EDO-Net: Learning Elastic Properties of Deformable Objects from Graph Dynamics

Alberta Longhini*[1], Marco Moletta*[1], Alfredo Reichlin[1], Michael C. Welle[1], David Held[2], Zackory Erickson[2], and Danica Kragic[1]

***Abstract*— We study the problem of learning graph dynamics of deformable objects that generalizes to unknown physical properties. Our key insight is to leverage a latent representation of elastic physical properties of cloth-like deformable objects that can be extracted, for example, from a pulling interaction. In this paper we propose EDO-Net (*Elastic Deformable Object - Net*), a model of graph dynamics trained on a large variety of samples with different elastic properties that does not rely on ground-truth labels of the properties. EDO-Net jointly learns an *adaptation* module, and a *forward-dynamics* module. The former is responsible for extracting a latent representation of the physical properties of the object, while the latter leverages the latent representation to predict future states of cloth-like objects represented as graphs. We evaluate EDO-Net both in simulation and real world, assessing its capabilities of: 1) generalizing to *unknown* physical properties, 2) transferring the learned representation to new downstream tasks.**

I. Introduction

Manipulation of deformable objects is a fundamental skill toward folding clothes, assistive dressing, wrapping or packaging [1], [2], [3]. In these scenarios, deformables are subject to variations of physical properties such as mass, friction, density or elasticity, that influence the dynamics of the manipulation [4], [5]. Despite the progresses made in robotic manipulation of deformable objects, modelling, learning, and transferring skills remain open challenges [6]. The complexity of the problem arises from the following two factors characterizing deformable objects [7]: *i)* their state is high dimensional and difficult to represent canonically; *ii)* their interaction dynamics are often non-linear and influenced by physical properties usually not known a priori.

To address *i)*, analytical models often employ particle-based representations such as graphs extracted from point clouds [8], [9]. These representations, combined with current advancements in Graph Neural Network (GNN), have shown promising results in learning complex physical systems [10], [11], [12]. However, current methods assume that the physical properties are known a priori, which may not hold when robots operate in human environments. Thus, addressing problem *ii)* is of fundamental importance. The field of *intuitive physics* [13] tackles this challenge by learning predictive models which distill knowledge about the physical properties from past experience and interaction observations [14]. This line of research has so far focused mostly on rigid objects, but recent advances of data-driven techniques for deformable objects manipulation suggest that interactions such as whipping or pulling may be relevant to learn an intuitive physics model of these objects [7], [15].

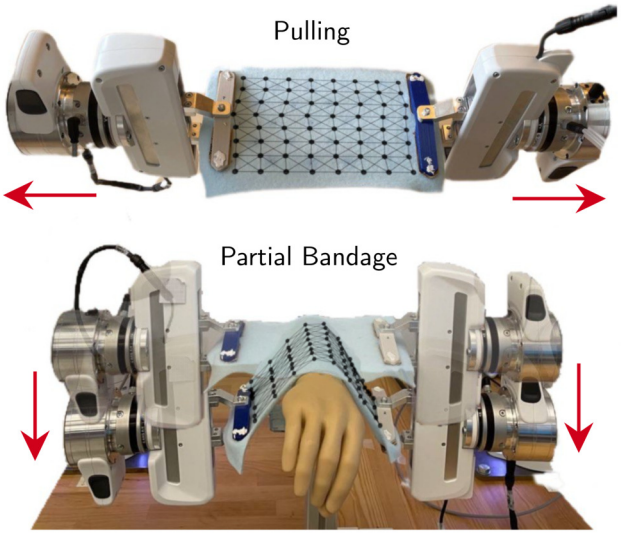


Fig. 1: A pulling interaction is leveraged by EDO-Net to explore the elastic properties of the object, which improves the performance in subsequent tasks such as partial bandage.

In this paper, we study the problem of learning graph dynamics of deformable objects that generalize to objects with unknown physical properties. In particular, we focus on elastic properties of cloth-like deformable objects, such as textiles, that we explore through a pulling interaction (Fig. 1). We propose EDO-Net (*Elastic Deformable Object - Net*), a model trained on a large variety of samples with different elastic properties, without relying on ground-truth labels of these properties. EDO-Net jointly learns an *adaptation* module, responsible for extracting a latent representation of the physical properties of the object, and a *forward-dynamics* module, that leverages the latent representation to predict future states, represented as graphs.

We evaluate our approach both in simulation and in the real world, showing how EDO-Net accurately predicts the future states of a deformable object. We also validate the quality of the learned representation by retrieving the ground truth physical properties from the simulation environment using a weak learner. In summary, our contributions are:

- EDO-Net, a model to learn graph dynamics of cloth-

*Contributed equally (listed in alphabetical order)

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

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

like deformable objects and a latent representation of their physical properties without explicit supervision;
- a procedure to train EDO-Net on a large variety of samples with different elastic properties, enabling generalization to objects with *unknown* physical properties;
- extensive evaluations, both in simulation and in the real world, of the quality of the latent representation and of the dynamics prediction.

