AdaptiGraph: Material-Adaptive Graph-Based Neural Dynamics for Robotic Manipulation

Kaifeng Zhang*, Baoyu Li*, Kris Hauser, Yunzhu Li
University of Illinois Urbana-Champaign

Contents

I. Related Work

A. Model Learning for Robotic Manipulation

Analytical physics-based models facilitate a wide span of robotic manipulation tasks [14, 31, 48]. However, building accurate physics models is often infeasible in the real world due to unobservable physics properties such as mass, friction, and stiffness, occluded surfaces of geometry, sensitivity to parameter estimates, and the high computational expense of simulating deformable objects. To mitigate these issues, recent approaches apply learning-based techniques to obtain dynamics models directly from sensory inputs [7, 30, 46, 15, 10, 2]. Graph-based representations and GNNs have been proven effective in modeling the complex behaviors of non-rigid objects due to their ability to capture spatial relational bias [3, 32, 20, 24, 34, 42]. Prior work has explored the application of graph-based dynamics models on a variety of material types, including rigid bodies [20, 16, 26], plasticine [36, 37], clothes [33, 24, 32, 27], fluids [19, 34], and granular matter [42]. However, nearly all of these approaches focus on a single type of material and fail to consider variation in physical properties, thus limiting their generalization and adaptation capabilities. In contrast, our method considers a wider range of materials and variations in physical properties in a single property-conditioned graph-based neural dynamics models, and this enables our approach to adaptively estimate the unknown physical properties of unseen objects through interaction.

B. Physical Property Estimation and Few-Shot Adaptation

Estimating and adapting to different physical properties of unseen objects is an inherent challenge in many robotic tasks. Previous works have attempted to extract physical properties including mass, friction coefficient, and moment of inertia of rigid objects by few-shot exploratory interactions [1, 18, 40, 35, 45, 8, 6]. However, these works are limited to rigid objects or require expensive sensors to acquire the physical property information. The commonsense reasoning ability of large pre-trained vision-language models can also be utilized to infer object physical properties from static visual observations [12, 41], but they are limited to giving rough estimations and do not involve actual physical interactions. For deformable objects, physical properties can be estimated by adjusting parameters in a physics-based simulation [22, 11, 38, 44, 47, 39] or learning a few-shot property estimation neural network based on simulated training data [21, 27]. However, these approaches typically require full-state information and a physics-based simulator closely aligned with the real world. In contrast, our method leverages a graph-based approximation of dynamics and performs property estimation of objects with diverse materials using a unified inverse optimization framework.

II. Methods

A. Material-Conditioned Graph-based Neural Dynamics Model

We proposed to instantiate the dynamics model with a graph neural network. Following prior work on graph-based neural dynamics (GBND) [19, 42, 36, 37], we define the environment state as a graph: $z_t \triangleq \mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where $\mathcal{V}_t$ is the vertex set representing object particles, and $\mathcal{E}_t$ denotes the edge set representing interacting particle pairs at time step t. Given the point cloud input, the object particle positions are determined by the farthest point sampling method [29], which ensures sufficient coverage of the object's geometry. We construct edges between particles based on a spatial distance threshold d. We also sample particles on the robot end-effector and construct relations between robot particles and object particles.

The main improvement of our model over previous works based on GBND is our material- and physical property-conditioning module. Suppose a vertex $v_{i,t} \in \mathcal{V}_t$ has material M_i and physical property variable ϕ_i. We incorporate this

*Denotes equal contribution.

material information into the vertex features along with the 3D position information $x_{i,t}$ and the vertex attribute $c^v_{i,t}$ which indicates whether the particle belongs to an object or the robot end-effector. Formally, $v_{i,t} = (x_{i,t}, c^v_{i,t}, \phi_i, M_i) \in \mathcal{V}_t$. The relation features between a pair of particles is denoted as $e_{k,t} = (w, u, c^e_{k,t}) \in \mathcal{E}_t$, where $1 \leq w, u \leq |\mathcal{V}_t|$ are the receiver particle index and the sender particle index of the k^{th} edge respectively. The edge attribute $c^e_{k,t}$ contains information such as whether the sender and receiver belong to the same or different objects.

