Learning Visible Connectivity Dynamics for Cloth Smoothing

Xingyu Lin*
Robotics Institute
Carnegie Mellon University
xlin3@andrew.cmu.edu

Yufei Wang*
Robotics Institute
Carnegie Mellon University
yufeiw2@andrew.cmu.edu

David Held
Robotics Institute
Carnegie Mellon University
dheld@andrew.cmu.edu

***Abstract*—Robotic manipulation of cloth remains challenging for robotics due to the complex dynamics of the cloth, lack of a low-dimensional state representation, and self-occlusions. In contrast to previous model-based approaches that learn a pixel-based dynamics model or a compressed latent vector dynamics, we propose to learn a particle-based dynamics model from a partial point cloud observation. To overcome the challenges of partial observability, we infer which visible points are connected on the underlying cloth mesh. We then learn a dynamics model over this visible connectivity graph. Compared to previous learning-based approaches, our model poses strong inductive bias with its particle based representation for learning the underlying cloth physics; it is invariant to visual features; and the predictions can be more easily visualized. We show that our method greatly outperforms previous state-of-the-art model-based and model-free reinforcement learning methods in simulation. Furthermore, we demonstrate zero-shot sim-to-real transfer where we deploy the model trained in simulation on a Franka arm and show that the model can successfully smooth different types of cloth from crumpled configurations. Videos can be found on our anonymous project website.**[1]

***Index Terms*—TBD**

I. Introduction

Robotic manipulation of cloth has wide applications across both industrial and domestic tasks such as laundry folding and bed making. However, cloth manipulation remains challenging for robotics due to its complex dynamics, lack of low-dimensional representation, and self-occlusions.

One approach to cloth manipulation explored by previous work, which we also adopt, is to learn a cloth dynamics model and then use the model for planning to determine the robot actions. However, it is unclear what is the appropriate state representation for the cloth dynamics model. One choice is to use a mesh model of the entire cloth [3]. However, fitting a full mesh model to an arbitrary crumpled cloth configuration is fairly difficult. Recent work have proposed to compress the cloth representation into a fixed-size latent vector [7], [11]–[13]. Such compressed vector representations lose much of the spatial information of the cloth. Another previous approach is to learn a visual dynamics model in pixel space [2]. However, a pixel-based dynamics model is somewhat disconnected from the real physics of the cloth dynamics.

In contrast to a pixel-based dynamics model, particle-based models have recently been shown to be able to learn dynamics for fluid and plastics [4], [9], [10]. A particle-based dynamics representation has the following benefits: first, it captures the inductive bias of the underlying physics, since real-world objects are composed of underlying atoms that can be modeled on the macro-level by particles. Second, particle-based models are invariant to visual features such as object colors or patterns. Last, such models are interpretable, since a person can visualize the predicted movement of the object particles. As such, in this paper we aim to learn a particle-based dynamics model for the cloth. However, the challenges in applying the particle-based model to cloth are that we cannot directly observe the underlying particles composing the cloth nor their mesh connections. The problem is made even more challenging due to the partial observability of the cloth due to self-occlusions when it is in a crumpled configuration.

Our insight into this problem is that, rather than fitting a mesh model to the observation, we should learn the *visible connectivity dynamics (VCD):* a dynamics model based on the connectivity structure of the visible portion of the cloth. To do so, we first learn to estimate the *visible connectivity graph*: we estimate which points in the point cloud observation are connected in the underlying cloth mesh. Estimating the mesh connectivity of the observation is a simplification of the problem of fitting a single full mesh model of the entire cloth to the observation; however, it is significantly easier to learn, since we do not need to find a globally consistent explanation of the observation; to estimate the mesh connectivity of the observation, we only need to consider the local cloth structure. We further learn a dynamics model directly on the inferred visible connectivity graph.