II. Related Work

We discuss the related work from the perspective of learning physical properties, representations and dynamics of deformable objects, as well as robotic tasks that could benefit from our proposed approach.

Learning physical properties: a common approach to extract physical properties like mass, moment of inertia or friction coefficients is to use different exploratory actions such as pushing, tilting or shaking [5], [16], [4]. In [5] mass and friction of rigid objects are learned through tactile exploration. Similarly, [17], [4] use a multi-step framework to encode physical properties of rigid objects from pushing tasks using dense pixels representations. Related to deformable objects, in [18] the authors propose how to predict properties of cloth-like objects to perform real2sim by learning to align real world and simulated behaviors through a differentiable simulator. In contrast to [18], we want to learn a representation of physical properties without relying on simulated behaviors and elastic parameters.

Representations and dynamics of deformable objects: regarding the challenge of finding canonical representations of deformable objects, an approach is to encode their high dimensional observations in structured latent spaces where to perform planning and control. Few examples are [19], [20], in which image observations are mapped in a latent space represented as a graph using constrastive learning, allowing to perform visual action planning to solve a folding task. A complementary way to leverage graphs is to use them to represent the state of the cloth and to learn its dynamics through GNNs [10], [21]. A particular line of research on learning graph dynamics of deformable objects addressed the challenge of partial observations [9], [22]. Other work instead looked into physics priors provided by differentiable simulators to better capture the complex dynamics models of deformable objects [23]. None of these works, however, focus on learning graph dynamics across a wide range of physical properties of deformable objects without relying on ground truth labels.

Tasks with cloth-like deformable objects: a substantial part of the work in manipulation of cloth-like deformable objects focuses on solving various robotic tasks including cloth folding [24], [1], [25], cloth smoothing [26], [9], [27], as well as healthcare applications like assisted dressing [2], [28], [29] and bedding manipulation [30], [31]. In these tasks, the different elastic properties of clothes influence the manipulation strategy that the robot has to execute. However, none of the aforementioned methods account for these variations, meaning that they could benefit from EDO-Net latent representation to adapt the manipulation strategy to different elastic properties and improve their generalization. We plan to explore this direction in future work.

III. Problem Formulation

In our formulation, we refer to the object's elastic properties as $\mathcal{T}_i \sim \mathcal{T}$, where $\mathcal{T}$ is the distribution of all possible physical properties. We explore $\mathcal{T}_i$ by collecting a sequence of observations O^i through an Exploratory Action (EA) [4], [5]. An *adaptation module* is responsible for extracting a latent representation z_i of the physical properties $\mathcal{T}_i$ from the observations O^i, which can be subsequently leveraged by a *forward dynamics module* to generalize its predictions across different $\mathcal{T}_i \sim \mathcal{T}$. We define the state of a deformable object with physical properties $\mathcal{T}_i$ as a graph $G^i = (V^i, E^i)$ with nodes $v \in V^i$ and edges $e \in E^i$. The features of the node v describe the 3D Cartesian position of the nodes, while the features of the edge e characterize the interaction properties among nodes. Given these, the aim of EDO-Net is to learn a graph dynamics model of cloth-like deformable objects g_θ conditioned on a latent representation z_i of the underlying physical properties $\mathcal{T}_i$ and the robot control action a_t:

$$\delta \hat{G}^i_t = g_\theta(G^i_t, a_t, z_i). \tag{1}$$

The latent representation z_i can be obtained through a learned function f_ϕ that takes as input sequence of observations O^i and an initialization z_0 of the representation:

$$z_i = f_\phi(O^i, z_0), \tag{2}$$

where the initialization z_0 is learned together with the model's parameters θ and ϕ. In what follows, we will describe in detail the method to implement and train the graph dynamics g_θ and adaptation f_ϕ functions, respectively.

IV. Method

An overview of the proposed EDO-Net is shown in Fig. 2. In particular, for each deformable object with unknown physical properties $\mathcal{T}_i$, the robot has to adapt the initialization z_0 by using a sequence of exploratory observations O^i. From O^i, the adaptation module f_ϕ first extracts a latent representation z_i of the physical properties $\mathcal{T}_i$. The extracted representation z_i is subsequently used in the forward dynamics module g_θ to obtain accurate predictions of the future states of $\mathcal{T}_i$ conditioned on different interactions a_t.

We focus on the scenario where the physical properties $\mathcal{T}_i$ are not directly observable from the initial state of the object. We assume that the state G^i_t of the deformable object with physical properties $\mathcal{T}_i$ is directly observable, which in real-world applications can be extracted from point clouds observations [32], [9]. We plan to relax this assumption in future work, for example by relying on approaches that tackle the challenge of partial observability using GNNs [9], [22].

