SpeedFolding: Learning Efficient Bimanual Folding of Garments

Yahav Avigal*,1, Lars Berscheid*,1,2, Tamim Asfour[2], Torsten Kröger[2], and Ken Goldberg[1]

***Abstract*— Folding garments reliably and efficiently is a long standing challenge in robotic manipulation due to the complex dynamics and high dimensional configuration space of garments. An intuitive approach is to initially manipulate the garment to a canonical smooth configuration before folding. In this work, we develop SpeedFolding, a reliable and efficient bimanual system, which given user-defined instructions as folding lines, manipulates an initially crumpled garment to (1) a smoothed and (2) a folded configuration. Our primary contribution is a novel neural network architecture that is able to predict pairs of gripper poses to parameterize a diverse set of *bimanual* action primitives. After learning from 4300 human-annotated and self-supervised actions, the robot is able to fold garments from a random initial configuration in under 120 s on average with a success rate of 93 %. Real-world experiments show that the system is able to generalize to unseen garments of different color, shape, and stiffness. While prior work achieved 3-6 Folds Per Hour (FPH), SpeedFolding achieves 30-40 FPH.**

See `https://pantor.github.io/speedfolding` for code, videos, and datasets.

I. Introduction

These tasks are largely performed by humans due to the complex configuration space as well as the highly non-linear dynamics of deformable objects [1], [2]. Additionally, folding is a long horizon sequential planning problem, as it requires to first flatten or smooth the garment, and then follow a sequence of steps [3], [4] or sub-goals [5] to achieve the desired fold.

Prior work has mainly focused on single-arm manipulation [2], [6], [7], [8] or on complex iterative algorithms [3], [4], [9], requiring a large number of interactions and resulting in long execution times. Recently, Ha et al. [10] proposed a method for smoothing cloth that computes the pick points for a high-velocity dynamic fling action directly from overhead images, and can smooth garments to 80 % coverage in 3 actions on average. However, the proposed 4 degrees of freedom (DoFs) action parameterization constrains the two pick poses significantly, in particular by discrete distances and a fixed rotation in between.

We present SpeedFolding, an end-to-end system for fast and efficient garment folding. At first, a novel BiManual Manipulation Network (BiMaMa-Net) learns to predict a pair of gripper poses for bimanual actions from an overhead RGBD input image to smooth an initially crumpled garment. Once the garment has been smoothed to a desired level, determined by a learned smoothing classifier, SpeedFolding executes a folding pipeline (see Fig. 1). This paper contributes:

[1]AUTOLab at UC Berkeley {yahav_avigal, goldberg}@berkeley.edu

[2]Karlsruhe Institute of Technology (KIT) {lars.berscheid, asfour, torsten}@kit.edu

*Equal contribution: order determined by shirt folding skills.

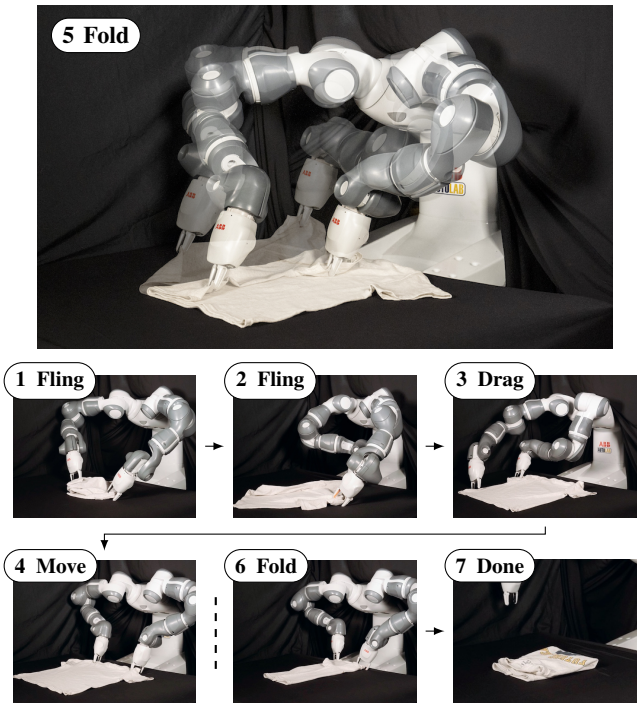


Fig. 1: SpeedFolding learns to fold garments from arbitrary configurations: Given a crumpled t-shirt, the robot unfolds using fling actions (1, 2), smooths it with a drag action (3) until it is *sufficiently smoothed*. It then moves the t-shirt for better reachability (4), and applies folds (5-6) to achieve the user-defined configuration (7).

1) The BiMaMa-Net architecture for bimanual manipulation that computes two *corresponding* planar gripper poses without any spatial restrictions, with an automated calibration procedure to account for robot reachability constraints.
2) An end-to-end robotic system for efficient smoothing and folding. First, the system learns to smooth a garment to a sufficiently smoothed configuration through self-supervision. Then, the robot folds the garment according to user-defined folding lines.
3) An experimental dataset from physical experiments that suggests the system can fold garments with a success rate of over 90 %, including garments unseen during training that differ in color, shape and stiffness. Folding a t-shirt takes under 120 s on average, improving baselines by 30 to 47 % and prior works by 5 to 10 ×.

