Attention-Based Cloth Manipulation from Model-free Topological Representation

Attention-Based Cloth Manipulation from Model-free Topological Representation

Kevin Galassi[1], Bingbing Wu[2], Julien Perez[3], Gianluca Palli[1], Jean-Michel Renders[2]

***Abstract*— The robotic manipulation of deformable objects, such as clothes and fabric, is known as a complex task from both the perception and planning perspectives. Indeed, the stochastic nature of the underlying environment dynamics makes it an interesting research field for statistical learning approaches and neural policies. In this work, we introduce a novel attention-based neural architecture capable of solving a smoothing task for such objects by means of a single robotic arm. To train our network, we leverage an oracle policy, executed in simulation, which uses the topological description of a mesh of points for representing the object to smooth. In a second step, we transfer the resulting behavior in the real world with imitation learning using the cloth point cloud as decision support, which is captured from a single RGBD camera placed egocentrically on the wrist of the arm. This approach allows fast training of the real-world manipulation neural policy while not requiring scene reconstruction at test time, but solely a point cloud acquired from a single RGBD camera. Our resulting policy first predicts the desired point to choose from the given point cloud and then the correct displacement to achieve a smoothed cloth. Experimentally, we first assess our results in a simulation environment by comparing them with an existing heuristic policy, as well as several baseline attention architectures. Then, we validate the performance of our approach in a real-world scenario. Project website: link**

I. Introduction

The manipulation of deformable objects, such as clothes and fabrics, remains a significant challenge for robotic systems. Unlike rigid objects, deformable materials exhibit rich dynamic behavior, making their manipulation particularly intricate and non-trivial. Cloth manipulation, in particular, represents a complex problem due to the various folds, wrinkles, and interdependencies of different cloth regions. Addressing this challenge requires a combination of scene understanding and planning, taking into account the changing behavior of deformable objects. Moreover, deformable objects are characterized by high degree of freedom, resulting in a complicated and challenging manipulation task characterized by lack of repeatability of the experiments and an overall increased complexity compared to rigid objects. Also, the lack of fixed features, e.g. shapes, enhances the complexity of the task itself.

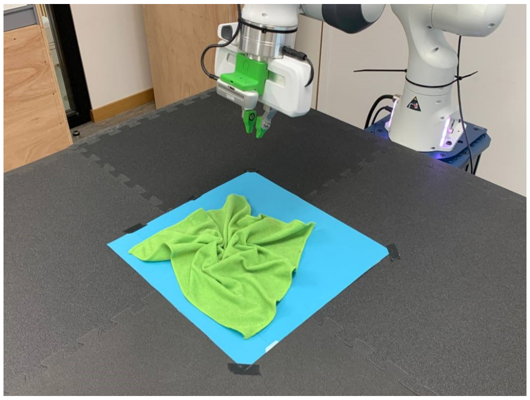


Fig. 1: Robotic Setup used for experimental valuation

The primary objective of this work is to develop a robust and efficient model-free learning approach for cloth smoothing using an autonomous robotic manipulation system. To this end, we introduce a novel transformer-based manipulation policy, which uses point clouds extracted solely from RGBD images. Self-attention is a state-of-the-art deep learning architecture originally introduced for natural language processing tasks and recognized for its capability to handle complex input structures. By leveraging this model architecture for cloth manipulation, we want to enhance the robot's understanding and control of deformable objects, leading to improved efficiency in cloth smoothing processes.

Our approach uses a simulated environment designed for cloth manipulation tasks based on a mass-spring-damper system. This environment includes realistic cloth physics simulation, enabling us to study and analyze the behavior of the cloth under various conditions. Then, we use an oracle policy which leverages the cloth's point masses to generate a sufficient number of training samples. Each training sample is a trajectory composed of a sequence of picks, displacements and releases, starting from an arbitrary initial configuration of the cloth in order to smooth it on top of a targeted surface.

We use the topological representation of the cloth, namely the cloth's point mass positions available only in the simulation, to execute our oracle policy. We record trajectories associated with its resulting behavior and use them to train an attention-based model to predict the robot action. However, to ensure a successful sim-to-real transition, another key idea of our work consists in switching the input of our trained policy to the cloth point cloud obtained from the depth of the scene. The resulting policy exhibits the ability to

1 DEI - Department of Electrical, Electronic and Information Engineering, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy.

2 NAVER LABS Europe, 6 Chemin de Maupertuis, Meylan, 38240, France. firstname.lastname@naverlabs.com

3 EPITA Research Laboratory (LRE) FR-94276 Le Kremlin-Bicêtre France

This work was co-supported by Naver Labs Europe as part of an Internship. This work was co-supported by the Horizon Europe project *IntelliMan - AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics* [grant number 101070136].

