Robotics: Science and Systems 2023
Daegu, Republic of Korea, July 10-July 14, 2023

One Policy to Dress Them All: Learning to Dress People with Diverse Poses and Garments

Yufei Wang, Zhanyi Sun, Zackory Erickson*, David Held*
Robotics Institute, Carnegie Mellon University
{yufeiw2, zhanyis, zerickso, dheld}@andrew.cmu.edu
*Equal Advising

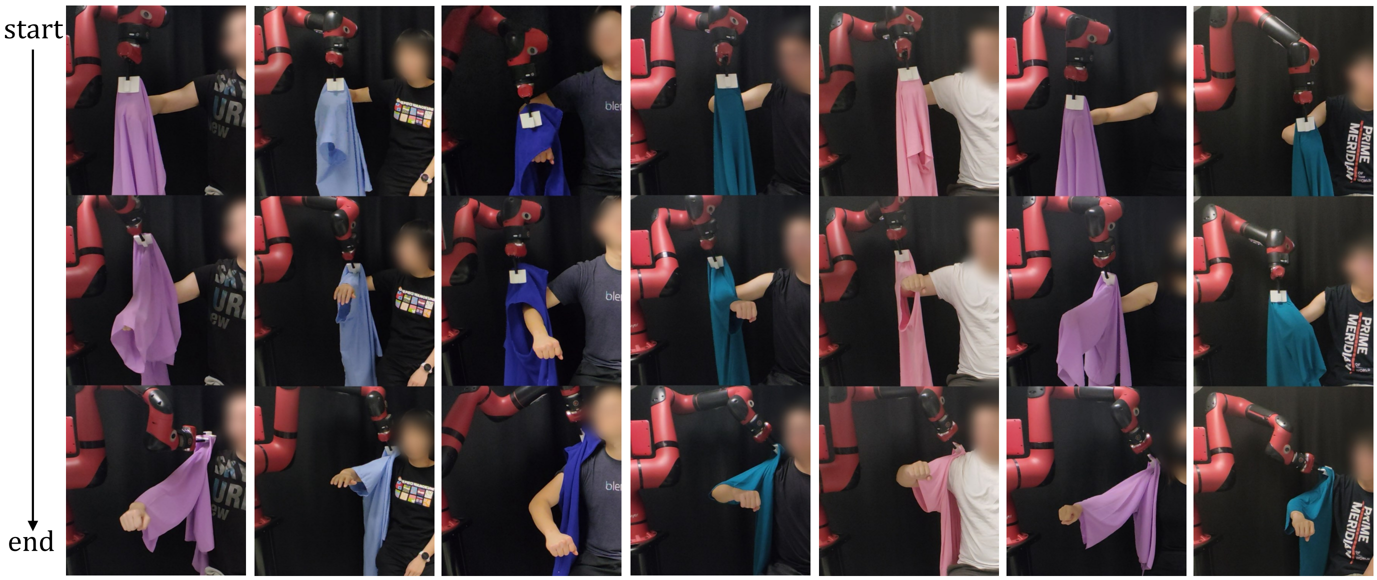

Fig. 1: We develop a robot-assisted dressing system based on a single learned policy that is able to dress different people with diverse poses and garments. Each column shows snapshots from a trajectory of our policy. We learn the dressing policy using reinforcement learning with point cloud observations to generalize to diverse garments. We use policy distillation to combine multiple policies that each work for a small range of arm poses into a single policy that works for a large variety of arm poses.

***Abstract*—Robot-assisted dressing could benefit the lives of many people such as older adults and individuals with disabilities. Despite such potential, robot-assisted dressing remains a challenging task for robotics as it involves complex manipulation of deformable cloth in 3D space. Many prior works aim to solve the robot-assisted dressing task, but they make certain assumptions such as a fixed garment and a fixed arm pose that limit their ability to generalize. In this work, we develop a robot-assisted dressing system that is able to dress different garments on people with diverse poses from partial point cloud observations, based on a learned policy. We show that with proper design of the policy architecture and Q function, reinforcement learning (RL) can be used to learn effective policies with partial point cloud observations that work well for dressing diverse garments. We further leverage policy distillation to combine multiple policies trained on different ranges of human arm poses into a single policy that works over a wide range of different arm poses. We conduct comprehensive real-world evaluations of our system with 510 dressing trials in a human study with 17 participants with different arm poses and dressed garments. Our system is able to dress 86% of the length of the participants' arms on average. Videos can be found on our project webpage [1].**

[1]https://sites.google.com/view/one-policy-dress

I. Introduction

Robot-assisted dressing could benefit the lives of many people. Dressing is a core activity of daily living. A 2016 study by the National Center for Health Statistics [17] shows that 92% of all residents in nursing facilities and at-home care patients require assistance with dressing. Such needs will likely keep growing due to aging populations. A robot-assisted dressing system could not only improve life quality for older adults and individuals with disabilities by helping them maintain independence and privacy, but also help mitigate the growing shortage of nursing staff, and provide some much needed respite for caregivers.

Despite its potential societal impact, robot-assisted dressing remains a challenging task for robotics for the following reasons. As in many deformable object manipulation tasks, there is no compact state representation of cloth, and the dynamics of garments are non-linear and complex [60]. Compared to table-top cloth manipulation tasks such as folding and smoothing that can be solved via pick-and-place actions, the dressing task demands more dexterous manipulation actions in 3D space. Furthermore, people with disabilities and older adults usually have a limited range of motion and thus the dressing robot needs to generalize to a diverse range of arm poses. Finally, there are many different types of garments with varying geometries and material properties, e.g., short-sleeve hospital gowns and long-sleeve jackets. It is non-trivial for a robot to be able to assist in dressing all these garments, since dressing a slender long-sleeve elastic jacket might require very different end-effector trajectories compared with dressing a wide, short-sleeve non-elastic hospital gown. All these factors render robot-assisted dressing a very challenging problem.

There have been many prior works that investigate robot-assisted dressing, yet they make certain assumptions that limit their ability to generalize. Most prior works [10, 23, 13, 11, 20, 59, 12] assume that the robot is dressing a known fixed garment and thus does not generalize to dressing diverse garments. Other prior works simplify the assistive dressing problem by assuming that the person holds a single pose [10, 37], or can move his arm into a pose that has high probability of dressing success [23, 20, 13]. We aim to improve upon these prior works to build a dressing system that can generalize to diverse garments and human poses.

In this work, we learn a single policy that is able to dress diverse garments on different people holding diverse poses from partial point cloud observations using a single depth camera. We use point clouds as input to our policy in order to represent and generalize to different garments and human arm poses. We show that with proper design of the policy and Q-function architectures, reinforcement learning (RL) can be used to learn effective policies with partial point cloud observations that work well for dressing diverse garments. We further leverage policy distillation to combine multiple policies trained on different ranges of human arm poses into a single one that works over a wide variety of different poses. Finally, for robust sim2real transfer, we employ "guided domain randomization", which trains a policy with randomized observations by imitating policies trained without any randomization. The final domain-randomized policy can be reliably transferred to a real robot manipulator and dress real people. We conducted a human study with 17 participants, and performed 510 total dressing trials with different arm poses and garments. On average, our system is able to dress 86% of the length of the participants' arms, and achieves a statistically significant difference in responses as compared to alternative baselines in a 7-point Likert item. We also provide a comprehensive analysis of our dressing system's performance in the human study to understand its strength and weakness.

In summary, we make the following contributions:

- We develop a robot-assisted dressing system based on a learned policy with partial point cloud observations that generalizes to different people, arm poses, and garments.
- We show the effectiveness of policy distillation to increase the effective working range of the policy.
- We perform comprehensive real-world evaluations of our system on a manikin, as well as with 510 dressing trials in a human study with 17 participants of varying arm poses and dressed garments.

