UniFolding: Towards Sample-efficient, Scalable, and Generalizable Robotic Garment Folding

Anonymous Author(s)
Affiliation
Address
email

Abstract: The paper discusses developing a scalable, generalizable robotic system, UniFolding, for folding diverse garments. It identifies three challenges: long-horizon processes, unanticipated corner cases, and real-world data adoption for refinement. UniFolding, utilizing a proposed UFONet neural network, integrates the unfolding and folding decisions into one policy model, adaptable to various garment types and states. Rather than relying on heuristic rules, it handles corner cases by augmenting training data continuously. UniFolding's design is built on a garment's partial point cloud, promoting generalization and reducing sensitivity to texture and shape differences. It's tested on two garment types, long and short-sleeve shirts. Training data is collected via a human-centric process with offline and online stages. Offline recording involves human folding actions via VR, and online uses a real-world model fine-tuned with a human-in-the-loop approach. System performance is assessed on 20 shirts, evaluating success rate.

More experiments and videos can be found in the supplementary materials and on the website: https://sites.google.com/view/unifolding.

Keywords: Garment Unfolding, Garment Folding

1 Introduction

Garment folding has been a long-standing task in the robotics community, with the potential to automate this process and enhance the quality of life by reducing human labor. However, despite recent advancements in learning methods for garment folding [1, 2, 3], they still struggle to efficiently handle the wide variety of garments within the same category that differs in size, texture, and material. This limitation hampers the scalability and applicability of these methods in real-world applications.

We identify three challenges to building a scalable and generalizable folding system: (1). **Long horizon.** Garment folding is a classic long-horizon decision process, which takes multiple stages (*i.e.*, *unfolding* and *folding*) to complete a folding task. An automatic folding system should be able to determine the appropriate action (*e.g.*, *fling*, *pick-and-place*, *drag*, and *mop*) based on the observed state of the garment. The policy model should be adaptive to different garments and arbitrary states. (2). **Corner cases.** Heuristic rules for *unfolding* and *folding*, which are widely adopted in previous works [1, 4, 2] cannot cover unexpected corner cases. To cover these, the rule system may grow too complex. (3). **Real-world data**. Data acquisition strategy that works in simulation with a large number of trials is infeasible in the real world. However, due to the sim-to-real gap, many garment states and dynamics cannot easily be covered in simulation. Thus adopting real-world data for fine-tuning is desired, but how to utilize and annotate the real-world data is non-trivial.

To address these challenges, we propose a novel robot folding system, **UniFolding**. It leverages an end-to-end neural network **UFONet** to make action decisions (i.e., classify the primitive action to perform and regress the actioning points) based on the input of the partial point cloud of the garment. In this way, it unifies the decision-making process of *unfolding* and *folding* into a single policy model. In contrast, previous systems [1, 5, 4, 2] treat them separately. Another merit of the

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

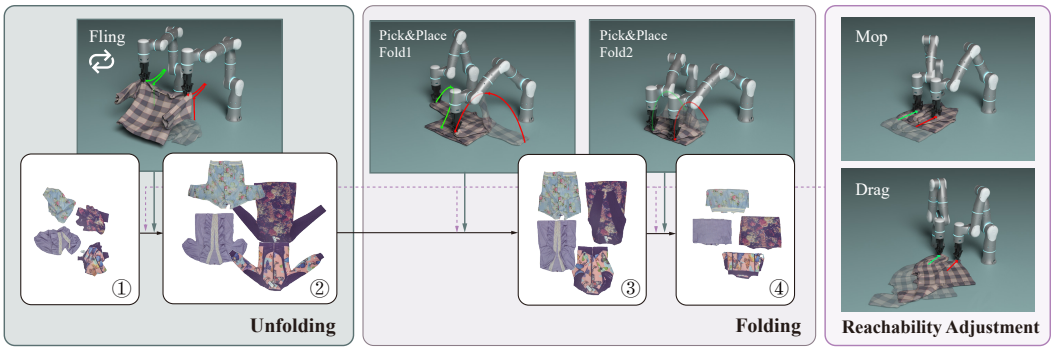

Figure 1: The pipeline of UniFolding system. It contains two stages to fully fold a garment from a random initial state, namely *Unfolding* and *Folding*.

unified policy design is that instead of making increasingly complex heuristic rules, it can overcome countless corner cases by continuing to add the training data. Besides, the point cloud-based design is less sensitive to the texture or shape diversity than the 2D-based solutions [1, 2], and thus it can greatly enhance the generalization ability of the system.