A. Exploratory Action and Adaptation

To collect information about the physical properties $\mathcal{T}_i$, the robot needs to observe the response of the object during a dynamic interaction. To this end, we evaluate a *pulling* interaction shown in Fig. 3, where a two-arm robotic manipulator

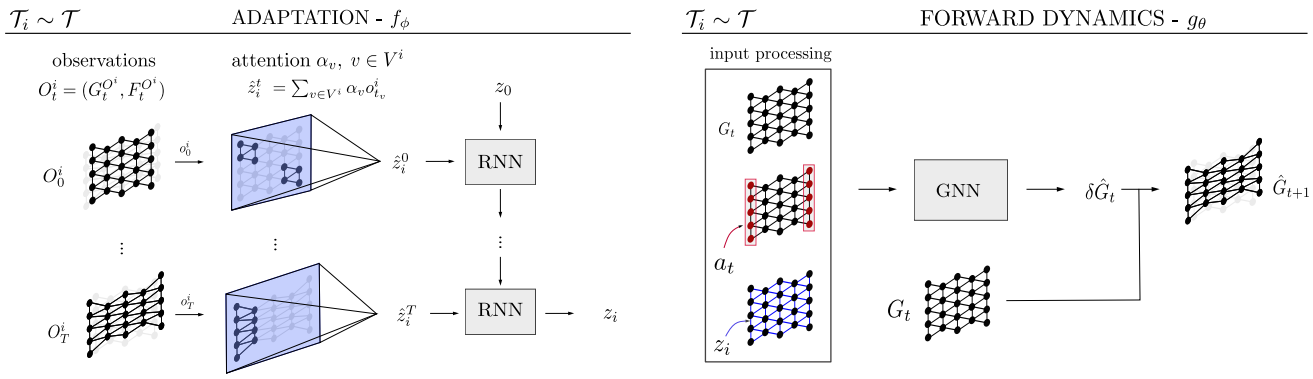


Fig. 2: Scheme of the overall model. Given a deformable object $\mathcal{T}_i$ with *unknown* physical properties, the adaptation module f_ϕ updates the initialization z_0 of the latent representation of the physical properties $\mathcal{T}_i$ from sequences of observations $O_t^i|_{t=1,\dots,T}$ processed by an attention layer and a RNN. In a second phase, the forward dynamics module g_θ, implemented as a GNN, uses z_i obtained from the adaptation module to predict future states $\hat{G}_t$ of the deformable object.

grasping a deformable object from its edges exerts tension stress on the object by pulling its edges along opposite directions, similarly to what is done in [15]. During the pulling Exploratory Action (EA), we record a set of T observations $O^i = O_t^i\ |_{t=1,\dots,T}$ where each $O_t^i = (G_t^{O^i}, F_t^{O^i})$ consists of the object state $G_t^{O^i}$ and the force $F_t^{O^i}$ recorded from the robot sensors at time t. The information contained in O^i about the physical properties $\mathcal{T}_i$ is subsequently input to the learned function f_ϕ to update the initialization z_0. The implementation of the adaptation function f_ϕ is the following: for each observation O_t^i, we encode $(G_t^{O^i}, F_t^{O^i})$ into a latent embedding o_t^i through a Multi-Layer Perceptron (MLP). We subsequently obtain an estimate $\hat{z}_i^t \in \mathbb{R}^p$ of z_i from o_t^i by learning a node's aggregation function through an attention layer, which aggregates the nodes as:

$$\hat{z}_i^t = \sum_{v \in V^i} \alpha_n o_{t_v}^i, \tag{3}$$

where α_n is the attention weight of the node n. For details about the implementation of the attention mechanism, we refer the reader to [33]. The set of $\hat{z}_i^t\ |_{t=1,..,T}$ is used to obtain the latent representation $z_i \in \mathbb{R}^p$ by recursively updating the initial belief z_0 through a Recurrent Neural Network (RNN), yielding the following update rule:

$$z_i = \text{RNN}(\hat{z}_i^t\ |_{t=1,..,T}, z_0). \tag{4}$$

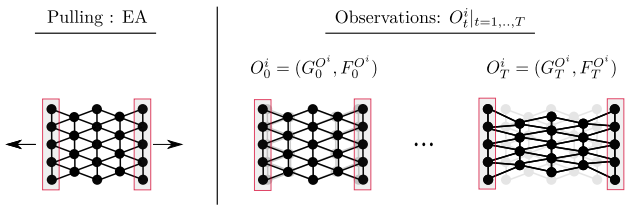


Fig. 3: Pulling Exploratory Actions to observe graphs and forces.

