AdaFold: Adapting Folding Trajectories of Cloths via Feedback-loop Manipulation

Alberta Longhini[1], Michael C. Welle[1], Zackory Erickson[2], and Danica Kragic[1]

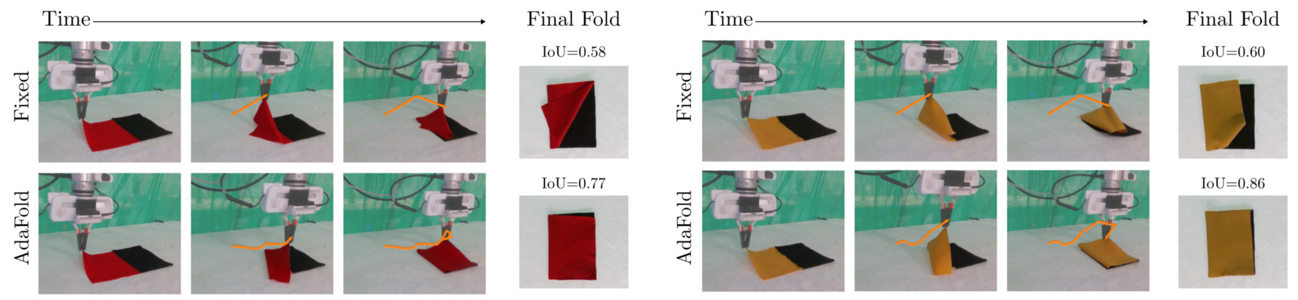

Fig. 1: AdaFold successfully adapts the folding trajectories of the two cloths with different physical properties, achieving a better folding than a predefined triangular trajectory.

***Abstract*— We present AdaFold, a model-based feedback-loop framework for optimizing folding trajectories. AdaFold extracts a particle-based representation of cloth from RGB-D images and feeds back the representation to a model predictive control to re-plan the folding trajectory at every time-step. A key component is the use of semantic descriptors extracted from visual-language models to distinguish between ambiguous point clouds of differently folded cloths. Our experiments demonstrate AdaFold's ability to adapt folding trajectories to cloths with varying physical properties and generalize from simulated training to real-world execution.**

I. Introduction and Related Work

Generalizing robotic manipulation skills requires adapting to object variations such as pose, shape, and physical properties [1]. Feedback-loop manipulation represents a class of methods to adapt to these variations [2]. Yet, within the realm of deformable objects such as cloth, feedback-loop manipulation remains under-explored as its effectiveness is heavily contingent upon the robot's ability to perceive and track the state of the object accurately.

Recent advances in model-based manipulation have shown promise in planning pick-and-place interactions for folding and flattening by learning cloth dynamics [3], [4]. Despite progress, these methods often rely on open-loop planning and predefined manipulation trajectories, limited by the practical challenges of estimating and tracking cloth states during manipulation. A promising approach to overcome these challenges involves integrating semantic knowledge in the state representation of the cloth, extracted from pre-trained visual-language models (VLMs) [5]. However, the effectiveness of VLMs in accurately interpreting a wide range of cloth configurations is yet to be fully explored.

[1]The authors are with the Robotics, Perception and Learning Lab, EECS, at KTH Royal Institute of Technology, Stockholm, Sweden `albertal, mwelle, dani@kth.se`

[2]The authors are with are with Carnegie Mellon University, Pittsburgh, USA `zerickso@andrew.cmu.edu`

This study introduces AdaFold, a model-based framework for feedback-loop manipulation of cloth to optimize folding trajectories. Utilizing a particle-based state representation and a learned cloth model, AdaFold employs model-predictive control to adapt to cloth variations by re-planning folding trajectories after every timestep. We integrate semantic descriptors of the cloth's upper and bottom layers from RGB images using pre-trained VLMs to address the challenges of tracking the cloth state.