II. Related Work

A. Robot Assisted Dressing

There have been many prior works in robot-assisted dressing [57] which can be categorized as follows. Some works make the assumption that the person being dressed is collaborative in performing the dressing task [23, 7], while others do not [10, 13]. We also do not assume a collaborative human in this work. Instead, we simplify the problem by assuming that the human is holding a fixed pose throughout the dressing procedure.

A large body of works focus on user modeling and building a personalized dressing plan for each human participant [23, 58, 13, 12, 4, 21, 59, 26]. In contrast to these prior works, we do not focus on user modeling and aim to learn a single policy that is able to generalize to diverse poses and body sizes of different human participants. We leave the integration of user modeling for future work.

Another line of work studies haptic perception and simulation during robot-assisted dressing [9, 53, 22, 45]. In contrast, we demonstrate how a robot can perform dressing assistance using only partial point clouds from a single off-the-shelf depth camera. Some prior works focus on learning where along a garment to grasp in preparation for dressing [37, 56]. In this work we make the assumption that the garment has already been grasped by the robot, but our method can be combined with these prior work to remove this assumption.

Reinforcement learning has been used in some prior works for dressing backpacks [20] or hospital gowns [7]. However, they assume a fixed garment and thus the policy has limited generalization towards different garments. Our policy can generalize to dressing diverse garments as we use point cloud as the garment representation.

B. Robotic Deformable Object Manipulation

Deformable object manipulation has long been a core task for robotics. It is challenging due to the complicated dynamics of deformables, high-dimensional state representation, and perception complexities such as self-occlusion. Many prior deformable object manipulation tasks such as cloth smoothing [29, 14, 51, 49], cloth folding [46, 1, 25], bedding manipulation [41, 34], dough rolling [28], rope reconfiguration [42, 27], and bag manipulation [2, 42, 6] focus on the 2D table-top setting. Many of these prior cloth manipulation tasks can be solved by simple pick-and-place actions or the use of other pre-defined motion primitives [29, 46, 41, 42, 49, 18, 14, 51]. Our work looks at the challenging problem of dressing assistance, which involves manipulating a deformable garment suspended in 3D space with complex closed-loop actions during physical human-robot interaction.

C. Policy Learning for Manipulation From Point Clouds

There has been much recent work that aims to learn robotic manipulation policies from point cloud observations. Some of these works propose new algorithms for imitation learning from point clouds for grasping and manipulating tools or articulated objects [50, 43, 8, 31]. Some recent works explore applying RL with point cloud observations for grasping or dexterous hand manipulation with rigid objects [44, 38, 30, 19]. Our work differs from these as we apply RL from point cloud observations for deformable cloth manipulation.

III. Task Definition and Assumptions

As shown in Fig. 1, the task we study in this paper is single-arm dressing, where the goal is to fully dress the sleeve of a garment onto a person's arm, and the task is considered to be complete when the sleeve of the garment covers the person's shoulder. The person can hold different arm poses before the dressing starts. The arm pose is defined by three joint angles $\phi = [\phi_1, \phi_2, \phi_3] \in \mathbb{R}^3$, where ϕ_1 is the lifting and lowering angle of the shoulder, ϕ_2 is the inwards-outwards bending angle of the elbow towards the body, and ϕ_3 is the lifting and lowering angle of the elbow (Fig. 5 illustrates these joint angles). A depth camera is used to record depth images of the scene, and partial point clouds P can be computed from the depth image. The ultimate goal of the paper is to learn a policy π to dress diverse garments $g \in G$ for people holding a diverse range of arm poses $\phi \in \Phi$. At each time step, the policy takes as input the point cloud P and outputs an action $a = \pi(P)$ as the delta transformation for the robot end-effector. The garment set G we consider includes hospital gowns and common everyday garments such as vests and cardigans with different sleeve lengths, geometries, and materials. The set of arm poses Φ is specified by the min and max values for each joint angle: $\Phi = \{[\phi_1, \phi_2, \phi_3] \mid \phi_i \in [\phi_i^{min}, \phi_i^{max}], i = 1, 2, 3\}$. Fig. 5 shows the garments and arm poses in Φ that we test in the real-world human study.

We make two assumptions for the dressing task. First, we assume that the robot has already grasped a part of the garment around the opening of the garment shoulder in preparation for dressing, since grasping is not the focus of our work. Besides, there has been some prior work that learn where along a garment to grasp for dressing [37, 56], and our system can be combined with these prior work to remove this assumption. This assumption has been used in prior work as well [10, 23]. Second, we assume that the person holds the pose static during the dressing process. This assumption helps address the visual occlusion of the arm caused by the cloth during the dressing process; with this assumption we can obtain the static arm point cloud before the dressing starts. This assumption has also been validated in other studies with participants with impairments [23]. We leave for future work adapting to human motion during the dressing process.

IV. Method

A. System Overview

As illustrated in Fig. 2, our robot-assisted dressing system consists of the following components. We first use reinforcement learning (RL) to learn the dressing policy π in a physics-based simulation by formulating the robot assisted dressing problem as a Partially Observable Markov Decision Process (POMDP). We design a special policy network architecture for efficient training of effective RL policies from partial point cloud observations. As it is hard to train a single policy that generalizes to a diverse range of arm poses Φ, we divide the arm pose range into multiple sub-ranges $\{\Phi^{sub}\}_{i=1}^N$ and train a policy π_i on each of them. We then use policy distillation to combine these different policies into a single policy π^* that works for the wide range of arm poses Φ. For robust sim2real transfer, we further train the policy π^* with domain randomization in the policy observation, with a behaviour cloning loss to imitate policies that are trained without any randomization, a procedure which we name as "guided domain randomization". Finally, we deploy the domain-randomized policy to a real robot that successfully dresses different participants with diverse poses and garments in a human study.

B. Learning to Dress with Reinforcement Learning

We use reinforcement learning in simulation to train policies for the dressing task. We formulate the robot-assisted dressing task as a POMDP $\langle S, A, O, R, T, U \rangle$, where S is the state space, A is the action space, O is the observation space, R is the reward function, T is the transition dynamics, and U is the measurement function that generates the observation from the state. We now detail the design of the core components of the POMDP as follows.

Observation Space O: Due to the lack of a compact state representation for cloth, the garment naturally requires a high-dimensional representation such as an image, mesh, or point cloud. To facilitate sim2real transfer, we use the partial point cloud P of the dressing scene computed from a depth camera as the policy observation. We perform cropping on the point cloud to only keep points that belong to the right arm of the human, denoted as P^h, and those that belong to the garment, denoted as P^g. In simulation we can easily perform such cropping using the ground-truth simulator information; see Section VI-C for details on how we perform such cropping in the real world. We further add a single point at the position of the robot end-effector to the point cloud to represent the robot gripper, denoted as P^r. The policy observation is then the concatenation of the arm, garment, and robot gripper points: $o = [P^h; P^g; P^r]$. See the middle part of Fig. 2 for an example of the segmented point cloud.

Action Space A**:** The action $a \in SE(3)$ is the delta transformation of the robot end-effector. We represent the delta transformation as a 6D vector, where the first 3 components are the delta translation, and the second 3 components describe the delta rotation using axis-angle. We set the roll rotation to be 0 in the action as it is not necessary for the dressing task.

