Learning Latent Graph Dynamics for Deformable Object Manipulation

Xiao Ma, David Hsu, Wee Sun Lee
National University of Singapore
{xiao-ma, dyhsu, leews}@comp.nus.edu.sg

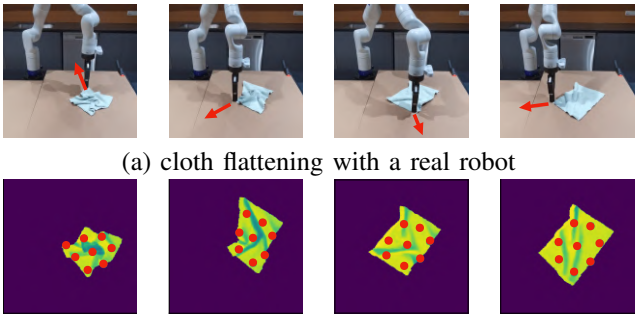
***Abstract*—Manipulating deformable objects, such as cloth and ropes, is a long-standing challenge in robotics: their large degree of freedom (DoFs) and complex non-linear dynamics make motion planning extremely difficult. This work aims to learn latent *Graph dynamics for DefOrmable Object Manipulation* (G-DOOM). To tackle the challenge of many DoFs and complex dynamics, G-DOOM approximates a deformable object as a sparse set of interacting *keypoints* and learns a *graph neural network* that captures abstractly the geometry and interaction dynamics of the keypoints. Further, to tackle the perceptual challenge, specifically, object self-occlusion, G-DOOM adds a recurrent neural network to track the keypoints over time and condition their interactions on the history. We then train the resulting recurrent graph dynamics model through contrastive learning in a high-fidelity simulator. For manipulation planning, G-DOOM explicitly reasons about the learned dynamics model through model-predictive control applied at each of the keypoints. We evaluate G-DOOM on a set of challenging cloth and rope manipulation tasks and show that G-DOOM outperforms a state-of-the-art method. Further, although trained entirely on simulation data, G-DOOM transfers directly to a real robot for both cloth and rope manipulation in our experiments. More details are available online at https://sites.google.com/view/g-doom.**

***Index Terms*—Deformable Object Manipulation, Graph Neural Networks**

I. Introduction

Robot manipulation for rigid-body objects has achieved significant progress in recent years, including grasping novel objects in clutter [13, 14], pushing novel objects [8], and solving Rubik's cube [1]. Nevertheless, many daily objects we interact with are non-rigid, from folding clothes to packing grocery bags. Extending existing rigid-body object manipulation algorithms to deformable objects remains challenging because: 1) the degree of freedom of a deformable object is too large for traditional models, making explicit planning very difficult; 2) the dynamics of deformable objects are highly complex and non-linear due to the microscopic interactions in the object itself, which is difficult to model and leads to unpredictable behaviors [27]; 3) the deformation of an object leads to partial observability. Consider the example of flattening a cloth in Fig. 1: it is unclear how to mathematically specify the state of the cloth, and predicting the exact motion of the cloth is difficult even for a human, given the self-occlusion.

Pioneering works for deformable object manipulation rely on the low-dimensional geometric features to specify the object states and perform decision making by planning with a predefined dynamic model [16, 23, 25]. However, handcrafted geometric features often generalize poorly to unseen configurations, and predefined dynamics introduce accumulative error during long-horizon planning [5, 12]. With the recent advances in model-free visual policy learning [17, 18, 20], learning-based methods improve the deformable object manipulation performance. They avoid modeling the object state by directly mapping raw visual inputs to robot actions [15, 24, 27]. In particular, Yan et al. [28] introduces Contrastive Forward Models (CFMs) that learn a latent dynamic model from visual inputs and improve the sample efficiency of learning-based algorithms by explicit reasoning. Nevertheless, the geometric structures, which are essential to understanding the dynamics of a deformable object [9], have been neglected.



(a) cloth flattening with a real robot

(b) detected keypoints from depth images

Fig. 1. Robot cloth flattening. (a) G-DOOM flattens a piece of crumpled cloth by reasoning a dynamic model learned from simulation data. (b) G-DOOM approximates the cloth, which has a large degree of freedom, by a low-dimensional keypoint-based graph generated from unsupervised learning.

We argue that understanding the geometric structure of a deformable object is useful, but accurately modeling the dynamics of the entire deformable object is unnecessary for a manipulation task. Consider how a human manipulates a deformable object: to fold a piece of cloth, instead of considering the dynamics of the entire object, human only considers keypoints of the cloth, e.g., the collar, shoulders, and sleeves. However, detecting the keypoints and modeling their non-linear spatio-temporal interactions remain non-trivial.