B. Forward Dynamics Module

We model the forward graph dynamics g_θ with a GNN conditioned on the latent representation z_i of the physical properties $\mathcal{T}_i$. We trained g_θ to predict state differences δG_t^i receiving as input the control action of the robotic manipulator a_t, and the initial state of the object G_t^i. We integrate z_i as features of the edges of the input graph as shown in the *input processing* block in Fig. 2. We use MLPs to encode nodes and edges before propagating the information among the nodes using a standard M-steps message passing GNN with the following update rule at step m [34]:

$$h_v^m = \Phi\left(\sum_{s \in \mathcal{N}_v^1 \cup \mathcal{N}_v^2} \Psi\left(h_v^{m-1}, h_s^{m-1}, z_i\right)\right) \quad \forall v \in V^i, \tag{5}$$

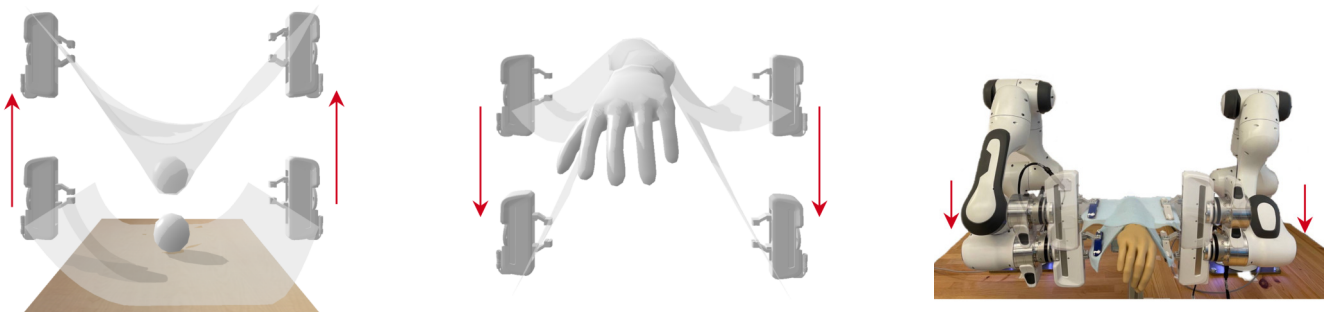
where Ψ and Φ are the learned message and update functions, and $\mathcal{N}_v^1$ and $\mathcal{N}_v^2$ are the sets of 1st and 2nd order neighbors of the node $v \in V^i$. To decrease the number of steps needed to propagate the information along the graph, we parallelize the computation of 1st and 2nd order neighbors as suggested in prior work [35]. Finally, the M-th hidden nodes are passed through a decoder MLP to the prediction of the graph displacement $\delta\hat{G}_t^i$.

C. Training Loss

The overall model can be learned using a dataset of exploratory observations $\mathcal{D}^O = \{\mathcal{D}^{O^i}\}_{\mathcal{T}_i \sim \mathcal{T}}$ and a dataset of interactions $\mathcal{D} = \{\mathcal{D}_i\}_{\mathcal{T}_i \sim \mathcal{T}}$. The parameters ϕ, θ and the initialization z_0 can be optimized using a loss on the prediction of the state difference $\delta\hat{G}_t^i$ obtained from g_θ for each training sample with physical properties $\mathcal{T}_i \sim \mathcal{T}$. The loss function $\mathcal{L}$ can be defined as follows:

$$\mathcal{L} = \mathbb{E}_{\substack{\mathcal{T}_i \sim \mathcal{T} \\ G_t^i, a_t, \delta G_t^i \sim \mathcal{D}_i}} \left[d(\delta G_t^i, g_\theta(G_t^i, a_t, z_i))\right], \tag{6}$$

where $z_i = f_\phi(O^i, z_0)$ with $O^i \sim \mathcal{D}^{O^i}$, and d is the Mean-Squared Error (MSE) between the ground truth state displacement of the deformable object and the model's prediction. Equation 6 optimizes the parameters θ to learn a forward dynamic model conditioned on different representations z_i of physical properties $\mathcal{T}_i$, implicitly driving the parameters ϕ to learn to encode z_i of different samples without supervision from ground truth labels of the physical parameters. Moreover, training across multiple $\mathcal{T}_i \sim \mathcal{T}$



(a) *Lifting* - Simulation (b) *Partial Bandage* - Simulation (c) *Partial Bandage* - Real world

Fig. 4: The environments employed to evaluate EDO-Net.

enforces the model to learn how to generalize to deformable objects with unknown physical properties.

V. Environment and Implementation Details

In this section, we introduce the simulated and real-world environments designed to evaluate the proposed method, along with its implementation and training details.