Correspondence author:Kevin Galassi *kevin.galassi2@unibo.it*

generalize scenes more effectively and achieves robust results with smaller datasets, outperforming other methods in both simulated and real-world scenarios. Our list of contributions can be summarized as follows:

- We propose a novel attention-based approach for learning a smoothing policy directly from a point cloud, characterized by a simple setup and aiming at improving sample efficiency.
- We compare our results with existing approaches in literature achieving superior results in coverage and efficiency.
- We evaluate the policy in real world using an Intel realsense 3D camera device and a Franka Emika Panda arm.

II. Related Works

Cloth smoothing is an interesting framework for the evaluation of deformable object manipulation approaches. In the context of dual arm manipulation, the task of cloth smoothing was successfully tested over the last couple of years. Several studies have focused on robotic manipulation of cloth for various applications. These methods differ depending on the actual robotic system and the control approaches involved in the definition of the task solution. Commonly, various methods have been developed to utilize robot arms grasping cloth in two distinct positions, lifting it and, using a controlled swinging motion, aligning the cloth with a surface while it is suspended in the air. This process ensures a smooth and controlled placement of the cloth onto the surface. This concept is present in [1] where a dual high-speed linear sliding robot is used to hold the tissue. In [2], [3], the same idea is developed but using a dual-arm robot instead. Dual-arm manipulation were also investigated for clothing assistance tasks [4] where Dynamic Movement Primitives (DMP) were used specifically for putting clothing on human subjects. In [5], starting from the assumption of a smoothed cloth, the proposed system folds the garments based on the detected cloth categories, e.g. trouser, t-shirt, jumpers, derived from the segmentation of the scene and a polygonal model of the garments. For dual-arm approaches, several benchmarks have been released for the evaluation of algorithms on diverse manipulation tasks such as spreading a tablecloth, folding a towel and dressing [6], the complexity levels and quality measures were defined for each task, and baseline solutions were evaluated according to the proposed metrics.

A mechanical system that can grasp, unfold, and position hemmed fabrics using a single-armed robot has also been investigated [7]. This system demonstrated the ability to handle fabrics through a series of motions.

The use of dedicated tools and equipments has also been adopted for solving such task, and cameras are commonly used as primary sensory information for the task execution. Alternatively, other approaches have investigated the deployment of highly specialized end-effectors for the manipulation and the perception of the cloth itself. As an example, in [8], [9], the authors developed grippers with roller fingertips for clothes manipulation, addressing the problem of retrieving fabric that is in danger of slipping away from the grasp. The idea of corner finding with a dedicated tool is also developed in alternatives works such as [10] or in [11], where several algorithmic methods for corner finding were reviewed. An alternative sensor involves the use of tactile sensing technologies placed on gloves, which can be used to provide additional feedback during manipulation [12].

In recent years, there have been numerous developments in statistical learning algorithms for deformable object manipulation tasks. In [13], the authors present a system capable of autonomously transforming a randomly crumpled clothing item into a folded state. They achieve this by using fiducial markers placed on the fabric to enhance and contextualize the observation space of the robot. Single arm cloth smoothing was studied in [14], where the authors learn a smoothing policy from imitation learning. Alternatively, in [15], a model of the cloth interaction from pixels is learned. Subsequently, in [16], a single-arm swing motion is trained instead of a more classical pick and place approach. In this context, attention-based models [17] are gaining rapid diffusion in robotics, following other fields, including computer vision [18], and natural language processing. The main reason for the popularity of such models is their capability to successfully generalize and understand complex scenes and seamlessly take inputs with variable sizes [19].

Transformers have also been successfully deployed in the context of cloth manipulation. In particular, in [20], the authors used Behavior Cloning (BC) from real-world demonstrations, proposing a pick-and-place approach to fold clothes. The use of real experiments to train the policy can be seen as a way to mitigate the Sim-To-Real problem. This constitutes an important topic in robotics, especially in deformable objects, considering the difficulties in accurately simulating the objects' behavior. The problem is emphasized in cases where the objective is to learn from pixels, in which the texture of the objects can be different. Consequently, it is possible to find methods that aim to transfer the learned policy [21] or introduce adaptation modules [22] to increase the robustness of the method in unseen scenarios. As an alternative, the use of point clouds mitigates the issue and the proposed method is capable of being transferred directly to real robot applications.