To train **UFONet**, we first standardize the complete folding procedures for the garment categories we focus on in this work: long-sleeve shirts and short-sleeve T-shirts. As depicted in Figure 1, the garment initialized with arbitrary configuration is unfolded by *fling* actions, and then it is folded with a two-step *pick-and-place* process. Consequently, in the **UFONet**, we have three primitive actions to identify based on the current observation, including *fling*, *pick-and-place (fold1)*, and *pick-and-place (fold2)*. For each action, **UFONet** predicts several keypoint candidates and selects a pair to operate with the highest score under a certain criterion. Aside from the actions directly decided by **UFONet**, we also have *drag* and *mop* actions, which are triggered when the prediction of the pick or place points on the garment is beyond the reachability of robots.

As for the data collection, we consider making real-world fine-tuning feasible as the first priority and design a human-centric pipeline: offline data collection in simulation with human demonstration, and online data collection in the real world with human-in-the-loop. In the offline data collection phase, we record operation sequences of human folding garments via a VR interface. The data collected in this phase is to train an initial model. In the online data collection phase, we deploy the initial model in the real world and fine-tune the model by re-ranking the points from the predicted candidates. To note, humans only need a few annotations in the online data collection phase can largely improve the keypoint prediction performance. In comparison with self-exploration guided by reward functions, it is more sample-efficient. And in comparison with pure heuristic rules, it is more adaptive.

To evaluate the folding system, we conduct experiments using long-sleeve shirts and short-sleeve shirts. We measure the success rate to assess the performance of our approach in Sec. 5.

We summarize our contribution as follows:

- We propose a novel robotic folding system, **UniFolding**, that can support the complete garment folding pipeline including *unfolding* and *folding*.
- We propose **UFONet**, a value-based end-to-end policy model to make decisions on action type and actioning points for the complete folding pipeline.
- We conduct extensive real robot experiments on 10 *unseen* long-sleeve shirts and 10 *unseen* short-sleeve T-shirts to demonstrate the generalization ability and robustness of our system.

2 Related Works

Robotic garment folding, a key research topic in computer vision and robotics, often starts with simple square-shaped items. Both heuristics-based [6, 7, 8] and learning-based approaches [9, 10, 11] prove effective in flattening or folding such items.

In the comprehensive garment folding pipeline outlined in previous work [4], the process begins with an *unfolding* phase, moving the garment from an arbitrary to a preferred configuration, aiding the *folding* phase. The diverse and complex configurations needing handling make *unfolding* more challenging.

Early *unfolding* studies focused on garment geometric features like corners and wrinkles [4], using heuristic policies. Later, adaptive manipulation policies were developed using human demonstrations [3] and self-supervised learning [5]. Notably, recent work Speedfolding [2] combines human demonstration with self-learning to unfold and smooth garments.

Folding research has shifted from heuristic [4, 8, 12, 13, 14] to learning-based methods [15, 16], usually framed as goal-conditioned RL problems with predefined goals.

Learning-based complete pipelines [4, 2, 1] for *unfolding* and *folding* exist but are separate processes. Unlike previous pipelines, our work integrates *unfolding* and *folding* into a unified end-to-end policy model, the **UFONet**.