II. Related Work

Bimanual robotic manipulation has been studied extensively in fields from surgical robotics to industrial manipula-

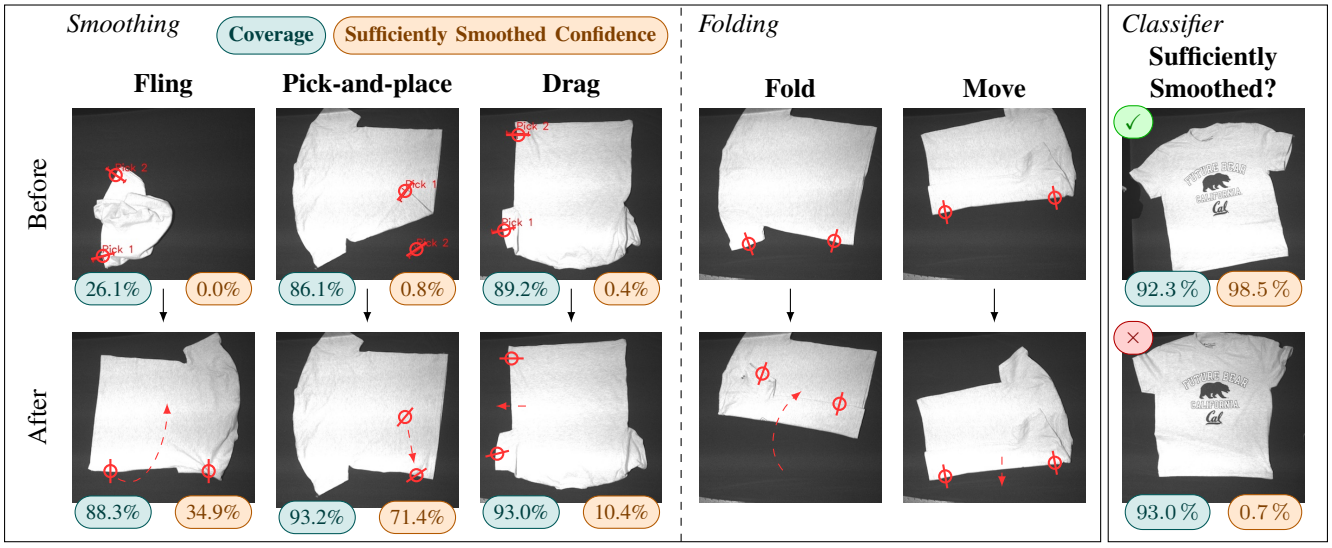


Fig. 2: **Action Primitives.** Given an overhead RGBD image, BiMaMa-Net selects a smoothing action from a discrete set of primitives (**left** box), and computes a pair of end-effector poses. Coverage calculation (in blue) is insensitive to wrinkles in the fabric (**right** box, **bottom**) and therefore BiMaMa-Net learns to classify configurations as Sufficiently Smoothed (**right** box, **top**) (in yellow). Folding a t-shirt from a smooth configuration is done through a sequence of folding primitives (**center** box).

tion [11]. A dual-arm system extends the workspace, allows for increased payload and for more complex behaviours than a single arm system [5], [12], [13], [14], but comes at the cost of higher planning complexity due to the additional DoFs and self-collisions [15]. A promising line of research is to employ dual-arm systems for garment manipulation [16]. Garments are especially difficult to control and manipulate due to their large configuration space, self-occlusions, and complex dynamics [2]. Recent works have mainly focused on garment smoothing from arbitrary configurations [10], or garment folding, assuming the garment has been initially flattened [5]. We present an end-to-end approach to smoothing and then folding garments from initial crumpled configurations.

Garment smoothing aims to transform the garment from an arbitrary crumpled configuration to a smooth configuration [7]. Prior works have focused on extracting and identifying specific features such as corners and wrinkles [3], [4], [17], [18]. Recent methods have used expert demonstrations to learn garment smoothing policies in simulation [2], [6], [7], however these methods learn quasi-static pick-and-place actions that require a large number of interactions on initially crumpled garments. Ha et al. [10] introduced a novel 4 DoF dynamic fling action parameterization learned in simulation that can achieve $\sim 80\,\%$ garment coverage within 3 actions. However, this parameterization is (1) limited to fling actions, (2) fails to fully smooth garments, and (3) induces grasp failures in more than $25\,\%$ of actions. In this work we use expert demonstrations and self-supervised learning purely in the physical world to train a novel bimanual manipulation neural network (NN) architecture to smooth a garment such that it is ready to be folded.

Garment folding has many applications in hospitals, homes and warehouses. Early approaches rely heavily on heuristics and can achieve high success rates, but have long cycle times on the order of 10 to 20 min per garment [3], [4], [9], [19], [20]. Recent methods have been focusing on learning goal-conditioned policies in simulation [5], [6], [21], [22] and directly on a physical robot [23]. In this work, we compare an instruction-based folding approach that can reliably fold smoothed garments, with a novel folding approach that can fold a t-shirt directly from a non-smooth configuration given prior knowledge about its dimensions.

III. Problem Statement

Given a visual observation $o_t \in \mathbb{R}^{W\times H\times C}$ of the garment's configuration s_t at time t, the objective is to compute and execute an action a_t to transfer the garment from an arbitrary configuration to a desired user-defined s^* goal configuration. In particular, s^* is invariant under the garment's position and orientation in the workspace. We assume an overhead observation with a calibrated pixel-to-world transformation, as well as a garment that is easily distinguishable from the workspace.

We consider a dual-arm robot with parallel-jaw grippers executing actions of type $m \in \mathcal{M}$ from a discrete set of pre-defined action primitives. In particular, we parameterize each primitive by two planar gripper poses

$$a_t = \langle m, (x_1, y_1, \theta_1), (x_2, y_2, \theta_2)\rangle$$

for each arm respectively, in which (x_i, y_i) are coordinates in pixel space, and θ_i is the end-effector rotation about the z axis. We further assume a flat obstacle-free workspace and a motion planner that computes collision-free trajectories for a dual-arm robot.