III. Manipulation Environments

For simulating the cloth and recording the associated trajectories from the oracle policy, we employ an already proven approach that describes the model of the deformable object as a collection of mass-spring-damper systems [23]. This type of modeling has been well-known in the literature and has been successfully applied in the field of deformable planar objects [14] and similarly in the scenario of deformable linear objects (DLOs) [24]. The tissue is generated on a flat surface, and a fixed square shape is used as the target for the smoothing policy. In the simulation, we model the cloth as a square tissue drape composed of $n_s = 25$ points per side. The particles are connected by means of a spring-damping system, whose mechanical parameters are

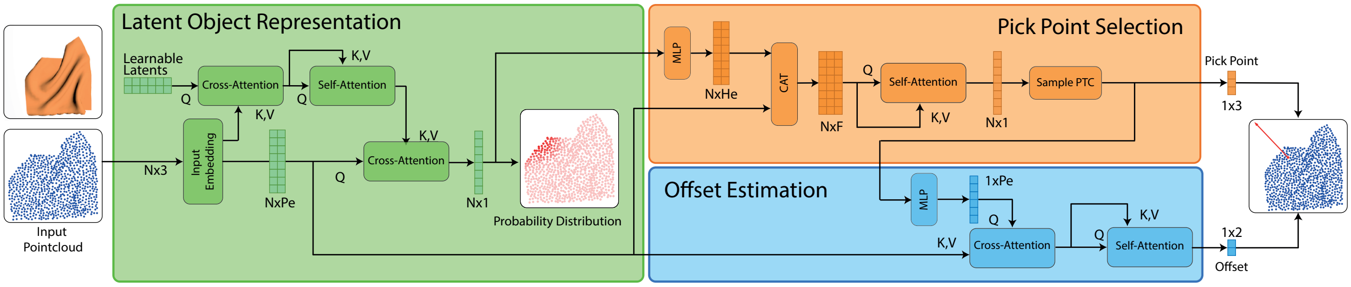


Fig. 2: Details of the three blocks of our network architecture. During the first training step, the probability decoder is trained with the position encoding. In a second step, the offset decoder is trained with the rest of the network kept frozen.

empirically determined, forming a mesh whose dynamical properties resemble those of a tissue. We refer to this collection of points $p_t \in \mathbb{R}^{n_s^2 \times 3}$ as the *topological* representation of the cloth since, based on its structure, we can identify a cloth as a planar deformable object. This set of points, available only in simulation, represents the state of the cloth. From the rendered RGBD image, we can obtain the point cloud, which will represent the observation of the environment. Note that the point cloud is expressed as a set of points $p_{ptc} \in \mathbb{R}^{n_p \times 3}$ in the space, with in general, $n_p < n_s^2$. Note that the point cloud offers only a partial representation of the cloth, since some points of the cloth can not be captured due to occlusion of the cloth itself and can not capture the exact structure of the cloth. In the environment, each action is defined by a pick-and-place sequence that the robot executes. More precisely, the action is described by the following variables:

$$a_t = \langle x, y, \Delta x, \Delta y \rangle \tag{1}$$

Namely, each action a_t executed at step t corresponds to picking up the object at coordinates x, y and releasing it at a new position found by adding the displacement offset $\Delta x, \Delta y$. The coverage value, represented as C, is traditionally calculated by comparing the positions of the outermost boundary points of the fabric with a reference shape that simulates the object when it's laid flat on the workspace. It quantifies the extent to which the cloth surface covers the target shape. This measure is used to determine the success (or failure) of the tissue-smoothing task completion.

To collect our training trajectories, we use an oracle-based policy derived from [14]. At each step, given the cloth's topological points, we identify the corner points. Based on the distance from the desired final position, the furthest point is then chosen and moved. In each episode, the initial state is generated as follows: a cloth is spawned within the scene from a randomly selected high position represented by z and is subsequently released. Following the release, a random "pick-and-place" operation is carried out on the cloth to introduce additional randomness to the initial configuration. Additionally, the entire cloth is rotated by a random angle within $\alpha \in [-\pi/6, \pi/6]$. This last addition is made to force the corner points to be further from the final desired position and to introduce more variety in the initial configurations. At each episode, the actions are applied until either the target coverage threshold of $> 90\%$ is reached or the total amount of available steps $= 10$ is executed. In total, a set of 15K steps is collected and used [1].

IV. Model and Algorithm

Our model, depicted in Fig. 2, consists of a dual-step predictor where the selection of the point to pick and the determination of the translational offset to apply are handled by two dedicated architectures invoked sequentially. As an insight, we conceive this sequence of predictions as the choice of the releasing position depending on the picked points. Both prediction modules, pick position and displacement depend on a prior representation module that feeds a latent representation of the object to them.

