PlaNet-ClothPick: Effective Fabric Flattening Based on Latent Dynamic Planning

Halid Abdulrahim Kadi[1,*] and Kasim Terzić[1]

Abstract— **Why do Recurrent State Space Models such as PlaNet fail at cloth manipulation tasks? Recent work has attributed this to the blurry prediction of the observation, which makes it difficult to plan directly in the latent space. This paper explores the reasons behind this by applying PlaNet in the pick-and-place fabric-flattening domain. We find that the sharp discontinuity of the transition function on the contour of the fabric makes it difficult to learn an accurate latent dynamic model, causing the MPC planner to produce pick actions slightly outside of the article. By limiting picking space on the cloth mask and training on specially engineered trajectories, our mesh-free PlaNet-ClothPick surpasses visual planning and policy learning methods on principal metrics in simulation, achieving similar performance as state-of-the-art mesh-based planning approaches. Notably, our model exhibits a faster action inference and requires fewer transitional model parameters than the state-of-the-art robotic systems in this domain. Other supplementary materials are available at: https://sites.google.com/view/planet-clothpick.**

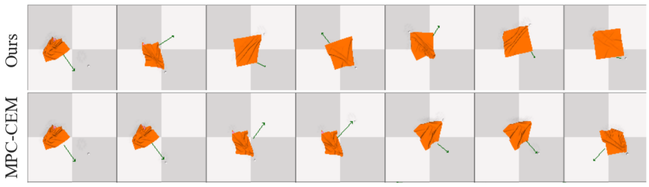


Fig. 1: Flattening trajectories of ClothMaskPick-MPC (ours) and MPC with cross entropy method (MPC-CEM). The head and end of the green arrows represent each step's pick and place positions. The fundamental reason PlaNet fails in cloth flattening is that the latent dynamic model cannot accurately model the transition function's sharp discontinuity on the cloth's contour. By limiting the first picking sampled actions to fall inside the cloth mask, PlaNet-ClothPick achieves SoTA performance in this domain.

I. Introduction

Deep reinforcement learning methods based on the Recurrent State Space Model (RSSM), such as PlaNet [1] and Dreamer [2], [3], [4] have achieved state-of-the-art (SoTA) asymptotic performance and data efficiency in both continuous control `dm_control` [5] and discrete-action Atari 2600 [6] benchmark environments. However, many authors have noted that RSSM-based models struggle with a canonical task in cloth-shaping: fabric flattening, where one or more end-effectors operate on a piece of square fabric to unfold it on a surface [7], [8], [9], [10].

Most successful data-driven methods, such as imitation learning [8], [11] and reinforcement learning [12], [9], [13], [14], [15] for fabric-flattening focus on quasi-static pick-and-place (P&P) manipulation. Despite pick-and-fling and pick-and-blow primitives being operationally more effective than quasi-static P&P primitives [16], [17], P&P is cost-effective as it only requires one robot arm and a camera.

Deep Planning Network (PlaNet) [1] is a model-based reinforcement learning algorithm that uses model-predictive control (MPC) to plan on a latent dynamic model (LDM) trained based on the RSSM. However, PlaNet keeps failing on fabric flattening [13], [18], [19]; it has been argued that this may be due to the blurry observation reconstruction of the LDM [9], [14]. In this paper, we investigate PlaNet's performance on the domain in simulation benchmark SoftGym [19] to understand the causes of poor performance on this task. We note four contributions of this paper:

(1) We propose a new reward function for fabric flattening, which leads to better performance than the normalised coverage and the reward adopted by Hoque et al. (2022) [9].

(2) Inspired by Lin et al. (2022) [14] and Hoque et al. (2022) [9], we suggest a domain-specific planning method ClothMaskPick-MPC that samples the first pick action on the cloth-mask to improve planning accuracy and efficiency (Figure 1); this helps to overcome the difficulty in accurately modelling the sharp discontinuity on the article's contour.

(3) LDMs need a large amount of data to overcome the enormity and complexity of the cloth's dynamic. Apart from the expert, random, corner-biased trajectories [9], we collect trajectories where the policy samples small dragging actions on an almost flattened fabric. This helps to boost the robustness and performance of the ClothMaskPick-MPC on the LDM. Besides, we employ data augmentation techniques, such as rotation [20] and flipping, to upsample the training trajectories to improve the robustness of the model.

(4) We also observe that PlaNet's LDM cannot learn a good latent prior distribution due to the complex non-linear behaviour of cloth-like deformable objects (CDOs), so we adopt *KL balancing* [3] to improve both posterior and prior learning quality. We further improve the planning performance by incorporating prior reward learning.

[1] School of Computer Science, University of St Andrews, Jack Cole Building, North Haugh, St Andrews, KY16 9SX United Kingdom

[*] Correspondence author, E-mail: `ah390@st-andrews.ac.uk`; Tel.: `+44 1334 46 1630`

We demonstrate that these four improvements lead to PlaNet-ClothPick achieving SoTA performance compared to mesh-based planning methods regarding primary metrics – normalised coverage (NC) and normalised improvement (NI) against action steps, and it outperforms visual planning and policy learning methods (Figure 4). It also showcases a one-order-of-magnitude advantage regarding the action inference time and transitional model parameters compared to the previous SoTA robotic systems in this domain (Figure 5). The strong inductive biases of our method are introduced in ClothMaskPick-MPC and the specially engineered offline dataset, hence the name PlaNet-ClothPick. This paper shows that RSSM-based algorithms can play an important role in a wider range of application domains.

II. Related Work

Model-based reinforcement learning (MBRL) applications in P&P cloth-flattening are mainly Type I [21]: planning or trajectory optimisation algorithms, where the agent needs access to the dynamic model of the environment for generating imaginary rollouts. The dynamic model can be either a known dynamic or a learned dynamic. Planning algorithms used in fabric-flattening are mainly based on model-predictive control (MPC), which can be further categorised into goal-conditioned MPC [9], [18] and reward-based MPC systems [13], [14], [15].

