Learning Predictive Representations for Deformable Objects Using Contrastive Estimation

Wilson Yan[1,†], Ashwin Vangipuram[1], Pieter Abbeel[1], and Lerrel Pinto[1,2]

***Abstract*—Using visual model-based learning for deformable object manipulation is challenging due to difficulties in learning plannable visual representations along with complex dynamic models. In this work, we propose a new learning framework that jointly optimizes both the visual representation model and the dynamics model using contrastive estimation. Using simulation data collected by randomly perturbing deformable objects on a table, we learn latent dynamics models for these objects in an offline fashion. Then, using the learned models, we use simple model-based planning to solve challenging deformable object manipulation tasks such as spreading ropes and cloths. Experimentally, we show substantial improvements in performance over standard model-based learning techniques across our rope and cloth manipulation suite. Finally, we transfer our visual manipulation policies trained on data purely collected in simulation to a real PR2 robot through domain randomization.**

I. Introduction

Robotic manipulation of rigid objects has received significant interest over the last few decades, from grasping novel objects in clutter [6, 5, 14, 9, 2] to dexterous in-hand manipulation [4, 1, 20]. However, the objects we interact within our daily lives are not always rigid. From putting on clothes to packing a shopping bag, we constantly need to manipulate objects that deform. As a result, there has been a growing interest in algorithms that can tackle deformable object manipulation [18, 3, 10, 11, 12, 19, 13, 7, 15].

Deformable object manipulation presents two key challenges for robots: no direct reprsentation of state and non-linear, complex dynamics. To address these issues, we introduce a new visual model-based framework that uses contrastive optimization to jointly learn both the underlying visual latent representations and the dynamics models for deformable objects. We hypothesize that using contrastive methods for model-based learning achieves better generalization and latent space structure do to its inherent information maximization objective. We re-frame the objective introduced in contrastive predictive coding [8] to allow for learning effective model dynamics and latent representations. Once the latent models for representations and dynamics are learned across offline random interactions, we use standard model predictive control (MPC) with one-step predictions to manipulate deformable objects to desired visual goal configurations. We demonstrate substantial improvements in multi-task deformable object manipulation over other model learning approaches. Finally, we show the applicability of our method to real robot rope and cloth manipulation tasks by using sim-to-real transfer without additional real-world training data. Videos of our real robot runs and reference code can be found on the project website: https://sites.google.com/view/contrastive-predictive-model.

[1]Department of EECS, University of California, Berkeley.
[2]Department of Computer Science, New York University.
[†]Correspondence to wilson1.yan@berkeley.edu

II. Contrastive Forward Modeling (CFM)

A. Contrastive Models

In our contrastive learning framework, we jointly learn an encoder $g_\theta(o_t) = z_t$ and a forward model $f_\phi(z_t, a_t) \approx z_{t+1}$. We use the InfoNCE contrastive loss described by Oord et al. [8].

$$\mathcal{L} = -\mathbb{E}_{\mathcal{D}}\left[\log \frac{h(\hat{z}_{t+1}, z_{t+1})}{\sum_{i=1}^{k} h(\hat{z}_{t+1}, \tilde{z}_i)}\right] \tag{1}$$

where h is some similarity function between the computed embeddings from the encoder. The $\tilde{z}_i$ represents negative samples, which are incorrect embeddings of the next state, and we use k such negative samples in our loss. The motivation behind this learning objective lies with maximizing mutual information between the predicted encodings and their respective positive samples. Within the embedding space, this results in the positive sample pairs being aligned together but the negative samples pushed further apart, as seen in Figure **??**. Since we are jointly learning a forward model that seeks to minimize $\|f_\phi(z_t, a_t) - z_{t+1}\|^2$, we use the similarity function:

$$h(z_1, z_2) = \exp(-\|z_1 - z_2\|^2) \tag{2}$$

where the norm is a ℓ_2-norm. After learning the encoder and dynamics model, we plan using a simple version of Model Predictive Control (MPC), where we sample several actions, run them through the forward model from the current z_t, and choose the action a_t that produces $\hat{z}_{t+1}$ closest (in ℓ_2-distance) to the goal embedding.

III. Experimental Evaluations

In this section, we experimentally evaluate our method in various rope and cloth manipulation settings, both in simulation and in the real world.