AdaFold's effectiveness is demonstrated through extensive evaluation in simulation and real-world environments using a single-arm manipulator for the half-folding task [6]. Our results confirm AdaFold's ability to optimize folding trajectories (Fig.1), successfully accounting for variations in physical properties.

II. AdaFold

The problem we address is feedback-loop manipulation of cloth, focusing on optimizing the manipulation trajectory within a set of pick-and-place positions $x_{\text{pick}}, x_{\text{place}} \in \mathbb{R}^3$. These positions can be provided by a dedicated planner [7], [8], [9]. We consider the half-folding task proposed in [6] as a representative evaluation task, with the goal of folding a rectangular cloth in half. The task is performed with the assumption of quasi-static manipulation, which implies that the forces and torques acting on the cloth are in static equilibrium at each time-step. Under these conditions, we decouple the problem of feedback-loop manipulation into cloth perception and trajectory optimization and propose a framework to tackle it, denoted as AdaFold. An overview of AdaFold can be seen in Fig.2.

Cloth Perception: The state of the cloth at time t is described as a 3D point cloud P_t representing the observable points of the cloth. The subscript t is omitted when the specific time-step of the state is not necessary. To disambiguate different cloth configurations, we introduce semantic information in the point cloud, allowing us to

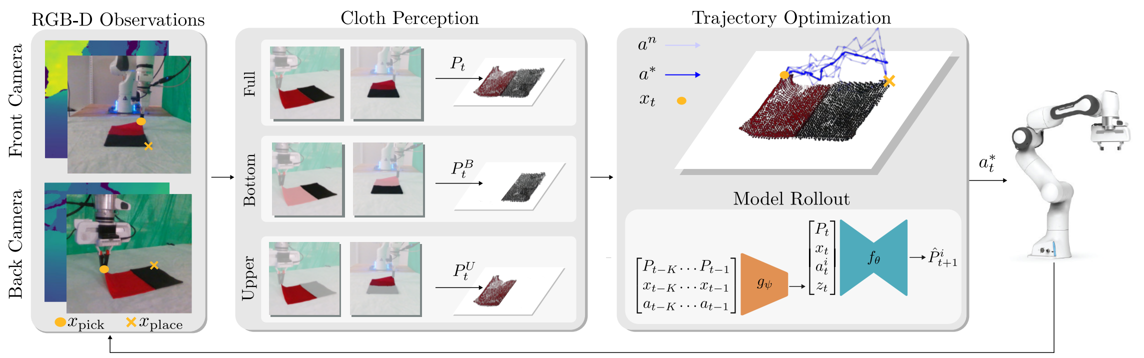

Fig. 2: Overview of AdaFold for feedback-loop manipulation of cloths. Given a set of pick-and-place positions (x_{pick}, x_{place}), AdaFold optimizes the best folding action a_t^* at each time-step t. RGB-D observations from different calibrated cameras are used to extract point cloud representations with semantic descriptors. The semantic descriptors *Upper* and *Bottom* are provided by VLMs. The optimal folding action a_t^* is obtained with MPC, which uses the forward and adaptation modules f_θ and g_ψ to evaluate the candidate trajectories a^n (light blue) and update the optimal control sequence a^* (dark blue).

cluster the points into two sets: P^U and P^B, where P^U corresponds to the **U**pper layer points of the cloth, P^B to the **B**ottom layer, and $P = P^U \cup P^B$. Specifically, the point cloud P is obtained from RGB-D observations using the camera's intrinsic matrix. We then find the segmentation masks *full*, *upper* and *bottom* from the RGB observation using Grounding-DINO [5] and Segment Anything (SAM) [10] and text prompts. To overcome the challenges encountered in robustly segmenting the upper and bottom layers of cloth from one specific prompt, we adopted an ensemble of prompts followed by a heuristic to perform the mask selection. This heuristic aggregates masks generated from diverse prompts through logical operations and color thresholding, acting as a soft voting mechanism.