Visual Foresight (VSF) [9] by Hoque et al. (2022) and the Contrastive Forward Model (CFM) [18] by Yan et al. (2020) are the two example applications of goal-conditioned MPC to fabric flattening, where the cost function is calculated based on the difference between the current and goal states. Note that VSF's cost function is calculated according to Visual MPC [22] framework, while the one of CFM is the distance of the two states at the latent space.

reward-based MPC, on the other hand, selects top trajectories based on reward prediction from the prior rollout trajectories [1], [13], [14]. In contrast to goal-conditioned MPC, the application domain of reward-based MPC is limited by the reward prediction function given to the algorithm. DefOrmable Object Manipulation (G-DOOM) is a latent reward-based MPC method that generates the prior rollout trajectories in the latent space trained with unsupervised-keypoint graph dynamics [13]. In contrast, Visible Connectivity Dynamics (VCD) by Lin et al. (2022) [14] is a mesh-based reward-based MPC method that applies rollout on the reconstructed mesh representation using a learned mesh dynamic [23]; they also proposed VCD Graph Imitation (VCD-GI), where a teacher dynamic model learns with the complete information of the cloth and distil the knowledge to the vision-based student.

Mesh-based reward-based MPC methods, such as VCD, VCD-GI [14], and MEDOR [15], outperform goal-conditioned MPC and latent reward-based MPC methods in cloth-flattening. Although they are invariant to the cloth shape, colour and camera pose, these methods cannot be easily applied to manipulating other kinds of objects, since the dynamic model is specially trained for CDOs. Most recent robotic systems focus on closing the simulation-to-reality gap of the mesh-based planning methods on garment-flattening tasks [24], [25], [26] by improving the mesh tracking accuracy in real-world trials.

Deep Planning Network (PlaNet) [1] is a latent reward-based MPC method that employs a learned latent dynamic based on a Recurrent State Space Model (RSSM). While it performs well on continuous control benchmark environments like the `dm_control` suite [5], numerous experiments have found it unsuitable for fabric flattening [18], [13], [19]. A possible reason is that the reconstructed observation from the visual model is fuzzy, which makes planning based on reconstructed vision hard due to the lack of precision around the edges and corners of the article [14].

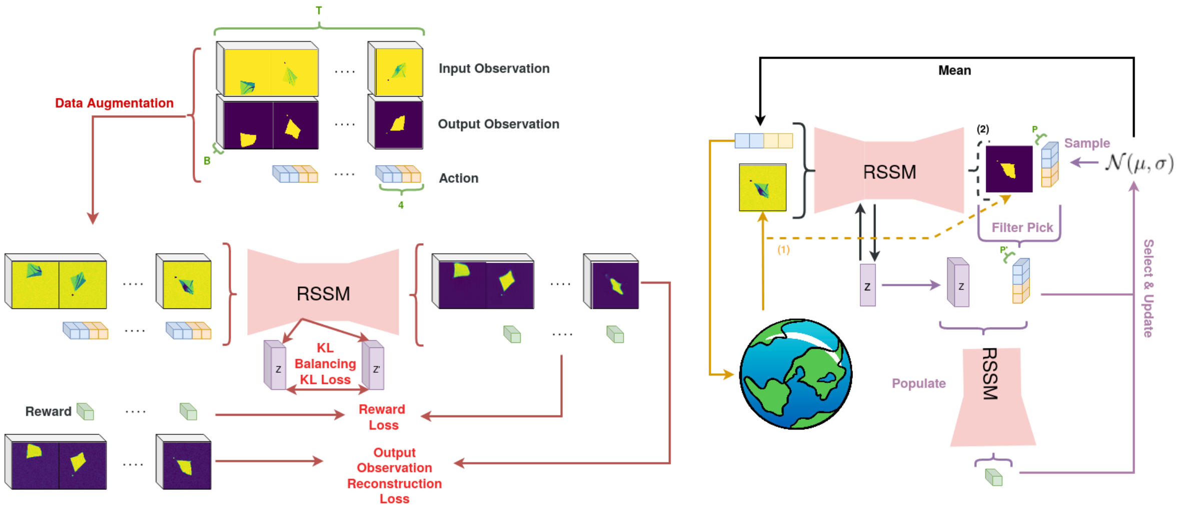
Dreamer [2], [3], [4] is a model-based actor-critic (AC) reinforcement learning algorithm that uses RSSM [1] for representation learning. It shows significant data efficiency and performance improvement compared to the PlaNet and AC baselines, and Dreamer V2 [3] is the first MBRL algorithm to achieve super-human performance with a single GPU in discrete-action Atari benchmarks. The architecture and objective of the LDM of Dreamer are directly inherited from PlaNet's [1]. Dreamer V2 [3] leverages categorical latent state space representation and *KL balancing* to learn the LDM.

Stochastic Latent Actor-Critic (SLAC) [27] combines the Soft Actor-Critic's (SAC) [28] maximum-entropy RL objective [29], [30] with latent dynamic representation learning to solve *Partially Observable Markov Decision Processes* (POMDP) [31]. In contrast to RSSM-based algorithms such as PlaNet and Dreamer, it only learns stochastic latent representation. It exhibits stability, data efficiency and improved performance compared to SAC and PlaNet in several benchmarks [32] but has never been tested on CDO manipulation tasks. We evaluate the performance of SLAC's policy learning and planning on the LDM in this paper.

Note that we did not examine imitation learning approaches because we mainly focus on reinforcement learning methods in this paper. Nevertheless, Hoque et al. (2022) [9] have shown that VSF outperforms the behaviour-cloning approach DAgger [8] regarding primary metrics.

III. Method

We aim to investigate the failure of PlaNet in fabric flattening to develop a model capable of handling this domain. A latent dynamic model (LDM) for pick-and-place (P&P) fabric-flattening must accurately predict future states based on a sequence of future action trajectories. This allows a planning algorithm to generate a trajectory of candidate actions that minimises a cost function. We can formulate the model learning problem as a partially observable Markov decision process (POMDP). Our environment is built on SoftGym's [19] cloth-flattening task with P&P action extension, which comprises 4 parameters $(x_{pick}, y_{pick}, x_{place}, y_{place})$ defined on continuous pixel space [-1, 1] for a single-picker operation [12], [13]. We did not use pick-and-drag action primitive [9],



(a) Training of RSSM

(b) Action Inference of ClothMaskPick-MPC

Fig. 2: PlaNet-ClothPick. We use I2O to denote the different variants; for example, D2Mask represents the model's input as a depth image and the output as a cloth-mask image. The red line depicts training data flow, the purple line represents action optimization in planning, the black line signifies internal state updates, and the yellow line illustrates environment-agent input/output. In training, PlaNet-ClothPick applies batch-wise rotation and vertical flipping on the input/output observations and actions before feeding them into the RSSM model; observation noise is only applied if the observations are RGB and/or depth images. During the inference time, ClothMaskPick-MPC samples pick-and-place actions from a normal distribution (initialised with mean as 0 and standard deviation as 1), then it filters them through an estimated cloth mask, which can be obtained in two different ways: (1) thresholding from the depth image of the environment or (2) predicting from the RSSM model if the decoding includes mask prediction. Then, it selects the top 10% candidates based on the reward the RSSM predicted from the last-step posterior latent state and the sampled actions to update the normal distribution for the next optimisation iteration. After the planning, the method uses the mean of the distribution as an action to execute.

