Efficiently Learning Single-Arm Fling Motions to Smooth Garments

Lawrence Yunliang Chen*[1], Huang Huang*[1], Ellen Novoseller[1], Daniel Seita[2], Jeffrey Ichnowski[1], Michael Laskey[3], Richard Cheng[4], Thomas Kollar[4], and Ken Goldberg[1] * equal contribution

[1] University of California, Berkeley, Berkeley, CA 94720, USA[1]⋆,
[2] Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Electric Sheep Robotics, San Francisco, CA 94131, USA
[4] Toyota Research Institute, Los Altos, CA 94022, USA
{yunliang.chen, huangr} @berkeley.edu

Abstract. Recent work has shown that 2-arm "fling" motions can be effective for garment smoothing. We consider single-arm fling motions. Unlike 2-arm fling motions, which require little robot trajectory parameter tuning, single-arm fling motions are very sensitive to trajectory parameters. We consider a single 6-DOF robot arm that learns fling trajectories to achieve high garment coverage. Given a garment grasp point, the robot explores different parameterized fling trajectories in physical experiments. To improve learning efficiency, we propose a coarse-to-fine learning method that first uses a multi-armed bandit (MAB) framework to efficiently find a candidate fling action, which it then refines via a continuous optimization method. Further, we propose novel training and execution-time stopping criteria based on fling outcome uncertainty; the training-time stopping criterion increases data efficiency while the execution-time stopping criteria leverage repeated fling actions to increase performance. Compared to baselines, the proposed method significantly accelerates learning. Moreover, with prior experience on similar garments collected through self-supervision, the MAB learning time for a new garment is reduced by up to 87%. We evaluate on 36 real garments: towels, T-shirts, long-sleeve shirts, dresses, sweat pants, and jeans. Results suggest that using prior experience, a robot requires under 30 minutes to learn a fling action for a novel garment that achieves 60–94% coverage. Supplementary material can be found at https://sites.google.com/view/single-arm-fling.

Keywords: Deformable Object Manipulation, Garment Smoothing

1 Introduction

Garment smoothing [1] is useful in daily life, industry, and clothing stores as a first step toward garment folding [2–4], and hanging, as well as assistive dressing [5, 6]. However, almost all prior research in robot garment manipulation has

---

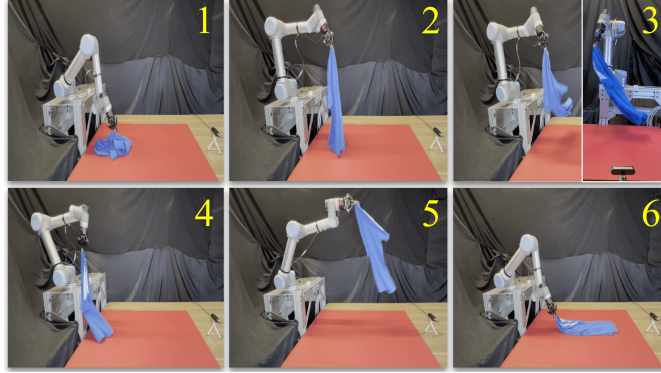⋆ [1]The AUTOLab at UC Berkeley (automation.berkeley.edu).

Fig. 1: **Single-arm dynamic garment flinging.** A UR5 robot flings a blue T-shirt. **Top row**: the robot holding the garment (1), lifts (2) and shakes it vertically (3 left) and horizontally (3 right) to loosen the garment using gravity. **Bottom row**: The robot flings the garment using the learned parameterized trajectory (Section 4) to smooth it on the surface. This cycle repeats until a stopping criterion is satisfied.

focused on quasistatic pick-and-place motions [7–13], which are relatively slow and limited by the robot workspace. Yet, robots can perform dynamic actions such as casting, throwing, vaulting [14–16], and air blowing [17] to efficiently place objects such as ropes, garments, and bags in desired configurations, even when much of the object lies outside the robot workspace. Dynamic manipulation of deformable objects is challenging [18–20] as such objects are partially observable, have infinite-dimensional configuration spaces, and have uncertain and complex dynamics due to deformability, elasticity, and friction. To address these modeling challenges, this work explores a learning-based approach for garment smoothing via high-speed dynamic actions that use inertia to move garments beyond the robot workspace.

In a pioneering paper, Ha and Song [16] proposed FlingBot, which used dynamic 2-arm "fling" motions to smooth garments. Using scripted motions at constant speed and air resistance, they achieved over 90% coverage for T-shirts. However, dual-arm manipulators may be too costly and bulky for home use. Many mobile manipulators, such as the HSR [21] and the Fetch [22], only have 1 arm. In this work, we show that a robot can use 1-arm fling motions to achieve a coverage of 60–94 % for 6 garment types.

Single-arm flings are more difficult than 2-arm flings, since they cannot straighten one garment edge prior to flinging. We propose a single-arm dynamic smoothing pipeline with novel parameterized primitives including shaking and flinging, and we propose an efficient method to learn these parameters. We begin with the robot gripping a fixed location on a crumpled garment. The robot lifts the garment and shakes it horizontally and vertically to loosen and reduce wrinkles. The robot then executes a learned fling motion. It retries the fling until a stopping rule determines that the coverage is sufficient. We propose a self-supervised learning method that uses the multi-armed bandit (MAB) framework

with Thompson Sampling (TS) to efficiently find a good candidate fling motion and then refines the fling parameters using the cross-entropy method (CEM).

The robot learns through real-world experiments. While prior work [15, 16] shows that simulation can facilitate learning for robot manipulation, state-of-the-art deformable object simulators have a large reality gap [23], and dynamic garment flings may require modeling additional effects such as material properties and air resistance. Consequently, we directly optimize flinging through real-world online learning and improve data efficiency by introducing a stopping rule for online training to threshold the expected performance improvement.