The constructed vertex and edge features are first fed into the vertex encoder $f^{enc}_{\mathcal{V}}$ and the edge encoder $f^{enc}_{\mathcal{E}}$ respectively to get the latent vertex and edge embeddings $h_{v_i,t}$ and $h_{e_k,t}$:

$$h^0_{v_i,t} = f^{enc}_{\mathcal{V}}(v_{i,t}), \quad h^0_{e_k,t} = f^{enc}_{\mathcal{E}}(v_{w,t}, v_{u,t}). \tag{1}$$

Then, an edge propagation network $f^{prop}_{\mathcal{E}}$ and vertex propagation network $f^{prop}_{\mathcal{V}}$ performs iterative update of the vertex and edge embeddings to perform multi-step message passing. Specifically, for $l = 0, 1, 2, \cdots, L-1$, a single message passing step is as follows:

$$h^{l+1}_{e_k,t} = f^{prop}_{\mathcal{E}}(h^l_{w,t}, h^l_{u,t}), \tag{2}$$

$$h^{l+1}_{v_i,t} = f^{prop}_{\mathcal{V}}(h^l_{v_i,t}, \sum_{j \in \mathcal{N}(v_{i,t})} h^{l+1}_{e_j,t}), \tag{3}$$

where $\mathcal{N}(v_{i,t})$ indicates the index set of edges in which vertex i is the receiver at time t, and L is the total number of message passing steps. Finally, one vertex decoder $f^{dec}_{\mathcal{V}}$ predicts the system's state at the next time step: $\hat{v}_{i,t+1} = f^{dec}_{\mathcal{V}}(h^L_{v_i,t})$.

To regulate the cumulative dynamics prediction error, we supervise the model's prediction results on K prediction steps and perform backpropagation through time to optimize model parameters. In practice, we choose $K = 3$ for all tasks for balancing efficiency and performance. We use MSE loss on predicted object particle positions as the loss function:

$$\mathcal{L} = \sum_t ||z_{t+1} - f(z_t, u_t; \phi, M)||^2_2. \tag{4}$$

To obtain training data at scale, we generate diverse object trajectories by randomizing robot actions and object configurations using physics-based simulators. Most importantly, we randomize the material configuration for each instance in the dataset. To achieve this, we identify the physics property ϕ and randomize the property over a wide range of feasible values.

B. Few-Shot Physical Property Adaptation

After learning the material-conditioned GBND model, we deploy the model to objects with unknown physical properties in the real world. Inspired by human's ability to reason about objects' physical properties by interacting with them, we design an inverse optimization pipeline through few-shot curiosity-driven interaction.

Specifically, to estimate the physical property variable, the robot actively interacts with the object. In each iteration, it selects the action that maximizes the predicted displacement of the object. Intuitively, the action that maximizes displacement is likely to reveal more information about the object's physical properties than random actions would. After each interaction, the robot updates its estimate of the object by minimizing the dynamics prediction error from previous interactions. As the robot undergoes several interactions, the estimation of physical property tends to stabilize, reaching the final optimized value.

In our experiments, we adopt a fixed number of iterations for adaptation. We measure the displacement of the object by computing the Chamfer Distance (CD) between the current state z_t and the predicted state $\hat{z}_t$:

$$\mathcal{L}_{CD}(\hat{z}_t, z_t) = \sum_{x \in z_t} \min_{y \in \hat{z}_t} ||x - y||^2_2 + \sum_{y \in \hat{z}_t} \min_{x \in z_t} ||x - y||^2_2. \tag{5}$$

The actions for curiosity-driven interactions are optimized using the Model-Predictive Path Integral (MPPI) [43] trajectory optimization algorithm to maximize the above Chamfer Distance.