[18], [14] for bounding the place position of the fabric on the observation space.

A. Deep Planning Network (PlaNet)

Recurrent State Space Model (RSSM) [1] is defined under the POMDP setting with the following latent state dynamic: (1) recurrent dynamic model $\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{z}_{t-1}, \boldsymbol{a}_{t-1})$, (2) representation model $\hat{z}_t \sim q(\hat{z} \mid \boldsymbol{h}_t, \boldsymbol{x}_t)$, and (3) transition predictor $\tilde{z}_t \sim p(\tilde{z} \mid \boldsymbol{h}_t)$, where $\boldsymbol{x}$ represents observation, $\boldsymbol{a}$ represents action, $\boldsymbol{h}$ represents the deterministic latent representation, and $\hat{z}$ and $\tilde{z}$ represent the prior and posterior stochastic latent states.

PlaNet learns the RSSM to generate accurate observations and rewards from a prior latent distribution for MPC planning. The dynamic model is trained by minimising the KL-divergence between prior and posterior latent states as well as maximising the maximum likelihood of reconstruction of the observation and reward, where it includes an observation predictor $\hat{\boldsymbol{x}}_t \sim p(\boldsymbol{x} \mid \boldsymbol{h}_t, z_t)$ and a reward predictor $\hat{r}_t \sim p(r \mid \boldsymbol{h}_t, z_t)$:

$$\mathscr{L}_{PlaNet} = \sum_{t=1}^{T} \Bigg(- \mathop{\mathbb{E}}_{q(z_t|\boldsymbol{x}_{1:t},\boldsymbol{a}_{1:t})} \Big[\log p(\boldsymbol{x}_t|z_t) + \log p(r_t|z_t) \Big] + \mathop{\mathbb{E}}_{q(z_{t-1}|\boldsymbol{x}_{1:t-1},\boldsymbol{a}_{1:t-1})} \Big[KL\big[q(z_t|\boldsymbol{x}_{1:t},\boldsymbol{a}_{1:t-1}) || p(z_t|z_{t-1},\boldsymbol{a}_{t-1}) \big] \Big] \Bigg) \tag{1}$$

PlaNet adopts mean-square-error (MSE) to learn observation reconstruction and reward prediction from the Gaussian posterior latent space and Kullback–Leibler (KL) divergence for prior learning.

Model predictive control (MPC) is a set of advanced control methods that usually require a learned/known dynamic model to predict the future behaviour of the controlled system and a cost function to optimise the sampled trajectories. MPC with Cross-Entropy Method (MPC-CEM) is a common variation that samples actions from a multivariate Gaussian distribution and iteratively optimises the distribution's mean and variance from the top trajectories determined by the cost function. PlaNet employs MPC-CEM to produce the policy at run time by unrolling and maximising the accumulative future rewards from the latent prior distributions. It iteratively refines its LDM by exploring the environment and collecting new trajectories generated by the planner.

B. *PlaNet-ClothPick*

Our PlaNet-ClothPick method is built upon the original PlaNet and trained with our domain-specific reward function (Section III-B.1). We train the LDM for cloth-flattening offline using a special data collection script (III-B.4) so that we bypass the exploration of reinforcement learning, which is a hard problem to address for cloth-like deformable objects [12], [18]. We also adopt *KL balancing* [3] (Section III-B.2) to enhance latent prior and posterior learning quality. In addition, we apply data augmentation — observation noise, rotation [20], and vertical flipping — to improve the learning efficiency and robustness of the method. Finally, we adopt the prior reward learning and the domain-specific planning method ClothMaskPick-MPC (Section III-B.3) to further improve manipulation performance. Figure 2 illustrates the further details of the method with its different input/output (I/O) variants.

1) Reward Function: We extend the reward function presented by Hoque et al. (2022) [9], which is based on the relative coverage improvement between two consecutive states. We impose penalties as -0.5 for mispicking, large absolute action values (when any equal to or greater than 0.7), and steps that lead to unflattening. Conversely, we assign bonuses as 0.5 to steps that lead to states with high coverage.

2) KL balancing: In Equation 1, the KL-divergence term aims to learn the prior from the posterior representation and regularises the posterior representation with the prior. To avoid regularising the posterior representation towards poor priors, Dreamer V2 [3] proposes *KL balancing* that prioritises learning of the prior over regularising the posterior. Combining the two components with an interpolation factor $\alpha = 0.8$, *KL balancing* achieves the former by stopping the gradient on the posterior representation and the latter by stopping the gradient on the prior representation. *KL balancing* is a significant factor for improving the asymptotic performance and learning efficiency of Dreamer [3], [4].

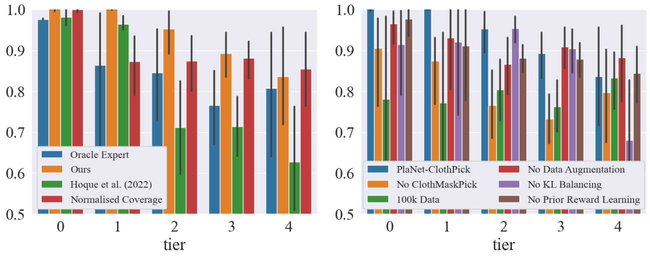
3) ClothMaskPick-MPC: Building upon original MPC-CEM planning, we restrict the sampling of the first picking action to the cloth mask, which can be extracted from the depth observation of the environment or estimated from the RSSM model if the decoding includes mask prediction. We set the planning horizon to 1, the population of the samples to 5000, and optimisation iterations to 100.