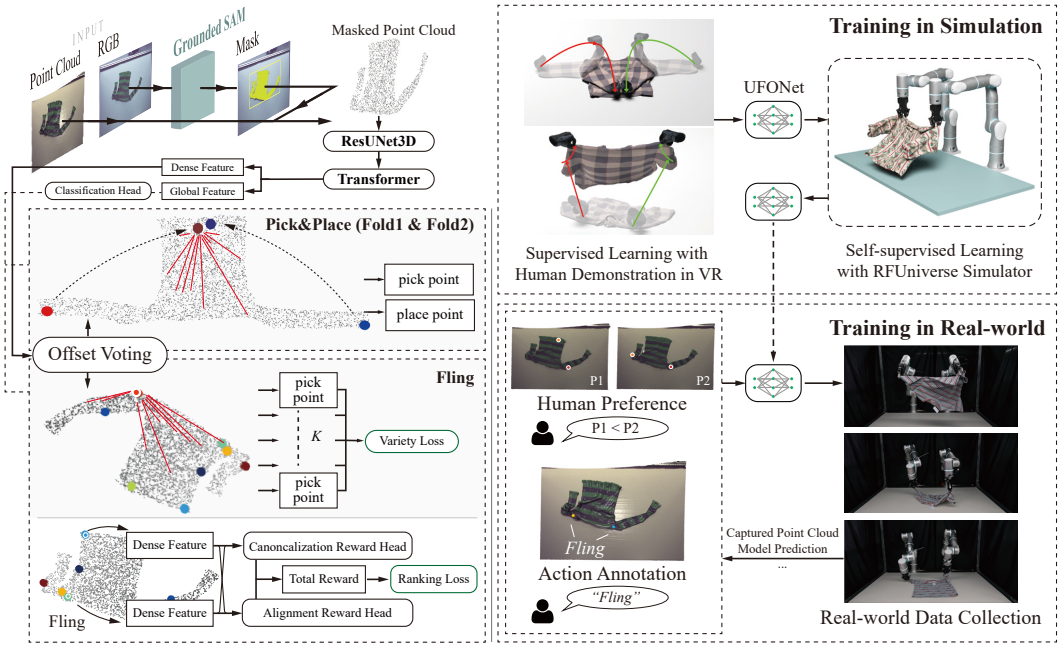

Figure 2: **Left**: UFONet takes a masked point cloud of the observed garment state as the input, classifies the primitive action to be performed, and regresses the actioning points. **Right**: The offline and online training strategies for UFONet.

3 UniFolding

Starting with an RGB-D observation $\boldsymbol{I}_0 \in \mathbb{R}^{W \times H \times 4}$ of a garment's configuration $\boldsymbol{s}_0$, UniFolding employs a dual-arm robot to sequentially transform it to the desired state $\boldsymbol{s}^*$ using UFONet to determine the action type $m \in \mathcal{M}$ and its parameters from evolving RGB-D observations $\boldsymbol{I}_t$. These parameters contain pick points $\boldsymbol{p}_i = (x_i, y_i, z_i)$ and placement locations $\boldsymbol{q}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$ where $i = 1, 2$ represents the left and right arm respectively. If the points are unreachable, UniFolding employs two primitive actions, *drag* and *mop* to ensure the process works smoothly.

The pipeline is shown in Fig. 1. The primitive actions, UFONet's design, and its training are detailed in Sec. 3.1, 3.2, and 3.3. respectively.

3.1 Action Primitive

Fling The ABB YuMi robot's fling operation in SpeedFolding is adapted to the Flexiv Rizon robot arm by modifying force thresholds, velocity, and trajectory parameters. The fling parameters are $\boldsymbol{a}_f = (\boldsymbol{p}_1; \boldsymbol{p}_2)$.

Pick-and-Place (fold 1 & fold 2) Given two pick points and two place locations from the camera space, the arms first pick the pick points, move to a certain height above the place locations, and release the grasp. The parameters of *pick-and-place* is $\boldsymbol{a}_{p\&p} = (\boldsymbol{p}_1, \boldsymbol{q}_1; \boldsymbol{p}_2, \boldsymbol{q}_2)$.

Drag & Mop If pick points or place locations are out of dual-arm reach, we change the garment position using rule-based points and trajectory, making unfolding and folding actions feasible.

3.2 UFONet for Garment Unfolding and Folding

For garment observation $\boldsymbol{I}_t$ at time step t, **UFONet** will first convert $\boldsymbol{I}_t$ into a point cloud $\boldsymbol{o}_t$ and random sample it so that $\boldsymbol{o}_t \in \mathbb{R}^{N\times 3}$, where N is the number of points. Then, it predicts the next action type and corresponding parameters from three primitive actions: *fling*, *fold1*, *fold2*. As the actions do not share the same parameter space, we predict $\boldsymbol{a}_{f,t}$ and $\boldsymbol{a}_{p\&p,t}$ in different branches. The overall framework design is illustrated in Fig. 2.