A. Environments Setup

For the simulation experiments we use Pybullet [36], [37], where we create the two environments displayed in Fig. 4a and Fig. 4b. Both environments include two free-floating Franka-Emika Panda end-effectors equipped with Force/Torque sensors. In the first environment, called *Lifting*, the robot lifts a sphere located on a cloth-like deformable object from an initial resting position on the table to a predefined height, by applying a displacement control action $a_t \in [0, D_{max}]$. In the second environment, called *Partial Bandage*, the robot holds and pulls the cloth downward over a human arm, applying a force control action $a_t \in [0, F_{max}]$. For both environments, we uniformly discretize the action intervals into 30 instances with fixed step size, while the pulling EA is implemented as shown in Fig. 3 We extract the graph G_t^i and force F_t^i observations at each time step of the simulation and we downsample it to a grid of 8×8 nodes. Furthermore, we smooth the force profiles using a Savitzky–Golay filter [38] with a window size of 21 and a third-grade polynomial. We generate a large variety of physical properties of the cloth by varying both the *stiffness* and the *bending* parameters of the simulator. We empirically selected the elasticity parameters in the range $[10, 45]$ with a step size of 3.0, and the bending parameters in the range $[0.01, 5.01]$ with a step size of 0.5, for a total of 143 unique elastic deformable objects.

We replicate the *Partial Bandage* environment in the real world as visible in Fig. 4c. We collect the pulling EA and the interaction trajectories on 40 textile samples with different elastic properties where the dataset characteristics correspond to the one in prior work [39]. The real-world dataset and the graph extraction procedure are shown in Fig. 5.

B. Network Architecture and Training details

We used an MLP with one hidden layer of size 32 to implement the node encoder of the adaptation module,

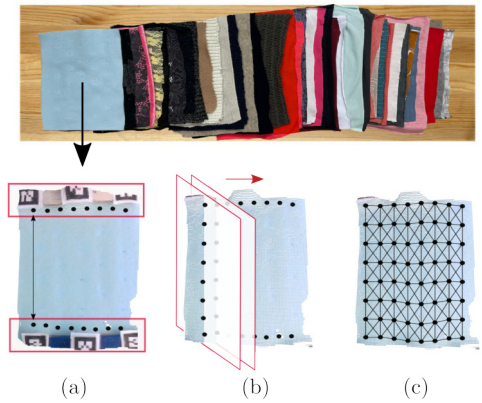


Fig. 5: Textile dataset samples (top) and procedure to extract graphs from point cloud (bottom). First we represent the grippers as 8 equidistant nodes (a). We then slice the point cloud with a plane passing through 2 corresponding nodes of the grippers, obtaining 6 additional equidistant nodes (b). We obtain the final graph by connecting the neighbors of each node as shown in (c).

while the attention layer is implemented as a learned linear projection from the encoded node $o_v \in \mathbb{R}^{32}$ to the attention value $\alpha_v \in \mathbb{R}$. The RNN architecture has one hidden state which starts with initialization $z_0 \in \mathbb{R}^p$ and outputs $z_i \in \mathbb{R}^p$ as its last hidden state, where $p = 32$. Hyperparameters were chosen empirically based on the highest overall performance across the evaluations. Regarding the forward model, we propagate the information for $M = 4$ steps, while we implement the node's encoder, the final linear projection, and the message and the aggregation functions as MLPs with one hidden layer of size 32. We use ReLU as non-linearity for all the modules except the RNN, where we use tanh as the activation function. The models are trained on the datasets normalized to zero mean and unit variance for 5000 epochs and batch size equal to 8. We used Adam [40] with a learning rate of 10^{-3} and weight decay equal to 10^{-5}. The simulation dataset consists of 30 datapoints for each of the 143 unique samples for *Lifting*, *Partial Bandage* and the pulling EA. We split the 143 samples into a train (80%), validation (10%), and test (10%) samples. In the real world, instead, the dataset consists of 30 datapoints for each of the 40 unique samples for the *Partial Bandage* and the pulling EA. We split the 40

samples into a training (80%) and test samples (20%).

VI. Experiments

In this section we evaluate the performance of EDO-Net, regarding its adaptation module f_ϕ, forward module g_θ and its generalisation capabilities. To this aim, we:

1) examine in simulation how accurately we can decode physical properties from the latent representation z_i by learning to predict ground-truth parameters from z_i;
2) quantitatively evaluate in simulation whether the latent representation z_i transfers between environments (from *Partial Bandage* to *Lifting*) or to different downstream tasks, such as learning an inverse model to predict the control action between two states;
3) analyze both in simulation and real-world environments the generalization capabilities of *EDO-Net*, testing the model over a set of deformable objects with *unseen* elastic physical properties $\mathcal{T}_i \sim \mathcal{T}$.

We compare EDO-Net with a *Non-Conditioned* (NC) baseline model, which trains the forward model g_θ without conditioning on z_i. We also consider an ablation of EDO-Net trained on a single exploratory observation, rather than a sequence of interactions, which we denote by EDO1. Moreover, we include three oracle models in simulation to set an upper-bound performance for the tasks: two *Oracle* models conditioned on the ground-truth simulation parameters, respectively (OI) and (OF) for inverse and forward dynamics models, and an *Oracle Supervised* forward model (OS), trained with an additional supervised loss term over z_i, to directly predict the ground-truth simulation parameters during the training procedure.