For inverse optimization at the t^{th} interaction step, we adopt gradient-free optimizers including Bayesian Optimization (BO) for single-dimensional physical property variables and CMA-ES for multi-dimensional variables. We instantiate the optimization problem described in the main paper by specifying the cost function cost$(\hat{z}_{i+1}, z_{i+1})$ as the Chamfer Distance between the dynamics prediction and the true outcome after each interaction:

$$\hat{\phi}_t = \arg\min_{\phi} \sum_{i=0}^{t-1} \mathcal{L}_{CD}(\hat{z}_{i+1}, z_{i+1}), \tag{6}$$

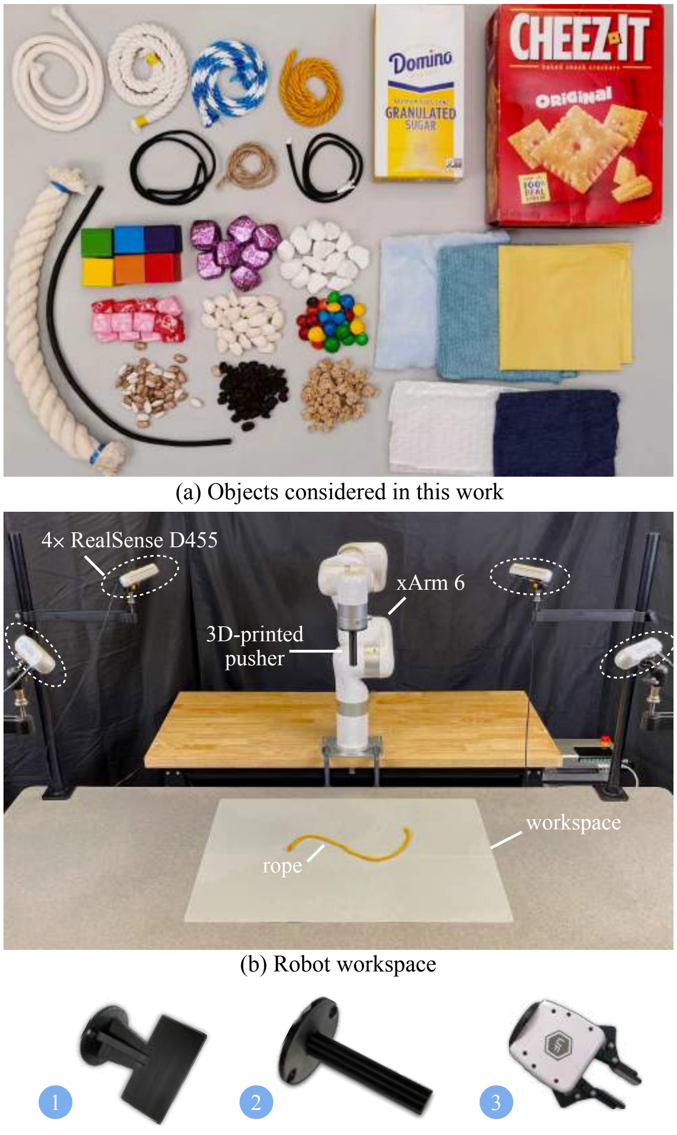
where, again, $\hat{z}_{i+1} = f(z_i, u_i; \hat{\phi}_t, M)$.

For some materials whose physical properties span a large range (e.g., stiffness for ropes), the test object can potentially fall outside the training distribution of the model. Our material-conditioned model allows for generalization beyond the training domain by directly setting the domain of $\hat{\phi}$ at the adaptation stage to be an extension of the maximal range of ϕ in the training data. Specifically, the minimum value and maximum value for $\hat{\phi}$ is $\phi_{min} - 0.2(\phi_{max} - \phi_{min})$ and $\phi_{max} + 0.2(\phi_{max} - \phi_{min})$, where ϕ_{max} and ϕ_{min} are the maximum and minimum value of ϕ in the training dataset.