4) Data Collection: We generate 1,000 random fabric instances in the SoftGym [19] environment to cover a wide range of shapes and positions. We reserve 100 episodes for assessing the manipulation system; the rest are for developing the method. We also produce 56,100 episodes of 20-step trajectory data from the developing settings for training the LDMs. We delegate 100 episodes for testing the LDM and 56,000 episodes (1.12 million transitional steps) for training the models.

To cover a wide range of scenarios of P&P actions on fabrics, we heuristically generate 10% purely random policy, 10% corner-biased random policy [9], [33], 40% Oracle expert flattening and various folding policies, 30% noisy expert policies and 10% mix policy for the first 50,000 trajectories. The remaining 6000 trajectories are generated from a highly flattened initial state (above 85% coverage), where 20% of the data are produced from expert flattening policy, 20% from noisy expert flattening policy and 60% from cloth-mask small-random-dragging policy.

TABLE I: Difficulty tiers regarding normalised coverage of initial states. We allocate 57 of the instances to the corresponding tiers regarding the generated distribution.

Tier	NC mean ± std	NC (min, max)	No. Eps
0	97.64 ± 0 %	(97.49%, 97.93%)	5
1	87.78 ± 9.78%	(73.12%, 93.50%)	4
2	56.47 ± 3.97%	(51.82%, 62.75%)	15
3	40.88 ± 1.58%	(38.28%, 43.50%)	25
4	28.39 ± 0.92%	(27.12%, 29.53%)	8



(a) Reward Study

(b) Ablation Study

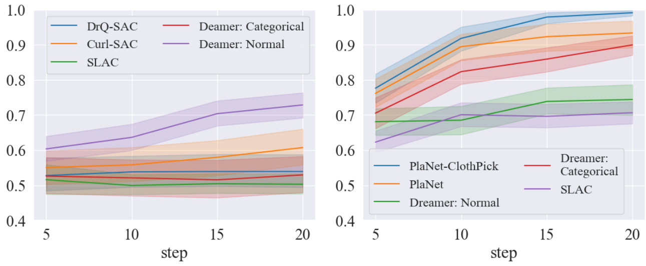
Fig. 3: Normalised coverage of PlaNet-ClothPick at step 10 among different tiers. Each constituent element of PlaNet-ClothPick is essential for avoiding unflattening high-coverage articles and reaching higher final coverage. ClothMaskPick-MPC and the specially engineered large dataset are critical for achieving effective flattening in general. Our method even beats the Oracle expert policy used to generate the dataset.

The manipulation outcome is extremely sensitive to the pick signal relative to the fabric. Oracle expert flattening, expert folding policies and the corner-biased policy are introduced to guide the pick action operating on the corner of the fabric. Noisy expert policies are designed to account for situations where the picking occurs slightly inside or outside the fabric's corners – within 5% of errors. The purely random policy addresses picking actions outside of the fabric, while the cloth-mask small-dragging policy is specifically designed for picking on the fabric surface. While most of these policies accommodate a wide range of fabric dragging scenarios, the cloth-mask small-dragging policy is particularly crucial for emphasising small-dragging actions.

The condition of the fabric itself is another crucial factor affecting the operation's outcome. We employ expert folding, noisy expert folding, random folding, and mix policies to include scenarios where the fabric becomes crumpled from a flattened state. The mixed policy is also introduced for diversifying the action types in a single trajectory.

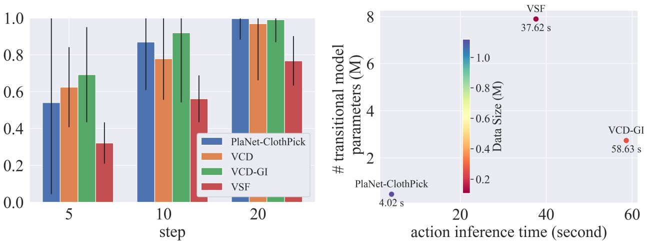
TABLE II: Numerical principal metrics of PlaNet-ClothPick's input/output variants across different tiers; we also include the performance of Oracle expert and heuristic method *Wrinkle* [7], [8] at the bottom of the table; and the best performance among different steps for all tiers is highlighted with bold text and an asterisk. By default, all these variants use cloth masks fetched from the environment, and we use `fm` to denote the ones estimated from the model. The depth-to-depth (D2D) variant performs worst, while other variants perform similarly. Fetching the accurate estimation of the cloth mask for ClothMaskPick-MPC is important for selecting the most effective picking position on the fabric.