Reward Function R: Fig. 3 illustrates the main reward r_m we use for the dressing task, which measures the progress of the task. The dressing task can be divided to two phases. Before the garment is dressed onto the person's forearm, the reward r_m is the negative distance from the garment shoulder opening center p^g_{center} to the finger of the person p^h_{finger}: $r_m = -||p^g_{center} - p^h_{finger}||_2$. After the garment is dressed onto the person's forearm, we compute the reward r_m based on the distance the garment has been dressed onto the person's arm. To compute the dressed distance, we first

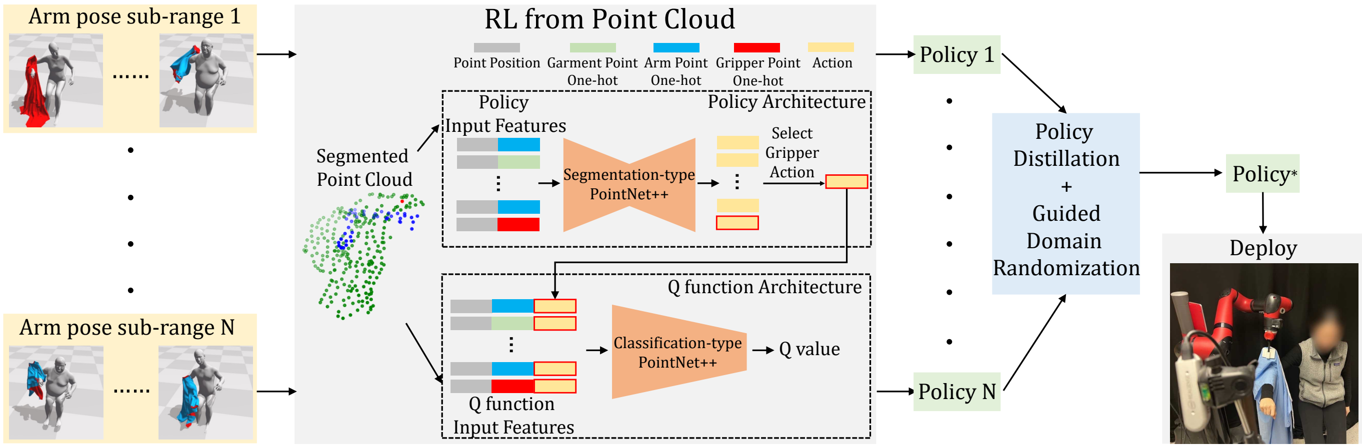

Fig. 2: An overview of our proposed robot-assisted dressing system. (Left) In simulation, we divide the diverse arm pose range into multiple sub-ranges to ease policy learning. (Middle) We propose a new policy architecture and a corresponding Q function architecture for reinforcement learning from partial point cloud observations to learn effective dressing policies on each of the divided pose sub-ranges. (Right) We then leverage policy distillation to combine policies working on different pose sub-ranges into a single policy that works for a diverse range of poses. We also perform guided domain randomization for sim2real transfer, and we deploy the distilled policy to a real-world human study dressing real people.

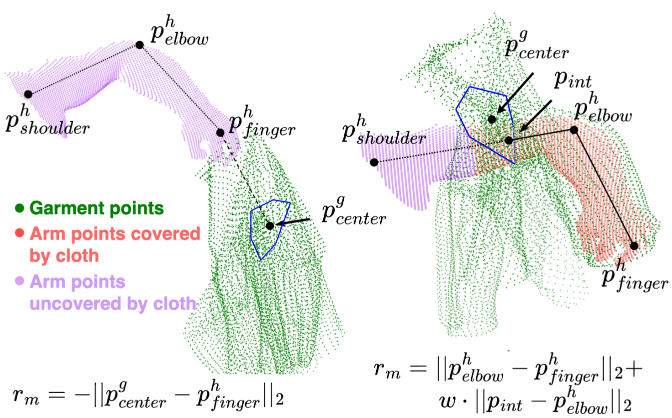

Fig. 3: Illustration of the main reward for the dressing task. (Left) The garment is not dressed onto the person's arm. (Right) The garment is dressed onto the person's arm.

approximate the opening of the garment shoulder as a hexagon (shown in blue in Fig. 3). Then, we represent the person's forearm as a line that connects the elbow point p^h_{elbow} and the finger point p^h_{finger}, and we represent the upper arm as another line that connects the shoulder point $p^h_{shoulder}$ and the elbow point p^h_{elbow}. Next, the intersection point p_{int} between the garment shoulder opening hexagon and the two arm lines are computed. If the intersection point p_{int} is on the forearm, r_m equals the dressed distance, which is the distance between the intersection point and the person's finger point: $r_m = ||p_{int} - p^h_{finger}||_2$. If the intersection point p_{int} is on the upper arm, the dressed distance is the length of the forearm plus the distance between the intersection point and the elbow point, and r_m is a weighted combination of these two: $r_m = ||p^h_{elbow} - p^h_{finger}||_2 + w \cdot ||p_{int} - p^h_{elbow}||_2$. We set $w = 5$ to encourage the policy to turn at the elbow and dress the upper arm.

In addition to the main reward term that measures the task progression, we also have three additional reward terms. The first is a force penalty r_f that prevents the robot from applying too much force through the garment to the person. The second is a contact penalty r_c that prevents the robot end-effector from moving too close to the person. The last reward term is a deviation penalty r_d that discourages the garment center from moving too far away from the arm. The full reward is given by: $r = r_m + r_f + r_c + r_d$. More details of how these terms are computed can be found in Appendix Section A. Note that the reward function is only available in simulation, as it requires access to the ground-truth garment and human mesh information, which is non-trivial to estimate in the real world.

Policy and Q function Architecture: Most prior works [30, 38, 44] that train RL policies with point cloud observation use a classification-type PointNet-like [35, 36] network architecture for the policy, which encodes the whole partial point cloud to a single action vector. There have been some recent works showing that instead of compressing the whole point cloud into a single action vector, inferring the action from a dense output leads to better performance [54, 47, 55, 48, 43, 8, 14, 5]. We follow the dense action representation idea and propose a new policy architecture named *Dense Transformation* for reinforcement learning from point clouds.

As shown in Fig. 2, the input to the policy is a segmented point cloud $P = \{p_i\}_{i=1}^M$ of size M, which contains the garment points P^g, human right arm points P^h, and a single point at the robot end-effector position representing the robot gripper P^r. The features for each point p_i include its 3D position, and a 3-dimensional one-hot vector indicating the class of the point, i.e., whether the point belongs to the garment, the human arm, or the robot gripper. Instead of using a classification-type neural network architecture (e.g., a classification-type PointNet++) that compresses the whole point cloud into a single action vector, the Dense Transformation policy uses a segmentation-type neural network architecture (e.g., a segmentation-type PointNet++) that outputs a dense per-point action vector $\{a_i\}_{i=1}^M$ (see Fig. 2). Among these M action vectors, we only execute the action vector a^* corresponding to the gripper point P^r, i.e, we select the action $a^* = a_j$, where j is the index of the gripper point. We could alternatively use a classification-type PointNet++ that encodes the whole point cloud to a single action, but we find that using a segmentation-type network leads to slightly better performance (though the difference is relatively small).

For the Q function, given the current point cloud P and the action sampled from the Dense Transformation policy $a^* \sim \pi(P)$, we concatenate a^* as an additional feature to every point p_i in the input point cloud, so the feature of each point includes its 3D position, the one-hot vector of the class type, and the action a^*. We then use a classification-type neural network architecture (i.e., classification-type PointNet++) to output a scalar Q value. This Q function architecture has also been used in prior work [44], although with a different policy architecture. We compare to other ways of representing the Q function in experiments and show that this one works the best. We use SAC [15] as the underlying RL algorithm, and we use PointNet++ [36] for the policy and Q function network.

C. Generalization to Diverse Human Poses via Policy Distillation

Learning a dressing policy that works on a diverse range of arm poses can be viewed as a multi-task learning problem, where each small range of poses can be considered as an individual task. The same holds true for learning a policy that works for diverse garments – each garment can be treated as an individual task. In our experiments, we find it is possible to learn a single universal policy for a diverse set of garments, and the performance is similar to that of learning an individual policy for each garment. However, we find it challenging to learn a single policy that works well for a diverse range of arm poses, possibly due to imbalanced learning speed for different tasks (e.g., some poses are easier to learn compared with others), conflicting gradients from different tasks (the desired trajectory for one arm pose might contradict to another), and other potential issues. Inspired by [39, 40], we use policy distillation to address this issue.

Policy Distillation. Given a set of policies that were each trained for a single task, policy distillation [39] can be used to combine the set of policies into a single policy that works for all of the tasks. It has been shown that this often outperforms directly training a single policy for all of the tasks. In our case, we train individual policies each for dressing a human arm pose sub-range; we then distill these policies into a single policy that works for the full diverse range of arm poses Φ.