We present *latent Graph dynamics for DefOrmable Object Manipulation* (G-DOOM), a framework for keypoint-based deformable object manipulation with only visual observations. Instead of explicitly modeling the entire object, G-DOOM abstracts the state of a deformable object as a low-dimensional keypoint-based graph with learned latent features. G-DOOM models the complex non-linear keypoint interactions by *Graph*

Neural Networks (GNNs) directly learned from data. Such a formulation explicitly represents the state of the object, simplifies the non-linear dynamics, and avoids accumulative errors of hand-crafted models. Specifically, G-DOOM discovers salient keypoints in an unsupervised manner from depth images with Transporter Networks [7] and extracts keypoint-centric feature vectors by masking the features with keypoint-based attention maps. The observed keypoints are grouped into a graph and high-level interactions among keypoints are effectively captured by Graph Neural Networks (GNNs). Nevertheless, abstraction into keypoints unavoidably leads to information loss. Moreover, the deformation of the object leads to self-occlusion and inaccurate keypoint detections, which eventually introduce partial observability to the task. To tackle these issues, we introduce a hybrid-scheme, *Recurrent Graph Dynamics*. It tracks the *belief*, i.e., the sufficient global statistics of the deformable object, with a Recurrent Neural Network (RNN), and predicts the next belief conditioned on the current belief and current graph state. By combining graph-based modeling with belief tracking, G-DOOM can successfully estimate a global state of a deformable object. Contrastive learning is further used to obtain a robust latent space that is accurate for planning. For effective decision making, we reason the learned dynamic model with a graph-based Model-Predictive Control (MPC), which bootstraps the search with the keypoint positions.

We evaluate G-DOOM on three deformable manipulation tasks: rope straightening, cloth flattening, and cloth folding. We first show that in a realistic simulator, NVidiaFlex [2], G-DOOM outperforms the state-of-the-art model-based deformable object manipulation methods on all three tasks. Besides, we show that our learned dynamic model using simulated data successfully transfers to a real-world scenario with a Kinova Gen3 robot.

II. G-DOOM

We introduce *latent Graph Dynamics for DefOrmable Object Manipulation* (G-DOOM). (Fig. 2). In this section, we only provide an overview to the approach. A detailed description can be found in our website, https://sites.google.com/view/g-doom.

In contrast to standard latent models which represent the state with a single vector [4, 11, 28], G-DOOM abstracts a deformable object as a set of keypoints grouped into a graph G_t with learned node features, which provides rich semantics including keypoint positions, depths, textures, etc. However, modeling the high-level keypoint interactions is non-trivial. Consider manipulating a straightened rope: if we pull one end towards the other end, the rope loosens and the other end remains at its original position; if we drag one end towards the opposite direction, the rope remains straightened and both ends move together. This makes manually constructing a dynamic model of the keypoint-based representation difficult. We parameterize the interactions as attention-based Graph Neural Networks (GNNs) learned directly from data, which avoids manually constructing the model and improves the predictive accuracy. In addition, to tackle the partial observability caused by self-occlusion and inaccurate keypoint detection, G-DOOM adopts a hybrid approach, *Recurrent Graph Dynamics*. Besides the graph state G_t, it introduces an additional *belief* state, h_t, which tracks the sufficient statistics of an object by summarizing the graph states history $G_1, G_2, \ldots, G_t$. The graph state update is then conditioned on the belief. The hybrid state representation performs implicit belief tracking and provides a global understanding during spatial interaction. For decision making, a simple yet effective graph-based MPC is used, which initializes the search based on the detected keypoints positions to improve the sample efficiency and convergence.

III. Experiments

We first evaluate the proposed G-DOOM on a set of rope straightening and cloth manipulation tasks in a high-fidelity simulator, NVidia-Flex [2]. To minimize the sim-to-real gap, we use masked depth images as the input, and we show that our learned dynamic model transfers directly to a real-robot.

We compare G-DOOM with the state-of-the-art (SOTA) model-based deformable object manipulation method, *Contrastive Forward Model* (CFM) [28], and a SOTA general-purpose model-based RL method, PlaNet [4]. For all baselines, we use the publicly available implementations. We show that: 1) G-DOOM generally outperforms all baselines in all tasks; 2) the recurrent graph dynamics improves the quality of the learned dynamics; 3) contrastive learning improves the accuracy of learned dynamics; 4) graph-based MPC generally improves overall performance.

In this section, we only report the real robot experiment results. Additional simulation experiments, setups, and ablation studies can be found in our full paper available on our website.

A. Real Robot Experiment

We further evaluate our learned model on a Kinova Gen3 robot, as shown in Fig. 3.a. To collect high-quality depth images, we mount a top-down Kinect 2.0 camera over the workspace. We observe that high-quality depth images and the simplified pick-and-place action model help to minimize the sim-to-real gap, and our trained models transfer directly to the real robot.

