



Duan, L. and Aragon-Camarasa, G. (2022) Recognising Known Configurations of Garments For Dual-Arm Robotic Flattening. In: 2022 4th International Conference on Robotics and Computer Vision (ICRCV 2022), Wuhan, China, 25-27 September 2022, pp. 340-344. ISBN 9781665481700 (doi: [10.1109/ICRCV55858.2022.9953186](https://doi.org/10.1109/ICRCV55858.2022.9953186))

There may be differences between this version and the published version. You are advised to consult the published version if you wish to cite from it.

<http://eprints.gla.ac.uk/273547/>

Deposited on 20 June 2022

Enlighten – Research publications by members of the University of Glasgow
<http://eprints.gla.ac.uk>

Recognising Known Configurations of Garments For Dual-Arm Robotic Flattening

Li Duan
School of Computing Science
University of Glasgow
Glasgow, United Kingdom
l.duan.1@research.gla.ac.uk

Gerardo Aragon-Camarasa
School of Computing Science
University of Glasgow
Glasgow, United Kingdom
gerardo.aragoncamarasa@glasgow.ac.uk

Abstract—Robotic deformable-object manipulation is a challenge in the robotic industry because deformable objects have complicated and various object states. Predicting those object states and updating manipulation planning is time-consuming and computationally expensive. In this paper, we propose learning *known configurations* of garments to allow a robot to recognise garment states and choose a pre-designed manipulation plan for garment flattening.

Index Terms—pattern recognition; robot vision; depth images; deformable objects

I. INTRODUCTION

Robots have been widely applied in many aspects of life, for example in robotic assistance for humans, life-threatening rescue, and manufacturing. Among these applications, robotic deformable object manipulation plays a critical part. Autonomous garment laundry and sorting, soft-object manipulation in life-threatening rescue and manufacturing, and robotic threading in the textile industry require an efficient and robust robotic deformable-object manipulation strategy.

However, robotic manipulation of deformable objects remains an open problem in robotics because deformable objects can take unpredictable object states (crumpled, stretched, or bent configurations). That is, their states frequently vary when they are being manipulated and can not be predicted precisely and timely. Deformable objects are easily slipped from robots and take a long time to be manipulated because of their properties. Therefore, current research efforts focus on reducing object-state or robotic action-state complexity.

Current approaches that reduce deformable object complexity for robotic manipulation can be divided into two categories: those that propose to find simplified models to represent object states or action states first [1]–[6], and those that rely on model-free reinforcement learning solutions where the agent learns to model the object while interacting with it [7]–[9]. When deformable object models are simplified, it is possible to reduce the computational costs, making it possible to predict the object states of manipulated deformable objects close to real-time. An efficient prediction of future object states enables a robot to update its manipulation plan on-the-fly, which is critical to manipulating deformable objects in a real-time setting. However, current research is geared towards simple geometries such as ropes, towels or cube-shaped objects. When a rein-

forcement learning solution is used, data-driven reinforcement learning agent learns manipulation policies using a reward-based mechanism. Robots are rewarded if their manipulation policies produce actions that contribute to a step closer to the object’s goal state. However, most of the reinforcement learning approaches rely on simulated environments of which the physics of deformable objects are simplified.

In this paper, we propose taking advantage of gravity to model and learn the *known configurations* of deformable objects (garments in this paper), which can be later used to choose a pre-designed manipulation plan based on the recognised *known configurations* to flatten these objects. Our approach does not need to find simplified models that provide good representations of object states for deformable objects. Similarly, our approach does not require manual action labelling compared to reinforcement-learning approaches. In this paper, we focus on recognising the *known configurations* of garments and develop a pipeline to flatten garments, see Figure 1.

In summary, our main contributions in this paper are two-fold: we propose learning the *known configurations* of garments, which is the foundation for a full pipeline of an effective robotic garment flattening pipeline. Compared with state-of-art, our approach has a higher accuracy (89% versus 73% in [5], [6]) and we captured a *known configurations* database which comprises RGB and depth images of garments

We introduce the related work in section II, and our methodology in section III. Our experiment results are in section IV and a discussion in section V. Finally, we conclude this paper and suggest future work in section VI.