A. Latent object representation

The object representation module produces a latent input that is used in the following *pick-point selection* and *offset estimation* modules. To enforce an informative latent representation, we train these layers to contextualize each point inside the cloth by introducing a probability distribution associated with each point. The heuristic policy produces a discrete action represented as a "one-hot encoding" vector, with all values equal to zero except for the point selected according to the policy described in Sec.III. Since choosing the exact oracle's point is not mandatory for the task, and all points sufficiently close to the oracle are eligible as well, we use a Gaussian distribution $P_i \sim \mathcal{N}(\mu_i, \sigma^2)$ to model the groundtruth probability distribution of being the selected point to pick, where μ_i corresponds to the oracle's position. We set $\sigma = 3$ to ensure that approximately one-third of the cloth is covered by the distribution.

During the training of these layers, the network uses the n_s^2 topological points, while during evaluation, we assess the performance of the network using the n_p points from the point cloud. As output, the layers produce a latent representation of the cloth that approximates the probability distribution of selecting each input point. This representation is then passed to both the *pick-point selection* module and the *offset estimation* module. We used a Perceiver [25] to

[1] This is to be compared with the approach based on imitation learning described in [14] that required 50K steps and, alternatively, the model-based approach of [15] that required around 100K steps.

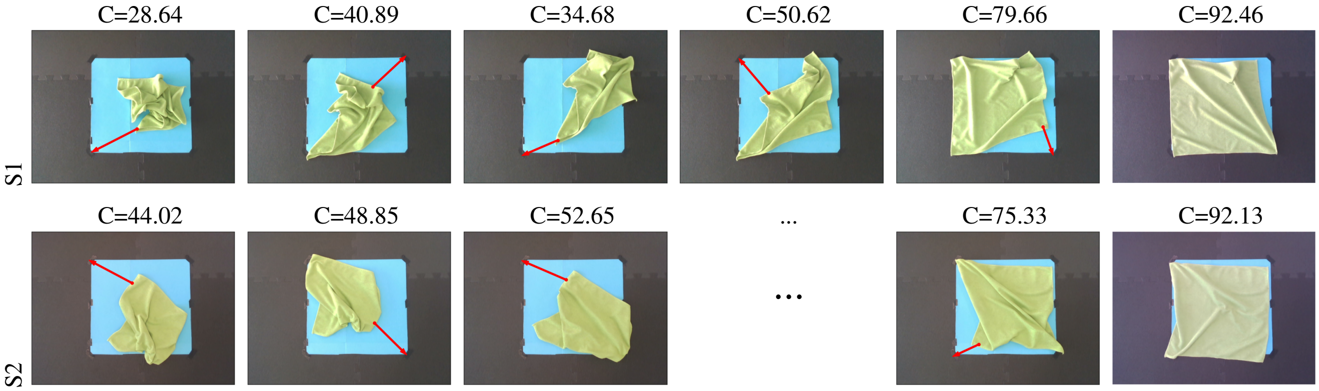


TABLE I: Two episodes starting from diverse configurations. The relative action executed. The coverage C with respect to to the target surface depicted in blue is reported at each step.

estimate this probability, using the embedded $n \times 3$ points as input. The use of this scalable neural architecture permits the reduction of the network's computational complexity while improving prediction. The input points are represented as arrays of size $n \times p_e$, with n the number of points (equal either to n_s^2 at training stage or to n_p at evaluation stage), while p_e is the embedding dimension. They serve as the keys and values for a cross-attention module. The queries are represented by a learnable array of parameters with dimensions $n_l \times p_e$, followed by self-attention layers. As output, a layer of cross-attention is used, followed by a fully-connected layer and a softmax layer to reduce the number of features to $n \times 1$.

B. Point Prediction Module

Even if the latent representation of the object is designed to capture the density distribution as being selected as the pick point, it appears that it is not precise enough to be directly used as a final selection policy, but only at isolating the most promising modes of this distribution. The Point Prediction module will leverage this knowledge to further focus on these promising areas. To this aim, we utilize a Transformer encoder layer. As input, the network incorporates information from the previously pre-trained layers. More specifically, we combine by concatenation the positions of the embedded cloth's points, represented as $n \times p_e$, with the predicted probability distribution of each point. The latter is processed through a set of fully connected layers to transform it into the desired final dimension of $n \times p_e$. At the final stage, the features are reduced to a vector of dimensions $n \times 1$ through a softmax layer and compared using a cross-entropy loss with the point-selecting groundtruth vector coming from the oracle policy. At evaluation time, the point from the point cloud with the highest probability is selected as the point to pick, represented as a 1×3 array.