		Normalised Coverage ↑						Normalised Improvement ↑					
input/output	steps	tiers: 0	1	2	3	4	all	0	1	2	3	4	all
RGB2RGB	5	1.0 ± 0.01	0.93 ± 0.1	0.77 ± 0.16	0.76 ± 0.14	0.63 ± 0.12	0.78 ± 0.16	1.1 ± 0.36	-0.12 ± 1.47	0.48 ± 0.34	0.59 ± 0.25	0.49 ± 0.17	**0.54 ± 0.5***
	10	1.0 ± 0.01	1.0 ± 0.01	0.95 ± 0.11	0.89 ± 0.14	0.84 ± 0.18	0.92 ± 0.14	1.1 ± 0.36	1.01 ± 0.03	0.89 ± 0.26	0.82 ± 0.24	0.77 ± 0.26	**0.87 ± 0.26***
	20	1.0 ± 0.01	1.0 ± 0.01	1.0 ± 0.01	0.99 ± 0.02	0.96 ± 0.06	0.99 ± 0.03	1.1 ± 0.36	1.01 ± 0.03	1.0 ± 0.02	0.99 ± 0.04	0.95 ± 0.09	1.0 ± 0.11
D2D	5	0.99 ± 0.01	0.95 ± 0.07	0.69 ± 0.15	0.69 ± 0.15	0.65 ± 0.16	0.73 ± 0.17	0.48 ± 0.5	0.26 ± 0.77	0.3 ± 0.34	0.47 ± 0.25	0.52 ± 0.22	0.42 ± 0.35
	10	0.98 ± 0.02	0.94 ± 0.07	0.84 ± 0.15	0.8 ± 0.15	0.78 ± 0.14	0.83 ± 0.15	0.23 ± 0.73	0.15 ± 0.87	0.62 ± 0.33	0.67 ± 0.25	0.7 ± 0.2	0.59 ± 0.41
	20	0.99 ± 0.02	0.96 ± 0.09	0.93 ± 0.12	0.92 ± 0.12	0.86 ± 0.15	0.92 ± 0.12	0.35 ± 0.78	0.51 ± 1.05	0.84 ± 0.25	0.87 ± 0.19	0.8 ± 0.22	0.78 ± 0.41
RGBD2RGBD	5	0.98 ± 0.04	0.84 ± 0.21	0.75 ± 0.18	0.74 ± 0.19	0.68 ± 0.08	0.76 ± 0.18	0.37 ± 1.58	-0.02 ± 0.53	0.42 ± 0.4	0.57 ± 0.31	0.56 ± 0.11	0.47 ± 0.55
	10	1.0 ± 0.01	0.98 ± 0.02	0.87 ± 0.19	0.92 ± 0.14	0.87 ± 0.15	0.91 ± 0.15	1.01 ± 0.52	0.74 ± 0.3	0.68 ± 0.47	0.86 ± 0.24	0.82 ± 0.21	0.81 ± 0.35
	20	1.0 ± 0.01	0.99 ± 0.01	0.99 ± 0.03	0.99 ± 0.04	0.98 ± 0.04	0.99 ± 0.03	1.01 ± 0.52	0.87 ± 0.18	0.99 ± 0.06	0.99 ± 0.07	0.98 ± 0.05	0.98 ± 0.16
RGB2Mask	5	0.99 ± 0.01	0.95 ± 0.03	0.78 ± 0.19	0.76 ± 0.17	0.68 ± 0.19	**0.79 ± 0.18***	0.46 ± 0.47	0.32 ± 0.53	0.51 ± 0.41	0.6 ± 0.29	0.56 ± 0.26	0.54 ± 0.35*
	10	0.99 ± 0.01	0.96 ± 0.06	0.94 ± 0.1	0.92 ± 0.11	0.93 ± 0.08	**0.94 ± 0.1***	0.39 ± 0.62	0.45 ± 0.92	0.88 ± 0.22	0.87 ± 0.19	0.91 ± 0.11	0.81 ± 0.36
	20	1.0 ± 0.01	0.98 ± 0.04	1.0 ± 0.04	1.0 ± 0.02	0.99 ± 0.03	**1.0 ± 0.03***	1.04 ± 0.37	0.66 ± 0.57	1.0 ± 0.09	1.0 ± 0.04	0.98 ± 0.04	0.97 ± 0.19
RGB2Mask-fm	5	0.99 ± 0.02	0.94 ± 0.07	0.8 ± 0.18	0.7 ± 0.16	0.67 ± 0.1	0.77 ± 0.17	0.34 ± 0.84	0.15 ± 1.07	0.55 ± 0.41	0.5 ± 0.27	0.53 ± 0.14	0.48 ± 0.45
	10	1.0 ± 0.01	0.96 ± 0.04	0.94 ± 0.1	0.88 ± 0.16	0.94 ± 0.1	0.92 ± 0.13	0.9 ± 0.42	0.68 ± 0.32	0.85 ± 0.23	0.8 ± 0.27	0.92 ± 0.13	0.83 ± 0.26
	20	1.0 ± 0.01	0.98 ± 0.03	0.98 ± 0.06	0.98 ± 0.05	1.0 ± 0.02	0.98 ± 0.05	0.9 ± 0.42	0.77 ± 0.33	0.95 ± 0.13	0.96 ± 0.09	1.0 ± 0.03	0.94 ± 0.17
D2Mask	5	0.8 ± 0.27	0.91 ± 0.09	0.76 ± 0.2	0.72 ± 0.18	0.69 ± 0.25	0.75 ± 0.2	-7.12 ± 11.33	-0.29 ± 1.17	0.44 ± 0.49	0.52 ± 0.31	0.57 ± 0.35	-0.22 ± 3.75
	10	0.89 ± 0.25	0.88 ± 0.25	0.83 ± 0.18	0.83 ± 0.21	0.85 ± 0.2	0.84 ± 0.2	-3.37 ± 9.89	-0.53 ± 3.1	0.61 ± 0.43	0.71 ± 0.35	0.8 ± 0.29	0.25 ± 3.0
	20	1.0 ± 0.01	0.99 ± 0.04	1.0 ± 0.02	0.98 ± 0.08	1.01 ± 0.01	0.99 ± 0.06	0.99 ± 0.62	0.8 ± 0.44	1.0 ± 0.05	0.96 ± 0.14	1.01 ± 0.01	0.97 ± 0.22
D2Mask-fm	5	0.87 ± 0.15	0.94 ± 0.1	0.65 ± 0.17	0.63 ± 0.18	0.69 ± 0.15	0.69 ± 0.19	-5.03 ± 7.15	0.13 ± 1.57	0.18 ± 0.42	0.37 ± 0.31	0.57 ± 0.22	-0.14 ± 2.5
	10	0.74 ± 0.26	0.97 ± 0.08	0.84 ± 0.14	0.81 ± 0.18	0.81 ± 0.18	0.82 ± 0.17	-10.5 ± 10.75	0.46 ± 1.13	0.63 ± 0.35	0.68 ± 0.3	0.74 ± 0.25	-0.32 ± 4.31
	20	0.97 ± 0.06	1.01 ± 0.01	0.94 ± 0.12	0.9 ± 0.16	0.92 ± 0.11	0.93 ± 0.13	-0.25 ± 3.07	1.08 ± 0.07	0.85 ± 0.28	0.83 ± 0.28	0.89 ± 0.15	0.77 ± 0.91
D2RGB	5	0.96 ± 0.06	0.95 ± 0.06	0.77 ± 0.19	0.69 ± 0.17	0.59 ± 0.16	0.74 ± 0.19	-0.55 ± 2.75	0.27 ± 0.83	0.47 ± 0.43	0.47 ± 0.28	0.43 ± 0.23	0.36 ± 0.87
	10	0.99 ± 0.03	1.0 ± 0.01	0.91 ± 0.14	0.89 ± 0.14	0.83 ± 0.12	0.9 ± 0.13	0.73 ± 1.13	0.94 ± 0.08	0.79 ± 0.32	0.82 ± 0.24	0.76 ± 0.17	0.8 ± 0.38
	20	1.0 ± 0.01	1.0 ± 0.01	1.01 ± 0.01	0.98 ± 0.08	1.0 ± 0.01	**1.0 ± 0.05***	1.2 ± 0.45	0.95 ± 0.08	1.01 ± 0.02	0.97 ± 0.13	1.01 ± 0.02	**1.01 ± 0.16***
Oracle Expert	5	0.98 ± 0.0	0.82 ± 0.23	0.73 ± 0.25	0.62 ± 0.2	0.52 ± 0.14	0.68 ± 0.23	-0.03 ± 0.18	-1.3 ± 3.12	0.4 ± 0.55	0.35 ± 0.35	0.34 ± 0.2	0.21 ± 0.92
	10	0.98 ± 0.0	0.86 ± 0.25	0.85 ± 0.24	0.77 ± 0.24	0.81 ± 0.25	0.82 ± 0.23	-0.01 ± 0.17	-0.77 ± 3.27	0.66 ± 0.53	0.61 ± 0.4	0.73 ± 0.35	0.49 ± 0.94
	20	0.98 ± 0.0	0.98 ± 0.01	0.92 ± 0.19	0.89 ± 0.22	0.97 ± 0.05	0.92 ± 0.18	0.03 ± 0.17	0.81 ± 0.13	0.82 ± 0.4	0.81 ± 0.38	0.96 ± 0.07	0.77 ± 0.4
Wrinkle	5	0.98 ± 0.0	0.87 ± 0.12	0.69 ± 0.16	0.58 ± 0.13	0.52 ± 0.14	0.65 ± 0.19	0.0 ± 0.0	-0.83 ± 2.08	0.29 ± 0.36	0.28 ± 0.22	0.33 ± 0.19	0.19 ± 0.61
	10	0.98 ± 0.0	0.9 ± 0.09	0.75 ± 0.14	0.63 ± 0.17	0.58 ± 0.14	0.7 ± 0.19	0.0 ± 0.0	-0.37 ± 1.34	0.43 ± 0.32	0.37 ± 0.29	0.42 ± 0.2	0.31 ± 0.46
	20	0.98 ± 0.0	0.92 ± 0.07	0.88 ± 0.12	0.7 ± 0.17	0.68 ± 0.2	0.78 ± 0.18	0.0 ± 0.0	-0.14 ± 1.02	0.73 ± 0.28	0.48 ± 0.3	0.55 ± 0.27	0.47 ± 0.43