IV. Method

SpeedFolding uses BiMaMa-Net, a learned garment-smoothing method to bring an initially crumpled garment to a

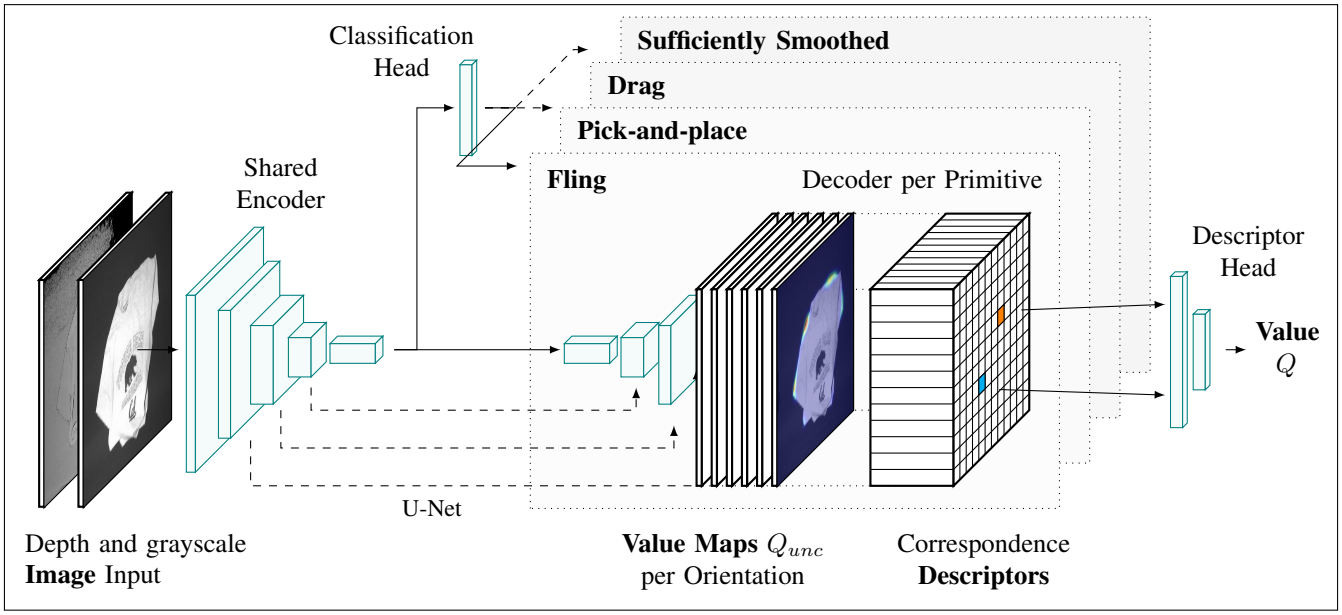


Fig. 3: **BiManual Manipulation Net (BiMaMa-Net)** architecture first maps an image to a manipulation primitive type via its shared encoder and classification head. Given a primitive, it predicts dense unconditioned value maps for a discrete set of gripper orientations. It then calculates pixel-wise correspondence descriptors. A descriptor pair, representing a bimanual action, is combined in the descriptor head to predict its joint value.

sufficiently smooth configuration, followed by an instruction-based garment folding pipeline.

A. Action Primitives

We are interested in the set of quasi-static and dynamic action primitives that enable the robot to (1) transfer an arbitrary garment configuration s_t to a folded goal configuration s^* (completeness), (2) reducing the number of action steps (efficiency), and (3) with a reduced number of primitives (minimality). Each action primitive is defined through a pair of poses as well as a motion trajectory. All primitives share a common procedure to reliably grasp the garment with parallel jaw grippers: Each gripper moves $4\,\mathrm{cm}$ above the grasp pose a_t, rotates $8°$ so that one fingertip is below the other, and moves $1\,\mathrm{cm}$ towards the direction of the higher fingertip. This motion improves the success for grasping in particular at the edge of the garment. We define following learned primitives (Fig. 2 left box):

Fling: Given two pick poses, the arms first pick those points, lift the garment above the workspace and stretch it until a force threshold is reached, measured using the arms' internal force sensors. Next, the arms apply a dynamic motion, flinging the garment forward and then backward while gradually reducing the height toward the workspace. Similar to [10], we find the fling motion to be robust under change of velocity and trajectory parameters, and therefore we keep these parameters fixed. The fling primitive allows to significantly increase the garment's coverage in a few steps, but often does not yield a smooth configuration.

Pick-and-place: Given a pick and a corresponding place pose, a single arm executes this quasi-static action, while the second arm presses down the garment at a point on a line extending the pick from the place pose. Pick-and-place enables the robot to fix local faults such as corners or sleeves folded on top of the garment.

Drag: Given two pick points, the robot drags the garment for a fixed distance away from the garment's center of mask, leveraging the friction with the workspace to smooth wrinkles or corners, e.g. sleeves folded below the garment.

We define heuristic-based primitives (Fig. 2 center box):

Fold: Both arms execute a pick-and-place action simultaneously to fold the garment. The heuristic for calculating the pick and place poses is explained in Sec. IV-E.

Move: While similar to drag, this primitive's pick poses and its drag distance are calculated by a heuristic so that the garment's center of mask is moved to a goal target point. Usually, the robot drags the garment towards itself to mitigate reachability issues in subsequent actions. Sec. IV-E provides details about the pose calculation.

We define an additional learned primitive to switch from garment-smoothing to folding (Fig. 2 right box):

Sufficiently Smoothed: We find that deciding whether a garment is ready to be folded purely from coverage computation, as done in prior works [6], [7], [10], is not sufficient. In particular, even a high coverage is prone to wrinkles or faults that might reduce the subsequent fold quality significantly (as described in Fig. 2). Instead of relying on the coverage, BiMaMa-Net returns a smoothness value given an overhead image. While this primitive is not used to change the configuration of the garment, it is used to switch from garment-smoothing to folding.