C. Offset Prediction Module

This last module predicts the displacement offset to be applied to the point chosen during the previous step. The network embeds the selected point into a dimension of $1 \times p_e$ and uses this embedding as the input query for a cross-attention layer. The embedded cloth points, represented as $n \times p_e$ and obtained from the latent representation layers, serve as key-value pairs. During the training of this module, we pass the ground-truth picked point, corresponding to one of the four cloth corners selected according to the oracle policy. The keys and values are obtained from the previous layers with frozen weights.

A first remark on this model architecture choice is that the resulting model is trained sequentially, not end-to-end. It should also be noted that the offset prediction module is not directly related to the pick selection part of the overall model.

V. Experiments

Our evaluations address the following questions: 1) Can a manipulation policy for fabric smoothing be learned using an attention-based model? How does changing input parameters, such as number of points of the point cloud, affect the performance of the policy? 2) How robust is our displacement network to possible errors in predicting the point to grasp? 3) How can the policy be transferred from simulation to the real world?

All our evaluations were conducted with the same fixed seeds for repeatability, using a cloth of dimensions 1m for simplicity. We used a workstation equipped with an NVidia A2000 with 12GB of memory. As an optimizer, we used *LAMB* [26] with a starting learning rate of $1e^{-4}$ and a batch size equal to 128. The trainable latent parameters of the Perceiver were initialized using a normal distribution $\mathcal{N}(0, 0.2)$ as suggested in [27], while a GELU has been used as activation function inside the attention layers. The overall dataset length is composed by 15K steps. We split the dataset between 80% of training and 20% for the validation set.

A. Ablation study

We start the evaluation by comparing our model with a Transformer model to predict a vector of dimensions 1×4, which corresponds to the pick points and the relative offset. Then, we predict which cloth points should be moved by switching to a prediction of $n \times 1$, with n being the same dimension as the input point cloud. This prediction output, therefore, results in two vectors of dimensions $n \times 1$ and 1×2, which are obtained from two separate feedforward layers. We consider describing the probability of picking the points in two possible ways: (1) A delta Dirac function with 1 in the correspondence of the ground truth picking point; therefore, we use a classification problem with a cross-entropy loss (CE); (2) we describe the points as a Gaussian distribution, as explained in Sec. IV, leading to the use of a mean-square error loss. For the prediction of the offset in both models, we use an MSE loss, which is then added to the point prediction loss. As depicted in Fig.2, we split the prediction into two consecutive steps using a dual Transformer architecture. The output's dimension remains the same with a vector of point probabilities followed by the offset predicted based on the chosen point. A final layer predicts the probability distribution of each point to be selected for picking, and we combine the features by summation (SUM) or concatenation (CAT). This distribution is used as the ground truth for the first step of the training to output the probability distribution in the final two model architectures. In the first case, it will be summed to the embedded points, and in the second case, the features are concatenated.

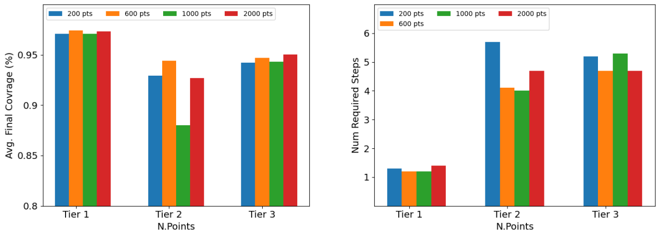
The ablation of the networks was performed with the same number of layers and parameters, meaning that the same number of attention layers, attention heads, and latent dimensions were used across different architectures. Table. II depicts the results of the training in terms of the average distance from the picked point and the displacement offset of the ground truth in an evaluation dataset.

Method	**Avg. Pt. Pick Distance [m]**	**Avg. Pt. Release [m]**
Transformer	0.221	0.084
Transformer Point (CE)	0.799	0.064
Transformer Point (MSE)	0.788	0.061
Dual Transformer	0.167	0.011
Heat. Transformer (Sum)	5.47×10^{-2}	1.50×10^{-2}
Heat. Transformer (CAT)	5.35×10^{-3}	6.33×10^{-3}

TABLE II: Comparison of attention models tested with the proposed architecture. We present the average distance of the prediction from the ground truth picking points and release points obtained during training, (lower is better).