Image Processing We adopt the Grounded-SAM [17, 18] model to segment the RGB image from $\boldsymbol{I}_t$ with prompt "cloth", multiply the mask by the depth image and convert the masked depth to point cloud $\boldsymbol{o}_t$ based on camera intrinsic parameters.

Feature Extraction We adopt a ResUNet3D [19] model to extract features from $\boldsymbol{o}_t$. The ResUNet3D [19] model is an efficient 3-D CNN architecture based on sparse convolution that is well-suited for extracting high-resolution features from 3D data. The extracted features are then passed to a self-attention module based on Transformer [20], which processes the features and produces two sets of outputs: global features $\mathcal{F}_g \in \mathbb{R}^{128}$ and per-point dense features $\mathcal{F}_d \in \mathbb{R}^{N\times 128}$.

Action Classification $\mathcal{F}_g$ generated by the Transformer model is fed into a 3-layer MLP classification head. Since the parameter spaces for each primitive action are not the same, the classification result will be used for selecting the action parameter set predicted from the dense feature $\mathcal{F}_d$.

Pick-And-Place Point Prediction We predict $\boldsymbol{a}_{p\&p,t}$ based on $\mathcal{F}_d$. Intuitively, humans prefer to select picking points $\boldsymbol{p}_t$ from the semantic-rich areas such as cuff, collar, waistline *etc*. With the definition of standard folding procedures within a category, both pick points $\boldsymbol{p}_t$ and place locations placement locations $\boldsymbol{q}_t$ tend to concentrate on a few areas. Thus, we predict one set of $\boldsymbol{a}_{p\&p,t}$ in this branch.

Fling Point Prediction We predict $\boldsymbol{a}_{f,t}$ based on $\mathcal{F}_d$. Unlike the *pick-and-place* operation, the pick point selection for the *fling* operation is much more ambiguous. It seems we can only judge whether the fling point prediction is good until it actually performs on the points. That's the main reason why previous works [5, 1] adopt the self-exploration approaches for fling point prediction. However, after analyzing the statistics from human demonstration data through VR, we surprisingly find that humans also have strong preferences for the *fling* operation: humans also tend to grasp semantic-rich areas for *fling* action (please see the supplementary file for more details). Thus, we choose to directly learn keypoint prediction from human demonstration data. However, due to the ambiguous nature, the keypoint distribution for the *fling* is not as concentrated as the *pick-and-place* operation. Thus, we leverage the multi-modal distribution feature and learn to predict K sets of $\boldsymbol{p}$ in this branch, where $\boldsymbol{P} = \{\boldsymbol{p}^{(j)}\}_{j=1,\ldots,K}$, supervised by a variety (Minimum-over-N) loss [21]:

$$L_{kp}(\boldsymbol{P}, \boldsymbol{p}^*) = \min_{\{\boldsymbol{p}^{(1)},\ldots,\boldsymbol{p}^{(K)}\}\in\boldsymbol{P}} \left\{ d\left(\boldsymbol{p}^*, \boldsymbol{p}^{(1)}\right), d\left(\boldsymbol{p}^*, \boldsymbol{p}^{(2)}\right), \ldots, d\left(\boldsymbol{p}^*, \boldsymbol{p}^{(K)}\right) \right\} \tag{1}$$

where $\boldsymbol{p}^*$ is the human-preferred point, and $d(\cdot,\cdot)$ is the distance metric. Intuitively, L_{kp} only supervises the predicted keypoint closest to the human-preferred point, which encourages the variety of the K predicted keypoints.

The prediction of keypoint candidates $\boldsymbol{P}$ is constructed by directly regressing keypoints in 3D camera space through attention-based offset voting [22]:

$$\boldsymbol{p}^{(j)} = \frac{1}{N}\sum_{k=1}^{N} w_{k,j}\left(\boldsymbol{x}_k + \boldsymbol{u}_{k,j}\right), \quad s.t. \sum_{k=1}^{N} w_{k,j} = 1 \tag{2}$$

where $\boldsymbol{p}^{(j)}$ is the j-th keypoint prediction, $w_{k,j} \in [0,1]$ is the attention score, $\boldsymbol{x}_k \in \boldsymbol{o}_t$ is the k-th point in the input point cloud $\boldsymbol{o}_t$, and $\boldsymbol{u}_{k,j}$ is the 3D offsets of the j-th keypoint $\boldsymbol{p}^{(j)}$ with respective

to the k-th point $\boldsymbol{x}_k$. The attention score $w_{k,j}$ and offsets $\boldsymbol{u}_{k,j}$ are predicted by MLP with dense features $\mathcal{F}_d$ as input.