Unlike quasistatic actions, each fling results in significant variance in coverage due to the inherent *aleatoric* uncertainty, i.e., non-repeatable variation in outcomes. Thus, we propose a *second* stopping rule for execution time that stops the robot flinging when the coverage exceeds a learned threshold or maximum number of attempts. This stopping rule differs from the training-time stopping rule, which increases data efficiency.

This paper makes 4 contributions:

1. A formulation of single-arm garment flinging and a robot system that uses self-supervised learning in real to efficiently smooth garments.
2. An online learning framework that operates under large aleatoric uncertainty and enables the robot to quickly adapt to novel garments by leveraging a learned garment category prior and by optimizing fling parameters in a coarse-to-fine manner.
3. A training-time stopping rule based on epistemic uncertainty and execution-time stopping rules that exploit the aleatoric uncertainty of dynamic actions to increase expected execution-time fling coverage.
4. Experiments with 36 real garments (30 for training, 6 for testing) suggesting that the robot can learn effective single-arm fling actions achieving average coverage of 60–94 % on novel garments in 30 minutes.

2   Related Work

Deformable object manipulation has a long history in robotics research [18, 24, 25]. This paper focuses specifically on manipulation of garments. Much prior research on cloth manipulation uses geometric methods for perception and planning [26–30]. While these works show promising results, they make strong assumptions on garment configuration. To simplify planning, many of these works "funnel" garments into known states, for instance iteratively re-grasping the lowest hanging corner or using gravity for vertical smoothing [2, 31].

Learning Garment Manipulation The complexities of crumpled garments make representing their states difficult [32]. Researchers have thus bypassed precise state estimation by using data-driven methods such as imitation learning [33] and reinforcement learning. Prior learning-based approaches for garments mainly use quasistatic actions (e.g., pick-and-place) for smoothing [7, 9–13, 34] and folding [8, 35, 36]. While attaining promising results, they may require many short

actions and are limited by the robot's reach. In contrast, dynamic actions can manipulate garments with fewer actions, including out-of-reach portions.

Dynamic Manipulation In dynamic manipulation, robots leverage high-speed actions [19] such as tossing [20], swinging [37], and blowing [17]. There has been considerable progress on deformable dynamic manipulation, for instance high-speed cable knotting [38, 39] and learning-based approaches for high-speed manipulations of cables with a fixed endpoint [14] and a free endpoint [15, 40, 41]. Work on dynamic *garment* manipulation includes Jangir et al. [42] and Hietala et al. [43], which both leverage simulators, with the latter transferring to real; in contrast, we do not use simulation due to the reality gap and challenges in Sim2Real transfer [23].While there have been great strides in deformable simulation with quasistatic actions, simulating the dynamic movement of deformable objects including the effect of aerodynamics remains challenging [44].

Recently, Ha and Song presented FlingBot, which learns to perform a fixed-parameter dynamic fling motion to smooth garments using two UR5 arms. Fling-Bot is initially trained in SoftGym simulation [45] and fine-tuned in real, and learns grasp points for both arms. The results suggest significant efficiency benefits in attaining high coverage compared to quasistatic pick-and-place actions. In contrast, we consider using a *single-arm* manipulator and directly learn actions in real. While it is much harder to smooth a garment with only one arm—since two can effectively stretch one side of the garment so that the fling must only straighten the other side—single-arm fling motions can reduce the hardware cost or increase efficiency via multiple arms running in parallel.

3   Problem Statement

Workspace Assumptions We define a Cartesian coordinate frame containing a single-arm UR5 robot and a manipulation surface parallel to the xy-plane (Fig. 3), faced by an overhead RGB camera. We estimate garment coverage via the overhead RGB images with a manually-measured HSV threshold for each garment, and using the OpenCV package. As in some prior work on dynamic robot manipulation [14, 15, 43], we assume that the robot gripper holds the garment at a fixed point. During experiments, the initial grasp is set by a human.

We select the fixed grasping point by studying the performance of various grasp points with a T-shirt (see Section 6.3). The *back collar* (Fig. 5) yields the most promising results. Thus, in all subsequent experiments, the robot grasps the *back collar* of shirts and dresses. For towels, the robot grasps the center top point, while for jeans and pants, the robot grasps the center of the waistband's backside, as these locations are most analogous to the back collar of a T-shirt.

Task Objective The objective is to efficiently learn single-arm robot fling motions that maximally increase the coverage of an arbitrarily-crumpled garment. We assume access to training and testing sets of garments, denoted G_{train} and G_{test}. The fling motions are learned on the training set. Once training concludes, the robot executes the learned fling motions at execution time. During execution, repeated flinging is allowed up to a repetition limit.

We evaluate performance via the average garment coverage achieved by the learned fling action, similar to prior work on dynamic garment manipulation [16] and garment manipulation more generally [12, 13]. We also aim to reduce the number of flinging trials needed during training, where each trial corresponds to the execution of a fling cycle as described in Section 4. For each test garment, we record the number of flings the algorithm takes to learn the best fling action.

4   Motion Primitives and Pipeline

For each selected fling action, we execute one or more fling cycles to smooth the garment. A fling cycle consists of a reset motion, two shaking motions, and a dynamic arm motion. The gripper is closed during this process.

Reset Procedure To learn effective fling motions for different initial garment states, the self-supervised learning procedure performs a *reset* motion at the start of each fling cycle to randomize the garment's initial state. The reset has two parts: 1) the robot crumples the garment by lifting to a fixed joint configuration $\mathbf{q}_{\text{up}}$ and then placing it down at $\mathbf{q}_{\text{down}}$; and 2) the robot returns to $\mathbf{q}_{\text{up}}$ to help smooth the garment using gravity [2]. While the garment is not in an identical state after each reset (see website), and the folds can assume different shapes and directions, the reset places the garment in a stable position with its center of mass below the gripper, increasing repeatability of subsequent dynamic actions.