II. RELATED WORK

Research on robotic deformable-object manipulation is divided into two categories: model-based approaches and model-free approaches. Model-based approaches focus on engineering methods, simplifying models representing object and action states or applying dynamics modelling rather than state-space modelling. On the other hand, model-free approaches concentrate on devising data-driven methods mainly relating to reinforcement or imitation learning. Robots learn to manipulate deformable objects by applying manipulation policies, usually trained on simulated environments.

The main two approaches of models-based approaches are object-state modelling and action-state modelling. Lin *et al.* [10] have proposed to use finite element methods (FEM) to model the dynamics of bottles, where they defined a ‘liftability’ test. A bottle can only be lifted if its deformation passes the ‘liftability’ test, so the bottle can be stably manipulated without slipping from the robot’s gripper. A mass-spring system has been applied in [4] to grasp deformable objects using a multi-fingered robotic hand. The Nvidia PhysX simulator has also been utilised in [11] to teach a robot for assisted human dressing. Miller *et al.* [12] have proposed a robotic laundry folding approach by introducing a quasi-static cloth model that represents states of garments and an algorithm that outputs a motion planning based on a 2D cloth polygon generated from the quasi-static cloth model and a desired sequence of folds. Yamakawa *et al.* [13] have also proposed learning a simplified model to represent the state space of ropes and derive motion planning from the inverse dynamics of rope states. McConachie *et al.* [14] have suggested learning reduced state spaces by using a classifier to bias a planner away from state-action pairs that are not feasible under general robotic working environments.

The problem with model-based approaches is that due to the high dimensionality of the deformable object and action spaces, only objects with simple geometries (such as towels, ropes, and sponges) can be modelled and manipulated. Objects with complex geometries such as garments can not be used in these approaches because modelling and manipulation are time-consuming and computationally costly. An alternative to simplify model representations is to convert unknown object states to known object states. Li *et al.* [6] [5] have proposed to estimate the pose of hanging garments and apply manipulation plans based on the recognised garments’ poses. Poses only relate to grasping points rather than initial object states of garments, effectively converting garments from unknown object states to known object states. But their approach requires re-grasping and rotating garments to recognise the poses, adding a time overhead for executing a garment manipulation task.

Data-driven approaches have been recently proposed, namely imitation and reinforcement learning. That is, Pignat *et al.* [7] have proposed to encode sensory information and motor commands as a joint distribution in a hidden semi-Markov model, of which parameters are learned from a set of human demonstrations. Each model set represents a sensorimotor pattern whose sequencing can produce complex behaviours. Balaguer *et al.* [9] have presented a combined approach of imitation and reinforcement learning, where human demonstrations are used to reduce the search space of the reinforcement learning agent and allow the solution to converge quickly. Tsumine *et al.* [15] have proposed a deep reinforcement learning for robotic cloth manipulation with a smooth policy update. Their networks combine the nature of smooth policy updates in value-function-based reinforcement learning with automatic feature extraction from high-dimensional observations in deep neural networks to enhance the sample efficiency and learning stability with fewer samples. Twardon *et al.* propose restricting

robotic action-space modelling to the actions that are safe for robotic hands and arms, garments and the head when a robot is trained to put a knit cap on a Styrofoam head. A direct policy-search algorithm finds appropriate trajectories in the restricted action space in order to enable the robotic knit-cap operation. Their approach has demonstrated an effective manipulation planning under a restricted action space. McConachie *et al.* [16] have proposed formulating a deformable object manipulation task as a multi-armed bandit problem, with each arm representing a model of the deformable object. Matas *et al.* [17] use a tailored version of the deep deterministic policy gradients (DDPG [18]) for deformable object manipulation, where they have learned a policy from simulation data while testing the policy on real data. The limitation of imitation and reinforcement learning approaches is that they need several training iterations to learn tasks. Matas *et al.* [17] trained a robot to fold a towel and drape a piece of cloth through a hanger by reinforcement learning. The robot is trained in a simulated environment for around 250 iterations. Training such a robot in real environments is difficult because some actions need to be validated in a simulated environment. Hazardous actions (actions that may harm robots or human beings) are easy to handle in simulated environments but need to be avoided in real environments.