Finally, we should select a keypoint pair from $\boldsymbol{P}$ to obtain $\boldsymbol{a}_{f,t}$. We design an evaluation module to score any two input keypoints. Specifically, for any two points with the indices of j and k in $\boldsymbol{P}$, we generate embeddings by Eq. 3:

$$\boldsymbol{e}_{j,k} = \text{MLP}([\boldsymbol{F}_j, \boldsymbol{p}^{(j)}, \boldsymbol{F}_k, \boldsymbol{p}^{(k)}]) \tag{3}$$

where $\boldsymbol{F}$ is the feature vector, defined as the weighted sum from the per-point dense feature $\mathcal{F}_d$.

Inspired by ClothFunnels [1], we predict two factorized scores given the embedding $\boldsymbol{e}_{j,k}$ as input, namely **Canoncalization score** R_{C} and **Alignment score** R_{A}. Please see the supplementary files for more details on R_{C} and R_{A}. Finally, we calculate the total score R_{CA} by Eq. 4.

$$R_{\text{CA}} = (1-\beta)R_{\text{C}} + \beta R_{\text{A}} \tag{4}$$

In the inference phase, we calculate $R_{\text{CA}}(\boldsymbol{e}_{j,k})$ for all $K(K-1)/2$ pairs of keypoint candidates, and choose the pair with the highest R_{CA} score as the final pick points.

3.2.1 Discussion: Action Poses Beyond Reachability

Prior methods [2] often use reachability masks to filter out unreachable poses on the action value map. While effective for small or singular garments, it fails for varied shapes and sizes. To resolve this, our model utilizes an active movement strategy: when the policy model's optimal action points are unreachable, it automatically switches to *drag* (in *unfolding* stage) or *mop* (in *folding* stage). *drag*'s pick points are selected from the closest keypoints to the robot bases, while *mop*'s are chosen via simple heuristics, as folded garments are typically well-shaped. These actions reposition the garment until it's 30cm from the dual-arm robot.

3.3 Data Collection & Network Training

Supervised Training with Human Demonstrations in VR We first train UFONet with the human demonstration data in VR. The keypoint candidates for *fling* action are supervised with variety loss defined in Eq. 1. The grasping and releasing points of *Pick-and-Place* action are supervised with Smooth-L1 loss. The VR dataset of human demonstrations contains 1218 manipulation videos for 203 short-sleeve T-shirts and 1575 videos for 315 long-sleeve shirts. Each video contains ~ 5 action steps on average to fully smooth and fold the garment. The total data collection time in VR takes about 16 hours. The training time for supervised training takes about 4 hours on one single RTX 3090 GPU for each category.

Self-supervised Training for Unfolding As shown in Fig. 2 (right), we continue to perform self-supervised training in simulation on the pre-trained model from supervised learning. It is used for training the score prediction head for *fling* action in Fig. 2 (left). The score R_{CA} is supervised with Smooth-L1 loss. We use RFUniverse [23] as the simulation environment, and the cloth simulation is based on ClothDynamics [24], which is a GPU-based cloth-specific physics engine in Unity [25]. The garment mesh models for simulation are selected from CLOTH3D [26] dataset. The training process in this stage is similar to ClothFunnels [1], except that our training is based on a pretrained model and we only need to choose pick points from a sparse set of keypoint candidates. Thus, our training is surprisingly sample-efficient, which only takes about 12 hours for data collection and model training on one single RTX 4090 GPU. In comparison, the training process of ClothFunnels [1] takes 2 days with 4 RTX 3090 GPUs.