Shaking Motions Different garment layers can often stick together due to friction, making it more difficult for a fling action to expand the garment. In addition, garments often exhibit twists while hanging, resulting in correlated initial states between trials. Thus, immediately following the reset, we use two shaking motions—horizontal and vertical, shown in Fig. 1—to loosen parts of garments that may be stuck together and to reduce inter-trial correlation. Both shaking motions have a predefined amplitude, and each is repeated 3 times. The period T is chosen to be as brief as possible subject to robot motion control constraints. Vertical shaking plays the more important role of loosening the garments, while horizontal shaking mainly helps to reduce any twists.

Fling Parameterization Inspired by observing humans performing single-arm flings, we parameterize the fling motion using a trajectory as shown in Fig. 3. We control the robot's movement through both the spatial position and angle of joint 3. Joint 3 starts at point 1 (P1), pulls back to point 2 (P2) to create more space for the actual fling motion, accelerates from P2 to point 3 (P3) to perform the fling, and finally lays down the garment by moving to point 4 (P4). All four points lie on the yz-plane (i.e., $x = 0$). We accelerate learning by reducing the parameter space dimension. During preliminary experiments, varying P2 and P4 did not significantly affect results, and hence we fix those points.

The fling motion is therefore parameterized by 7 learnable *fling parameters*, $(v_{max}^{23}, v_{max}^{34}, P_{3,y}, P_{3,z}, \theta, v_\theta, a_\theta) \in \mathbb{R}^7$, where v_{max}^{23} and v_{max}^{34} are maximum joint velocities for all joints (Fig. 3) while traveling from P2 to P3 and from P3 to P4,

respectively; θ and (v_θ, a_θ) are the angle, velocity and acceleration of joint 3 at P3; and $(0, P_{3,y}, P_{3,z})$ gives the (x, y, z)-coordinates of P3. In FlingBot [16], contrastingly, the authors use a universal scripted fling motion for all garments and instead focus on learning the grasp points for their two arms. We use Ruckig [46], a time-optimal trajectory generation algorithm, to compute a trajectory passing through all waypoints with the desired velocities and accelerations at Points 1–4, while limiting velocity, acceleration, and jerk.

5   Self-Supervised Learning Pipeline for Flinging

To accomplish the task objective, we propose a self-supervised learning procedure to learn the fling parameters for a given garment, optionally using prior experience with similar garments to accelerate learning.

5.1   Learning Procedure of the Fling Action

Due to the high aleatoric uncertainty and continuous parameter space, we propose to explore the fling parameter space in a coarse-to-fine manner. First, we use a multi-armed bandit (MAB) [47] algorithm on discretized bins to quickly identify the best region of the parameter space. Then, we fine-tune the parameters with the cross-entropy method (CEM) [48]. The MAB framework is well-suited for efficiently identifying well-performing actions under the high aleatoric uncertainty in fling outcomes, while CEM does not require assuming a specific form for the underlying distribution.

For MAB, we form a set of actions $\mathcal{A} = \{a_1, \ldots, a_K\}$ by dividing the flinging parameter space into a uniform grid consisting of K grid cells, where each action $a_k \in \mathbb{R}^7$ is a grid cell center and parameterizes a flinging motion. Each parameter's range is defined based on the robot's physical limits and human observation (see Section 4). During each learning iteration t, the MAB algorithm samples an action $a^{(t)} \in \mathcal{A}$ to execute in the next fling cycle and observes a stochastic reward $r_t \in [0, 1]$ measuring post-fling garment coverage, computed using the overhead camera and expressed as a percentage of the garment's full surface area in its fully-smoothed state. After T steps, the observed actions and rewards form a dataset, $\mathcal{D}_T := \{(a^{(1)}, r^{(1)}), \ldots, (a^{(T)}, r^{(T)})\}$. Learning continues for a predefined iteration limit or until satisfying a *stopping rule* based on the estimated expected improvement. After the MAB learning procedure, we refine the MAB-chosen action using CEM, a continuous optimization algorithm, where we constrain CEM within the fling parameter grid cell corresponding to the MAB-chosen action. At each iteration, CEM samples a batch of actions from a normal distribution, which are then executed by the robot. The top few actions with the highest coverage (termed the "elites") are used to update the mean and variance of the CEM sampling distribution for the next CEM iteration.

Multi-Armed Bandit Algorithm We use Thompson Sampling (TS) [49] as the MAB algorithm. TS maintains a Bayesian posterior over the reward of each

action $a \in \mathcal{A}$: $p(r \mid a, \mathcal{D}_t)$. We use a Gaussian distribution to model both the prior and posterior reward of each action, and initialize the prior to $\mathcal{N}(0.5, 1)$ for all actions. On each iteration t, TS draws a reward sample $\tilde{r}_i \sim p(r \mid a_i, \mathcal{D}_t)$ for each action $a_i \in \mathcal{A}$, $i \in \{1, \ldots, K\}$. Then, TS selects the action corresponding to the highest sampled reward: $a^{(t)} = a_j, j = \operatorname{argmax}_i\{\tilde{r}_i \mid i = 1, \ldots, K\}$.