In this work, we focus on the task of smoothing a piece of cloth from a crumpled configuration. We propose a method that infers the observable particles and their connections from the point cloud, learns a visible connectivity dynamics model for the observable portion of the cloth, and uses it for planning to smooth the cloth. We show that planning with a visible connectivity dynamics model can be effective in smoothing the cloth and our method greatly outperforms state-of-the-art methods that use a fixed-size latent vector representation or learn a pixel-based visual dynamics model. Furthermore, we demonstrate zero-shot sim-to-real transfer where we deploy

* Equal contribution, order by dice rolling.

[1] https://sites.google.com/view/vcd-cloth

the model trained in simulation on a Franka arm and show that the learned model can be used to successfully smooth different types of cloth from crumpled configurations.

II. Method

A. Graph Representation of Cloth Dynamics

We represent the state of a cloth with a graph $\langle V, E\rangle$. The nodes $V = \{v_i\}_{i=1\ldots N}$ represent the particles that compose the cloth, where $v_i = (x_i, \dot{x}_i)$ denotes the particle's current position and velocity, respectively. There are two types of edges E in the graph, representing two types of interactions between the particles: mesh edges and collision edges. The mesh edges, E^M, represent the connections among the particles on the underlying cloth mesh. The mesh connectivity is determined by the structure of the cloth and does not change throughout time. Each edge $e_{ij} = (v_i, v_j) \in E^M$ connects nodes v_i to v_j and models the mesh connection between the two nodes. The other type of edges are collision edges, E^C, which model the collision dynamics among two particles that are nearby in space. These collision edges are dynamically constructed at each time step based on the following criteria:

$$E_t^C = \{e_{ij} \big| \; ||x_{i,t} - x_{j,t}||_2 < R\}, \qquad (1)$$

where R is a distance threshold and $x_{i,t}, x_{j,t}$ are the positions of particles i, j at time step t. Additionally, we assume that $E^M \subset E^C$, since a mesh edge connects nodes that are close to each other and hence should also satisfy Eqn. 1.

B. Inferring Visible Connectivity from a Partial Point Cloud

In the real world, we observe the cloth as a partial point cloud, and we need to infer the connectivity of particles for the visible portion of cloth. We first pre-process the point cloud by filtering it with a voxel grid filter: we overlay a 3d voxel grid over the observed point cloud and then take the centroid of the points inside each voxel to obtain a voxelized point cloud $P = \{x_i\}_{i=1,\ldots,N_p}$. This preprocessing step is done both in simulation training and in the real world evaluation.

Given the voxelized point cloud, the collision edges are then inferred by applying the criterion from Eqn. 1. However, inferring the mesh edges is less straightforward, since in the real world we cannot directly perceive the underlying cloth mesh connectivity. To overcome this challenge, we use a graph neural network to infer the mesh edges from the voxelized point cloud. Given the positions of the points in the voxelized point cloud P, we first construct a graph $\langle P, E^C\rangle$ with only the collision edges based on Eqn. 1. Based on our assumption that $E^M \subset E^C$, we train a classifier to estimate whether each collision edge $e \in E^C$ is also a mesh edge. Our classifier takes the form of a graph neural network (GNN) [1]. We term this classifier the edge GNN, and denote it as G_{edge}. It takes as input the graph $\langle P, E^C\rangle$, and outputs a binary label for each edge $e \in E^C$, indicating whether such a collision edge is also a mesh edge. We take the network architecture in previous work [10] (referred to as GNS) for the edge GNN G_{dyn}.

C. Modeling Visible Connectivity Dynamics with a GNN

In order to predict the effect of a robot's action on the cloth, we must model the cloth dynamics. We learn a dynamics model based on the voxelized point cloud and its inferred visible connectivity (Sec. II-B). Formally, given the cloth graph $G_t = \langle V, E\rangle$, a dynamics GNN G_{dyn} predicts the particle states in the next time step, which includes both the positions and the velocities. Here, E refers to inferred visible connectivity that includes both the predicted mesh edges E^M as well as the non-mesh collision edges. We also use the GNS architecture for our dynamics GNN G_{dyn}.