Real-world Fine-tuning We develop an online learning framework for real-world fine-tuning. As shown in Fig. 2, in each episode, we first use the current policy model to automatically collect data for garment *unfolding*. Then human annotates the collected data samples, which will be used for training the policy model for the next episode. Specifically, the human annotator provides their preference on (1) the best action type and actioning points for the current garment state. (2). evaluation of the keypoint candidates. As the ground-truth R_{CA} score cannot be obtained in real-world scenarios, we fine-tune the evaluation model from human feedback. Given T randomly selected

Figure 3: Real-world hardware setup and **unseen** garment instances for testing.

keypoint pairs from all possible pairs. The annotator ranks the pairs by making M comparison. For each comparison, we denote the keypoint pair σ, and the two pairs to be compared σ_1 and σ_2. The annotator gives a label μ, where $\mu \in \{(0,1),(1,0),(0.5,0.5)\}$. The odds that σ_1 is superior than σ_2 is calculated by Eq. 5:

$$\hat{P}\left[\sigma_1 \succ \sigma_2\right] = \exp(\hat{R}_{\mathrm{CA}}(\sigma_1)) / (\exp(\hat{R}_{\mathrm{CA}}(\sigma_1)) + \exp(\hat{R}_{\mathrm{CA}}(\sigma_2))) \tag{5}$$

We use the following cross-entropy loss [27] in Eq. 6 to supervise the evaluation module, with the collection of annotations denoted as A:

$$\mathrm{loss}(\hat{r}) = -\sum_{(\sigma_1,\sigma_2,\mu)\in A} \mu(1)\log\hat{P}\left[\sigma_1 \succ \sigma_2\right] + \mu(2)\log\hat{P}\left[\sigma_2 \succ \sigma_1\right] \tag{6}$$

In total, we collected 2040 data samples in the real-world and annotated $M = 8$ comparisons for each data sample. The whole data collection and annotation process takes about 17 hours. Please see the supplementary files for more details on the process of human preference annotation and learning.

4 Real-world Experimental Setup

4.1 Garments in Real World

We tested two garment types: long-sleeve shirts, and short-sleeve T-shirts, choosing 60 real-world pieces of varying shapes, sizes, textures, and materials. Sizes ranged from $38cm \times 60cm$ to $80cm \times 167cm$, aspect ratios varied between $0.2695 : 1$ and $1.1167 : 1$, and materials such as cotton, polyester, spandex, nylon, viscose, wool and so on. See the supplementary files for additional garment details. The garments were divided into train/test sets at a $2 : 1$ ratio, with real-world fine-tuning garments selected from the training set, and the testing set garments remaining unseen in all experiments. Fig. 3 (b) showcases some test set examples.

4.2 Robots Setup

Fig. 3 shows our setup: two Flexiv Rizon robots with AG-95 grippers, equipped with torque sensors and sandpaper for increased friction, are placed at a rigid Polypropylene (PP) board table. A high-precision depth camera, Photoneo MotionCam3D, and an RGB camera, MindVision SUA202GC, are mounted above.

The *Contact Grasp* function from Flexiv control API allows adaptive control of gripper height during operations, ideal for our PP board surface. An automatic grasp failure detection mechanism is implemented to halt and retry action if required. For data collection, we randomly generate crumpled garment states by grasping and lifting at heights $[0.5m - 1.0m]$.

5 Experiment Results

5.1 Metrics

IoU (Intersection over Union) IoU measures the quality of garment unfolding by comparing the garment mask with the target T-shaped mask.

Normalized Coverage This is the ratio of the current top-down pixel count of the garment mask to the maximum count at the target T-shaped pose.