A. Environments and Tasks

To simulate deformable objects such as cloth and rope, we used the Deep Mind Control [16] platform with MuJoCo 2.0 [17]. We use an overhead camera that renders $64 \times 64 \times 3$ RGB images as input observations for training our method.

1. Rope: The rope is represented by 25 geoms in simulation with a four-dimensional action space: the first 2 are the pixel pick point on the rope, and the last 2 are the x, y delta direction to perturb the rope.

TABLE I: Quantitative comparisons between different model-based learning methods on rope and cloth manipulation tasks in the simulator. The metric is the sum of pairwise geom distances between the final observation and goal state, where lower distance is more accurate.

	Rope Horizontal	Rope Vertical	Rope 45°	Rope 135°	Rope Random	Cloth Flat	Cloth Random
Random Policy	4.75	4.93	4.80	4.87	5.73	7.98	10.12
Autoencoder	1.72 ± 0.31	3.24 ± 1.28	$\mathbf{2.11 \pm 0.51}$	2.49 ± 0.64	4.308 ± 1.16	3.24 ± 0.29	4.82 ± 0.0
PlaNet	1.81 ± 0.13	3.36 ± 0.78	2.31 ± 0.72	2.38 ± 0.20	3.037 ± 0.24	4.12 ± 0.21	5.06 ± 0.02
Joint Dynamics Model	2.13 ± 0.66	4.33 ± 0.85	3.88 ± 0.95	4.02 ± 0.85	1.78 ± 0.09	4.24 ± 0.06	4.70 ± 0.03
Visual Forward Model	2.09 ± 0.13	$\mathbf{2.65 \pm 0.27}$	2.55 ± 0.34	2.27 ± 0.17	4.77 ± 0.18	$\mathbf{2.20 \pm 0.05}$	4.65 ± 0.10
CFM (Ours)	$\mathbf{0.58 \pm 0.09}$	3.08 ± 1.19	2.29 ± 1.42	$\mathbf{2.24 \pm 0.90}$	$\mathbf{1.52 \pm 0.10}$	2.69 ± 0.25	$\mathbf{3.97 \pm 0.16}$

	Rope (With DR) Horizontal	Rope (With DR) Vertical	Rope (With DR) 45°	Rope (With DR) 135°	Rope (With DR) Random	Cloth (With DR) Flat	Cloth (With DR) Random
Random Policy	4.75	4.93	4.80	4.87	5.73	7.975	10.12
Autoencoder	3.29 ± 1.08	3.70 ± 1.47	3.19 ± 1.14	3.30 ± 1.14	4.31 ± 1.16	6.26 ± 1.23	7.08 ± 2.22
PlaNet	2.35 ± 0.56	4.06 ± 1.84	3.73 ± 1.66	3.58 ± 1.46	3.04 ± 0.24	8.74 ± 0.55	10.10 ± 1.56
Joint Dynamics Model	1.01 ± 0.40	2.29 ± 0.10	1.35 ± 0.59	1.82 ± 0.50	1.78 ± 0.09	4.17 ± 0.17	4.64 ± 0.20
Visual Forward Model	3.05 ± 0.45	5.65 ± 0.37	5.37 ± 0.90	5.11 ± 1.04	4.77 ± 0.18	6.64 ± 0.66	6.07 ± 0.52
CFM (Ours)	$\mathbf{0.88 \pm 0.21}$	$\mathbf{1.20 \pm 0.07}$	$\mathbf{0.99 \pm 0.07}$	$\mathbf{0.99 \pm 0.17}$	$\mathbf{1.38 \pm 0.03}$	$\mathbf{3.99 \pm 0.15}$	$\mathbf{4.40 \pm 0.06}$

TABLE II: The maximum intersection area in pixels between the goal image and observation images averaged over all seeds

Robot Experiments (Intersection in pixels)	Rope (Horizontal)	Rope (Vertical)	Rope (45°)	Rope (135°)	Rope (Squiggle)	Cloth (Flat)
Random Policy	6.880	14.727	13.662	4.266	0.049	462.513
Autoencoder	5.526	3.334	3.862	7.499	3.419	603.927
Joint Dynamic Model	17.722	23.636	33.631	21.267	18.311	772.303
Contrastive Forward Model (Ours)	**32.827**	**36.387**	**33.891**	**38.952**	**20.711**	**1001.082**

2. Cloth: The cloth is represented by a 9×9 grid of geoms in simulation with a five-dimensional action space: the first 2 are the pixel pick point on the cloth, and the last 3 are the x, y, z delta direction to perturb the cloth.