Stopping Rule during Learning Physical robot learning can be expensive due to the high execution time and potential risk of collision, making data efficiency important. During MAB learning, we balance efficiency with performance by stopping online learning when the estimated expected improvement over the highest posterior mean reward of any action falls below a given threshold [50]. Otherwise, the algorithm continues learning until an iteration limit. We calculate the expected improvement analytically [50] as follows, where $\phi(\cdot)$ and $\Phi(\cdot)$ are the PDF and CDF of the standard normal distribution, respectively, and $\mu(a_i)$ and $\sigma(a_i)$ are the posterior mean and standard deviation of action a_i:

$$\text{EI}(a_i) = \left(\mu(a_i) - \max_i\{\mu(a_i)\}\right)\Phi(Z) + \sigma(a_i)\phi(Z), \;\; Z = \frac{\mu(a_i) - \max_i\{\mu(a_i)\}}{\sigma(a_i)}.$$

5.2 Adapting to New Garments

To efficiently adapt to novel garments, we propose a learning algorithm that leverages prior experience from previously-seen garments. The algorithm consists of a training stage followed by an online learning stage. During training, we collect data from a training dataset G_{train} of m garments from all categories by executing the MAB algorithm and recording the final converged distributions of each action's coverage after MAB learning. During the online learning stage, we use the collected distribution data from garments in G_{train} to set an *informed* prior for the MAB, rather than using the *uninformed* initial prior of $\mathcal{N}(0.5, 1)$ for all actions. We then apply the MAB with the informed prior for a new test garment from test dataset G_{test}. We compare two methods for setting the prior mean and standard deviation based on prior experience: *All Garment* and *Category*. **All Garment** uses the empirical mean and standard deviation of coverage across all training garments for those corresponding actions. **Category** uses garments of only the same category as the test garment.

5.3 Execution-Time Stopping Rules

Given a novel garment, we execute the best action $a_{\text{best}} := \operatorname{argmax}_{a \in \mathcal{A}} \mu(a)$ identified by the online learning procedure in Section 5.2; this is the action with the highest posterior mean coverage. However, due to aleatoric uncertainty arising from the garment state and air friction, there is significant variance in the outcome. To mitigate this uncertainty, the robot repeatedly flings the garment until the current coverage is above a learned threshold. We propose 3 methods for deciding when to stop. All methods use the model posterior over a_{best}'s coverage, learned during training: $p(r \mid a_{\text{best}}, \mathcal{D}_{\text{train}}) = \mathcal{N}\left(\mu(a_{\text{best}}), \sigma^2(a_{\text{best}})\right)$, where $\mathcal{D}_{\text{train}}$ is the dataset collected during training. In this section, *stopping rule* refers exclusively to execution-time.

The first two methods set a hard stopping threshold: if the coverage is above a threshold, the robot stops. As different garments may have different performance levels, for Stopping Rule 1, we set the threshold based on a z-score determined from a_{best}'s posterior: $\mu(a_{\text{best}}) + z\sigma(a_{\text{best}})$ for some $z > 0$. For Stopping Rule 2, we set the threshold based on the one-step expected improvement calculated from the posterior $p(r \mid a_{\text{best}}, \mathcal{D}_{\text{train}})$ and stop if it falls below a given threshold.

Stopping Rule 3 accounts for a maximum flinging budget B. We decide to stop by estimating the expected improvement in coverage achievable during the remaining budget. We estimate this quantity using posterior sampling. Let r_j^{exec} be the coverage at step j of execution. We draw L sets of samples $\tilde{r}_{j+1,i}^{\text{exec}}, \ldots, \tilde{r}_{B,i}^{\text{exec}} \sim p(r \mid a_{\text{best}}, \mathcal{D}_{\text{train}})$, $i \in \{1, \ldots, L\}$, and approximate the expected improvement as: $EI(a_{\text{best}}) \approx \frac{1}{L}\sum_{i=1}^{L}\left[\max\{\tilde{r}_{j+1,i}^{\text{exec}}, \ldots, \tilde{r}_{B,i}^{\text{exec}}\} - r_j^{\text{exec}}\right]$. One could straightforwardly penalize $EI(a_{\text{best}})$ based on the cost of performing each fling action. Without such a penalty, however, this more-optimistic stopping rule will tend to fling more times than the one-step lookahead rules.

6   Experiments

We use a UR5 robot arm to perform experiments across a total of 36 garments from 6 categories: T-shirts, long-sleeve shirts, towels, dresses, sweat pants, and jeans, with 30 garments in G_{train} and 6 garments in G_{test}. See the project website for examples of garments in G_{test} at various smoothness levels. We calculate coverage via color thresholding on RGB images from a Logitech Brio 4K webcam. At the start of each experiment batch with a given garment, the human supervisor manually fixes the grasp location. The gripper never opens during the experiments. We use the following parameter ranges: $v_{max}^{23} \in [2\text{m/s}, 3\text{m/s}]$, $v_{max}^{34} \in [1\text{m/s}, 3\text{m/s}]$, $P_{3,y} \in [0.55\text{m}, 0.7\text{m}]$, $P_{3,z} \in [0.4\text{m}, 0.55\text{m}]$, $\theta \in [-40°, 20°]$, $v_\theta \in [-1\text{m/s}, 1\text{m/s}]$, $a_\theta \in [-20\text{m/s}^2, 20\text{m/s}^2]$. Each fling cycle takes ~ 45 s.

To evaluate problem difficulty and our algorithm's effectiveness, we compare with five human subjects. Results suggest that the robot achieves similar performance to humans. Details of this experiment are shown in the project website.

6.1   Comparison of Learning Methods

We first study different learning methods. We use a T-shirt and compare the learning performance of MAB+CEM with 3 baselines on that T-shirt: CEM, Bayesian optimization (BO) [50], and random actions. CEM and BO both perform continuous optimization without MAB, and BO uses a Gaussian Process model and the expected improvement acquisition function. The random baseline samples a random action from the continuous parameter range in each trial.