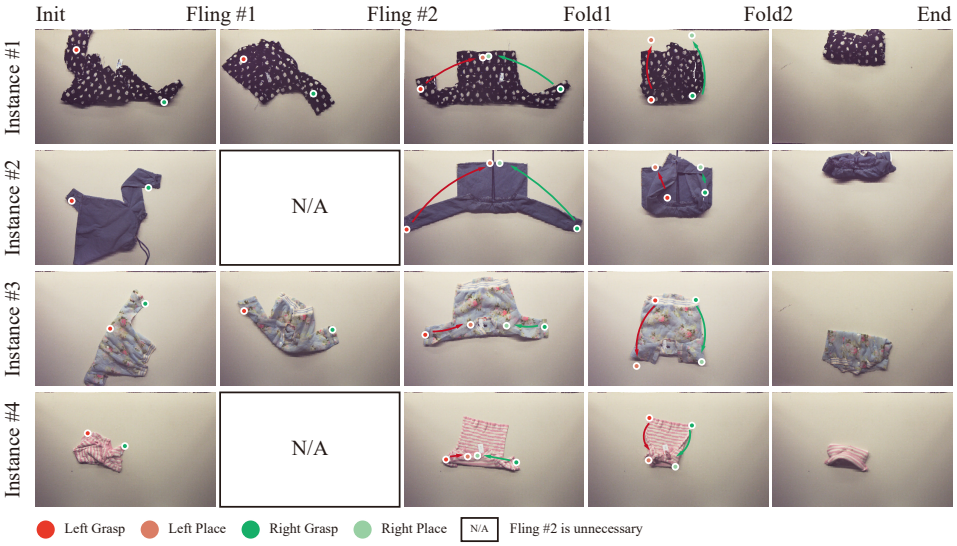

Figure 4: The figure illustrates the shape transformations of different types and sizes of clothes after applying each primitive action of the UniFolding system under various initial conditions. This process demonstrates the scalability, efficiency and generalization of our system.

Success Rate for End-to-end Folding Success rate averages over 10 trials, with failure defined by a crumpled initial state, unsuccessful folding as judged by three reviewers, or exceeding a 10-step action limit.

Table 1: The metrics of the end-to-end folding process for each **unseen** garment in the testing set.

Category	Long-sleeve Shirts										
ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	**Mean**
IoU	0.521	0.578	0.557	0.564	0.636	0.508	0.530	0.525	0.572	0.679	0.567
Coverage	0.732	0.782	0.750	0.776	0.815	0.698	0.831	0.765	0.702	0.878	0.772
Folding Success	50%	60%	60%	80%	70%	50%	80%	60%	70%	80%	66%
Category	Short-sleeve T-shirts										
ID	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	**Mean**
IoU	0.599	0.744	0.656	0.625	0.576	0.678	0.616	0.635	0.578	0.574	0.628
Coverage	0.867	0.760	0.789	0.735	0.729	0.862	0.759	0.695	0.892	0.875	0.796
Folding Success	30%	40%	60%	40%	60%	50%	30%	40%	50%	50%	45%

5.2 Unfolding Results

Tab. 1 shows some quantitative results of the metrics for unfolding task (*i.e.*, IoU and Normalized Coverage). We find that Shirts with high stiffness (*e.g.*, garment instance $\#5, \#7$) generally have larger IoU and Coverage during the unfolding process. In contrast, garments made of very soft materials (*e.g.*, garment $\#1$ and $\#6$) may have more trouble during the unfolding process, because they can easily be crumpled during manipulation.

5.3 End-to-end Folding Results

Qualitative results for end-to-end folding are illustrated in Fig. 4. Tab. 1 also shows the overall success rate for the whole manipulation process. The success rate of folding short-sleeved T-shirts is considerably lower than that of long-sleeved shirts, despite having higher IoU and Coverage. This is due to the high flatness required in the folding process of short sleeves. If the sleeves are curled up or covered, it may not affect the IoU or Coverage significantly, but it can cause significant difficulties in the subsequent folding process. We also observe a large variance in success rate on garments within and across categories, which indicates that the shape and physical material could greatly affect the difficulty of subsequent folding.

(a) Failure Cases (b) Generalization of *fold* primitive

Figure 5: (a) Failure cases and (b) *fold* primitive generalization across categories.

5.4 Generalization between categories

Flingbot[5] has proved that *fling* action could be transferred between objects with different shapes (*e.g.*, tower and T-shirt). What about transferring the learned folding actions between categories? In our setting, the original folding actions are slightly different for short-sleeve and long-sleeve shirts (*e.g.*, the folding direction of *fold2* are opposite to each other). We swap the pre-trained UFONet models for long-sleeved shirts and short-sleeved T-shirts, respectively attempting to predict the action poses of *fold1* and *fold2* in the other category. The visualization results of model predictions are shown in Fig. 5(b). We can see that the learned *fold* primitives can be directly transferred to other categories with different shapes, and they can even create new folding patterns in this way (*e.g.*, the long sleeves are folded towards the collar rather than the waist).

5.5 Ablation Study

Table 2: The mean metrics of all garments under different training settings.