To employ policy distillation, we first need to decompose the diverse arm pose range Φ into smaller ranges $\{\Phi_i^{sub}\}_{i=1}^N$ where RL can be directly used to learn effective policies on these smaller pose ranges. As aforementioned, the arm pose range Φ is specified by min and max values for each of the three joint angles $\Phi = \{[\phi_1, \phi_2, \phi_3] \mid \phi_i \in [\phi_i^{min}, \phi_i^{max}], i = 1, 2, 3\}$. We perform the decomposition by dividing each joint angle range $[\phi_i^{min}, \phi_i^{max}]$ into a number of smaller intervals. For example, we can uniformly divide the range $[\phi_i^{min}, \phi_i^{max}]$ to $\Phi_i^j = [\phi_i^{min} + (j-1)\delta, \phi_i^{min} + j\delta], j = 1, ..., L$, where $\delta = (\phi_i^{max} - \phi_i^{min})/L$. This will then result in L^3 sub-ranges $\{\Phi^{sub} = \Phi_1^j \times \Phi_2^k \times \Phi_3^l\}_{j=1,k=1,l=1}^L$.

After we decompose the diverse arm pose range Φ into N smaller ranges, we train N "teacher" policies $\{\pi_i^t\}_{i=1}^N$, one for each pose sub-range, using RL as described in Section IV-B. We then distill these N teacher policies $\{\pi_i^t\}_{i=1}^N$ into a single "student" policy π_s by training the student policy with a combination of the RL loss and a policy distillation loss, shown in Eq. 1.

Specifically, let θ_s denote the parameters of the student policy π_s, let θ_t^i denote the parameters of the i^{th} teacher policy π_i^t, and let $\mathcal{N}(\mu_\theta(o), \sigma_\theta(o)) = \pi_\theta(o)$ denote the Gaussian distribution of the action output by policy π_θ on observation o. Given a batch of samples collected by the student policy π_s, $\{o_n, a_n, r_n, o_n'\}_{n=1}^B$, we use the following loss to train π_s:

$$L(\theta_s) = L_{SAC}(\theta_s) + \beta \sum_{i=1}^N L_{distill}(\theta_s, \theta_t^i), \tag{1}$$

where $L_{SAC}(\theta_s)$ is the standard SAC training loss, $L_{distill}$ is the policy distillation loss (described below), and β weighs the two terms. The policy distillation loss computes the distance between the student action distribution and the teacher action distribution, and thus the student learns to imitate the teacher's behaviors by minimizing such a loss. Specifically, we compute the Earth Mover's distance between the action distribution of the student and the teacher policy:

$$L_{distill}(\theta_s, \theta_t^i) = \sum_{n=1}^B \left(\mu_{\theta_s}(o_n) - \mu_{\theta_t^i}(o_n)\right)^2 + \left(\sqrt{\sigma_{\theta_s}(o_n)} - \sqrt{\sigma_{\theta_t^i}(o_n)}\right)^2 \tag{2}$$

The loss in Eq. 1 has also been used in prior work [40]; however, they used the KL divergence instead of Earth Mover's distance for the distillation loss. Our experiments indicate that earth-mover distance leads to significantly improved performance. We set $\beta = 0.01$ in our experiments.

D. Guided Domain Randomization Learning for Sim2real Transfer

We find that there is a huge difference between the simulated garment point cloud and the real-world garment point cloud, due to large variations in simulation vs. real garment geometries, and also since the simulator does not perfectly model the dynamics of garment deformations in the real world. To make the observation more aligned between simulation and the real world, and to make the policy robust to the observation difference, we add randomizations to the policy observation (described at the end of this section). Let o denote the original non-randomized point cloud observation, and let $\tilde{o}$ denote the randomized observation. Naively training the policy with randomized observation $\tilde{o}$ will usually fail or lead to degraded performance, since the randomizations make policy learning more difficult. To mitigate this issue, we propose "guided domain randomization", where we first train teacher policies π_i^t without any domain randomization; we then distill the teachers into a student policy π_s trained with domain randomization on the observation. Let θ_s denote the parameters of the observation randomized policy π_s. To train the student policy π_s, we run the student to collect a batch of data that stores both the randomized and non-randomized observation $\{\tilde{o}_n, o_n, a_n, r_n, \tilde{o}'_n, o'_n\}_{n=1}^{B}$, and train it as follow:

$$L(\theta_s) = \tilde{L}_{SAC}(\theta_s) + \beta \sum_{i=1}^{N} \tilde{L}_{distill}(\theta_s, \theta_t^i),$$

$$\tilde{L}_{distill}(\theta_s, \theta_t^i) = \sum_{n=1}^{B} \left(\mu_{\theta_s}(\tilde{o}_n) - \mu_{\theta_t^i}(o_n)\right)^2 + \left(\sqrt{\sigma_{\theta_s}(\tilde{o}_n)} - \sqrt{\sigma_{\theta_t^i}(o_n)}\right)^2,$$

where $\tilde{L}_{SAC}(\theta_s)$ is the SAC loss with running the policy π_s on the randomized observations, and $\tilde{L}_{distill}(\theta_s, \theta_t^i)$ is the loss of imitating the teacher policy π_i^t trained without domain randomization. Importantly, note that the student receives the randomized observation $\tilde{o}_n$ whereas the teacher receives the non-randomized observation o_n. For the observation randomization, we perform random cropping, dropping, erosion, and dilation on the cloth point cloud, and we add random noise to the robot gripper position. More details on the observation randomization can be found in Appendix Section A.

V. Simulation Experiments

A. Experimental Setup

We train our policies using Softgym [27] based on the NVIDIA Flex simulator. We use SMPL-X [33] to generate human meshes of different body sizes and arm poses. To generate random poses, the shoulder joint ϕ_1 is uniformly sampled from $[-20, 30]$, the inwards-outwards elbow joint ϕ_2 is uniformly sampled from $[-20, 20]$, and the upwards-downwards elbow joint ϕ_3 is uniformly sampled from $[-20, 30]$. We decompose the arm pose range Φ into $N = 27$ regions for policy distillation, by dividing each joint angle range into 3 intervals. We randomly pick 5 garments for training from the Cloth3D dataset [3]: a hospital gown and 4 cardigans. The selected garments have different geometries such as varying sleeve lengths and widths. We randomly generate 50 human poses for each of the 27 pose sub-ranges, resulting in a total of $27 \cdot 50 \cdot 5 = 6750$ different configurations. Among the 50 poses for each sub-range, we use 45 poses for training and 5 poses for evaluation. At each training episode of SAC, we randomly sample one garment out of the five for training. The simulated dressing environment with the person holding different poses and with different garments is shown in Fig. 2. More details of the simulation experimental setup can be found in Appendix Section B.

B. Baselines and Ablations

We compare the following RL algorithms for learning the dressing policy from partial point cloud observations: **Dense Transformation policy (ours)** is our proposed method, which is described in Section IV. **Dense Transformation policy + latent Q function:** This baseline uses the same policy as our method. For the Q function, instead of concatenating the action to the input point cloud as an additional feature, this baseline first uses a PointNet++ to encode the point cloud observation into a latent vector, and then concatenates the action with the latent vector. An MLP then takes as input the concatenated latent vector and outputs a scalar Q value. **Direct Vector** uses a classification PointNet++ policy to encode the input point cloud to a single action vector. It uses a similar Q function network architecture as our method, where the action is concatenated to every point in the point cloud as an additional feature. **TD-MPC [16]**, a state-of-the-art model based RL algorithm. **Deep Haptic MPC** [10], which learns a force prediction model based on end-effector measurements, and combines it with MPC to plan forward actions that minimizes the predicted force during dressing. The actions in MPC are sampled based on a "moving forward" heuristic. Note that this method does not use any visual information of the arm or the garment.