Evaluation metric: We measure the distance-to-goal by counting the number of pixels within a goal region. Denoting the set of pixels covered by a deformable object as S_o, we define the score as follows. For rope straightening tasks, we define goal region S_g to be a rectangle centered in the middle of the image rotated for different degrees $(0°, 45°, 90°, 135°)$, and measure score $= |S_o \cap S_g|$; for cloth flattening, we simply compute the total number of pixels of the covered area by score $= |S_o|$; for cloth folding, we define the goal area to be half of the cloth in the initial frame and measure score $= -\,||S_o| - |S_g||$. All results are averaged over 3 random seeds.

Results: The quantitative results of the real robot experiments are given in Tab. I and visualizations are provided in Fig. 3.

G-DOOM generally outperforms the baselines. In real robot experiments, G-DOOM achieves higher scores than the baselines, which is consistent with our simulation results.

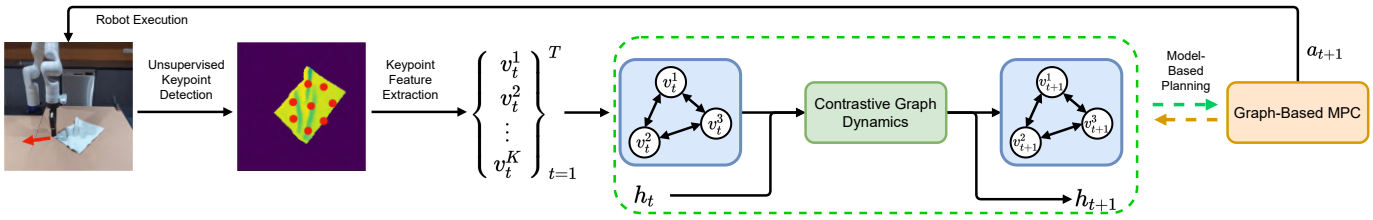
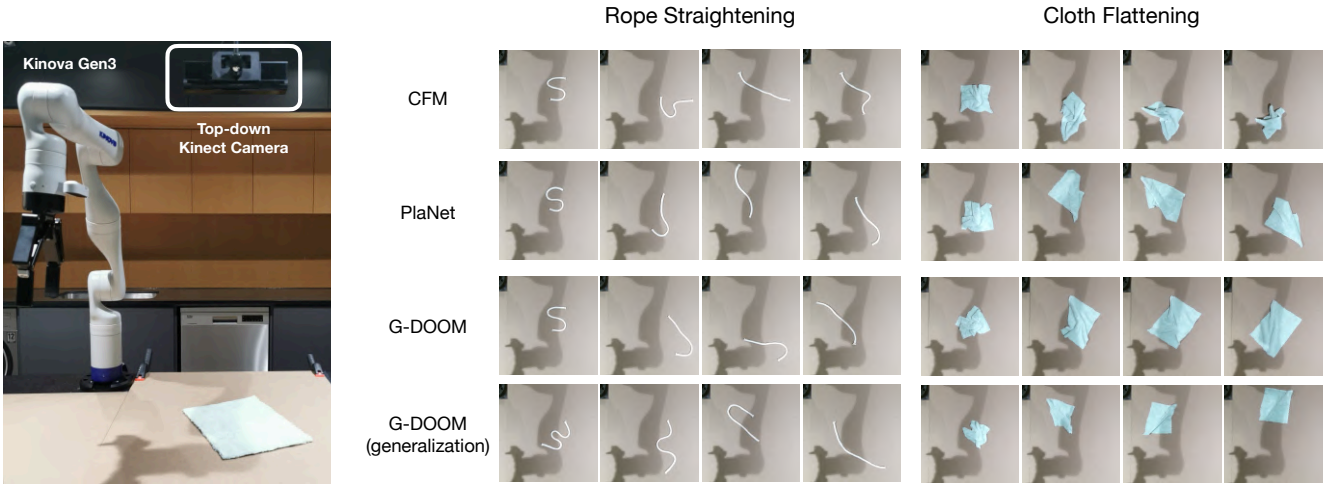


Fig. 2. G-DOOM Pipeline. G-DOOM performs unsupervised keypiont detection using depth images and extracts corresponding keypoint features $\{v_t^i\}_{i=1}^K$, which are composed into a graph according to the spatial relationships. Recurrent graph dynamics learns to predict the future states considering the spatio-temporal interaction among the local graph features $\{v_t^i\}_{i=1}^K$ and global statistics h_t, i.e., the belief. A graph-based Model-Predictive Control (MPC) reasons the learned graph dynamics and compute the action a_t conditioned on the detected keypoints.