Trajectory Optimization: We model the discrete-time dynamics of the observable points of a cloth similarly to [11], [12], where they jointly learn from synthetic data: 1) an adaptation module g_ψ to encode a recent history of observations into a latent representation z_t of the physical properties and adapt to different objects, 2) an approximate model f_θ of the cloth dynamics conditioned on the latent representation. Thus, $\hat{P}_{t+1} = f_\theta(P_t, x_t, a_t, z_t)$, where $\hat{P}_t = P_t^U \cup P_t^B$, x_t is the 3D position of the robot end-effector (EE), a_t is the robot action corresponding to a 3-DoF EE displacement. We use PointNet++ [13] as the backbone and a mean squared error loss for training. The folding trajectory from time $t = 0$ to $t = T$ is defined as $\tau_{0:T} = [P_0, P_1, \ldots, P_T; x_0, x_1, \ldots, x_T; a_0, a_1, \ldots, a_T]$ where T is the control horizon. The trajectory is optimized using MPC with reciding horizon H to find a sequence of actions $a_{0:T}^* = \arg\min_{a_{0:T}} \mathcal{J}(\tau_{0:T})$ minimizing a cost function $\mathcal{J}$. The complete optimization process is detailed in Algorithm 1, which outlines how N candidate open-loop control sequences $a_{t:t+H}$ are first sampled from a multivariate Gaussian distribution. The process then employs the MPPI algorithm to refine and update the optimal control input [14]:

$$a_h^* = \frac{1}{\sum_{n=1}^N \exp\left(-\frac{1}{\lambda}\mathcal{J}^n\right)} \sum_{n=1}^N \exp\left(-\frac{1}{\lambda}\mathcal{J}^n\right) a_h^n, \tag{1}$$

Algorithm 1: AdaFold

```
Result: Optimized folding actions $a^*_{0:T}$.
Input: Pick and place positions $\{x_{\text{pick}}, x_{\text{place}}\}$,
       Learned models $f_\theta$ and $g_\psi$, Horizon $H$,
       Number of action candidates $N$, Control
       hyper-parameters $\lambda, w_1, w_2$, Initial control
       sequence $a_{0:H}$, Control variance: $\Sigma$
1  for $t \leftarrow 0$ to $T$ do
2      $P_t = P_t^U \cup P_t^B \leftarrow$ Cloth Perception$(\{I_t^1, I_t^2\})$
3      for $n \leftarrow 1$ to $N$ do
4          $a^n_{t:t+H} \leftarrow \mathcal{N}(a_{t:t+H}, \Sigma)$
5          $z_t \leftarrow g_\psi(\tau_{t-K:t-1})$
6          for $h \leftarrow t$ to $t+H$ do
7              Compute $\tau_{h:h+1}$ unrolling $f_\theta$
8              $\mathcal{J}^n(\tau_{h:h+1}) \leftarrow w_1 c_1 + w_2 c_2$
9          end
10         $\mathcal{J}^n(\tau_{t:t+H}) \leftarrow \sum_{h=t}^{t+H} \mathcal{J}^n(\tau_{h:h+1})$
11     end
12     $a^*_{t:t+H} \leftarrow$ MPPI$(\{\mathcal{J}^n(\tau_{t:t+H})\}_{n=1}^N, \lambda)$  ▷ Eq. (1)
13     Execute $a_t^*$
14     Warm-start control sequence $a$ with $a^*$
15 end
```

for $h = t, .., t+H$ where $\mathcal{J}^n$ is the cost of the n-th trajectory, λ is a temperature parameter that controls exploration, and a_t^n is the control input of the n-th trajectory at time step t.

Cost function: The cost function is designed to maximize the alignment of the cloth's halves, using Intersection over Union (IoU) to assess the alignment. We further include a term penalizing large cloth displacements. Specifically, the cost is defined as a weighted sum of two terms:

$$\mathcal{J}(\tau_{t:t+H}) = w_1 c_1(\tau_{t:t+H}) + w_2 c_2(\tau_{t:t+H}), \tag{2}$$

with weight w_1, w_2. The term c_1 evaluates the progress towards the alignment of the two halves as:

$$c_1(\tau_{t:t+H}) = \sum_{j=1}^{H} \beta^{H-j} \text{IoU}(\hat{P}^U_{t+j}, P^B_{t+j}), \tag{3}$$

TABLE I: Final IoU obtained folding the cloths with different methods. The folding is executed 20 times for each cloth and method.