A. Decoding Physical Properties

The aim of this section is two-fold: 1) to evaluate whether it is possible to decode the ground truth physical properties $\mathcal{T}_i$ of the deformable object from the latent representation z_i, and 2) to analyse the influence of the length T of the sequence of exploratory observations used to extract z_i. In particular, we train an MLP with 3 hidden layers of size 64 and ReLU non-linearities which takes as input the learned representation z_i to predict the bending and elastic parameters of the simulator. We evaluate the predicted physical parameters in both the *Partial Bandage* and *Lifting* environments by evaluating the MSE between the ground truth normalized physical parameters and the model predictions. We distinguish between *seen* and *unseen* physical properties depending on whether $\mathcal{T}_i \sim \mathcal{T}$ was used to train the model or not. Fig. 6 shows the prediction results of the physical parameters from the learned z_i while varying the number T of exploratory observation O^i used to extract the representation. It can be noticed how increasing the number of observations improves the quality of the latent representation of the physical properties learned by f_ϕ, highlighting the relevance of using a sequence of dynamic interactions to encode physical properties. Moreover, the performance of EDO-Net is close to OS, suggesting that the loss in Eq. 6 implicitly trains the model to learn a latent representation of the physical properties without explicit supervision from the ground truth labels.

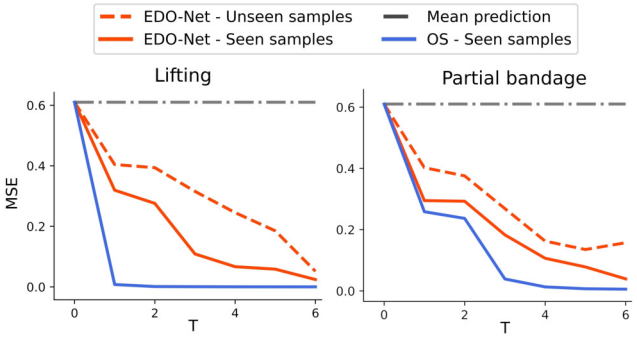


Fig. 6: MSE (in normalized units) of the prediction of the simulation parameters varying the length T of the sequence of exploratory observations.

B. Evaluation of the Adaptation Module f_ϕ

In this section, we further evaluate z_i by answering the following questions: can we 1) transfer z_i to efficiently learn forward models of different environments, and 2) transfer z_i to efficiently learn new downstream tasks, such as inverse dynamics prediction? To address these questions we pretrain *EDO-Net* on the *Partial Bandage* environment and designed the following scenarios:

1) *Bandage2Lifting*: we fine-tune the forward dynamics model g'_θ on the *Lifting* environment, while keeping the weights of f_ϕ fixed;
2) *Inverse Dynamics*: we train an inverse dynamics model g''_θ conditioned on z_i in the *Partial Bandage* environment to predict the control action a_i^t between the initial state of the deformable object G_0^i and the next state G_t^i while keeping the weights of f_ϕ fixed.

In the *Bandage2Lifting* scenario, we evaluate the performance of the fine-tuned model by computing the MSE between the state-differences $\delta\hat{G}_i^{t+1}$ (in normalized units) and the ground-truth δG_i^{t+1} of the *Lifting* environment for deformable objects with physical properties $\mathcal{T}_i \sim \mathcal{T}$ unseen during training. For the *Inverse Dynamics* scenario, we implement g''_θ by initially encoding graph nodes and physical properties z_i with an MLP, subsequently projecting their concatenation to a latent space of the same dimensionality of the action. We finally aggregate and average the projections to obtain the predicted action. The performance of the inverse model is evaluated by computing the MSE between the normalized versions of the predicted action $\hat{a}_i^t$ and the ground-truth action a_i^t. For both scenarios, we compare the model performance with the NC baseline, and EDO1. For the $Bandage2Lifting$ scenario we set as reference performances EDO-Net model trained directly on the *Lifting* environment (EDO(L)-Net) and OF, while for the *Inverse Dynamics* scenario we set as reference performances OS and OI. The results are presented in Table I. We observe in the *Bandage2Lifting* scenario that all the evaluated models outperform the NC baseline, indicating that the representation z_i