	IoU	Coverage	Folding Success
UFONet only trained with human demonstration data in VR	0.378	0.542	16%
UFONet w.o. real-world fine-tuning	0.423	0.652	38%
UFONet with real-world fine-tuning	**0.597**	**0.784**	**55%**

Tab. 2 shows the mean metrics of all garments in the testing set under different training settings. We can see that the model only trained from human demonstration data perform very badly in the real-world (1st row in Tab. 2), due to the lack of negative samples. After self-supervised training in simulation, the success rate increases by a large margin (2nd row in Tab. 2). Finally, real-world fine-tuning can be very crucial (3rd row in Tab. 2) to achieve better results in the real world.

5.6 Limitations and Failure Cases

In our current implementation, if the grasping point on the garment has multiple layers, the robot gripper can NOT only grasp a single layer of cloth. Fig. 5(a) shows some failure cases in our experiments. **1. Self-entanglement state**. In practice, we find that only relying on *Fling* action can not fully flatten the garment in a self-entanglement state. More dexterous manipulation skills are required for this problem. **2. The garment is folded in half**. Most attempts to manipulate garments in this folded state will always grasp two layers of cloth, which may be stuck in a loop forever and fail to finish the unfolding task. **3. The front and back of the garment are separated**. In this state, the robot should accurately grasp the recognizable cuff with only one single layer of cloth to fully smooth the garment. However, such behavior is hard to accomplish in our system. **4. Garments with open zipper or buttons**. In order to fully smooth garments with open zippers or buttons, more dexterous manipulation skills are required.

6 Conclusion

In this work, we propose a novel system, UniFolding, to address the significant challenges associated with automating garment folding. This system leverages an end-to-end neural network, UFONet. Our system is data-efficient, thanks to our human-centric data collection and training policy. It is scalable, owing to the unified policy model and data-driven paradigm, and it is generalizable, given its ability to handle garments with large variations in size, shape, texture, and materials. We believe UniFolding is on track toward achieving full automation of the robotic garment folding task. In the future, we are interested in extending the capabilities of the UniFolding system to accommodate more garment categories.

References

[1] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. *arXiv preprint arXiv:2210.09347*, 2022.

[2] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2022.

[3] 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)*, pages 11515–11522. IEEE, 2021.

[4] 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*, 32(6):1461–1478, 2016.

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

[6] L. Sun, G. Aragon-Camarasa, P. Cockshott, S. Rogers, and J. P. Siebert. A heuristic-based approach for flattening wrinkled clothes. In *Towards Autonomous Robotic Systems: 14th Annual Conference, TAROS 2013, Oxford, UK, August 28–30, 2013, Revised Selected Papers 14*, pages 148–160. Springer, 2014.

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

[8] 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 *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315. IEEE, 2010.

[9] 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. *arXiv preprint arXiv:2003.09044*, 2020.

[10] 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)*, pages 9651–9658. IEEE, 2020.

[11] 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*, pages 192–202. PMLR, 2022.

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

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

[14] D. J. Balkcom and M. T. Mason. Robotic origami folding. *The International Journal of Robotics Research*, 27(5):613–627, 2008.

[15] D. Tanaka, S. Arnold, and K. Yamazaki. Emd net: An encode–manipulate–decode network for cloth manipulation. *IEEE Robotics and Automation Letters*, 3(3):1771–1778, 2018.

[16] R. Lee, D. Ward, V. Dasagi, A. Cosgun, J. Leitner, and P. Corke. Learning arbitrary-goal fabric folding with one hour of real robot experience. In *Conference on Robot Learning*, pages 2317–2327. PMLR, 2021.

[17] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. 2023.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[19] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8958–8966, 2019.

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

[21] L. A. Thiede and P. P. Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9954–9963, 2019.

[22] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020.

[23] H. Fu, W. Xu, H. Xue, H. Yang, R. Ye, Y. Huang, Z. Xue, Y. Wang, and C. Lu. Rfuniverse: A physics-based action-centric interactive environment for everyday household tasks. *arXiv preprint arXiv:2202.00199*, 2022.

[24] UnityStore. URL https://assetstore.unity.com/packages/tools/physics/cloth-dynamics-194408.

[25] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018. URL https://github.com/Unity-Technologies/ml-agents.

[26] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: clothed 3d humans. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 344–359. Springer, 2020.

[27] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.