III. METHODOLOGY

As described in section I, state-of-art approaches conduct manipulations on tables or platforms, where deformable objects have complicated and various configurations (or object states) and result in computationally costly updates to find a manipulation plan. In contrast, we consider taking advantage of gravity to control the variety of object configurations and reduce the complexity of deformable object configurations.

In this paper, we start by assuming that garments of the same categories (e.g. jeans, towels, tshirts, etc.) lying on a table have different configurations. If the garments are grasped from similar grasping points, they will have similar configurations when the robot picks them up, which we call *known configurations* in this paper. For example two towels have different crumpled configurations on a table (i.e. starting configurations), but they have similar *known configurations* after a robot grasps them. These *known configurations* only depend on the grasping points because of gravity. That is, we convert complex starting configurations to simple *known configurations* from which the robot can follow pre-defined manipulation plans to flatten them. Recognising the *known configurations* of garments is crucial for the robotic garment flattening pipeline shown in Figure 1.

After the garment’s *known configuration* is obtained after grasping them from the table, we find the second grasping points based on how gravity effect on the garment. For example, as shown in Figure 1 step 1 in the *manipulation routine* box, the second grasping point is the lowest point of the towel. Different *known configurations* have different locations of the second grasping points, so recognising the garment’s *known configurations* are critical to localise the second grasping point.

After the robot finds the second grasping point, the next step (step two in Figure 1) is to find the third grasping point, which is the opposing ending corner of the towel. Then, the robot stretches the towel from the two grasping points in step three in Figure 1. Because of the stretching and gravity, the towel is in flattened state; therefore, the final step consists of placing the towel on the table by sliding it on the edge of the table (shown in steps four and five in Figure 1).

To recognise garment’s *known configurations*, we propose to use an off-the-shelf deep neural network (named KCNet in this paper) based on ResNet-18 [19]. KCNet is mathematically expressed as: $O = F(C(I))$, where O is the output of the KCNet, which is the recognised *known configurations* (represented as classification class), I is the input image that captures *known configurations*, C are convolutional layers (a ResNet-18 [19]), and F are fully connected layers. We use a negative log-likelihood loss (NLLLoss) to train this network: $L(\theta) = \sum_{i=1}^n (y_i \log \hat{y}_{\theta,i} + (1 - y_i) \log(1 - \hat{y}_{\theta,i}))$, where θ is the weight parameters of the network, y_i is the ground truth probability that the i th data point is positive, and $\hat{y}_{\theta,i}$ is the predicted probability that the i th data point is positive. Figure 1 shows how KCNet works in the *known configurations* pipeline.

As stated in section III, a *known configuration* highly depends on the grasping points, therefore we label *known configurations* in terms to its corresponding grasp points. *Known configurations* should be recognisable by KCNet, so they should be distinguishable from each other. Therefore, a *known configuration* represents all grasping points in a segmented area of a garment, rather than a single grasping point. We proposed to discretise garments into different segments and define a grasping point to represent one segment. Our experiments found that dividing a garment into ten segments can provide the most *known configurations* that are recognisable. In each segment, we locate its corresponding grasping point as the centre of the area.

State-of-art focuses on real-time policy updating in robotic deformable object manipulation. A robot will update its manipulation plan after observing a new object state. The updates are computationally intensive and time-consuming. In contrast, we propose designing manipulation plans in advance to avoid updating the manipulation plan to speed up the task execution.

A manipulation plan comprises sequences of 6D poses and the the robot’s gripper state (i.e. open or close). As described in section III, our manipulation strategy include finding the second and third grasping points, stretching to flatten garments and lifting garments down to the table. Our manipulation plan design is thus based on these strategy and ensures that each step requires the least actions. We defined these sequences in CSV files to transfer manipulation commands to the robot after a *known configuration* is recognised.