(a) Policy Learning Baselines (b) Planning Baselines

Fig. 4: Normalised coverage of different deep reinforcement learning algorithms on pick-and-place fabric flattening. Planning baselines generally shows better performance than policy learning methods on the principal metric, and Planet-ClothPick beats all other state-of-the-art deep reinforcement learning algorithms in fabric-flattening.

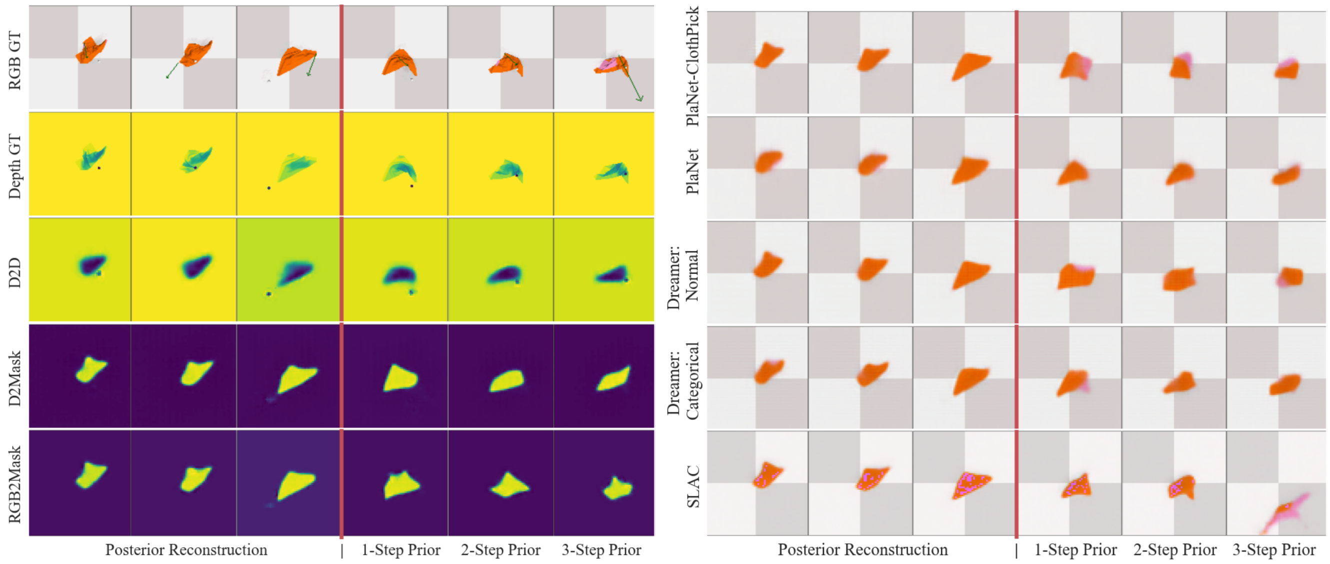


(a) Normalised Improvement (b) Secondary Metrics

Fig. 5: Comparison of PlaNet-ClothPick against state-of-the-art cloth-flattening robotic systems; the colour of the dots in subfigure (b) corresponds to the colour bar for indicating the size of the datasets. Our method achieves a similar level of fabric-fattening as the mesh-based planning methods and surpasses visual planning on the principal metric in simulation, and it showcases a one-order-of-magnitude advantage over the action inference time and transitional model parameters over these systems.