C. Closed-Loop Model-Based Planning

Using the estimated physics parameter $\hat{\phi}$, the learned model f adapts to new objects, yielding lower dynamics prediction errors on the few-shot, curiosity-driven online interaction dataset. Thus, we can also use the adapted model to perform closed-loop planning for material-specific manipulation tasks within a Model Predictive Control (MPC) framework [5]. The improved dynamics prediction accuracy after few-shot adaptation will help the robot manipulate the object more efficiently and effectively towards goal configurations.

Concretely, the model-based control pipeline is defined as follows: given the state space $\mathbb{Z}$ and the action space $\mathbb{U}$, the cost function is a mapping from $\mathbb{Z} \times \mathbb{U}$ to $\mathbb{R}$. For each starting state $z_0 \in \mathbb{Z}$, we iteratively sample actions $\{u_i\}^T_{i=1}$ in the action space, apply the learned dynamics model to predict the outcome, and apply the MPPI trajectory optimization algorithm for the action sequence $\{u_i\}$ that minimizes the cost function. In our experiments, the cost function includes a task-related



(a) Objects considered in this work

(b) Robot workspace

(c) Robot tools

Fig. 1: **Real-world setup:** **(a)** Our study involves 22 objects categorized into four types of materials, each with distinct physical characteristics: (i) 9 varieties of ropes, such as cotton ropes and cables, (ii) 9 granular materials, including items like toy blocks and coffee beans, (iii) 5 pieces of cloth made from different fabrics like cotton and synthetic fibers, (iv) 2 boxes of varying shapes, whose centers of pressure we alter by placing weights inside them. **(b)** The dashed white circles show four calibrated RGB-D cameras mounted at four corners of the table. The robot is outfitted with specialized end effectors to interact with the objects in its operational area. **(c)** We employ three different tools for specific tasks: (1) a flat pusher for granular piles gathering, (2) a cylindrical pusher for pushing rigid boxes and straightening ropes, (3) an xArm gripper for cloth relocating.

term that measures the distance from the current state to the desired target, along with other penalty terms for infeasible actions and collision avoidance.

III. Experiments

A. Evaluation Materials and Corresponding Tasks

To demonstrate the modeling power of our framework for diverse materials, we implement one task for each of the 4 material categories: rigid box pushing, rope straightening, granular pile gathering, and cloth relocating.

Rigid Box Pushing. The task is to use a point contact to push a box to a target position and orientation, which demands precise control over the translation and rotation motions in the presence of uncertainty of the center of pressure [49]. The physical property variable is defined to be the normalized 2D position of the center of mass from the top view. It is a 2-dimensional variable $\phi = [c_x, c_y]$ with range $c_x, c_y \in (-0.5, 0.5)$. We use the mean squared error as the cost function.

Rope Straightening. The task is to rearrange the rope to a target configuration on the tabletop. We consider the stiffness of the rope as the physical property variable and define it as a normalized continuous variable $\phi \in (0, 1)$ where $\phi = 0$ and $\phi = 1$ correspond to the minimal and maximal stiffness in the simulator, respectively.

Granular Pile Gathering. The target is defined as a region on the tabletop, and the task is to gather the granular piles in an arbitrary initial distribution into the target region. We consider the granular size/granularity as the physical property variable and use a normalized variable $\phi \in (0, 1)$ to represent the size of a single grain in the pile.