IV. EXPERIMENTS AND RESULTS

A. Experimental Methodology

Our *known configurations* pipeline consists of three stages: the robot recognises the *known configuration* of a grasped garment, based on the recognised *known configuration*, the

robot chooses a pre-defined manipulation plan, and then the robot flattens the garment.

To recognise garments’ *known configurations*, we captured a novel dataset comprising depth and RGB images of garments from five categories: jeans, shirts, sweaters, towels and tshirts. There are four garment instances in each category, and as described in section III, each garment instance has ten grasping segments represented by ten grasping positions. We collected 100 images for each position, where each image captures the *known configuration* of a garment grasped from a specific position. We captured a total of 19,269 depth and RGB images, respectively, with a resolution of 256×256 pixels. Figure 2 shows examples of images in this dataset. For each garment category, we pre-defined 10 manipulation plans for each the 10 segments as described in section III. Therefore, we have 50 manipulation plans.

With this dataset, we trained a KCNet which is a classification network that consists of a pre-trained ResNet 18 structure and a fully connected network. The detailed implementation of KCNet can be found at https://liduanatglasgow.github.io/known_configurations/. The learning rate is set to 10^{-3} and we allow it to decay during the training with a step scheduler (a decay factor of 0.1 and a step size of 8).

We have implemented a k-fold cross-validation approach (k-fold CVA) to train and test KCNet, rather than the traditional approach of train-validate-test splits. The k value in our experiment is set to four, which means that our database is split into four groups for the k-fold CVA training and testing sessions. There are four garment instances in a group. Three groups are assigned as training groups for each session and one group as a testing group. We ensured that the garment instances in the testing group were ‘unseen’ by KCNet. We iterate the testing group to include all garment instances in a category, and we averaged the classification accuracies as the classification accuracy for the network.

The robot in this experiment is a Baxter dual-arm robot with a table at the front to place the garments. We use a computer with Ubuntu 16.04 and an NVIDIA 1080 Ti GPU to train the KCNet. The robot is controlled by the Robot Operating System, and an Xtion camera captures the images. In section IV-B, we demonstrate an example of a five-step manipulation plan based on the recognised *known configurations* of a towel for flattening the towel. We will improve pre-designed manipulation plans in our future research.

B. KCNet Validation Results and Manipulation Demonstration

We compared the performance of KCNets trained on RGB, depth and RGBD images. Table I shows our results for each category in each k-fold cross-validation described in section IV-A. Table I shows classification accuracies that represent the percentages of *known configurations* recognised by the KCNet.

We found that training a KCNet on depth images (89%) outperforms a KCNet on RGB images (73%). Depth images capture structural and physical characteristics of garments (as found in [20]), and therefore, enables a KCNet to recognise