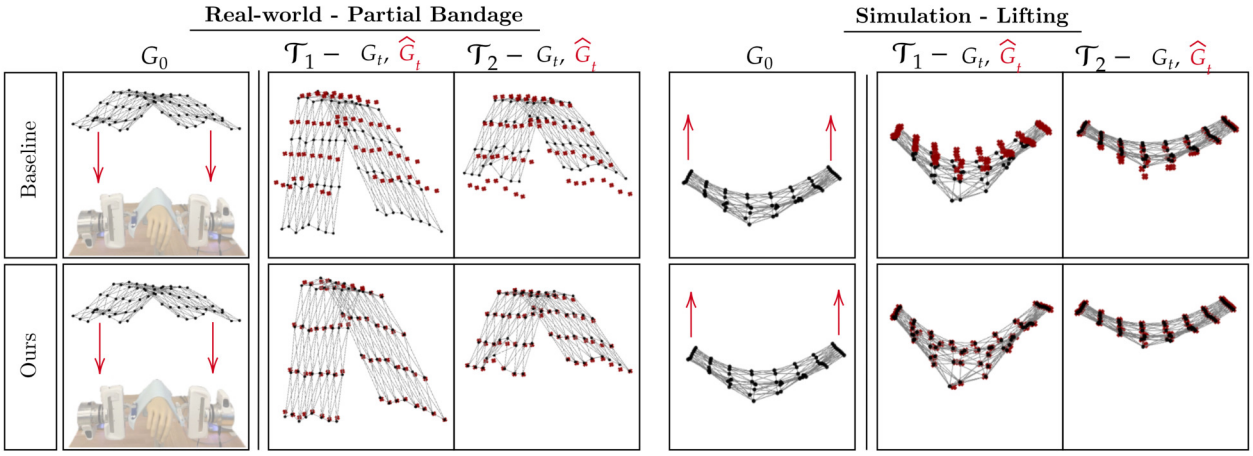


Fig. 7: Qualitative evaluation of the graph dynamics predictions $\hat{G}_t$ obtained by EDO-Net and the NC baseline starting from the inital graph G_0. For each environment we select two elastic samples with different physical properties $\mathcal{T}_1$, $\mathcal{T}_2$.

learned in the *Partial Bandage* environment is informative for the *Lifting* one. For the *Inverse Dynamics* scenario, instead, we observe that EDO-Net outperforms all the other baseline methods. These results suggest that our latent representation transfers to different environments and downstream tasks.

C. Generalization to Unseen Physical Properties

In this section, we evaluate the generalisation capabilities of EDO-Net. We consider both simulation and real-world environments, and we perform quantitative and qualitative tests of the model over a set of deformable objects with elastic physical properties $\mathcal{T}_i \sim \mathcal{T}$ unseen during training. We evaluated the model performance by computing the MSE of the model's predictions with respect to the ground truth for each testing sample. We compare the performance of our model with respect to the NC and the EDO1 baselines. In simulation we also compare to OS and OF. In Table II we report the mean and standard deviation of the MSEs evaluated across all the testing samples with physical properties $\mathcal{T}_i \sim \mathcal{T}$. In all scenarios, *EDO-Net* outperforms the baseline models both in terms of the average error and the standard deviation across samples with different elastic properties. The high standard deviation of the NC model is due to the large difference between the average elastic behavior and the extreme (rigid/elastic) ones. Moreover, *EDO-Net* achieves comparable performances with respect to OF. Qualitative visualizations of the relevance of our proposed method for both simulation and real-world environment are shown in Fig. 7. We can observe how the NC baseline does not distinguish among samples with different physical properties ($\mathcal{T}_1$ and $\mathcal{T}_2$), hindering its capability of predicting the outcome of the robot control actions. On the other hand, *EDO-Net* successfully leverages the latent representations (z_1 and z_2) provided by the adaptation module f_ϕ.

TABLE I: Results of *Bandage2Lifting* and *Inverse Dynamics* scenarios (in normalized units), with T=5. Lower is better.

(a) *Bandage2Lifting*

Model	MSE ($\times 10^{-3}$)
NC	6.668 ± 13.02
EDO1	0.233 ± 0.313
EDO-Net	0.270 ± 0.491
EDO(L)-Net	0.102 ± 0.068
OF	0.081 ± 0.040

(b) *Inverse Dynamics*

Model	MSE
NC	9.705 ± 11.11
EDO1	0.212 ± 0.331
EDO-Net	0.058 ± 0.096
OS	0.051 ± 0.041
OI	0.035 ± 0.057

TABLE II: Generalisation results of EDO-Net and the baselines in the simulated and real-world environments (in normalized units), with T=5. Lower is better.

Model	MSE ($\times 10^{-3}$) *Partial Bandage* simulation	MSE ($\times 10^{-3}$) *Lifting* simulation	MSE ($\times 10^{-3}$) *Partial Bandage* real world
NC	29.60 ± 65.29	6.585 ± 12.79	59.37 ± 57.50
EDO1	0.260 ± 0.197	0.171 ± 0.106	3.046 ± 1.603
EDO-Net	0.151 ± 0.125	0.102 ± 0.068	1.481 ± 0.500
OS	0.992 ± 1.480	0.321 ± 0.699	–
OF	0.122 ± 0.194	0.081 ± 0.040	–

VII. Conclusions

We presented EDO-Net, a data-driven model that learns a latent representation of physical properties of cloth-like deformable objects to generalize graph-dynamic predictions to objects with unseen physical properties. We assessed in simulation that it is possible to decode ground truth parameters from the learned representation, as well as to transfer the representation across different environments. Furthermore, we assessed both in simulation and real world how conditioning the forward dynamics model to the latent representation z_i helps in generalizing over unseen physical properties. The latent representation learned from EDO-Net is relevant for robotic tasks to generalize manipulation skills to a wide variety of cloth-like objects. Moreover, leveraging this framework with multiple exploratory actions could enable learning physical properties beyond elasticity and generalizing to different manipulation tasks.