Method ↑	10 Cloths ↑
Random	0.40 ± 0.17
Triangular	0.41 ± 0.02
DDPG-Critic	0.48 ± 0.09
Adafold-OL	0.48 ± 0.16
AdaFold	$\mathbf{0.78 \pm 0.11}$

where β^{H-j} progressively increases the importance of future cloth alignments, preventing the robot from greedily selecting actions that lead to aligning the halves as fast as possible. The second term c_2, instead, acts as a binary flag that penalizes actions moving the gripper outside the cloth's initial convex hull, thus leading to a larger movement of the entire cloth.

III. Experimental Results

Our experiments evaluate AdaFold's ability to optimize folding trajectories through feedback-loop manipulation, specifically assessing its improvement in folding outcomes and adaptability to varying cloth physical properties.

Half Folding Task: We consider the half-folding task proposed in [6], where the objective is to fold the cloth in half by aligning the corners. We evaluate the success of the folding execution by computing the 2D intersection over union (IoU) between the two halves of the cloth. We implemented the task in PyBullet [15] using a square cloth with elastic and stiffness parameters of 40 and 60. We replicated the task in the real world using two Realsense D435 cameras capturing different views of the scene and a Franka-Emika-Panda robot. The real-world cloth dataset consists of six rectangular cloths, each varying in physical properties and dimensions. The cloths 1 through 5 are arranged to reflect an increasing stiffness, measured accordingly to [16]. The sixth cloth is distinguished by its larger size.

Baselines: We compared AdaFold against four baselines: a fixed triangular trajectory (Triangular), a random action selection (Random), an open-loop AdaFold (AdaFold-OL), and a model-free learning method DDPG, trained offline on the same dataset as AdaFold. DDPG utilized its critic to select from the action candidates (DDPG-Critic), with the architecture and hyperparameters aligned with AdaFold.

Simulation Results: To assess the relevance of optimizing the folding trajectory and showcasing the benefits of feedback-loop optimization, we compared our method in simulation against the four baselines. We randomly selected 10 elastic and stiffness parameters to introduce variations not observed during training. Table I shows the folding results. It can be observed that methods that do not optimize the folding trajectory (Random and Fixed) yielded the lowest performance, thereby underscoring the advantage of trajectory optimization. While the model-free baseline (DDPG-Critic) and the open-loop approach (AdaFold-OL) improved the folding outcome, AdaFold outperformed all the baselines and achieved the best folds, showcasing the benefits of feedback-loop optimization.

TABLE II: Final IoU evaluated on real-world cloths $1-5$. The folding is repeated 5 times for each combination of cloth and method.

Cloth	Triangular ↑	AdaFold ↑
1	0.57 ± 0.03	0.81 ± 0.06
2	0.55 ± 0.03	0.81 ± 0.06
3	0.61 ± 0.01	0.79 ± 0.04
4	0.60 ± 0.03	0.82 ± 0.05
5	0.75 ± 0.02	0.83 ± 0.04
All	0.62 ± 0.08	$\mathbf{0.81 \pm 0.05}$

TABLE III: Evaluation of AdaFold's generalization to variations of cloth initial positions and size. The evaluation metric is the IoU. The folding is repeated 1 time for each random position and 5 times for cloth 6 for both evaluated methods.