	Dense Transform (Ours)	Dense Transform Latent Q	Direct Vector	TD-MPC	Deep Haptic MPC
1	$\mathbf{0.68 \pm 0.05}$	0.55 ± 0.05	0.63 ± 0.07	0.04 ± 0.04	0.31 ± 0.08
2	$\mathbf{0.55 \pm 0.06}$	0.42 ± 0.08	0.52 ± 0.09	0.00 ± 0.00	0.11 ± 0.02
3	$\mathbf{0.75 \pm 0.04}$	0.68 ± 0.07	0.73 ± 0.02	0.03 ± 0.05	0.37 ± 0.02

TABLE I: Upper arm dressed ratio of different policies on 3 pose sub-ranges. Results are averaged across 3 seeds.

Policy Distillation (Ours)	Policy Distillation (KL)	PCGrad	No Distillation	Heuristic Motion Planning
$\mathbf{0.68 \pm 0.012}$	0.45 ± 0.010	0.37 ± 0.063	0.34 ± 0.10	0.32

TABLE II: Upper arm dressed ratio of different ways to enable the policy to generalize to diverse poses. Results are averaged across 3 seeds. The Heuristic Motion Planning baseline does not have a standard deviation because there is no learning or randomness in this method.

We also compare different ways of enabling the policy to generalize to diverse poses. **Policy Distillation** is our proposed method, as described in Section IV-C. **Policy Distillation (KL)** replaces the Earth Mover's distance in Equation (2) with KL divergence. **PCGrad [52]** is a multi-task learning algorithm that balances gradients from different tasks; in our case, a "task" is defined as training on a different pose sub-range. **No Distillation** trains a single policy on the entire pose range, without any distillation. **Heuristic Motion Planning** finds a collision-free robot end-effector path along the human arm based on some heuristically designed constraints. More implementation details of these baselines can be found in Appendix Section B.

In simulation, we use the *upper arm dressed ratio* as the evaluation metric. The ratio is computed between the dressed upper arm distance and the actual upper arm length: $\frac{||p_{int} - p^h_{elbow}||_2}{||p^h_{shoulder} - p^h_{elbow}||_2}$; see Section IV-B and Fig. 3 for how these points are defined. This ratio is upper bounded by 1.

C. Does the Dense Transformation policy perform better than the alternative baselines for learning the dressing policy?

We first compare the performance of different RL algorithms. We randomly select 3 different sub pose-ranges, and for each method, we train a separate policy on these 3 pose sub-ranges to compare their performances. We train each method for 1e6 steps, evaluate at each 10K steps, and report

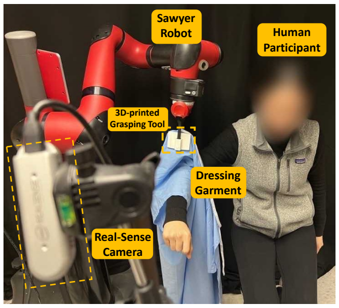

Fig. 4: Real-world human study setup.

the maximum performance. The results are shown in Table I and averaged across 3 seeds. As shown, our proposed method Dense Transformation policy achieves the best performances on all 3 pose sub-regions, although Direct Vector has similar performance. Using a "Latent Q" architecture leads to much worse performance. We find that TD-MPC [16] does not work at all for this dressing task, possibly since the algorithm is originally designed and tested on image and state observations instead of point clouds, and also since it might be hard to learn a good latent dynamics model for deformable garments. Deep Haptic MPC also does not work well. The original paper tested only one arm pose and garment, with no visual information of the garment or human pose, potentially explaining its low performance in our setting with diverse arm poses and garments. Based on this result, we use Dense Transformation as our base RL algorithm for the following experiments.

D. Does policy distillation improve the ability of our method to generalize to diverse poses?

We now compare the performance of different methods for learning a single policy on all 27 pose sub-ranges. We train all methods long enough until they converge (as different methods require different numbers of environment steps to converge), and report the maximum achieved performance. Results are averaged across 3 seeds. As shown in Table II, policy distillation with the Earth Mover's distance outperforms all other baselines. The baseline "No Distillation" and "PC-Grad [52]" both perform poorly, showing the difficulty of directly learning a single policy across diverse pose ranges without distillation. The performance of the "Heuristic Motion Planning" baseline is also low. We find it performs well when there is no sharp bending at the shoulder and the elbow, and worse when the bends are sharp, aligning with findings in prior work [23]. Interestingly, we find that using Earth Mover's distance noticeably outperforms KL divergence, even though KL divergence seems to be the most common choice for policy distillation in the literature [39, 40, 32].

VI. Real-world Experiments and Human Study

We perform guided domain randomization (Section IV-D) to obtain a distilled policy that is robust for sim2real transfer, and deploy it in the real world, both for dressing a manikin and in a real-world human study. Our real-world experiments aim to answer the following questions: (1) Does our distilled policy outperform the baseline in the real world? (2) Does our policy generalize to different people with diverse poses and different garments?

Fig. 5: Arm poses and garments used for the human study.

A. Setup

Fig. 4 shows the real-world setup that we use for the human study, and Fig. 6 shows the manikin setup. Fig. 5 shows the arm poses and dressing garments we test. The poses we test in the real world span the pose ranges we train in simulation. We test 5 garments: a sleeveless, non-elastic vest; a short-sleeve, non-elastic hospital gown; a short-sleeve, elastic pink cardigan; a medium-length sleeve, narrow opening, elastic green cardigan; and a long-sleeve, non-elastic purple cardigan. Note that the real-world garments are not calibrated to the training garments in simulation; for example, the policy is never trained on a sleeveless vest in simulation. We use a Sawyer robot to execute the dressing task and an Intel RealSense D435i camera to capture the depth images.

We use the following quantitative metrics to measure the performance of a dressing trial. **Dressing Success:** We put a marker on the participant's shoulder, at the position that is 80% of the length up the upper arm. If the marker is covered by the garment at the end of the dressing trial, we consider this dressing trial is a success; otherwise it is not. **Upper Arm Dressed Ratio:** We compute the ratio between the "dressed upper arm distance" and the length of the participant's upper arm. The "dressed upper arm distance" is measured as the distance from the elbow to the intersection point of the garment and the participant's upper arm (see Fig. 3). **Whole Arm Dressed Ratio:** We compute the ratio between the "dressed whole arm distance" (the upper arm dressed distance + the forearm dressed distance) and the length of the participant's whole arm (upper arm length + forearm length).

B. Comparison on a Manikin

In order to perform a strictly controlled comparison of the Distilled Policy and the No Distillation baseline, we first conduct experiments on a manikin shown in Fig. 6. The manikin has a fixed pose similar to pose 1 in Fig. 5. We conduct 10 dressing trials per garment (across 5 garments),

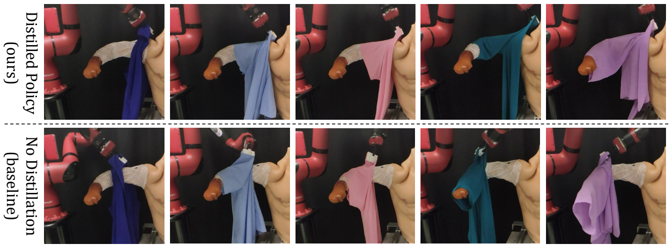

Fig. 6: Final state comparison of the Distilled Policy (Ours, top) and No Distillation baseline (bottom) on the manikin.

						Total
Distilled Policy (Ours)	10/10	10/10	10/10	10/10	10/10	50/50
No Distillation	0/10	0/10	0/10	0/10	0/10	0/50

TABLE III: Dressing success rate of the Distilled and No Distillation policies on 5 garments on the manikin.

resulting in 50 trials for each method. The numerical results are shown in Table III and Fig. 7. As shown, the Distilled Policy performs vastly better than the No Distillation baseline on all metrics, which resembles performance in simulation. The Distilled Policy successfully covered the marker on the shoulder every trial for all the garments. Fig. 6 shows the final state achieved by the Distilled Policy and the No Distillation baseline. For most of the trials, the No Distillation policy can only dress the forearm, and it has trouble correctly turning at the elbow to dress the upper arm.