For MAB, we define the action space $\mathcal{A}$ via a uniform grid discretization of the trajectory parameters. To keep $|\mathcal{A}|$ small, we fix θ, v_θ, and a_θ at the centers of their ranges and vary the other 4 parameters, since we notice that v_{max}^{23}, v_{max}^{34}, $P_{3,y}$, and $P_{3,z}$ most strongly affect fling performance. We divide this 4-parameter space into a 2^4-cell grid and use the resulting grid cell centers as bandit actions.
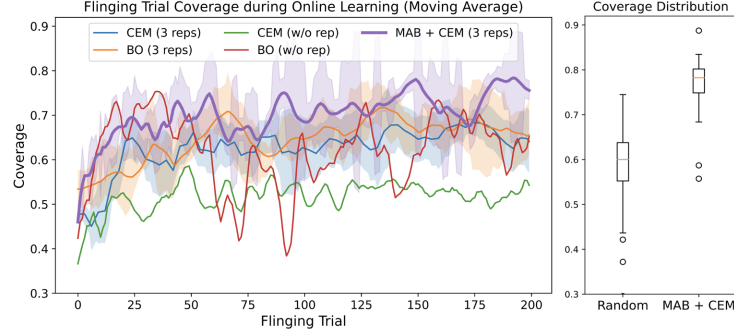
Fig. 2: Physical experiments showing the coverage of a T-shirt (fully-smoothed = 100%) vs. number of fling trials for MAB with CEM refinement (MAB+CEM) compared with baselines. **Left:** Coverage achieved over time for each optimization algorithm, smoothed by moving window averaging (window size = 10). For CEM and BO with 3 repetitions per sampled action, the shaded region shows the standard deviation among the 3 trials for each action. MAB+CEM converges more quickly than either continuous optimization algorithm alone. **Right:** The post-convergence coverage distribution of MAB+CEM significantly outperforms random action selection.

For BO and CEM, we optimize over the entire continuous parameter range. Since BO and CEM are noise sensitive, we repeat each action 3 times, averaging the coverage values. A CEM iteration samples 5 actions and forms the elite from the top 3. We run BO for 70 iterations ($70 \times 3 = 210$ trials) and run CEM for 14 iterations ($14 \times 5 \times 3 = 210$ trials). For MAB+CEM, we run MAB for 50 iterations and then CEM for 10 iterations ($50 + 10 \times 5 \times 3 = 200$ trials). We also compare BO and CEM without repetition, for 210 and 42 iterations respectively.

Results in Fig. 2 suggest that MAB+CEM outperforms CEM and BO with repetition. Due to the large aleatoric uncertainty in each fling trial, as can be seen from the shaded region and the box plot, BO and CEM without repetition perform the worst and are either unstable or fail to converge to a good action.

6.2 Comparison of Different Trajectory Parameterization Methods

To study the trajectory parameterization, we also compare the learning performance of the 7D parameterization with a 9D parameterization on the same T-shirt used in Section 6.1. The 9D parameter space consists of the original 7 parameters and two acceleration parameters: a_{max}^{23} and a_{max}^{34}, corresponding to the maximum acceleration for all joints while traveling from P2 to P3 and from P3 to P4, respectively. Results in Fig. 4 indicate that the 7D and 9D parameterizations exhibit similar performance for MAB+CEM, while the 9D parameterization makes BO less stable. MAB+CEM outperforms either CEM or BO alone in both parameterizations, suggesting generalizability of our proposed coarse-to-fine method.

6.3 Results for Different Grasp Points

The choice of grasping point can affect single-arm fling performance. We study the effectiveness of single-arm fling actions on a T-shirt with various grasp points:
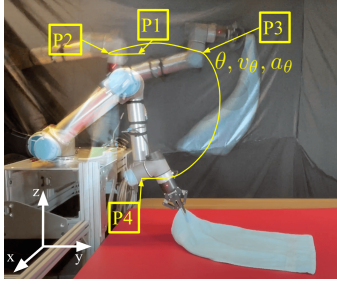
Fig. 3: 7-Dim trajectory parameterization. The robot end effector begins at point P1, moves to P2, moves (flings) to P3 at a high speed, and travels downwards to P4, bringing the garment to rest. θ is the angle of joint 3 at P3; (v_θ, a_θ) are the velocity and acceleration of joint 3 at P3.
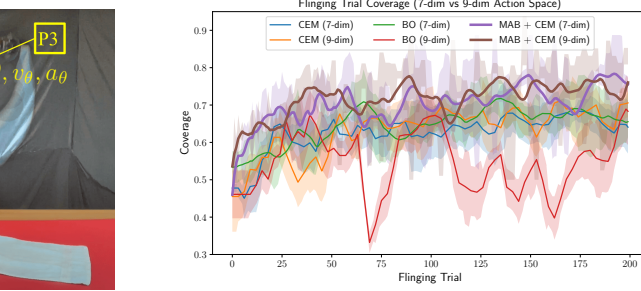

Fig. 4: We conduct physical experiments to compare optimizing in 7 and 9-dimensional parameter spaces, where in the latter, the acceleration parameters are also learnable. Each action is repeated 3 times for CEM and BO. We find that 7 parameters are sufficient.

the back collar, sleeve hem, sleeve joint, shoulder, side seam, bottom center (1 layer and 2 layers), and bottom corner (Fig. 5). We apply the MAB algorithm for each grasp point.

Fig. 5: 8 T-shirt grasp points.

Table 1: Average coverage at each grasp point by the median and best actions among the 16 bandit actions used for MAB learning.

Grasp Point	Median Coverage	Max Coverage
Bottom Center (1-layer)	44%	54%
Side Seam	47%	57%
Bottom Corner	49%	66%
Sleeve Joint	50%	65%
Shoulder	52%	66%
Bottom Center (2-layer)	55%	74%
Sleeve Hem	55%	70%
Back Collar	**61**%	**74**%

Table 1 reports the average coverage of the median and best actions, and indicates that the back collar is the best grasp point, followed by the sleeve hem and bottom center (2-layer). Thus, we focus on grasping the back collar, while selecting analogous grasp points for other garment types (Section 3).