D. Planning with Pick-and-place Actions

We plan in a high-level, pick-and-place action space. For each action, denoted by two positions $a = \{x_{pick}, x_{place}\}$, the gripper grasps the cloth at x_{pick}, moves to x_{place}, and then drops the cloth. Our goal is to smooth a piece of cloth from a crumpled configuration. To compute the reward function r, we treat each node in the graph as a sphere with radius R and compute the covered area of these spheres when projected onto the ground plane. Given the current voxelized point cloud of a crumpled cloth P, we first estimate the mesh edges using the edge predictor $E^M = G_{edge}(\langle P, E^C\rangle)$. These edges are kept fixed throughout the rollout of a pick-and-place trajectory. We then sample K high-level pick-and-place actions. For each sampled high-level action, we roll out our dynamics model using that action for H low-level steps and obtain the sequence of predicted point positions.

E. Training in Simulation

The simulator we use for training is Nvidia Flex, a particle-based simulator with position-based dynamics [6], [8], wrapped in SoftGym [5]. In Flex, a cloth is modeled as a grid of particles, with spring connections between particles to model the bending and stretching constraints.

One challenge that we must address is that the points in the observed partial point cloud do not directly correspond to the underlying grid of particles in the cloth simulator. This presents a challenge for obtaining the ground-truth labels used for training the dynamics GNN and the edge GNN, including the position and velocity for each point in the observed point cloud and the mesh edges among them. To address this issue, we perform bipartite graph matching to connect each point in the voxelized point cloud to a simulated particle. After we get the mapping from the points to the simulator particles, the ground-truth acceleration of each point is simply assigned to be the acceleration of its mapped particle, which is used for training the dynamics GNN. For training the edge GNN, we need to obtain the ground-truth of which collision edges are also mesh edges. During simulation training, a collision edge is assumed to be a mesh edge if the mapped simulation particles of the edge's both end points are connected by a spring in the simulator.

Algorithm# of pick-and-place actions	5	10	20	50	100
(Ours)	$\mathbf{0.620 \pm 0.252}$	$\mathbf{0.738 \pm 0.273}$	$\mathbf{0.879 \pm 0.202}$	$\mathbf{0.956 \pm 0.196}$	-
VSF [2]	0.337 ± 0.141	0.585 ± 0.194	0.750 ± 0.180	0.932 ± 0.102	-
CFM [13]	0.028 ± 0.070	0.053 ± 0.082	0.072 ± 0.122	0.122 ± 0.149	0.132 ± 0.177
MVP [12]	0.330 ±0.296	0.341 ±0.262	0.337 ±0.296	0.421 ±0.304	0.379 ±0.268

TABLE I

NORMALIZED PERFORMANCE OF ALL METHODS IN SIMULATION, FOR VARYING NUMBERS OF ALLOWED PICK AND PLACE ACTIONS.

Material# of pick-and-place actions	5	10	20	Best
Cotton	0.504 ± 0.155	0.682 ± 0.253	0.797 ± 0.297	0.903 ± 0.153
Silk	0.412 ± 0.164	0.457 ± 0.236	0.456 ± 0.236	0.648 ± 0.169

TABLE II

NORMALIZED PERFORMANCE OF OUR METHOD IN THE REAL WORLD ON TWO CLOTHS OF DIFFERENT MATERIALS.

III. EXPERIMENT

A. Experimental Setup

Simulation Setup As mentioned, we use the Nvidia Flex simulator, wrapped in SoftGym [5], for training. For the simulation experiments, we use a nearly square cloth. For all methods, we randomly generate 20 initial cloth configurations for training and another 40 initial configurations for testing.