C. Human Study Procedure

We recruit 17 participants in the human study, including 6 females and 11 males. The age of the participants ranges from 19 to 29. For each participant, we conduct 6 trials for each of the 5 garments, totaling 30 trials. Among the 6 trials per garment, we perform 5 trials using our distilled policy, corresponding to the 5 human poses shown in Fig. 5. To compare against the baseline, we perform the remaining trial using the No Distillation baseline on a pose randomly chosen from the 5 poses. We test the No Distillation baseline with only one pose for each garment as we find that most participants experience arm soreness after 30 trials; thus it is impractical to

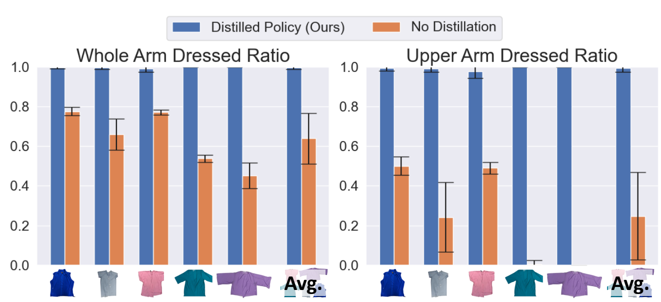

Fig. 7: The whole arm (left) and upper arm (right) dressed ratio of the Distilled Policy and the No Distillation baseline on 5 garments on the manikin, averaged across 10 trials for each garment.

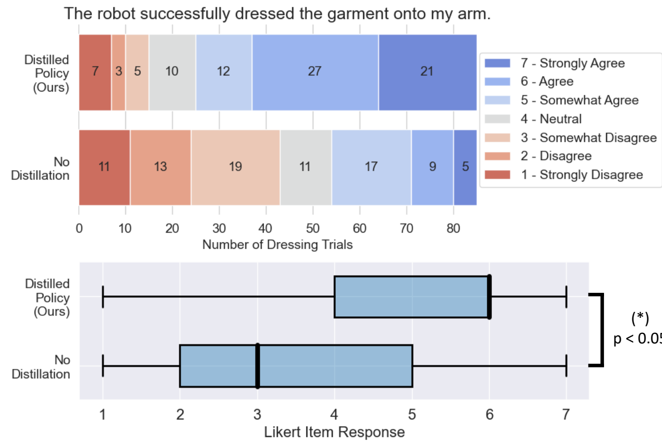

Fig. 8: Likert item responses of the Distilled Policy and the No Distillation baseline, on the 85 dressing trials for which both methods are evaluated on the same poses and garments, shown as a full distribution (top) or box plot (bottom). We perform the Wilcoxon signed rank test on the Likert item response and find a statistically significant difference ($p < 0.05$) between our distilled policy and the baseline.

	Whole Arm Dressed Ratio	Upper Arm Dressed Ratio	Success Rate
Distilled Policy (Ours)	$\mathbf{0.86 \pm 0.19}$	$\mathbf{0.74 \pm 0.35}$	$\mathbf{0.66 \pm 0.47}$
No Distillation	0.63 ± 0.15	0.26 ± 0.27	0.01 ± 0.11

TABLE IV: Comparison of the Distilled Policy and the No Distillation baseline in a real-world human study. Results are averaged over 17 participants, on the 85 trials for which both methods are evaluated on the same poses and garments.

test all poses with the baseline policy. Furthermore, since the No Distillation baseline shows poor performance on a static manikin as shown in Table III, it is less likely that it would work on real humans with diverse arm shapes and poses. We randomize the test order of the garments, the poses, and the policy for each participant. During the study, the participant is not aware of which policy we are testing.

The procedure of each trial is as follows: We first show the participant the pose they should imitate, and they lift their right arm to maintain the pose. We then capture a point cloud of the participant's right arm. We move the Sawyer's end-effector, which is already holding the garment, to be positioned near the participant's hand. We then capture a point cloud of only the garment using color thresholding. The color-based segmentation of the garment could be replaced by training a garment segmentation network, such as fine-tuning an existing segmentation network with a small amount of data in our setting [24]. The point clouds of the human arm (recorded statically before the trial), the garment (color thresholded at each timestep), and the end-effector position (obtained using forward kinematics of the robot) are concatenated as input to the policy. The participant holds their arm steady throughout the trial. The trial terminates if any of the following conditions holds true: (1) if a maximum time step of 75 is reached, (2) if

Whole Arm Dressed Ratio	Upper Arm Dressed Ratio	Success Rate
0.86 ± 0.17	0.71 ± 0.34	0.57 ± 0.49

TABLE V: Performance of our distilled policy, averaged over 17 participants and all 425 dressing trials (not just the 85 trial subset in Table IV).

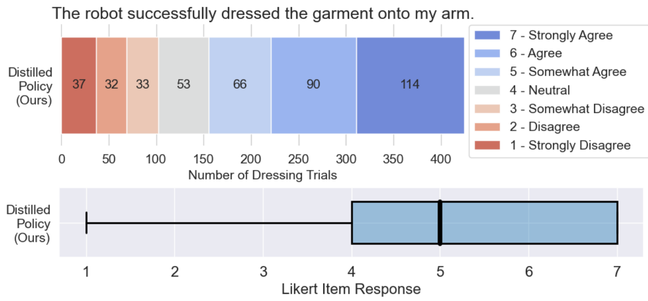

Fig. 9: (Top) Full distribution and (Bottom) box plot of the Likert item responses of the Distilled Policy on all 425 dressing trials (not just the 85 trial subset shown in Fig. 8).

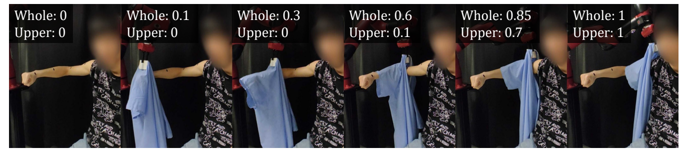

Fig. 10: An illustration of the dressed ratios during a trial.

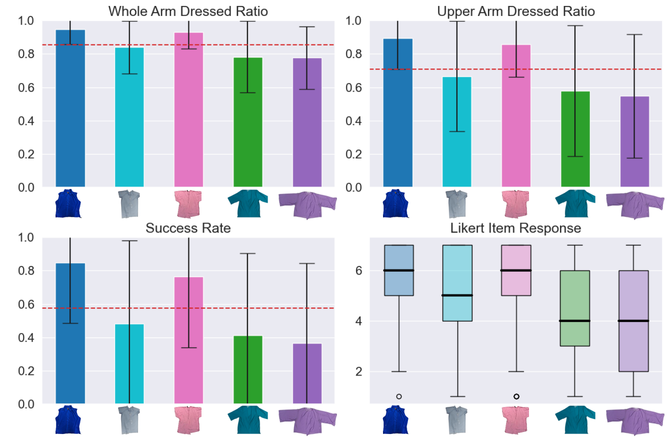

Fig. 11: Performance of our distilled policy on different garments. The dashed red line shows the average over all garments.

the participant's shoulder is covered by the garment, (3) if the policy is not making any progress in 15 consecutive time steps, and (4) if the participant wishes to have trial stopped. After the trial terminates, we measure dressing distances to compute our evaluation metrics. Finally, at the end of each trial, we provide the participant with a single 7-point Likert item statement "The robot successfully dressed the garment onto my arm", which ranges from 1='Strongly Disagree' to 7='Strongly Agree'. More details on the human study procedure can be found in Appendix Section C.