6.4   Training and Testing Across Different Garments

We use MAB+CEM to learn fling actions for different garments. For MAB, we set the threshold for the expected improvement (EI) to be 1.5 % because empirically, we find that the EI starts to decrease significantly more slowly past 1 %. We then refine via 2 iterations of CEM ($2 \times 5 \times 3 = 30$ trials).

For each garment category, the training set G_{train} includes 5 garments. For each garment in G_{train}, we run MAB with 16 actions for $N_{\text{train}} = 50$ trials. We do not perform the CEM step, as the goal is to form a prior on those 16 actions

which we use to accelerate the learning on test garments, instead of finding the best action on each training garment.
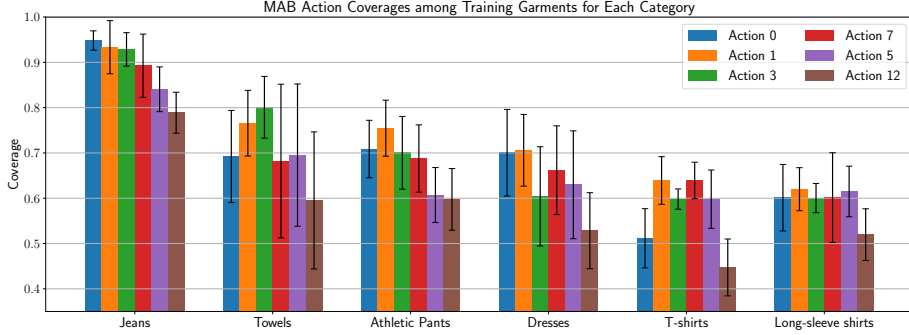

Fig. 6: From physical experiments, average coverage achieved for 6 of the 16 bandit actions among all garments in the training set G_{train}, separated by garment category (mean $\pm$ 1 std). Each category contains 5 garments. The standard deviations indicate performance variability among different garments within a category. While some actions tend to outperform others, the best actions differ for different garment categories.

Fig. 6 shows the average coverage for 6 of the 16 bandit actions, separated by category. We see that within each garment type, different actions lead to significant differences in coverage. Moreover, the relative performance of various bandit actions is different in each category, suggesting that different actions are better suited for different garment categories.

Using the data from G_{train} to inform the MAB prior, we evaluate performance on the garments in G_{test}, which contains one new garment from each category. We compare MAB performance with 3 priors: a) the original prior (i.e., "Uninformed"), b) a prior calculated using all the garments in G_{train} ("All-Garment"), and c) a prior calculated only using garments in G_{train} of the same category as the test garment ("Category"). The informed priors (b and c) set the prior mean and standard deviation for each bandit action using the empirical means and standard deviations from training. Table 2 reports results over 20 repetitions. While the different priors do not significantly affect the final coverage achieved by the best-identified action, the results suggest that the "Category" prior can significantly accelerate learning. In particular, with a more informative prior based on G_{train}, a new garment's EI falls below the termination threshold in 57–87 % fewer MAB iterations than when starting from an uninformed prior.

From Table 2, we also see that CEM refinement can usually increase performance by several percentage points over the best MAB-identified action, with the exception of the jeans and towel, where MAB already identifies high-coverage actions. The p-values test whether the improvements in mean coverage are statistically significant, given the large aleatoric uncertainty of each individual fling trial (whose coverages have standard deviation up to 7 %). We conclude that within 10 MAB iterations (10 trials) and 2 CEM iterations (30 trials), the robot

Prior	Type	#	MAB	+CEM	p-val.	Type	#	MAB	+CEM	p-val.
Uninf.		24	94±4%	94±1%	0.509		32	93±6%	93±7%	0.594
All	Jeans	9	92±1%	93±2%	0.003	Towel	16	93±6%	93±7%	0.594
Cat.		**3**	94±4%	94±1%	0.509		**12**	90±6%	88±4%	0.759
Uninf.		40	78±2%	78±2%	0.743		30	75±5%	78±4%	0.011
All	Dress	30	79±3%	82±3%	0.003	Sweat Pants	20	77±3%	80±3%	0.016
Cat.		**6**	80±1%	81±2%	0.035		**13**	77±3%	80±3%	0.016
Uninf.		32	72±6%	73±3%	0.257	Long-	48	57±5%	60±5%	0.060
All	T-shirt	19	73±6%	75±4%	0.115	sleeve	37	54±6%	61±5%	4e-4
Cat.		**7**	73±6%	75±4%	0.115	Shirt	**14**	57±4%	60±4%	0.018

Table 2: Physical experiments with 36 garments demonstrating the efficiency of learning fling actions on test garments using category priors. Under each MAB prior, we report the number (#) of bandit trials that MAB takes to reach a stopping threshold (EI < 1.5%), and columns 4, 5, and 9, 10 are the average coverage (mean ± std) achieved by the best MAB-identified action (prior to CEM) and by MAB+CEM, where we execute each action 20 times. We report identical coverage numbers for MAB+CEM instances in which MAB identifies the same best action, since the CEM procedure is the same. We compare the *Uninformed* prior (Uninf.) to the *All-Garment* (All) and *Category* (Cat.) priors, for which the decrease in MAB trials indicates high garment category informativeness. The p-values are from t-tests comparing whether the mean coverage of the action identified by MAB+CEM significantly improves on the MAB-identified action; p-values compare the difference in sample means with standard errors, which account for the sample size (20). Thus, while there is large aleatoric uncertainty, as seen from the large standard deviations among the 20 repetitions, the single-digit percentage increase in mean can be statistically significant.

can efficiently learn a fling action that achieves 60–94 % coverage for a new garment in under 30 minutes. For comparison, using two arms, Flingbot [16] achieves 79–93 % coverage on towels and 93–94 % coverage on T-shirts.