Method	10 Random Poses ↑	Cloth 6 ↑
Triangular	0.57 ± 0.03	0.55 ± 0.01
AdaFold	$\mathbf{0.76 \pm 0.09}$	$\mathbf{0.71 \pm 0.05}$

Real-World Results: We further assessed whether AdaFold can improve the fold quality compared to the Triangular trajectory for real-world cloths. We present the results in Table II. The variance that the Triangular trajectory exhibits highlights that identical folding trajectories can yield different outcomes based on the cloth's properties. Stiffer cloths (e.g. $4-5$) achieved on average, a better fold with respect to less stiff cloths. Conversely, AdaFold consistently produced satisfactory folds across different samples, improving the fold for samples where the fixed trajectory fell short. Figure 1 provides a qualitative comparison of the fixed and optimized trajectories for samples 1 and 4.

We extended this evaluation by testing AdaFold on variations in the initial positions and size of the cloth. We selected cloth 2 for the variation in position and randomly rotated its starting position on the table 10 times, with rotations ranging between $\pm 45^\circ$ degrees. For the variation in size, we selected cloth 6, which is larger than cloths $1-5$. We compared AdaFold against the Triangular baseline and showed the results in Table III. While AdaFold outperformed the Triangular trajectory in both scenarios, the improvement was not as high as in the previous evaluation. We attribute this to the higher misalignment between training and test conditions.

The overall outcomes confirm the adaptability of our method, underscoring the benefit of using particle-based representations along feedback-loop manipulation to adapt to different object variations.

IV. Conclusion and Future Work

We introduced AdaFold, a novel framework that leverages model-based feedback-loop manipulation to optimize cloth folding trajectories. Our evaluations on the half-folding task showcased the potential of coupling a robust perception module with data-driven optimization strategies to perform feedback-loop manipulation. We plan to extend these results to different goal and reward configurations, a broader spectrum of clothing items and tasks beyond folding.

References

[1] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *JMLR*, vol. 22, no. 1, pp. 1395–1476, 2021.

[2] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, “Scalable deep reinforcement learning for vision-based robotic manipulation,” in *CoRL*. PMLR, 2018, pp. 651–673.

[3] X. Lin, Y. Wang, Z. Huang, and D. Held, “Learning visible connectivity dynamics for cloth smoothing,” in *CoRL*. PMLR, 2022, pp. 256–266.

[4] X. Ma, D. Hsu, and W. S. Lee, “Learning latent graph dynamics for deformable object manipulation,” *arXiv preprint arXiv:2104.12149*, vol. 2, 2021.

[5] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” *arXiv preprint arXiv:2303.05499*, 2023.

[6] 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,” *RA-L*, vol. 5, no. 2, pp. 1111–1118, 2020.

[7] 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 *CoRL*. PMLR, 2021, pp. 2317–2327.

[8] K. Mo, C. Xia, X. Wang, Y. Deng, X. Gao, and B. Liang, “Foldsformer: Learning sequential multi-step cloth manipulation with space-time attention,” *RA-L*, vol. 8, no. 2, pp. 760–767, 2022.

[9] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, “Visuospatial foresight for physical sequential fabric manipulation,” *Autonomous Robots*, pp. 1–25, 2022.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, “Segment anything,” *arXiv preprint arXiv:2304.02643*, 2023.

[11] A. Longhini, M. Moletta, A. Reichlin, M. C. Welle, D. Held, Z. Erickson, and D. Kragic, “Edo-net: Learning elastic properties of deformable objects from graph dynamics,” *arXiv preprint arXiv:2209.08996*, 2022.

[12] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “Rma: Rapid motor adaptation for legged robots,” *arXiv preprint arXiv:2107.04034*, 2021.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.

[14] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, “Information theoretic mpc for model-based reinforcement learning,” in *ICRA*. IEEE, 2017, pp. 1714–1721.

[15] B. Ellenberger, “Pybullet gymperium,” https://github.com/benelot/pybullet-gym, 2018–2019.

[16] I. Garcia-Camacho, A. Longhini, M. Welle, G. Alenya, D. Kragic, and J. Borras, “Standardization of cloth objects and its relevance in robotic manipulation,” in *ICRA*. IEEE, 2024.