D. Human Study Results and Analysis

We first compare the performance of the Distilled Policy against the No Distillation baseline in Table IV. The comparison is made on the 85 dressing trials for which both methods are evaluated on the same poses and garments. As shown, the Distilled Policy outperforms the No Distillation baseline by a large margin under all metrics. Fig. 8 shows the Likert item response distributions of our distilled policy versus the baseline. Overall, participants agreed at higher rates that the distilled policy successfully dressed the garment, as compared to the baseline. On average, our distilled policy achieves a median response of 6.0, meaning that the participants "Agree" that the robot successfully dressed the garment onto their arm, while the No Distillation baseline achieves a median response of 3.0, meaning the participants "Somewhat disagree." Let d be the difference between the Distilled Policy's Likert item response and the No Distillation baseline's Likert item response. We perform the Wilcoxon signed rank test to test if the distribution of d is stochastically greater than a distribution symmetric about zero. We obtain a p-value of $p = 0.03125$; hence we find a statistically significant difference ($p < 0.05$) between our distilled policy and the baseline, and the median of the difference is positive.

We now analyze our policy's performance over all 425 dressing trials from the 17 participants. Table V shows the Distilled Policy's performance under all metrics. On average, our policy is able to dress 86% of the participant's whole arm, and 71% of the participant's upper arm, achieving a final dressing state similar to the 5th sub-image in Fig. 10. Fig. 1 shows snapshots of the dressing trajectories of our policy. In terms of Likert item responses, as shown in Fig. 9, our distilled policy achieves a median response of 5.0 ("Somewhat Agree" that the robot successfully dressed the garment onto the arm). Note that the metrics and Likert item responses are different from those in Table IV and Fig. 8 because here the numbers are averaged across all dressing trials, as opposed to the 85 trial subset used for the baseline comparison.

Fig. 11 shows the performance of our method across different garments. It is interesting to see a consistent ordering of the garments under all metrics. Such ordering aligns with human intuitions based on the garment geometry & materials (see Fig. 5): the vest is sleeveless and thus the easiest to perceive and dress (even though the sleeveless vest has never been trained in simulation). The green and purple cardigans, however, have longer sleeves and are harder in terms of both perception and dressing. Meanwhile, although the pink cardigan and the gown have similar sleeve length, the material of the pink cardigan is much more elastic than the hospital gown. This provides some allowance in the end-effector trajectories during the dressing process, leading to better performance. This also holds true for the purple versus green cardigan – although the opening of the purple cardigan is larger, it is less elastic, which results in the garment getting caught on the participants' arm more frequently, leading to a failure.

Finally, we analyze the performance of our distilled policy on different poses in Fig. 12. We notice that some poses are harder than others. For example, the fifth pose with sharp

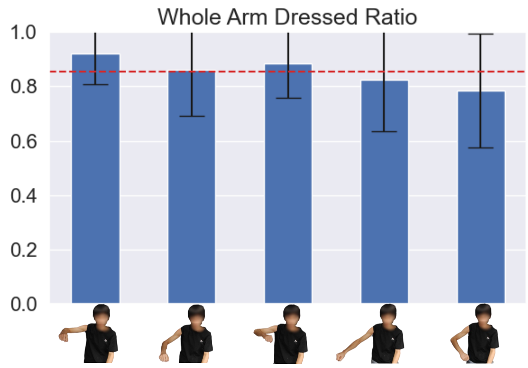

Fig. 12: Whole arm dressed ratio of our distilled policy on different poses. The dashed red line represents the average over all poses. Other metrics can be found in Appendix Section C.

bends at the shoulder and elbow has the lowest performance. On the other hand, the first pose, which does not have a sharp bend at either shoulder nor elbow, achieves the highest performance. Such a difference is likely due to the fact that sharp bends require more complex end-effector trajectories, and the garment is also more likely to get caught at sharp bends. We also note that although the first pose in the human study is similar to the pose of the manikin in Fig. 6, the success rate on this pose in the human study is lower than the one achieved on the manikin shown in Table III. We speculate that such a difference could be due to the diversity of the arm shape and sizes in the human study, as some participants' arm sizes and shapes might be out of our training distribution, resulting in lower performance. Another reason could be that compared to the manikin, real people unconsciously move their arms during the dressing trial, which violates our assumption that the human holds a static arm pose.

E. Failure Cases

Visual examples of failure cases can be found in our project webpage and also in Appendix Section C. The first failure case is that the policy gets stuck and cannot output actions that make further progress for the task, e.g., the policy oscillates between moving up and down and does not move forward. This might be due to that the arm pose of the participant is out of the training distribution, or other aspects of the observation cause a sim2real gap. Another failure case is the cloth getting caught on the arm. This usually happens when the participant unintentionally moves their arm too much in the dressing process, the policy actions move the gripper too high above the participant's arm, or the policy actions turn too early at the elbow. Due to limited fidelity of the simulator, the garment in simulation is more elastic and can still be dressed even if the robot is pulling high above the arm. However, the garments we test in the real world are less elastic than the simulated ones and they experience greater friction and get caught more easily. We believe that fine-tuning the policy in the real world can help address both failure cases. Since it is difficult to visually detect if a garment is undergoing high friction or has gotten caught on the body, we believe incorporating force-torque sensing can help alleviate these issues. We leave both as future work.

VII. Out-of-distribution evaluation and generalization of the system

We conducted a preliminary out of distribution evaluation of our system by relaxing the static arm assumption, i.e., the participants can move their arm during the dressing process. With experiments in both simulation and real world, we observe our system to be robust to small arm movements. We first evaluate how our system performs if the participants change their shoulder or elbow joint angles after we capture the initial arm point cloud. The simulation experiments are conducted on 1 pose sub-region, and the results show that our system is robust to 8.6 degrees of change in shoulder and elbow joint angles (averaged across 3 types of joint angle changes and 5 garments) while maintaining 75% of the original performance. Please refer to Appendix Section D for detailed experimental results. We also conducted 4 real-world dressing trials with one participant changing their joint angles, including lowering down the shoulder joint for 5 degrees, lowering down the elbow joint for 5 degrees, bending the elbow joint inwards for 5 degrees, and bending the elbow joint inwards for 10 degrees. Our system succeeds in the first 3 trials, and fails for the last trial since the elbow joint angle change is too large. We also evaluate our system with one participant performing constant arm motions during the dressing process in the real world. The participant is asked to perform 4 different kinds of arm motions during dressing, including constantly moving their forearm horizontally, constantly moving their forearm in a spherical motion, constantly moving their forearm up and down, and constantly moving their shoulder up and down. The maximum displacement of the arm during the motion was ± 10 centimeters. Our system succeeds in all 4 kinds of arm motions. Please see our website for videos of these real-world evaluations.

We also evaluate the generalization of the system towards dual-arm dressing. With the same single-arm dressing assumptions (static arm pose and robot already grasping the garment), our system generalizes to dual-arm dressing. The primary change is to control two robotic arms, one for each sleeve of the garment, which our Dense Transformation policy can handle well by extracting actions corresponding to both of the robot gripper points. We verified this in simulation with preliminary experiments where we successfully train policies to perform dual arm dressing of a hospital gown and a cardigan on a fixed pose (see visualizations on our project website). We note that dressing over a person's head, such as with a t-shirt, is more complex, due to the need for more dexterous trajectories and awareness of safety considerations. We leave such an extension to future work.

VIII. Conclusion

In this work, we develop a robot-assisted dressing system that is able to dress diverse garments on people with diverse poses from partial point cloud observations, based on a learned policy. We show that with careful design of the policy architecture, reinforcement learning (RL) can be used to learn effective policies with partial point cloud observations that work well for dressing diverse garments. We further leverage policy distillation to combine multiple policies trained on different ranges of human arm poses into a single policy that works over a wide variety of different poses. We propose guided domain randomization for effective and robust sim2real transfer. We perform comprehensive real-world evaluations of our system on a manikin, and in a human study with 17 participants of varying body size, poses, and dressed garments. On average, our system is able to dress 86% of the length of the participants' whole arm, and 71% of the length of the participants' upper arm across 425 dressing trials. We hope this work will serve as a foundation for future research to develop more robust and effective dressing systems.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grant No. IIS-2046491, and NVIDIA Corporation for their academic hardware grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or NVIDIA Corporation.