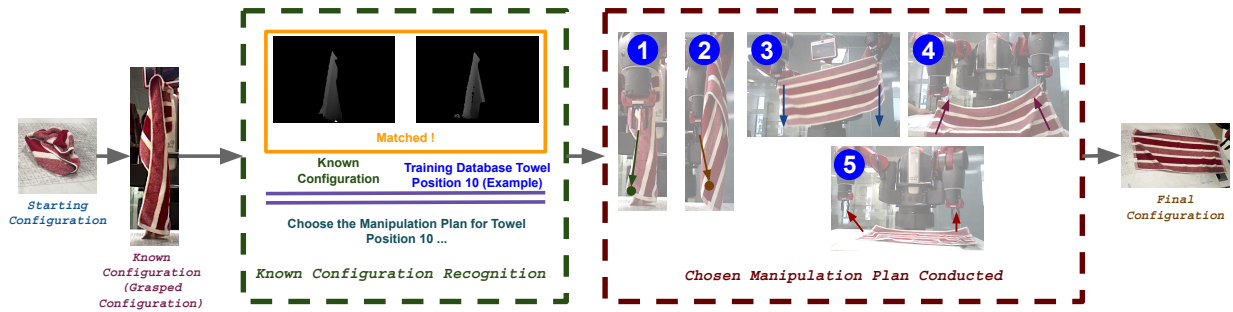


Fig. 1. *Known Configurations Pipeline*: a towel with a crumpled configuration (starting configuration) is grasped by a robot from a grasping point (the first grasping point as described in Section III), giving rise to a *known configuration*. The *known configuration* is recognised, and a pre-designed five-step manipulation plan (described in section III) can be chosen based on the recognised *known configuration*. The robot manipulates the towel with the chosen manipulation plan.



Fig. 2. *Database Examples*: There are five garment categories in our database: jeans, shirts, sweaters, towels and tshirts. Our database contains masked depth and RGB images of four garment instances in each category, which have five categories (Top: original images, Bottom: masked depth images).

the *known configurations* of unseen garment instances. The RGB images capture texture characteristics that depend on the lighting condition and shadows which hinders the ability of KCNet to predict *known configurations*. Meanwhile, We can observe an increase from 73% to 78.5% for the KCNet trained on RGBD images but still a significant gap behind the KCNet trained only on depth images (89.0%). Depth information facilitates a KCNet to recognise the *known configurations* of unseen garments, while RGB information may affect the effectiveness of depth information.

Based on this ablation study, we chose the KCNet trained on the depth images to conduct the manipulation experiments. Figure 3 shows an example of manipulations in our experiments. As described in section III, the robot firstly recognises the *known configurations* of the garments, then chooses a manipulation plan based on the recognised *known configurations*, and finally flattens the garments with the chosen manipulation plan. We can observe that the towel shown in Figure 3 are correctly flattened from its crumpled configuration. A robotic manipulation example can be found on our project website https://liduanatglasgow.github.io/known_configurations/.

V. DISCUSSION

In our experiments, we proposed a robotic garment manipulation by recognising garments' *known configurations* and choosing a pre-designed manipulation plans to flatten

garments. This paper introduces the first part of this pipeline: how to recognise the 'known configurations' of garments. The *known configurations* can be recognised because we took advantage of gravity and reduced deformable objects' complexity. Various and complicated initial configurations are converted into simple *known configurations* by taking advantage of gravity after the robot has grasped the garment.

VI. CONCLUSION AND FUTURE WORK

We proposed an effective robotic garment flattening approach by recognising the *known configurations* of garments and choosing a pre-designed manipulation plan to flatten them. Our approach also features training KCNets on depth images of real garments, resulting in higher recognition accuracy compared with previous work [6].

In our experiments, however, a robot can only recognise the *known configurations* of the garments with shapes from the five categories, while it is unable to recognise those of the garments with an unknown category. Our future work aims to devise a continual learning framework so that a robot learns unknown shapes when it deals with those shapes. Additionally, we believe that a robot can benefit from prior knowledge of shapes [21] or physics properties [20] of garments to recognise *known configurations*, so we propose learning that prior knowledge during the robot's grasping garments and improving recognition accuracy on *known configurations*.

TABLE I
K-FOLD CROSS VALIDATION EXPERIMENT RESULTS: COMPARISON BETWEEN DEPTH, RGB AND RGBD IMAGES (UNIT: %)