The performance metric we use for evaluation is the normalized increment of the covered area of the cloth in the top-down view. In brief, the normalized performance metric is 0 if no action is taken and 1 if the cloth is fully flattened. We compare our proposed method (Visible Connectivity Dynamics) with the following baselines which are state-of-the art methods for cloth smoothing: (1) VisuoSpatial Foresight (VSF) [2], which learns a visual dynamics model using RGBD data; (2) Contrastive forward model (CFM) [13], which learns a latent dynamics model via contrastive learning; (3) Maximal Value under Placing (MVP) [12], which uses model-free reinforcement learning with a specially designed action space.

We trained each of the baselines for at least as many pick-and-place actions as they were trained in their original papers. For training our method , we collect 1500 trajectories, each consisting of 1 pick-and-place action. Each method is evaluated on 40 test configurations and we report the mean and standard deviation of the results. For planning with VCD, we sample 500 pick-and-place actions, where the pick point is first uniformly sampled from a bounding box of the cloth and then projected to be on the cloth mask.

Real World Setup After training in simulation, we evaluate our trained dynamics model in the real world with a Franka Emika Panda robot arm with a standard panda gripper. We evaluate on two pieces of cloth: One is made of soft polyester satin (Silk). The other is made of cotton (Cotton). We use the same covered area as our reward function and the evaluation metric for the real world experiments. For each cloth, we evaluate 10 trajectories each with a maximum of 20 pick-and-place actions. For each trajectory, the robot stops if the normalized performance is higher than 0.95 or if the predicted rewards of all the sampled actions are smaller than the current reward. For each pick-and-place action, we sample 100 pick-and-place actions to be evaluated by our model.

B. Simulation Results

For each method, we report the achieved normalized performance after different numbers of actions (counting high-level pick-and-place actions). The trajectory ends if the normalized performance is above 0.95 or if the maximal number of actions is achieved. For CFM and MVP, we additionally test them with 100 pick-and-place actions to compensate for the typically small action size of these methods. The results are summarized in Table I. Under any given number of pick-and-place actions, greatly outperforms all of the baselines. Due to their use of RGB data, the baselines do not incorporate the inductive bias of the cloth structure into their dynamics model or policy. In contrast, our method learns a particle-based dynamics model that incorporates this inductive bias and leads to better performance.

C. Real-world Results

We also evaluate our method for smoothing in the real world. The point cloud representation allows VCD to easily transfer to the real world. The quantitative results of our method can be found in Table II and *visualizations of smoothing sequences can be found on our project website*. We note that, despite the drastic differences in visual appearances, as well as the different dynamics of the cotton and silk cloths, our model is able to smooth them. We also evaluate our model performance if we were able to terminate optimally in hindsight and choose the frame with the highest performance in each trajectory; the result is shown in the last column of Table II.

IV. CONCLUSION

In this paper, we propose the visible connectivity dynamics (VCD) model, under a framework that infers a visibility connectivity graph from the partial point cloud, learns a particle-based dynamics model over the graph, and apply it to plan action sequences for the task of cloth smoothing. has the advantage of posing strong inductive bias that fits the underlying cloth physics, being invariant to visual features, and being interpretable. We show that greatly outperforms previous state-of-the-art methods for cloth smoothing, and achieves zero-shot sim-to-real transfer when deployed on a Franka arm for smoothing different types of cloth from crumpled configurations.

References

[1] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[2] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. VisuoSpatial Foresight for Multi-Step, Multi-Task Fabric Manipulation. In *Robotics: Science and Systems (RSS)*, 2020.

[3] Pablo Jiménez and Carme Torras. Perception of cloth in assistive robotic manipulation tasks. pages 1–23. Springer, 2020.

[4] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.

[5] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, 2020.

[6] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[7] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. *Conference on Robot Learning (CoRL)*, 2018.

[8] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[9] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

[10] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.

[11] 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 From an Algorithmic Supervisor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[12] Wilson Wu, Yilin adn Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations. *Robotics Science and Systems (RSS)*, 2020.

[13] Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. 2020.