(a) Robot Setup (b) Execution Sequences

Fig. 3. (a) We use a Kinova Gen3 robot with a Robotiq gripper for the experiment. A top-down Kinect 2.0 camera is used to produce depth images. (b) Visualizations of execution trajectories on a real robot (Rope Straightening 135° and Cloth Flattening). G-DOOM (generalization) shows that our trained model generalizes to different objects with different initial configurations, e.g., longer ropes and smaller cloths.

TABLE I
REAL ROBOT EXPERIMENT RESULTS

	Rope				Cloth	
	0°	45°	90°	135°	Flatten	Fold
CFM	1.378	33.26	45.64	14.49	515.24	-158.15
PlaNet	40.93	**68.52**	36.92	11.51	946.82	-199.08
G-DOOM	**81.91**	67.56	**47.92**	**53.50**	**1,458.15**	**-42.15**

Graph-based dynamics allow G-DOOM to generalize better. In the simulation, PlaNet achieves reasonable performance on rope straightening tasks, while on a real robot, it fails on rope straightening 0° and 135°. In contrast, G-DOOM generalizes in all cases. This is potentially because by down-sampling an object into a keypoint-based graph, G-DOOM constructs an information bottleneck that filters the high-frequency noise and maintains a minimum amount of information for modeling the dynamics. Also, the recurrent graph dynamics compensate for the information loss. As shown in Fig.3 G-DOOM (generalization), our trained model can be directly applied to different objects, e.g., longer ropes and smaller cloths.

Due to the space limit, we visualize only two real robot tasks in Fig. 3.b.

IV. CONCLUSION

In this paper, we introduce G-DOOM, a graph-based approach for deformable object manipulation. We observe that modeling the full dynamics of a deformable object, which has a large degree of freedom, is unnecessary. G-DOOM performs unsupervised keypoint detection and feature extraction, model the spatio-temporal keypoint interactions using recurrent graph dynamics, and conducts model-based planning using the learned model. G-DOOM tackles the partial observability caused by the self-occlusion of deformable objects by belief tracking with a recurrent neural network. A graph-based MPC is introduced to improve the planning performance. We show that G-DOOM outperforms the SOTA deformable object manipulation methods in both simulation and a real robot.

However, the current framework generates inaccurate detection which introduces noise to the dynamic model. Future works could consider learning temporally and robust keypoints by jointly training the perception and the dynamics module, which might experience high GPU memory cost.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] Gianni Ciccarelli. Particle-based fluid simulation with nvidia flex. 2019.

[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

[4] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

[5] Peter Karkus, Xiao Ma, David Hsu, Leslie Pack Kaelbling, Wee Sun Lee, and Tomás Lozano-Pérez. Differentiable algorithm networks for composable robot learning. *Robotics: Science and Systems*, 2019.

[6] Thomas N. Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=H1gax6VtDB.

[7] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32:10724–10734, 2019.

[8] Jue Kun Li, Wee Sun Lee, and David Hsu. Push-net: Deep planar pushing for objects with unknown physical properties. In *Robotics: Science and Systems*, volume 14, pages 1–9, 2018.

[9] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJgbSn09Ym.

[10] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. *arXiv preprint arXiv:2011.07215*, 2020.

[11] Xiao Ma, Siwei Chen, David Hsu, and Wee Sun Lee. Contrastive variational model-based reinforcement learning for complex observations. *arXiv preprint arXiv:2008.02430*, 2020.

[12] Xiao Ma, Péter Karkus, David Hsu, and Wee Sun Lee. Particle filter recurrent neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 5101–5108. AAAI Press, 2020.

[13] Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken 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 *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1957–1964. IEEE, 2016.

[14] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[15] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 734–743. PMLR, 2018.

[16] Stephen Miller, Jur Van Den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research*, 31(2):249–267, 2012.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[18] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[20] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern*

recognition, pages 652–660, 2017.

[22] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[23] Mitul Saha and Pekka Isto. Manipulation planning for deformable linear objects. *IEEE Transactions on Robotics*, 23(6):1141–1150, 2007.

[24] Daniel Seita, Aditya Ganapathi, Ryan Hoque, Minho Hwang, Edward Cen, Ajay Kumar Tanwani, Ashwin Balakrishna, Brijen Thananjeyan, Jeffrey Ichnowski, Nawid Jamali, et al. Deep imitation learning of sequential fabric smoothing policies. *arXiv preprint arXiv:1910.04854*, 2019.

[25] Eric Torgerson and Frank W Paul. Vision-guided robotic fabric manipulation for apparel manufacturing. *IEEE Control Systems Magazine*, 8(1):14–20, 1988.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[27] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. *arXiv preprint arXiv:1910.13439*, 2019.

[28] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.