IV. Experiments

We standardise the pick-and-place (P&P) fabric-flattening simulated environment and conduct all experiments in SoftGym [19], which originally provided the basic functionality of the simulation and the performance benchmark on the velocity control. We assess manipulation performance through normalised coverage (NC) and normalised improvement (NI) across the action steps up to 20 steps, as it is the standard in the cloth-flattening literature. We additionally evaluate latent dynamic models (LDMs) through the observation reconstruction from posteriors, which gives a good indication of posterior representation learning, which is essential for providing good initial latent states for the planning (see Figure 6). Following VSF [9], we manually select a subset of testing states and classify them into five tiers based on initial coverage (see Table I). Note that all the methods in



(a) Depth and Mask Reconstruction of PlaNet-ClothPick

(b) RGB2RGB Reconstruction of LDMs

Fig. 6: Reconstruction quality of latent dynamic models compared to ground truth (GT). By incorporating *KL balancing*, PlaNet-ClothPick produces the best posterior and prior observation reconstruction quality.

this work are trained offline as other SoTA robotic systems [9], [13], [14].

A. Comparison against SoTA methods

We benchmark PlaNet-ClothPick (Section III-B) on fabric flattening with our reward function against SoTA policy learning deep reinforcement learning (DRL) algorithms, such as Curl-SAC [34], DrQ-SAC [35], Dreamer with normal and categorical distributions, and SLAC (Figure 4(a)), as well as ClothMaskPick-MPC (Section III-B.3) planning on the LDMs from PlaNet, Dreamer with both variants and SLAC (Figure 4(b)). Note that the policy learning DRL baselines are trained with 500,000 update steps, and LDMs with 100,000 update steps — we employ ClothMaskPick-MPC to plan on these LDMs for consistency. Both sets of baselines learn from our special dataset with the proposed reward function and 64×64 RGB images as input/output (I/O) observation by rescaling the values within the range of [-0.5, 0.5]. For controlling the variables while comparing against PlaNet-ClothPick, only the LDM baselines apply our data augmentation, as some of the policy learning methods come with their own.

We also compare our method against the reported performance of the previous SoTA cloth-flattening robotic systems (Figure 5), such as VSF, VCD and VCD-GI, from Lin et al. (2022) [14]. In addition, we experiment on the heuristic method *Wrinkle* proposed by Sun et al. (2013) [7] via integrating the corresponding implementation of Seita et al. (2020) [8] to our environment. Note that this implementation approximates the detection of wrinkles from the true particle positions of the fabric rather than from the depth camera as in the original method; hence, it only works in simulation. Then, we examine the action inference time of all successful methods on a GeForce RTX 3090 GPU with the systems' default setting.

Our result shows that PlaNet-ClothPick outperforms all the general DRL algorithms in fabric flattening. It also statistically surpasses SoTA NI-against-step performance of VSF and VCD, reaching the same level of competence as VCD-GI. Moreover, our method exhibits around 10× faster action inference time and 10× fewer transitional model parameters compared to the three baselines. However, it does need 10× more data to train.

B. Study on PlaNet-ClothPick

(i) How significant are the different components of the PlaNet-ClothPick?

We train our model on two other reward functions: normalised coverage and the reward from Hoque et al. (2022) [9]. Figure 3(a) indicates that our reward function is key to achieving near-perfect performance, especially for cases with low initial coverage. It also shows that PlaNet-ClothPick outperforms the expert policy used for data generation.

Figure 3(b) presents the significance of the different parts of the model during the training and inference time. We observe that cross-entropy model predictive control (MPC-CEM) often produces an action slightly outside the cloth, which always misses the cloth and makes the algorithm operationally inefficient (see Figure 1); the proposed ClothMaskPick-MPC is critical for achieving effective flattening in general. Sufficient data, data augmentation, *KL balancing* and prior reward learning are all essential for PlaNet-ClothPick to achieve near-perfect flattening.

(ii) How does KL balancing contribute to the success of the PlaNet-ClothPick? Figure 8 shows that *KL balancing* generates a latent space that leads to better observation and

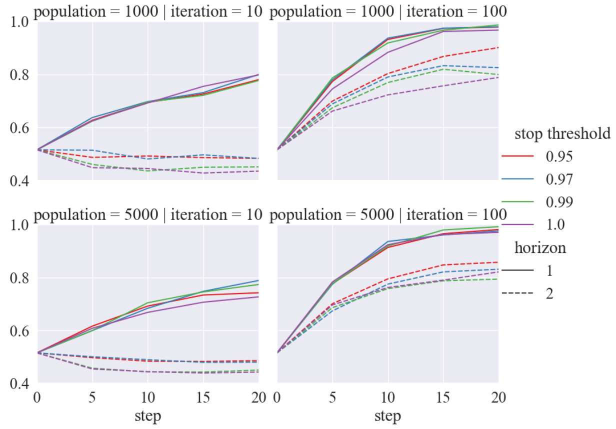
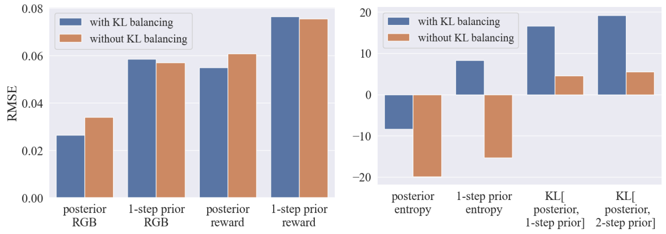


Fig. 7: Normalised coverage of ClothMaskPick-MPC with different hyperparameters. More optimisation iterations and populations produce better and more effective planning results. However, the proposed planning method struggles with multi-step horizons, as it cannot estimate the prior cloth mask for further planning steps.



(a) Prediction Error

(b) Latent Properties

Fig. 8: Effects of *KL balancing*. Although *KL balancing* increases the entropy of the latent space and the divergence between the posterior and prior representation, it produces better latent space reflected by the better accuracy of the observation and reward posterior prediction.

reward prediction accuracy, which provides a better initial state estimate for planning.

(iii) How does the RSSM model's input/output variants affect the performance of PlaNet-ClothPick?

Table II indicates that the D2D variant performs worse than the D2RGB and D2Mask variants; combining the reconstruction quality of the variants from Figure 6 suggests that compressing the depth-only information cannot produce good latent representation for achieving fabric-flattening. Besides, the depth or RGB input of the RSSM model does not affect the performance of PlaNet-ClothPick when the output observation is informative enough for learning good latent representation.