6.5   Stopping Time During Execution

To compare the 3 stopping rules proposed in Section 5.3 and study the trade-off between the number of flinging trials and performance, we apply each stopping rule with a range of thresholds by post-processing the execution-time results of the best action found for each garment. For each method, we bootstrap the experiment results and empirically estimate the stopping-time distribution under various stopping thresholds. From the results in Fig. 7, we can see that it takes on average 4 flings to achieve a coverage that is 1 standard deviation above the mean or that results in a one-step EI below 1 % absolute coverage. Stopping based on EI with a budget leads to more optimistic EI and significantly more trials. In practice, a heuristically-estimated cost for each additional fling can counter the tendency of this method to fling too many times.

7   Conclusion

We consider single-arm dynamic flinging to smooth garments. We learn fling motions directly in the real world to address complex, difficult-to-model garments
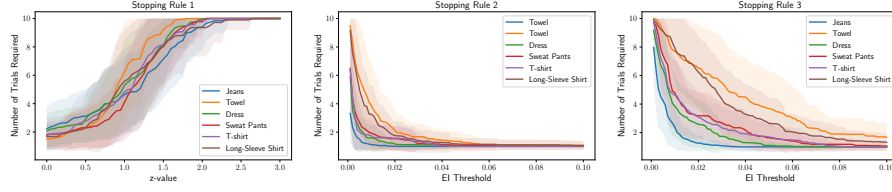
Fig. 7: Number of trials under each execution-time stopping rule as a function of the termination threshold. **Left**: stopping when the coverage exceeds one standard deviation above the mean. **Middle**: stopping when the one-step expected improvement falls below a threshold. **Right**: stopping when the expected improvement over the remaining trial budget falls below a threshold, using a total budget of 10 trials. With repeated fling trials and optimal stopping, the robot can achieve better-than-average results during execution, especially for garments with large aleatoric uncertainty.

such as long-sleeve shirts and dresses. We propose stopping rules during training and execution time. Results suggest that we can learn effective fling actions that achieve up to 94 % average coverage in under 30 minutes. In future work, we will enhance the current system with quasistatic pick-and-place actions to fine-tune smoothing [9, 12] and explore evaluation metrics that judge the quality of dynamic motions by the ease of subsequent quasistatic fine-tuning.

Acknowledgements

References

[1] Li Sun, Gerardo Aragon-Camarasa, Simon Rogers, and J. Paul Siebert. "Accurate Garment Surface Analysis using an Active Stereo Robot Head with Application to Dual-Arm Flattening". In: *IEEE International Conference on Robotics and Automation*. 2015.

[2] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. "Cloth Grasp Point Detection Based on Multiple-View Geometric Cues with Application to Robotic Towel Folding". In: *IEEE International Conference on Robotics and Automation*. 2010.

[3] Stephen Miller, Jur van den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. "A Geometric Approach to Robotic Laundry Folding". In: *International Journal of Robotics Research*. 2012.

[4] Benjamin Balaguer and Stefano Carpin. "Combining Imitation and Reinforcement Learning to Fold Deformable Planar Objects". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011.

[5]   Zackory Erickson, Vamsee Gangaram, Ariel Kapusta, C. Karen Liu, and Charles C. Kemp. "Assistive Gym: A Physics Simulation Framework for Assistive Robotics". In: *IEEE International Conference on Robotics and Automation.* 2020.

[6]   Zackory Erickson, Henry Clever, Greg Turk, C. Karen Liu, and Charles Kemp. "Deep Haptic Model Predictive Control for Robot-Assisted Dressing". In: *IEEE International Conference on Robotics and Automation.* 2018.

[7]   Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. "Learning Visible Connectivity Dynamics for Cloth Smoothing". In: *Conference on Robot Learning.* 2021.

[8]   Thomas Weng, Sujay Bajracharya, Yufei Wang, Khush Agrawal, and David Held. "FabricFlowNet: Bimanual Cloth Manipulation with a Flow-based Policy". In: *Conference on Robot Learning.* 2021.

[9]   Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. "Learning to Manipulate Deformable Objects without Demonstrations". In: *Robotics: Science and Systems.* 2020.

[10]   Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. "Learning Predictive Representations for Deformable Objects Using Contrastive Estimation". In: *Conference on Robot Learning.* 2020.

[11]   Martina Lippi, Petra Poklukar, Michael C. Welle, Anastasiia Varava, Hang Yin, Alessandro Marino, and Danica Kragic. "Latent Space Roadmap for Visual Action Planning of Deformable and Rigid Object Manipulation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2020.

[12]   Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, Katsu Yamane, Soshi Iba, John Canny, and Ken Goldberg. "Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2020.

[13]   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: *IEEE International Conference on Robotics and Automation.* 2021.

[14]   Harry Zhang, Jeff Ichnowski, Daniel Seita, Jonathan Wang, Huang Huang, and Ken Goldberg. "Robots of the Lost Arc: Self-Supervised Learning to Dynamically Manipulate Fixed-Endpoint Ropes and Cables". In: *IEEE International Conference on Robotics and Automation.* 2021.

[15]   Vincent Lim, Huang Huang, Lawrence Yunliang Chen, Jonathan Wang, Jeffrey Ichnowski, Daniel Seita, Michael Laskey, and Ken Goldberg. "Planar Robot Casting with Real2Sim2Real Self-Supervised Learning". In: *IEEE International Conference on Robotics and Automation.* 2022.

[16]   Huy Ha and Shuran Song. "FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding". In: *Conference on Robot Learning.* 2021.

[17]   Zhenjia Xu, Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. "DextAIRity: Deformable Manipulation Can be a Breeze". In: *Robotics: Science and Systems.* 2022.

[18]   Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. "Robotic Manipulation and Sensing of Deformable Objects in Domestic and Industrial Applications: a Survey". In: *International Journal of Robotics Research.* 2018.