For both rope and cloth environments, we evaluate our method by planning to a desired goal state image and computing the sum of the pairwise geom distances between the achieved and true goal states.

B. Data Collection

Since collecting real-world data on robots is expensive, our method seeks to address this problem by collecting randomly perturbed rope and cloth data in simulation. Using random perturbations allows for a diverse set of deformable objects and interactions for learning the latent space and dynamics model.

C. Does Using Contrastive Models Improve Performance?

We now compare the results of using our method with those of our baselines, analyzing the advantages and benefits that contrastive models bring over prior methods. Consider a naive baseline where we replace the InfoNCE loss with an MSE loss. However this would make the encoder encode all observations to a constant vector to achieve zero loss. To prevent this form of a degenerate solution, we are required to regularize our latent space in some way. Both prior methods and contrastive learning do this in different ways so we analyzed which methods performed better over others. Table I shows the quantitative results comparing our method against baselines in different rope and cloth environments, with and without domain randomization for robot transfer. Note that our method does better on all randomly sampled goals with domain randomization, indicating stronger generalization in latent spaces for planning.

D. Real Robot Experiments

We use a PR2 robot to perform our experiments and an overhead camera looking down on the deformable objects to get the RGB image inputs. To ensure the policy learned in the simulator transfers over to the real world, we apply domain randomization by changing the lighting, texture, friction, damping, inertia, and mass of the object during every training step within the simulator. We also use a pick and place strategy to mimic the same four-dimensional actions within the simulator.

To compute the actions, we employ a model predictive control (MPC) approach of replanning our action at each time step based on the previous image. We segment the rope/cloth against the background to get the list of valid pick locations of the object. We then generate possible actions by uniformly sampling 100 random deltas in $[-1, 1]$ combined with randomly chosen start locations. We feed these into our forward model along with the encoding of our start image to get the latent encoding for each of the next prospective states. To pick the optimal action, we find the location and delta that minimizes the Euclidean distance from these next states to our goal state and return this action to the robot. Our results in Table II comparing our method to the baselines include 4 different starting locations for the rope and 2 different colors for the cloth to represent different seeds for the model.

REFERENCES

[1] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint*, 2018.

[2] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *NeurIPS*, 2018.

[3] Dominik Henrich and Heinz Wörn. *Robot manipulation of deformable objects*. Springer Science & Business Media, 2012.

[4] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *ICRA*, 2016.

[5] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *ISER*, 2016.

[6] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *ICRA*, 2016.

[7] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *ICRA*, 2010.

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint*, 2018.

[9] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *ICRA*, 2016.

[10] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Generalization in robotic manipulation through the use of non-rigid registration. In *ISRR*, 2013.

[11] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *ICRA*, 2013.

[12] Daniel Seita, Nawid Jamali, Michael Laskey, Ajay Kumar Tanwani, Ron Berenstein, Prakash Baskaran, Soshi Iba, John Canny, and Ken Goldberg. Deep transfer learning of pick points on fabric for robot bed-making. *arXiv preprint*, 2018.

[13] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, Katsu Yamane, Soshi Iba, John Canny, and Ken Goldberg. Deep imitation learning of sequential fabric smoothing policies. *arXiv preprint*, 2019.

[14] Karun B Shimoga. Robot grasp synthesis algorithms: A survey. *IJRR*, 1996.

[15] Jan Stria, Daniel Prusa, Vaclav Hlavac, Libor Wagner, Vladimir Petrik, Pavel Krsek, and Vladimir Smutny. Garment perception and its folding using a dual-arm robot. In *IROS*, 2014.

[16] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint*, 2018.

[17] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

[18] Takahiro Wada, Shinichi Hirai, Sadao Kawamura, and Norimasa Kamiji. Robust manipulation of deformable objects by a simple pid feedback. In *ICRA*, 2001.

[19] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. *arXiv preprint*, 2019.

[20] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors and Actuators A: physical*, 2011.