| Depth | 1 | 2 | 3 | 4 | RGB | 1 | 2 | 3 | 4 | RGBD | 1 | 2 | 3 | 4 |
|----------------------|------|------|------|------|----------------------|------|------|------|------|----------------------|------|------|------|------|
| towel | 94.4 | 96.4 | 93.1 | 86.2 | towel | 67.0 | 55.9 | 74.1 | 64.5 | towel | 71.8 | 79.7 | 85.2 | 76.7 |
| tshirt | 86.8 | 87.2 | 96.3 | 94.7 | tshirt | 70.8 | 71.2 | 72.1 | 76.9 | tshirt | 74.6 | 68.3 | 84.4 | 83.9 |
| shirt | 78.2 | 80.3 | 75.9 | 92.5 | shirt | 87.9 | 58.4 | 75.4 | 91.4 | shirt | 87.9 | 58.4 | 68.8 | 91.9 |
| sweater | 78.4 | 85.4 | 87.0 | 86.2 | sweater | 54.6 | 42.0 | 68.1 | 85.2 | sweater | 52.7 | 51.6 | 74.7 | 77.0 |
| jean | 99.3 | 95.8 | 95.3 | 99.1 | jean | 76.4 | 87.8 | 87.0 | 98.8 | jean | 87.5 | 92.0 | 97.2 | 99.5 |
| average | 87.0 | 89.0 | 89.0 | 92.0 | average | 73.0 | 62.0 | 75.0 | 83.0 | average | 76.0 | 70.0 | 82.0 | 86.0 |
| AVERAGE: 89.0 | | | | | AVERAGE: 73.0 | | | | | AVERAGE: 78.5 | | | | |



Fig. 3. An Example of Manipulations: an example are demonstrated: firstly, the robot grasps the towel from the table to recognise its *known configurations*. Then a manipulation plan is chosen. Finally, the robot flattens the towel with the chosen manipulation plan.

VII. ACKNOWLEDGEMENT

We thank the members of the Computer Vision and Automation System group from the University of Glasgow for discussing the experimental results and providing experiment suggestions.

REFERENCES

- [1] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, "Learning predictive representations for deformable objects using contrastive estimation," *arXiv preprint arXiv:2003.05436*, 2020.
- [2] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6000–6006.
- [3] D. Agrawal, M. Choy, H. Va Leong, and A. K. Singh, "Maya: A simulation platform for distributed shared memories," in *Proceedings of the Eighth Workshop on Parallel and Distributed Simulation*, ser. PADS '94. New York, NY, USA: Association for Computing Machinery, 1994, p. 151–155. [Online]. Available: <https://doi.org/10.1145/182478.182583>
- [4] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, "Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand," *Robotics and Autonomous Systems*, vol. 95, pp. 196–206, 2017.
- [5] Y. Li, D. Xu, Y. Yue, Y. Wang, S.-F. Chang, E. Grinspun, and P. K. Allen, "Regrasping and unfolding of garments using predictive thin shell modeling," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1382–1388.
- [6] Y. Li, Y. Wang, Y. Yue, D. Xu, M. Case, S.-F. Chang, E. Grinspun, and P. K. Allen, "Model-driven feedforward prediction for manipulation of deformable objects," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1621–1638, 2018.
- [7] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.
- [8] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Robotics Research*. Springer, 2016, pp. 339–354.
- [9] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1405–1412.
- [10] H. Lin, F. Guo, F. Wang, and Y.-B. Jia, "Picking up a soft 3d object by "feeling" the grip," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1361–1384, 2015.
- [11] W. Yu, A. Kapusta, J. Tan, C. C. Kemp, G. Turk, and C. K. Liu, "Haptic simulation for robot-assisted dressing," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6044–6051.
- [12] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.
- [13] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2249–2254.
- [14] D. McConachie, T. Power, P. Mitrano, and D. Berenson, "Learning when to trust a dynamics model for planning in reduced state spaces," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3540–3547, 2020.
- [15] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," *Robotics and Autonomous Systems*, vol. 112, pp. 72–83, 2019.
- [16] D. McConachie and D. Berenson, "Estimating model utility for deformable object manipulation using multiarmed bandit methods," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 967–979, 2018.
- [17] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.
- [18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [20] L. Duan, L. Boyd, and G. Aragon-Camarasa, "Learning physics properties of fabrics and garments with a physics similarity neural network," 2021. [Online]. Available: <https://arxiv.org/abs/2112.10727>
- [21] L. Duan and G. Aragon-Camarasa, "Garnet: A continuous robot vision approach for predicting shapes and visually perceived weights of garments," 2022. [Online]. Available: <https://arxiv.org/abs/2109.07831>