[19]   Kevin M Lynch and Matthew T Mason. "Dynamic nonprehensile manipulation: Controllability, planning, and experiments". In: *The International Journal of Robotics Research* 18.1 (1999), pp. 64–92.

[20]   Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. "TossingBot: Learning to Throw Arbitrary Objects with Residual Physics". In: *Robotics: Science and Systems*. 2019.

[21]   Kunimatsu Hashimoto, Fuminori Saito, Takashi Yamamoto, and Koichi Ikeda. "A Field Study of the Human Support Robot in the Home Environment". In: *IEEE Workshop on Advanced Robotics and its Social Impacts*. 2013.

[22]   Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. "Fetch & Freight: Standard Platforms for Service Robot Applications". In: *IJCAI Workshop on Autonomous Mobile Service Robots*. 2016.

[23]   N. Jakobi, P.Husbands, and I. Harvey. "Noise and the Reality Gap: The use of Simulation in Evolutionary Robotics". In: *European Conference on Advances in Artificial Life*. 1995.

[24]   Julia Borras, Guillem Alenya, and Carme Torras. "A Grasping-centered Analysis for Cloth Manipulation". In: *arXiv preprint arXiv:1906.08202* (2019).

[25]   Jihong Zhu, Andrea Cherubini, Claire Dune, David Navarro-Alarcon, Farshid Alambeigi, Dmitry Berenson, Fanny Ficuciello, Kensuke Harada, Jens Kober, Xiang Li, Jia Pan, Wenzhen Yuan, and Michael Gienger. "Challenges and Outlook in Robotic Manipulation of Deformable Objects". In: *arXiv preprint arXiv:2105.01767* (2021).

[26]   Yunfei Bai, Wenhao Yu, and C. K. Liu. "Dexterous Manipulation of Cloth". In: *European Association for Computer Graphics*. 2016.

[27]   Yinxiao Li, Danfei Xu, Yonghao Yue, Yan Wang, Shih-Fu Chang, Eitan Grinspun, and Peter K. Allen. "Regrasping and Unfolding of Garments Using Predictive Thin Shell Modeling". In: *IEEE International Conference on Robotics and Automation*. 2015.

[28]   Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, and Sotiris Malassiotis. "Autonomous Active Recognition and Unfolding of Clothes Using Random Decision Forests and Probabilistic Planning". In: *IEEE International Conference on Robotics and Automation*. 2014.

[29]   Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. "Clothes State Recognition Using 3D Observed Data". In: *IEEE International Conference on Robotics and Automation*. 2009.

[30]   Yasuyo Kita, Toshio Ueshiba, Ee Sian Neo, and Nobuyuki Kita. "A Method For Handling a Specific Part of Clothing by Dual Arms". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009.

[31]   Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James F O'Brien, and Pieter Abbeel. "Bringing Clothing Into Desired Configurations with Limited Perception". In: *IEEE International Conference on Robotics and Automation*. 2011.

[32]   Cheng Chi and Shuran Song. "GarmentNets: Category-Level Pose Estimation for Garments via Canonical Space Shape Completion". In: *The IEEE International Conference on Computer Vision*. 2021.

[33]   Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. "A Survey of Robot Learning From Demonstration". In: *Robotics and Autonomous Systems* 57 (2009).

[34]   Daniel Seita, Nawid Jamali, Michael Laskey, Ron Berenstein, Ajay Kumar Tanwani, Prakash Baskaran, Soshi Iba, John Canny, and Ken Goldberg. "Deep

Transfer Learning of Pick Points on Fabric for Robot Bed-Making". In: *International Symposium on Robotics Research*. 2019.

[35] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. "VisuoSpatial Foresight for Multi-Step, Multi-Task Fabric Manipulation". In: *Robotics: Science and Systems*. 2020.

[36] Robert Lee, Daniel Ward, Akansel Cosgun, Vibhavari Dasagi, Peter Corke, and Jurgen Leitner. "Learning Arbitrary-Goal Fabric Folding with One Hour of Real Robot Experience". In: *Conference on Robot Learning*. 2020.

[37] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. "SwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020.

[38] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. "Motion Planning for Dynamic Knotting of a Flexible Rope With a High-Speed Robot Arm". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010.

[39] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. "Dynamic High-Speed Knotting of a Rope by a Manipulator". In: *International Journal of Advanced Robotic Systems*. 2013.

[40] Simon Zimmermann, Roi Poranne, and Stelian Coros. "Dynamic Manipulation of Deformable Objects with Implicit Integration". In: *IEEE Robotics and Automation Letters*. 2021.

[41] Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. "Iterative Residual Policy for Goal-Conditioned Dynamic Manipulation of Deformable Objects". In: *Robotics: Science and Systems*. 2022.

[42] Rishabh Jangir, Guillem Alenya, and Carme Torras. "Dynamic Cloth Manipulation with Deep Reinforcement Learning". In: *IEEE International Conference on Robotics and Automation*. 2020.

[43] Julius Hietala, David Blanco-Mulero, Gokhan Alcan, and Ville Kyrki. "Closing the Sim2Real Gap in Dynamic Cloth Manipulation". In: *arXiv preprint arXiv:2109.04771* (2021).

[44] Yashraj Narang, Yunzhu Li, Miles Macklin, Dale McConachie, and Jane Wu. *RSS 2021 Workshop on Deformable Object Simulation in Robotics*. https://sites.google.com/nvidia.com/do-sim. Accessed: 2021-09-12. 2021.

[45] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. "SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation". In: *Conference on Robot Learning*. 2020.

[46] Lars Berscheid and Torsten Kröger. "Jerk-limited Real-time Trajectory Generation with Arbitrary Target States". In: *Robotics: Science and Systems*. 2021.

[47] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2 (2002), pp. 235–256.

[48] R. Rubinstein and D. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag, 2004.

[49] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[50] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Neural Information Processing Systems*. 2012.