VIII. Acknowledgements

This work has been supported by the European Research Council (ERC-BIRD), Swedish Research Council and Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1–8.

[2] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, "Deep haptic model predictive control for robot-assisted dressing," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4437–4444.

[3] 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)*. IEEE, 2021, pp. 4568–4575.

[4] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "Densephysnet: Learning dense physical object representations via multi-step dynamic interactions," *arXiv preprint arXiv:1906.03853*, 2019.

[5] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5633–5640.

[6] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, X. Li, J. Pan, *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *arXiv preprint arXiv:2105.01767*, 2021.

[7] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *arXiv preprint arXiv:2203.00663*, 2022.

[8] Z. Weng, F. Paus, A. Varava, H. Yin, T. Asfour, and D. Kragic, "Graph-based task-specific prediction models for interactions between deformable and rigid objects," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5741–5748.

[9] X. Lin, Y. Wang, Z. Huang, and D. Held, "Learning visible connectivity dynamics for cloth smoothing," in *Conference on Robot Learning*. PMLR, 2022, pp. 256–266.

[10] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8459–8468.

[11] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning mesh-based simulation with graph networks," *arXiv preprint arXiv:2010.03409*, 2020.

[12] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4470–4479.

[13] M. McCloskey, "Intuitive physics," *Scientific american*, vol. 248, no. 4, pp. 122–131, 1983.

[14] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," *Advances in neural information processing systems*, vol. 28, 2015.

[15] A. Longhini, M. C. Welle, I. Mitsioni, and D. Kragic, "Textile taxonomy and classification using pulling and twisting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7564–7571.

[16] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *Advances in neural information processing systems*, vol. 29, 2016.

[17] J. K. Li, W. S. Lee, and D. Hsu, "Push-net: Deep planar pushing for objects with unknown physical properties." in *Robotics: Science and Systems*, vol. 14, 2018, pp. 1–9.

[18] P. Sundaresan, R. Antonova, and J. Bohg, "Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects," *arXiv preprint arXiv:2204.03139*, 2022.

[19] M. Lippi, P. Poklukar, M. C. Welle, A. Varava, H. Yin, A. Marino, and D. Kragic, "Latent space roadmap for visual action planning of deformable and rigid object manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5619–5626.

[20] ——, "Enabling visual action planning for object manipulation through latent space roadmap," *IEEE Transactions on Robotics*, 2022.

[21] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

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

[23] S. Chen, Y. Liu, S. W. Yao, J. Li, T. Fan, and J. Pan, "Diffsrl: Learning dynamical state representation for deformable object manipulation with differentiable simulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9533–9540, 2022.

[24] 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*. PMLR, 2022, pp. 192–202.

[25] G. Salhotra, I.-C. A. Liu, M. Dominguez-Kuhne, and G. S. Sukhatme, "Learning deformable object manipulation from expert demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8775–8782, 2022.

[26] 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, 2022.

[27] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*. PMLR, 2022, pp. 24–33.

[28] A. Kapusta, Z. Erickson, H. M. Clever, W. Yu, C. K. Liu, G. Turk, and C. C. Kemp, "Personalized collaborative plans for robot-assisted dressing via optimization and simulation," *Autonomous Robots*, vol. 43, pp. 2183–2207, 2019.

[29] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, "Multidimensional capacitive sensing for robot-assisted dressing and bathing," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2019, pp. 224–231.

[30] K. Puthuveetil, C. C. Kemp, and Z. Erickson, "Bodies uncovered: Learning to manipulate real blankets around people via physics simulations," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1984–1991, 2022.

[31] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making," in *Robotics Research: The 19th International Symposium ISRR*. Springer, 2022, pp. 275–290.

[32] X. Ma, D. Hsu, and W. S. Lee, "Learning latent graph dynamics for deformable object manipulation," *arXiv preprint arXiv:2104.12149*, 2021.

[33] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[34] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1205–1211.

[35] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *international conference on machine learning*. PMLR, 2019, pp. 21–29.

[36] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[37] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 169–10 176.

[38] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[39] A. Longhini, M. Moletta, A. Reichlin, M. C. Welle, A. Kravberg, Y. Wang, D. Held, Z. Erickson, and D. Kragic, "Elastic context: Encoding elasticity for data-driven models of textiles," 2022. [Online]. Available: https://arxiv.org/abs/2209.05428

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.