B. BiMaMa-Net for Bimanual Manipulation

Predicting a single pose from an overhead image is commonly done by first estimating a pixel-wise value map per gripper z-axis rotation θ, in which each pixel value represents a future expected reward (e.g., grasp success, increase in garment coverage, etc.), and then selecting the maximum greedily [24], [25], [26], [10]. Extending this approach to two *corresponding* planar poses $(x, y, \theta)_{1,2}$ conditioned on each other is however challenging primarily due to the exponential scaling of possible end-effector poses with the number of dimensions. In particular, this is a multi-modal problem, and the predicted unconditioned value maps $Q_{unc}(x, y, \theta)$ have multiple peaks (as in Fig. 6). While unconditioned value maps may provide information relevant for downstream bimanual tasks, such as the grasp success, they provide no information regarding their correspondences. To address this we define *correspondence descriptors*

$$\boldsymbol{d} = (Q_{unc}, x, y, \sin\theta, \cos\theta, m, \boldsymbol{e})$$

where $\boldsymbol{e} \in \mathbb{R}^M$ is a learned embedding for each pixel (disregarding orientations θ) concatenated with the unconditioned value Q_{unc}, positional encodings, and the action primitive type m. Then, the final conditioned value $Q(\boldsymbol{d}_1, \boldsymbol{d}_2)$ depends on a descriptor pair.

Fig. 3 shows the complete information flow of BiMaMa-Net: A shared encoder using a ResNext-50 [27] backbone maps an input image (e.g. depth and grayscale) to high-level features. First, a classification head predicts the manipulation primitive m. For a *Sufficiently Smoothed* primitive, no further action is required. For all other learned primitives, a U-Net [28] decoder predicts value maps for a discrete number N of end-effector orientations θ. We choose a U-Net architecture over fully convolutional NN used in prior robotics manipulation works [29], [30], [10], [25] as U-Nets are better suited for high-resolution inputs that we find necessary for detecting edges and wrinkles for garment smoothing.

Then, *BiMaMa-Net* samples a set of poses from the value map, where pixels with higher values are more likely to get sampled. During training, BiMaMa-Net samples from $p(a|s) \sim \sqrt{Q_{unc}(a, s)}$ to allow for sampling negative examples to better estimate the underlying distribution of action values. For inference, BiMaMa-Net samples from $p(a|s) \sim Q_{unc}(a, s)^2$ which emphasizes action poses with high values. It then calculates the correspondence descriptors for each pose, and a final NN head combines all descriptor pairs $\boldsymbol{d}_1$, $\boldsymbol{d}_2$ to output the final conditioned action value Q.

If two poses are interchangeable (e.g., during a fling or a drag action), a single decoder predicts the value maps per θ. However, if a certain relation between the poses must be maintained (e.g., the conceptual difference between the pick and the place poses in a pick-and-place action) then separate decoders compute two value maps Q^1_{unc} and Q^2_{unc}.

C. Reachability Calibration

As shown in Fig. 6, to ensure reliable garment smoothing and folding, the robot should compute the actions that maximize the expected reward within the reachable space.

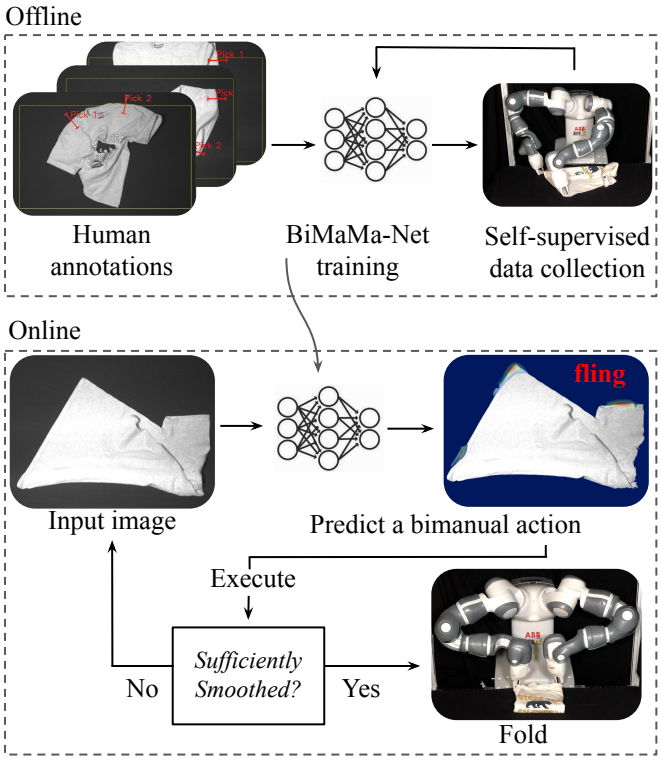


Fig. 4: **SpeedFolding Pipeline.** We start by manually annotating input images with primitives and gripper poses, train a NN and then iteratively use the NN for self-supervised data collection (**top**). During runtime, we use the NN to predict a primitive and a pair of poses given an input image and execute it on the robot. If the resulting garment configuration is classified as Sufficiently Smoothed the robot will fold the garment, otherwise it will repeat the process.