REFERENCES

[1] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. *arXiv preprint arXiv:2208.10552*, 2022.

[2] Arpit Bahety, Shreeya Jain, Huy Ha, Nathalie Hager, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Bag all you need: Learning a generalizable bagging strategy for heterogeneous objects. *arXiv preprint arXiv:2210.09997*, 2022.

[3] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.

[4] Gerard Canal, Guillem Alenyà, and Carme Torras. Adapting robot task planning to user preferences: an assistive shoe dressing example. *Autonomous Robots*, 43(6):1343–1356, 2019.

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

[6] Lawrence Yunliang Chen, Baiyu Shi, Daniel Seita, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. Autobag: Learning to open plastic bags and insert objects. *arXiv preprint arXiv:2210.17217*, 2022.

[7] Alexander Clegg, Zackory Erickson, Patrick Grady, Greg Turk, Charles C Kemp, and C Karen Liu. Learning to collaborate from simulation for robot-assisted dressing. *IEEE Robotics and Automation Letters*, 5(2):2746–2753, 2020.

[8] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[9] Zackory Erickson, Alexander Clegg, Wenhao Yu, Greg Turk, C Karen Liu, and Charles C Kemp. What does the person feel? learning to infer applied forces during robot-assisted dressing. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6058–6065. IEEE, 2017.

[10] Zackory Erickson, Henry M Clever, Greg Turk, C Karen Liu, and Charles C Kemp. Deep haptic model predictive control for robot-assisted dressing. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4437–4444. IEEE, 2018.

[11] Zackory Erickson, Henry M Clever, Vamsee Gangaram, Greg Turk, C Karen Liu, and Charles C Kemp. Multidimensional capacitive sensing for robot-assisted dressing and bathing. In *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pages 224–231. IEEE, 2019.

[12] Yixing Gao, Hyung Jin Chang, and Yiannis Demiris. User modelling for personalised dressing assistance by humanoid robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1840–1845. IEEE, 2015.

[13] Yixing Gao, Hyung Jin Chang, and Yiannis Demiris. Iterative path optimisation for personalised dressing assistance using vision and force information. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4398–4403. IEEE, 2016.

[14] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[16] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

[17] Lauren D Harris-Kojetin, Manisha Sengupta, Jessica Penn Lendon, Vincent Rome, Roberto Valverde, and Christine Caffrey. Long-term care providers and services users in the united states, 2015-2016. 2019.

[18] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Visuospatial foresight for physical sequential fabric manipulation. *Autonomous Robots*, 46(1):175–199, 2022.

[19] Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021.

[20] Pedro Ildefonso, Pedro Remédios, Rui Silva, Miguel

Vasco, Francisco S Melo, Ana Paiva, and Manuela Veloso. Exploiting symmetry in human robot-assisted dressing using reinforcement learning. In *EPIA Conference on Artificial Intelligence*, pages 405–417. Springer, 2021.

[21] Aleksandar Jevtić, Andrés Flores Valle, Guillem Alenyà, Greg Chance, Praminda Caleb-Solly, Sanja Dogramadzi, and Carme Torras. Personalized robot assistant for support in dressing. *IEEE transactions on cognitive and developmental systems*, 11(3):363–374, 2018.

[22] Ariel Kapusta, Wenhao Yu, Tapomayukh Bhattacharjee, C Karen Liu, Greg Turk, and Charles C Kemp. Data-driven haptic perception for robot-assisted dressing. In *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 451–458. IEEE, 2016.

[23] Ariel Kapusta, Zackory Erickson, Henry M Clever, Wenhao Yu, C Karen Liu, Greg Turk, and Charles C Kemp. Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Autonomous Robots*, 43(8):2183–2207, 2019.

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[25] Robert Lee, Daniel Ward, Vibhavari Dasagi, Akansel Cosgun, Juxi Leitner, and Peter Corke. Learning arbitrary-goal fabric folding with one hour of real robot experience. In *Conference on Robot Learning*, pages 2317–2327. PMLR, 2021.

[26] Shen Li, Nadia Figueroa, Ankit J Shah, and Julie A Shah. Provably safe and efficient motion planning with uncertain human dynamics. In *Robotics: Science and Systems*, 2021.

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

[28] Xingyu Lin, Carl Qi, Yunchu Zhang, Zhiao Huang, Katerina Fragkiadaki, Yunzhu Li, Chuang Gan, and David Held. Planning with spatial-temporal abstraction from point clouds for deformable object manipulation. In *6th Annual Conference on Robot Learning*, 2022.

[29] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022.

[30] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022.

[31] Chuer Pan, Brian Okorn, Harry Zhang, Ben Eisner, and David Held. Tax-pose: Task-specific cross-pose estimation for robot manipulation. *arXiv preprint arXiv:2211.09325*, 2022.

[32] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

[33] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[34] Kavya Puthuveetil, Charles C Kemp, and Zackory Erickson. Bodies uncovered: Learning to manipulate real blankets around people via physics simulations. *IEEE Robotics and Automation Letters*, 7(2):1984–1991, 2022.

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

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

[37] Jinge Qie, Yixing Gao, Runyang Feng, Xin Wang, Jielong Yang, Esha Dasgupta, Hyung Jin Chang, and Yi Chang. Cross-domain representation learning for clothes unfolding in robot-assisted dressing. In *Tenth International Workshop on Assistive Computer Vision and Robotics*, 2022.

[38] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2211.09423*, 2022.

[39] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[40] Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*, 2018.

[41] Daniel Seita, Nawid Jamali, Michael Laskey, Ajay Kumar Tanwani, Ron Berenstein, Prakash Baskaran, Soshi Iba, John Canny, and Ken Goldberg. Deep transfer learning of pick points on fabric for robot bed-making. In *The International Symposium of Robotics Research*, pages 275–290. Springer, 2019.

[42] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4568–4575. IEEE, 2021.

[43] Daniel Seita, Yufei Wang, Sarthak J Shetty, Edward Yao Li, Zackory Erickson, and David Held. Toolflownet:

Robotic manipulation with tools via predicting tool flow from point clouds. *arXiv preprint arXiv:2211.09006*, 2022.

[44] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022.

[45] Yufei Wang, David Held, and Zackory Erickson. Visual haptic reasoning: Estimating contact forces by observing deformable object interactions. *IEEE Robotics and Automation Letters*, 7(4):11426–11433, 2022.

[46] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.

[47] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial action maps for mobile manipulation. *arXiv preprint arXiv:2004.09141*, 2020.

[48] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial intention maps for multi-agent mobile manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8749–8756. IEEE, 2021.

[49] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. *arXiv preprint arXiv:1910.13439*, 2019.

[50] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance. *arXiv preprint arXiv:2204.02320*, 2022.

[51] Zhenjia Xu, Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Dextairity: Deformable manipulation can be a breeze. *arXiv preprint arXiv:2203.01197*, 2022.

[52] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[53] Wenhao Yu, Ariel Kapusta, Jie Tan, Charles C Kemp, Greg Turk, and C Karen Liu. Haptic simulation for robot-assisted dressing. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 6044–6051. IEEE, 2017.

[54] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[55] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[56] Fan Zhang and Yiannis Demiris. Learning grasping points for garment manipulation in robot-assisted dressing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9114–9120. IEEE, 2020.

[57] Fan Zhang and Yiannis Demiris. Learning garment manipulation policies toward robot-assisted dressing. *Science robotics*, 7(65):eabm6010, 2022.

[58] Fan Zhang, Antoine Cully, and Yiannis Demiris. Personalized robot-assisted dressing using user modeling in latent spaces. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3603–3610. IEEE, 2017.

[59] Fan Zhang, Antoine Cully, and Yiannis Demiris. Probabilistic real-time user posture tracking for personalized robot-assisted dressing. *IEEE Transactions on Robotics*, 35(4):873–888, 2019.

[60] Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, et al. Challenges and outlook in robotic manipulation of deformable objects. *IEEE Robotics & Automation Magazine*, 29(3):67–77, 2022.