B. Simulation results

Based on the results presented in Tab.II, we evaluate the final coverage achieved with the best model in simulation. Here, we compare the performance associated with two different types of inputs: (1) the cloth topological points n_t, as used in training, and (2) the point cloud n_{ptc} obtained from the depth image. The number of n_{ptc} points was reduced to a fixed length using the classic Furthest Points Sampling (FPS) algorithm. We report the results obtained in III and compare them with the heuristic policy and two related works [14], [15]. The first method uses Dagger [28] to learn a policy from images, while the second method employs a model-based approach to learn the model behavior and then uses a Model Predictive Control (MPC) approach to choose the best action. We compare the results obtained across different starting configurations of the simulated cloth, following an incremental order of difficulties. Our results do not show any significant difference in performance when utilizing point clouds compared to the explicit topological configuration of the manipulated fabric.



(a) Achieved Coverage

(b) Steps Required

Fig. 3: Average coverage achieved in various tiers (left), average number of steps required (right). Notably, the number of data points has no discernible impact on the network's performance within the simulated scenario.

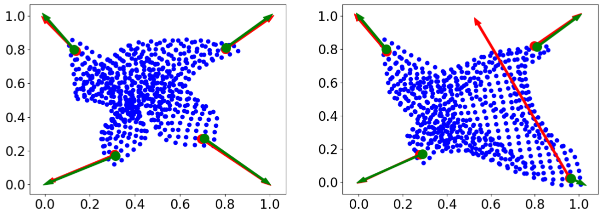
Tier	**Method**	**Coverage**	**No. of Steps**
1	Policy	96.61±1.88	1.80±0.85
1	Ours (top. points)	96.69±1.89	1.68±0.68
1	Ours (point cloud)	96.56±1.81	1.80±0.80
1	Dagger RGBD [14]	95.1±2.2	3.3 ± 3.2
1	VSF [15]	92.5±2.51	8.3±4.7
2	Policy	94.86±3.33	4.52±2.23
2	Ours (top. points)	95.15±1.66	3.68±1.09
2	Ours (point cloud)	94.09±1.35	4.60±1.96
2	Dagger RGBD [14]	91.7±7.1	5.4±4.2
2	VSF [15]	90.3±3.86	12.1±3.42
3	Policy	94.89±1.49	4.40±1.10
3	Ours (top. points)	95.04±1.76	4.56±1.20
3	Ours (point cloud)	94.90±1.53	4.92±1.29
3	Dagger RGBD [14]	87.7±10.1	7.2±2.3
3	VSF [15]	89.3±5.9	13.1±2.9

TABLE III: Comparisons of the network performance with the proposed oracle policy include coverage (higher is better) and the required number of steps (lower is better). The evaluation of the network's predictions was conducted across 25 episodes using a fixed seed, considering both mesh topological points and cloth point cloud (PTC).

Additionally, we test the performance of our model using various point cloud dimensions. We reduce the number of points using FPS to the following set of fixed numbers: $n = [200, 600, 1000, 2000]$. As a reference, the fixed number of points used in simulation was $n = 625$. During the evaluation, the predictions were consistent across the range of values without any noticeable differences; therefore, the number of points was limited to 600.

C. Robustness to diverse grasping prediction

To ensure reliable and consistent decisions from the model, we need to understand what occurs when the predicted point in the dual-stage approach deviates from the correct one. It is indeed possible for the predicted grasping position to differ from the reference one, even while maintaining a consistent displacement with the expert trajectory. Fig. 5 illustrates such situation. Given that the target displacement is readily accessible for the reference points, we adopt a



(a) (b)

Fig. 4: Comparison of displacement predictions (in red) with the expected displacements of corner points (in green). On the right, an erroneous behavior is evident, where the offsets of corner points close to the target are inaccurately predicted.

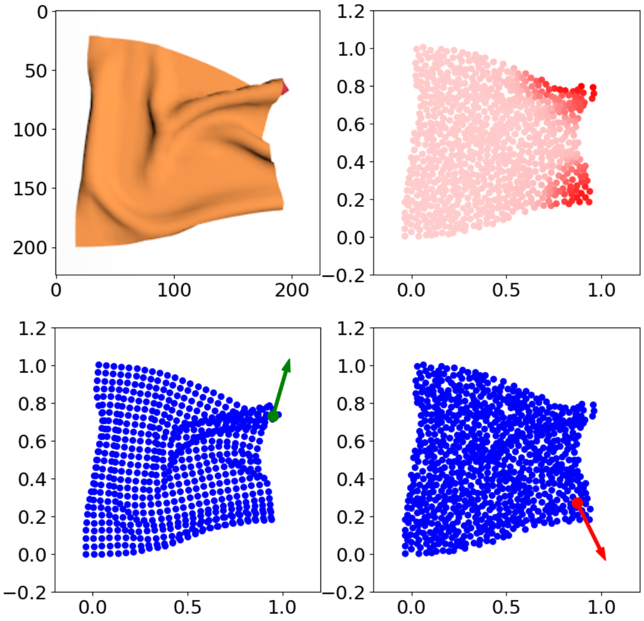


Fig. 5: The predicted picking distribution (top right) reveals two distinct significant areas: the network's prediction (bottom right) diverges from the heuristic policy (bottom left) but aligns with the intended smoothing behavior.