To find the robot's reachable space, we perform a one-time boundary search along a discretized grid in the action space (x, y, θ) for each gripper, assuming a constant height z above the table. The search, done separately for each θ, starts with a fixed lower value of y and increases x until the inverse kinematics fails to find a solution. Afterwards, it repeatedly increases y or decreases x so that the search is confined to the continuous boundary at which reachability fails. As a result, we get masks M_l and M_r for the left and right arms

$$M(x, y, \theta) \rightarrow \{0, 1\}$$

that can be incorporated into BiMaMa-Net as spatial binary constraints by restricting the action sampling to the masks. To ensure that each reachability mask contains at least one pose, we create up to four masked value maps from Q^1_{unc} (or Q^2_{unc}) by multiplying them with M_l or M_r: $\{Q^{1l}_{unc}, Q^{1r}_{unc}, Q^{2l}_{unc}, Q^{2r}_{unc}\}$. An action value $Q_{unc} = 0$ is ignored in the sampling process. We then sample and combine the correspondence descriptors from Q^{1l}_{unc} and Q^{2r}_{unc}, and vice versa for Q^{1r}_{unc} and Q^{2l}_{unc}.

We find that using a calibrated reachability mask for each end-effector orientation θ separately significantly reduces the number of false negatives that arise when using approximations, such as a circular mask. After selecting the final,

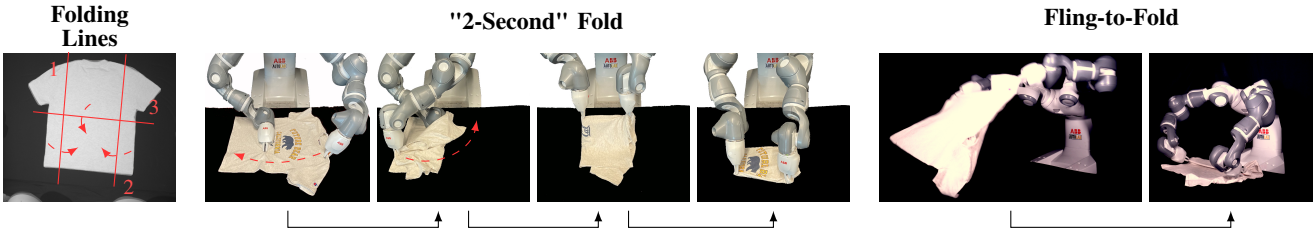


Fig. 5: **Folding Approaches.** We compare three approaches for folding. Left: A template mask with a sequence of folding lines that is compiled to a number of bimanual pick-and-place actions. Center: A so-called "2-second" folding heuristic that applies only very few steps however is for t-shirts only [31]. Right: A *fling-to-fold* primitive that combines a fling with an immediate folding action. Here, the garment does not need to be fully smoothed, however prior knowledge is required.

reachable poses $(x, y, \theta)_1$ and $(x, y, \theta)_2$ during runtime, we check for possible collisions due to inter-arm interaction. If a potential collision is detected, the next best action is selected until reachable and collision-free poses are found.

D. Training for Smoothing

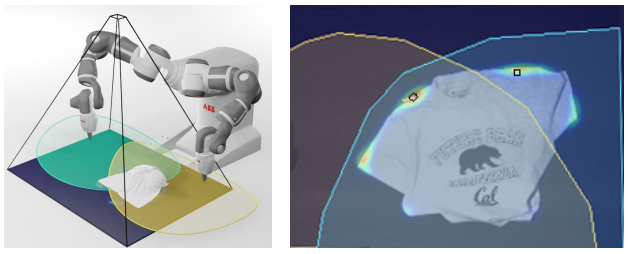
We train BiMaMa-Net via self-supervised real-world learning to predict the manipulation primitive type m and the corresponding action poses $(x, y, \theta)_1$ and $(x, y, \theta)_2$ given an overhead image of a garment.

In order to scale real-world interaction, the learning process is designed for minimal human intervention. First, we collect examples of smooth configurations to train a classifier outputting the confidence $p(\textit{Sufficiently Smoothed}|s)$. Additionally, let $\textit{cov}(s)$ be the coverage of the garment at configuration s observed from an overhead perspective, calculated by background subtraction and color filtering. We define the reward r:

$$r_t = \max\left(\tanh\left[\alpha\ \left(\textit{cov}(s_{t+1}) - \textit{cov}(s_t)\right)\right.\right.$$
$$\left.\left. +\ \ \beta\ \left(p(\textit{smoothed}|s_{t+1}) - p(\textit{smoothed}|s_t)\right)\right], 0\right)$$

as the sum of the *change* of coverage and *Sufficiently Smoothed* confidence with tuned weights α and β respectively. It is scaled to $r \in [-1, 1]$ first and then clipped to a non-negative value, so that no change equals zero reward. To ensure continuous training, the robot resets the garment configuration by grasping it at a random position on its mask and dropping it from a fixed height. We iteratively train a self-supervised data collection NN, interleaving training and execution (Fig. 4). The robot explores different actions by uniformly sampling from the set of N_s best actions.

To avoid a purely random and sample-inefficient initial exploration, we kickstart the training with human annotations. We differentiate between self-supervised and human annotated data within the training process in three ways: (1) We set the reward of human annotated data to a fixed r_h. (2) Besides training the value map at the specific annotated pixel position and orientation, we follow [26] and introduce a Gaussian decay centered around each pose as a global target value instead. (3) The classification head is trained only with data that has a reward higher than a tuned threshold $r \geq r_c$.



(a) Workspace (b) Reachability mask

Fig. 6: **Reachability.** (a) We perform a boundary search to compute separate reachability masks for the left (yellow) and right (blue) robot arms. (b) BiMaMa-Net guarantees at least one pick pose (black) from the value map within each mask.

E. Folding Pipeline

We compare three approaches for folding: *instruction-based folding*, which can be adapted to different garments and different folding techniques, *"2-second" fold*, a known heuristic for surprisingly fast t-shirt folding, and *fling-to-fold* (F2F), a novel technique that can increase the number of folds-per-hour (FPH) by leveraging prior knowledge about the t-shirt's dimensions.