In addition, comparing the results of RGB2Mask to RGB2Mask-fm and D2Mask to D2Mask-fm, we conclude that fetching the accurate estimation of the cloth mask for ClothMaskPick-MPC is important for selecting the most effective picking position on the cloth.

(iii) How robust is ClothMaskPick-MPC?

We examine the hyper-parameters of ClothMaskPick-MPC on the RGB2RGB variant of PlaNet-ClothPick. Figure 7 demonstrates that more optimisation iterations and larger populations produce better and more effective planning results. However, the proposed planning method struggles with multi-step horizons, as it cannot estimate the prior cloth mask for further planning steps.

V. Conclusion

This paper investigates the failure of latent dynamic models (LDMs) on fabric flattening. To our knowledge, this is the first time an Recurrent State Space Models (RSSM) based model has shown state-of-the-art (SoTA) performance on the fabric-flattening task. We find that the sharp discontinuity of the transition function on the fabric's contour makes it difficult to learn an accurate LDM, causing the Model Predictive Control (MPC) planner to produce pick actions slightly outside of the cloth. We employ ClothMaskPick-MPC, *KL balancing*, prior reward learning, data augmentation, and special data collection to improve the performance and robustness of PlaNet in this domain.

Our mesh-free method PlaNet-ClothPick achieves SoTA performance regarding primary metrics among all the reinforcement learning methods, an order-of-magnitude advantage over the action inference time and the number of transitional model parameters compared to the previous SoTA robotic systems in this domain.

In the future, we would like to investigate our method in real-world trials and extend its application to garment flattening. We also plan to reduce the inductive biases we introduced in the data collection by applying SoTA exploration strategies and those in planning algorithms by combining policy learning and planning. Finally, we will investigate the multi-step prediction ability of the RSSM models and make the planning algorithm more robust with multi-step planning, potentially making the flattening more operationally effective.

References

[1] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. Long Beach, CA, USA: PMLR, 10–15 June 2019, pp. 2555–2565.

[2] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, Addis Ababa, Ethiopia, 26–30 April 2020. [Online]. Available: https://openreview.net/forum?id=S1lOTC4tDS

[3] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=0oabwyZbOu

[4] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[5] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq *et al.*, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.

[6] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[7] 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*. Oxford, UK: Springer, 8–30 August 2013, pp. 148–160.

[8] 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.

[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*, vol. 46, no. 1, pp. 175–199, 2022a.

[10] H. A. Kadi and K. Terzić, “Data-driven robotic manipulation of cloth-like deformable objects: The present, challenges and future prospects,” *Sensors*, vol. 23, no. 5, p. 2389, 2023.

[11] 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)*. Xi’an, China: IEEE, 30 May–5 June 2021, pp. 4568–4575.

[12] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, “Learning to manipulate deformable objects without demonstrations,” *arXiv preprint arXiv:1910.13439*, 2019.

[13] X. Ma, D. Hsu, and W. S. Lee, “Learning latent graph dynamics for visual manipulation of deformable objects,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8266–8273.

[14] X. Lin, Y. Wang, Z. Huang, and D. Held, “Learning visible connectivity dynamics for cloth smoothing,” in *Conference on Robot Learning*. Auckland, New Zealand: PMLR, 5–18 December 2022, pp. 256–266.

[15] Z. Huang, X. Lin, and D. Held, “Mesh-based dynamics with occlusion reasoning for cloth manipulation,” *arXiv preprint arXiv:2206.02881*, 2022.

[16] Z. Xu, C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, “Dextairity: Deformable manipulation can be a breeze,” in *Proceedings of Robotics: Science and Systems (RSS)*, New York, NY, USA, 27 June–1 July 2022.

[17] H. Ha and S. Song, “Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding,” in *Conference on Robot Learning*. Auckland, New Zealand: PMLR, 15–18 Decemeber 2022, pp. 24–33.

[18] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, “Learning predictive representations for deformable objects using contrastive estimation,” in *Conference on Robot Learning*. London, UK: PMLR, 8–11 November 2021, pp. 564–574.

[19] X. Lin, Y. Wang, J. Olkin, and D. Held, “Softgym: Benchmarking deep reinforcement learning for deformable object manipulation,” in *Conference on Robot Learning*. London, UK: PMLR, 8–11 November 2021, pp. 432–448.

[20] 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*. London, UK: PMLR, 11 November 2021, pp. 2317–2327.

[21] T. M. Moerland, J. Broekens, and C. M. Jonker, “A framework for reinforcement learning and planning,” *arXiv preprint arXiv:2006.15009*, 2020.

[22] F. Ebert, C. Finn, S. Dasari, A. Xie, A. X. Lee, and S. Levine, “Visual foresight: Model-based deep reinforcement learning for vision-based robotic control,” *CoRR*, vol. abs/1812.00568, 2018. [Online]. Available: http://arxiv.org/abs/1812.00568

[23] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia, “Learning mesh-based simulation with graph networks,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=roNqYL0_XP

[24] Z. Huang, X. Lin, and D. Held, “Self-supervised cloth reconstruction via action-conditioned cloth tracking,” *arXiv preprint arXiv:2302.09502*, 2023.

[25] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, “Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5872–5879.

[26] W. Wang, G. Li, M. Zamora, and S. Coros, “Trtm: Template-based reconstruction and target-oriented manipulation of crumpled cloths,” *arXiv preprint arXiv:2308.04670*, 2023.

[27] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine, “Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 741–752, 2020.

[28] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*. Stockholm, Sweden: PMLR, 10–15 July 2018, pp. 1861–1870.

[29] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, “Modeling interaction via the principle of maximum causal entropy,” in *ICML*, 2010.

[30] S. Levine, “Reinforcement learning and control as probabilistic inference: Tutorial and review,” *arXiv preprint arXiv:1805.00909*, 2018.

[31] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[32] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, “Stochastic adversarial video prediction,” *arXiv preprint arXiv:1804.01523*, 2018.

[33] 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*. uckland, New Zealand: PMLR, 15–18 December 2022, pp. 192–202.

[34] M. Laskin, A. Srinivas, and P. Abbeel, “Curl: Contrastive unsupervised representations for reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[35] I. Kostrikov, D. Yarats, and R. Fergus, “Image augmentation is all you need: Regularizing deep reinforcement learning from pixels,” *arXiv preprint arXiv:2004.13649*, 2020.