quantitative approach in which all potential corner points are used as inputs to the model. Subsequently, the predicted offset is compared to the actual displacement provided by the simulation. The obtained results show consistency across the prediction of all the oracles' points, with an average of 0.08±0.15m. Fig. 4 illustrates an erroneous behavior exhibited by the network. After analysis, it appears that the oracle's predicted displacement is primarily affected when the prediction is located near their target positions. During the training process, the majority of the samples involved scenarios where the corner points were considerably distant from their expected positions. This suggests that refining the action prediction model through additional fine-tuning, using a dataset that incorporates such examples, is likely to improve and address this issue.

D. Transfer to real robot

For our real-world experiments, we used a Franka Emika Panda robot with custom fingertips to further reduce the fingertips' size.To obtain the point cloud, we use a realsense d435 camera mounted on gripper sides. The camera's precision sufficed for the task, but to ensure a reliable grasp despite depth information noise, we adjusted the grasping height to closely match the table. With more accurate sensors, such as the Photoneo mentioned in related works, it would be possible to grasp the cloth point at its exact position. The cloth used is a squared microfiber tissue of dimension 39 × 39cm.

	Method	**Start Cov. [%]**	**Final Cov.[%]**	**Steps.**
T1	VSF [15]	78.3 ± 6	93.4 ± 2	8.2 ± 4
T1	Dagger RGBD [14]	72.5 ± 4	95.0 ± 2	2.1 ± 1
T1	Dagger D [14]	77.9 ± 4	78.8 ± 24	5.5 ± 4
T1	**Ours**	56.7 ± 10.7	87.5 ± 7.1	4 ± 2
T2	VSF [15]	59.5 ± 3	87.1 ± 9	12.8 ± 3
T2	Dagger RGBD [14]	55.0 ± 5	91.3 ± 8	6.8 ± 3
T2	Dagger D [14]	58.7 ± 5	64.9 ± 20	8.3 ± 3
T2	**Ours**	50.1 ± 9.9	83.3 ± 9.8	6 ± 3
T3	VSF [15]	41.4 ± 3	75.6 ± 15	15.0 ± 0
T3	Dagger RGBD [14]	41.7 ± 2	83.0 ± 10	8.8 ± 2
T3	Dagger D [14]	47.0 ± 3	63.2 ± 9	10.0 ± 0
T3	**Ours**	44.3 ± 9.0	70.2 ± 16.2	10 ± 1

TABLE IV: Performances in coverage of the target surface, higher the better, and associated number of steps, lower the better.

Tab. IV compares the performance of our approach with the reported results from [14] and [15]. The results from the referenced studies were obtained using a different robotic platform composed of DaVinci Robots and a Photoneo, but with comparable perception capabilities and degrees of freedom (DoFs). Despite successfully training the reference models in simulation using the code reported by the authors, transferring them to the real robot proved more challenging. The results were significantly lower than expected; thus, the original results are reported to be more fair with respect to the original works. In general, during the evaluation, we noticed that the use of point clouds in this case is remarkably simple to transfer between various starting configurations, as the difference between simulation and reality can be more easily addressed by adding noise in the simulation. Another issue pertains to the depth map generated by the camera, which exhibited lower precision compared to its simulated counterpart. Despite introducing noise during the training process, the policy relying solely on depth information proved insufficient for successfully completing the task.

VI. Conclusion

In this paper, we aim to address the challenging problem of cloth manipulation by introducing a novel attention-based model for cloth smoothing. Our approach is designed for single-arm robotic manipulation in a simulated environment. We leverage topological points of the simulated cloth to train a model, which is then transferred and evaluated in the real world using a point cloud as input. In future work, we will study the use of topological points in model-based approaches. Finally, we plan to evaluate whether reinforcement learning can improve the achieved performance or serve as an alternative method for training.