Instruction-based Folding: As shown in Fig. 5 (left), given a mask of a smoothed garment, the robot iteratively folds the garment along user-specified *folding lines*. These allow to define the goal configuration of a smooth garment precisely without using high-dimensional visual goal representations [5], [6]. A complete user instruction includes: (1) A binary mask called template and (2) a list of folding lines relative to the template (Fig. 5 left). The folding direction is defined with respect to the line according to the right-hand-rule.

To execute the folding lines, a particle-swarm optimizer computes an affine transformation by registering the template with the current image. Afterwards, SpeedFolding calculates corresponding poses for a bimanual fold action: Let $\boldsymbol{p}_{ent}$ be the first and $\boldsymbol{p}_{exit}$ be the second intersection point of the line and the mask, where the line enters and exits the mask respectively. This splits the mask into a *base* and a *fold-on-top* part. On the contour of the latter, the algorithm finds two pick points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ so that the area of the four-sided polygon $(\boldsymbol{p}_{ent}, \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_{exit})$ is maximized. We use the normal at the pick point for the gripper orientation θ. The place poses are calculated by mirroring the pick poses at the folding line.

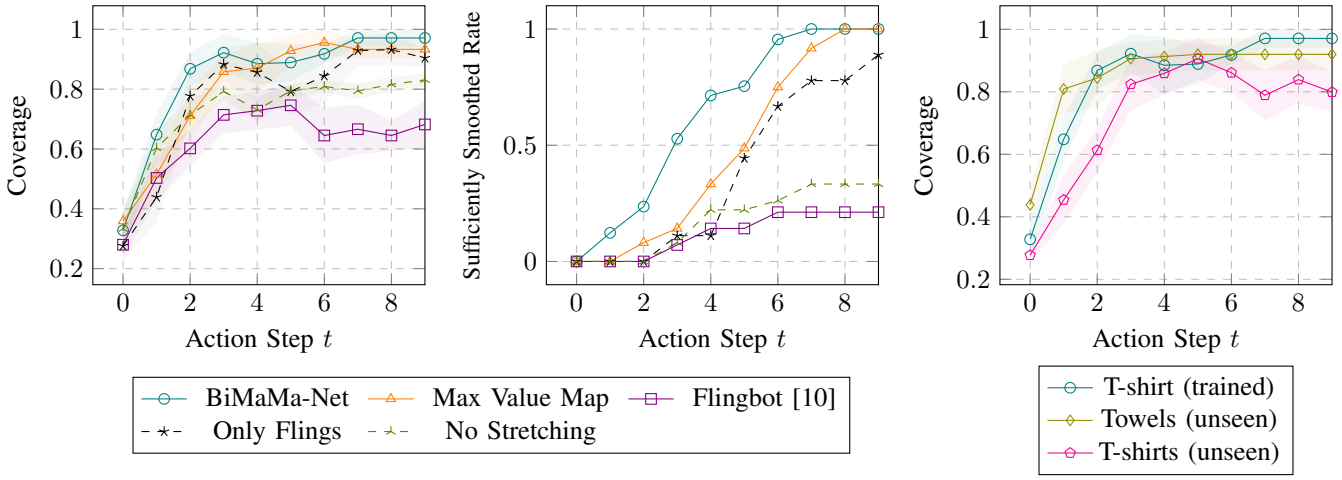


Fig. 7: **Garment smoothing** until it is Sufficiently Smoothed. We compare the normalized coverage (left) and prediction of the learned *Sufficiently Smoothed* classifier (center) over the number of action steps with different baseline methods. The system is able to generalize to unseen garments of different color, patterns, and material (right).

"2-Second" Fold: For specific garments such as t-shirts, there exist heuristics for efficient folding. Given a smooth configuration, the "2-second" fold follows a set of steps that requires using two arms simultaneously (Fig. 5), and is therefore well suited for a bimanual robot [31].

Fling-to-fold (F2F): We observe that (1) a fling action while grasping a sleeve and the non-diagonal bottom corner is especially effective and (2) the first fold action grasps the same points. We conclude that these two steps can be merged to reduce imaging and motion time. We implement F2F by adding a learned primitive to BiMaMa-Net that computes these pick points if visible. The primitive's motion is implemented by combining a fling with a consecutive fold action (Fig. 5). To ensure that the t-shirt is folded correctly, prior knowledge about the t-shirt's dimension is required to adapt the height of each arm prior to the fold.

V. Experiments

We experimentally evaluate the garment smoothing and folding performance of SpeedFolding on a known t-shirt, as well as on two garments unseen during training.

A. Experimental Setup

We perform experiments on a physical ABB YuMi robot with parallel-jaw grippers. The gripper's fingertips are extended by small 3D printed teeth to improve grasping. A thin sponge mattress is placed on the workspace to allow the grippers to reach below the garment without colliding. A Photoneo PhoXi captures overhead grayscale and depth images of the workspace, generating observations $o_t \in \mathbb{R}^{256\times192\times2}$. As the garment is frequently outside the camera's field of view, a 1080P GESMATEK RGB webcam is mounted above the workspace and used for coverage calculation. Computing is done on a system using an Intel i7-6850K CPU, 32GB RAM, and a NVIDIA GeForce RTX 2080 Ti.

We first perform data collection, and train IV-D on a single t-shirt. Initially, 600 scenes of random garment configuration were recorded and manually annotated in 1 h. After training a first NN, the robot collected 2200 self-supervised actions in 16 h. To include data of less frequently observed actions, we copied and re-annotated 1500 actions in 3 h, resulting in a dataset of 4300 actions in total. We used a single t-shirt shown in Fig. 1 throughout the training. We further perform data augmentation, including random translations, rotations, flips, resizes, brightness and contrast changes. We use $N = 20$ gripper orientations equally distributed over $[0, 2\pi)$ to implement the BiMaMa-Net decoder as described in Sec. IV-B. For training, we manually tune $N_s = 50$, $r_h = 0.8$ and $r_c = 0.3$ (see Sec. IV-D).

