.; Zanchettin, A.M.; Rocco, P. Flattening and folding towels with a single-arm robot based on reinforcement learning. *Robot. Auton. Syst.* **2023**, *169*, 104506. [CrossRef]
31. Lee, R.; Ward, D.; Dasagi, V.; Cosgun, A.; Leitner, J.; Corke, P. Learning arbitrary-goal fabric folding with one hour of real robot experience. In Proceedings of the Conference on Robot Learning, PMLR, London, UK, 8–11 November 2021; pp. 2317–2327.
32. Ha, H.; Song, S. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In Proceedings of the Conference on Robot Learning, PMLR, Auckland, New Zealand, 14–18 December 2022; pp. 24–33.
33. Sun, Z.; Wang, Y.; Held, D.; Erickson, Z. Force-Constrained Visual Policy: Safe Robot-Assisted Dressing via Multi-Modal Sensing. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4178–4185. [CrossRef]
34. Vecerik, M.; Hester, T.; Scholz, J.; Wang, F.; Pietquin, O.; Piot, B.; Heess, N.; Rothörl, T.; Lampe, T.; Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv* **2017**, arXiv:1707.08817.
35. Ho, J.; Ermon, S. Generative adversarial imitation learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
36. Tsurumine, Y.; Matsubara, T. Goal-aware generative adversarial imitation learning from imperfect demonstration for robotic cloth manipulation. *Robot. Auton. Syst.* **2022**, *158*, 104264. [CrossRef]
37. Fujimoto, S.; Meger, D.; Precup, D. Off-policy deep reinforcement learning without exploration. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2052–2062.
38. Salhotra, G.; Liu, I.C.A.; Dominguez-Kuhne, M.; Sukhatme, G.S. Learning Deformable Object Manipulation from Expert Demonstrations. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8775–8782. [CrossRef]
39. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
40. Bridle, P. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. *Neurocomputing* **1990**, *2*, 227–236.
41. Li, X.; Sun, X.; Meng, Y.; Liang, J.; Wu, F.; Li, J. Dice Loss for Data-imbalanced NLP Tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 465–476. [CrossRef]

42. Wu, T.; Tang, S.; Cai, Q.; Zhang, C.; Torr, P.H. Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022.
43. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
44. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.