References

[1] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Dynamic Manipulation of a Cloth by High-speed Robot System using High-speed Visual Feedback," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8076–8081, 1 2011. [Online]. Available: http://dx.doi.org/10.3182/20110828-6-IT-1002.00596

[2] H. Yuba, S. Arnold, and K. Yamazaki, "Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty," *Advanced Robotics*, vol. 31, no. 10, pp. 544–556, feb 9 2017. [Online]. Available: http://dx.doi.org/10.1080/01691864.2017.1285722

[3] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu, *Spreading of clothes by robot arms using tracing method*. Elsevier, 2007, pp. 77–80.

[4] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using Dynamic Movement Primitives," in *Proceedings of the Advances in Robotics*. ACM, jun 28 2017. [Online]. Available: http://dx.doi.org/10.1145/3132446.3134878

[5] J. Stria, D. Prusa, V. Hlavac, L. Wagner, V. Petrik, P. Krsek, and V. Smutny, "Garment perception and its folding using a dual-arm robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 9 2014. [Online]. Available: http://dx.doi.org/10.1109/IROS.2014.6942541

[6] I. Garcia-Camacho, M. Lippi, M. C. Welle, H. Yin, R. Antonova, A. Varava, J. Borras, C. Torras, A. Marino, G. Alenya, and D. Kragic, "Benchmarking Bimanual Cloth Manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1111–1118, 4 2020. [Online]. Available: http://dx.doi.org/10.1109/LRA.2020.2965891

[7] M. Shibata, T. Ota, Y. Endo, and S. Hirai, "Handling of hemmed fabrics by a single-armed robot," in *2008 IEEE International Conference on Automation Science and Engineering*. IEEE, 8 2008. [Online]. Available: http://dx.doi.org/10.1109/COASE.2008.4626476

[8] K. S. M. Sahari, H. Seki, Y. Kamiya, and M. Hikizu, "Clothes Manipulation by Robot Grippers with Roller Fingertips," *Advanced Robotics*, vol. 24, no. 1-2, pp. 139–158, 1 2010. [Online]. Available: http://dx.doi.org/10.1163/016918609X12586175245158

[9] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu, "Inchworm robot grippers for clothes manipulation," *Artificial Life and Robotics*, vol. 12, no. 1-2, pp. 142–147, 3 2008. [Online]. Available: http://dx.doi.org/10.1007/s10015-007-0456-6

[10] Sahari, "Edge Tracing Manipulation of Clothes Based on Different Gripper Types," *Journal of Computer Science*, vol. 6, no. 8, pp. 872–879, aug 1 2010. [Online]. Available: http://dx.doi.org/10.3844/JCSSP.2010.872.879

[11] P. Jimnez, "Visual grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview," *Robotics and Autonomous Systems*, vol. 92, pp. 107–125, 6 2017. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2017.03.009

[12] P. Maiolino, S. Denei, F. Mastrogiovanni, and G. Cannata, "A sensorized glove for experiments in cloth manipulation," in *2013 IEEE RO-MAN*. IEEE, 8 2013. [Online]. Available: http://dx.doi.org/10.1109/ROMAN.2013.6628484

[13] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 9 2011. [Online]. Available: http://dx.doi.org/10.1109/IROS.2011.6095109

[14] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658.

[15] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for physical sequential fabric manipulation," *Autonomous Robots*, pp. 1–25, 2022.

[16] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," 2022.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] X. Ma, D. Hsu, and W. S. Lee, "Learning latent graph dynamics for visual manipulation of deformable objects," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8266–8273.

[20] K. Mo, C. Xia, X. Wang, Y. Deng, X. Gao, and B. Liang, "Foldsformer: Learning sequential multi-step cloth manipulation with space-time attention," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 760–767, 2023.

[21] J. Hietala, D. Blanco-Mulero, G. Alcan, and V. Kyrki, "Learning visual feedback control for dynamic cloth folding," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2022. [Online]. Available: https://doi.org/10.1109%2Firos47612.2022.9981376

[22] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," 2021.

[23] S. M. Platt and N. I. Badler, "Animating facial expressions," in *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, 1981, pp. 245–252.

[24] N. Lv, J. Liu, X. Ding, J. Liu, H. Lin, and J. Ma, "Physically based real-time interactive assembly simulation of cable harness," *Journal of Manufacturing Systems*, vol. 43, pp. 385–399, 2017, special Issue on the 12th International Conference on Frontiers of Design and Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612517300146

[25] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer *et al.*, "Perceiver io: A general architecture for structured inputs & outputs," *arXiv preprint arXiv:2107.14795*, 2021.

[26] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," 2020.

[27] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[28] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.