We design a set of garment smoothing and folding experiments to evaluate SpeedFolding. Initial garment configurations are generated by environment resets as described in IV-D. Each experiment is averaged over 15 trials. We ignore experiments that terminate early due to a motion planning error, as this is not the focus of this paper. A trial is considered unsuccessful if the garment was not successfully folded according to the majority vote of three reviewers or the number of actions exceeded a maximal horizon of $H = 10$. We define a grasp success if the gripper holds the garment *after* an executed action. For known garments, BiMaMa-Net achieves a grasp success rate of over 96 %.

B. Sufficiently Smoothed

We evaluate garment smoothing using two metrics: The garment coverage, computed from an overhead image, and a binary Sufficiently Smoothed value, predicted using the Sufficiently Smoothed classifier. We compare BiMaMa-Net to two baseline (1) *Max Value Map*, a variant of BiMaMa-Net that computes the pick points directly from the value maps Q_{unc} by computing the maximum over the map to find two pick points without using correspondence descriptors, (2) *Only Flings*, a variant restricted to fling actions only, and (3) *Flingbot*, the pre-trained method from [10] (see Fig. 7). Results suggest that BiMaMa-Net is able to smooth

TABLE I: **End-to-end folding** for different NN architectures, folding approaches, and garments, averaged over 15 trials per experiment. The durations are averaged over successful folds, while the cycle time and FPH are averaged over both successful and unsuccessful folds.

Method	Folding Approach	Garment	Smoothing Actions	Duration [s]	Fold Success	Cycle Time [s]	Folds Per Hour (FPH)
Max Value Map	Instruction	T-shirt	5.1 ± 0.5	133.9 ± 7.4	80 %	167.4 ± 9.2	21.5 ± 1.2
BiMaMa-Net	Instruction	T-shirt	3.0 ± 0.4	108.7 ± 7.3	**93 %**	116.9 ± 7.9	30.8 ± 2.1
	”2-Second“ Fold		3.0 ± 0.4	97.3 ± 4.8	53 %	182.4 ± 5.4	19.7 ± 0.6
	Fling-to-fold		**1.8 ± 0.2**	**81.7 ± 4.3**	**93 %**	**87.9 ± 4.7**	**40.9 ± 2.2**
Garments Unseen During Training							
BiMaMa-Net	Instruction	Towel	1.7 ± 0.2	59.2 ± 3.8	87 %	68.1 ± 4.4	52.9 ± 3.4
		T-shirt	4.8 ± 0.4	141.1 ± 8.7	80 %	176.3 ± 10.9	20.4 ± 1.3

a known t-shirt to a Sufficiently Smoothed configuration in ~ 3 fewer steps compared to baselines requiring ~ 5. Although the increase of coverage is similar to Only Flings, the latter reaches a Sufficiently Smoothed configuration later or even fails to do so, confirming the need of additional action primitives to fully smooth a garment.

We note that the FlingBot baseline fails to reach an $80\,\%$ coverage as reported in [10], as we observe frequent grasp failures presumably due to differences in the physical setting. We ablate the stretching motion before a fling and observe that stretching leads to higher coverage.

C. Folds per Hour

Table I shows results of end-to-end garments folding experiments. BiMaMa-Net manages to (1) successfully fold garments in over $90\,\%$ of the trials on known garments and (2) $30\,\%$ faster than the Max Value Map baseline using the Instruction-based folding approach. The ”2-second“ fold achieves an additional speedup of $10.4\,\%$ when executed successfully, however we find that it is sensitive to t-shirt's orientation in a Sufficiently Smoothed configuration and suffers from a low fold success rate. With prior information on the t-shirt's dimensions, F2F uses $40\,\%$ less smoothing actions and imaging time. As a result, it achieves a speedup of over $25\,\%$ compared to the instruction-based approach, leading to 40.9 folds per hour on average. Calculating an action using BiMaMa-Net takes $(126.0 \pm 0.9)\,\mathrm{ms}$ on our hardware.

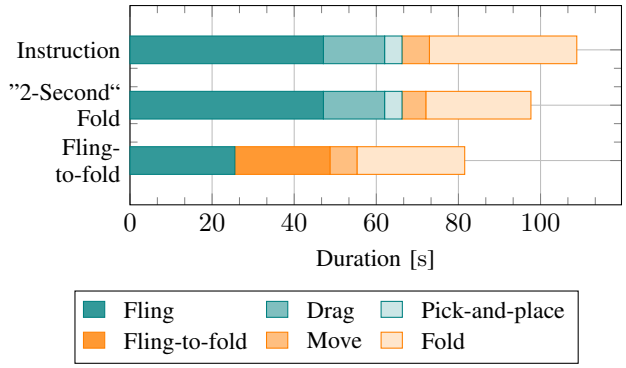


Fig. 8: **Timings** for calculating and executing the action primitive types depending on the folding approach. Instruction-based and ”2-Second“ fold share the same smoothing actions (blue), but differ in folding (orange). By introducing a combined Fling-to-fold primitive, a smoothed state is not required before folding. However, the ”2-Second“ fold is suited for manipulating a t-shirt only, and Fling-to-fold assumes prior knowledge of the garment.

D. Generalization to Unseen Garments

We explore how SpeedFolding, trained on a single t-shirt, can generalize to garments unseen during training. In these experiments we use (1) a t-shirt with a different color and stiffness and (2) a rectangular towel with a different color compared to the original t-shirt. We evaluate SpeedFolding on unseen garments using instruction-based folding, as this is the only approach that easily adapts to general garments. We run the same experiments on the unseen t-shirt with no changes to the BiMaMa-Net model or the folding template. In contrast, when we run the towel experiments we observe that the system fails to classify a Sufficiently Smoothed configuration, as the object's shape is different from that BiMaMa-Net was trained on. To address this, we add 20 Sufficiently Smoothed towel images to the dataset and re-train BiMaMa-Net. Table I suggests that SpeedFolding can generalize to garments with different color, stiffness and shape.

E. System Limitations

Grasp failures, especially during a fling motion, can decrease the garment's coverage dramatically. We find that most grasp failures happen due to losing the grip during the stretching motion prior to a fling action. This limitation can be mitigated using improved force feedback or by adding visual feedback. We observe a frequent failure case during top-down grasps while executing the first step of the ”2-second“ fold. These grasps may require different gripper jaws that are better suited for top-down grasps.

As common with data-driven methods, SpeedFolding can generalize to *similar* unseen garments. For example, textile patterns may be more challenging to detect and classify correctly. This limitation can be addressed through additional data augmentation. Generalization to different garment shapes may also be limited, and can be addressed by adding examples of Sufficiently Smoothed configurations to the dataset, as described in Sec. V-D for the towel example.

VI. Conclusion and Discussion

We presented SpeedFolding, a bimanual robotic system for efficient folding of garments from arbitrary initial configura-

tions. At its core, a novel BiMaMa-Net architecture predicts two conditioned poses to parameterize a set of manipulation primitives. After learning from 4300 human-annotated or self-supervised actions, the robot is able to fold garments in under $120\,\mathrm{s}$ on average with a success rate of $93\,\%$.

While prior works e.g. by Maitin-Shepard et al. [4] or Doumanoglou et al. [3] achieved high success rate for end-to-end cloth folding, cycle times for a single fold were on the order of $3-6$ Folds Per Hour (FPH), whereas SpeedFolding achieves 30 to $40\,\mathrm{FPH}$. Similar to Ha et al. [10], the fling primitive can unfold the garment in a few actions. In contrast however, we introduce additional action primitives that enable the robot to reach a sufficiently smoothed configuration. In future work, we will explore methods that can learn to manipulate a novel garment given a few demonstrations.

ACKNOWLEDGMENT

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. We thank Max Cao and Huy Ha for their helpful feedback.

REFERENCES

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics and Automation Magazine*, 2021.

[2] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali *et al.*, "Learning dense visual correspondences in simulation to smooth and fold real fabrics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 515–11 522.

[3] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[4] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. 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*. IEEE, 2010, pp. 2308–2315.

[5] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *Conference on Robot Learning*. PMLR, 2022, pp. 192–202.

[6] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *Robotics: Science and Systems (RSS)*, 2020.

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

[8] L. Yunliang Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," *arXiv e-prints*, pp. arXiv–2206, 2022.

[9] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1413–1419.

[10] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*. PMLR, 2022, pp. 24–33.

[11] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—a survey," *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[12] P. Lertkultanon and Q.-C. Pham, "A certified-complete bimanual manipulation planner," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1355–1368, 2018.

[13] S. Hayakawa, W. Wan, K. Koyama, and K. Harada, "A dual-arm robot that autonomously lifts up and tumbles heavy plates using crane pulley blocks," *IEEE Transactions on Automation Science and Engineering*, 2021.

[14] Y. Hu, L. Zhang, W. Li, and G.-Z. Yang, "Robotic sewing and knot tying for personalized stent graft manufacturing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 754–760.

[15] A. Edsinger and C. C. Kemp, "Two arms are better than one: A behavior based control system for assistive bimanual manipulation," in *Recent progress in robotics: Viable robotic service to human*. Springer, 2007, pp. 345–355.

[16] I. Garcia-Camacho, M. Lippi, M. C. Welle, H. Yin, R. Antonova, A. Varava, J. Borras, C. Torras, A. Marino, G. Alenya *et al.*, "Benchmarking bimanual cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1111–1118, 2020.

[17] L. Sun, G. Aragon-Camarasa, P. Cockshott, S. Rogers, and J. P. Siebert, "A heuristic-based approach for flattening wrinkled clothes," in *Conference Towards Autonomous Robotic Systems*. Springer, 2013, pp. 148–160.

[18] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4871–4876.

[19] D. J. Balkcom and M. T. Mason, "Robotic origami folding," *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 613–627, 2008.

[20] K. Tanaka, Y. Kamotani, and Y. Yokokohji, "Origami folding by a robotic hand," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2540–2547.

[21] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.

[22] D. Tanaka, S. Arnold, and K. Yamazaki, "Emd net: An encode–manipulate–decode network for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.

[23] R. Lee, D. Ward, A. Cosgun, V. Dasagi, P. Corke, and J. Leitner, "Learning arbitrary-goal fabric folding with one hour of real robot experience," *arXiv preprint arXiv:2010.03209*, 2020.

[24] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. 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)*. IEEE, 2018, pp. 4238–4245.

[25] L. Berscheid, P. Meißner, and T. Kröger, "Robot learning of shifting objects for grasping in cluttered environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 612–618.

[26] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, V. Viswanath, M. Laskey, J. Gonzalez, and K. Goldberg, "Untangling dense knots by learning task-relevant keypoints," in *Conference on Robot Learning*. PMLR, 2021, pp. 782–800.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[29] L. Berscheid, P. Meißner, and T. Kröger, "Self-supervised learning for precise pick-and-place without object model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4828–4835, 2020.

[30] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[31] "How to fold a t-shirt in two seconds," https://www.wikihow.com/Fold-a-T-Shirt-in-Two-Seconds, accessed: 2022-03-01.