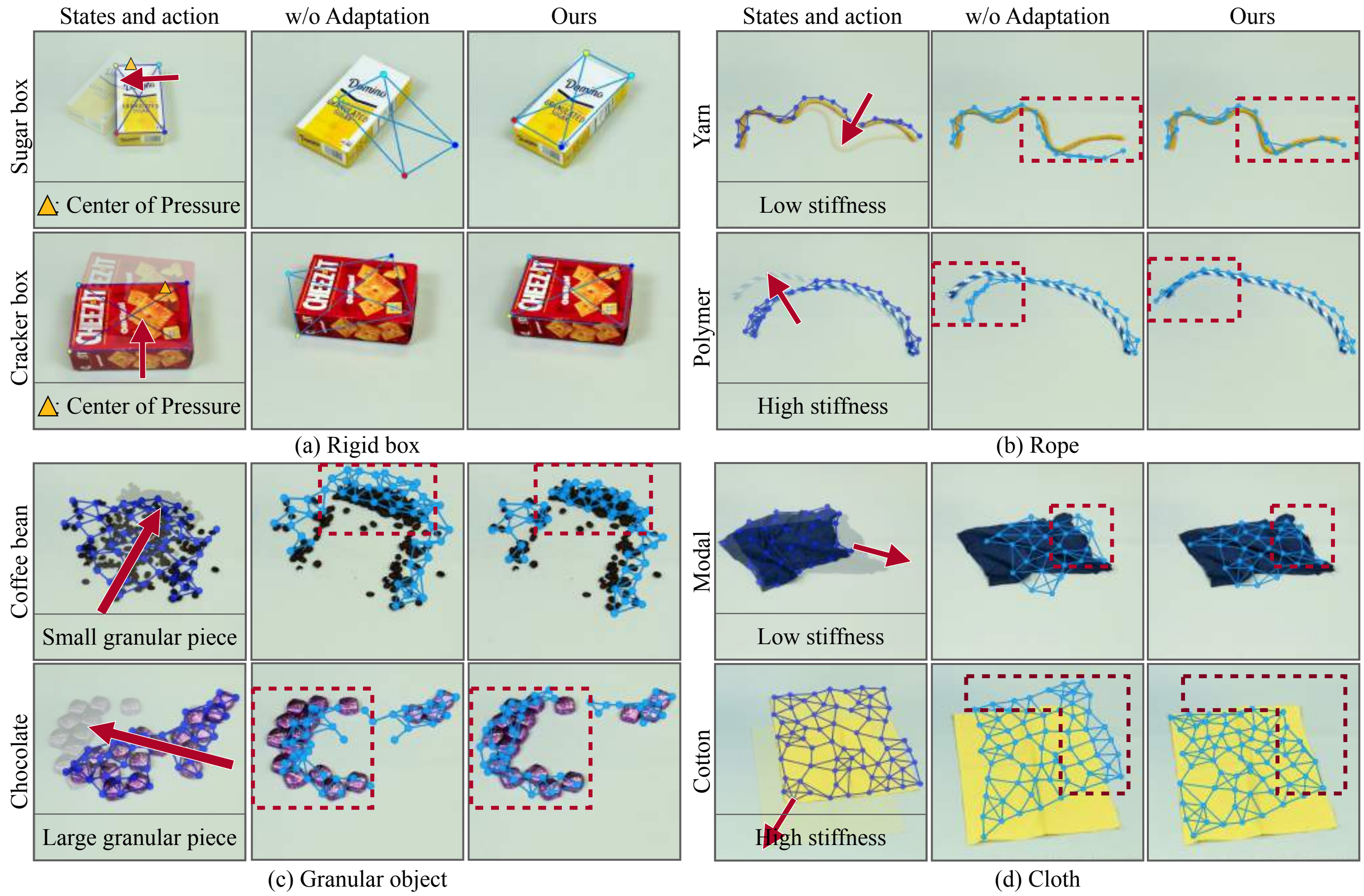
Cloth Relocating. The task is to use grippers to grasp the cloth and drag it on the table to place the cloth in the target configuration. We use a continuous variable $\phi \in (0, 1)$ to represent the stiffness of the cloth, which affects whether a piece of cloth will wrinkle or fold during a drag.

B. Environment and Evaluation Setup

Simulation. Simulations of deformable and granular materials are conducted using NVIDIA FleX [19, 28], a position-based simulation framework designed to model interactions between objects of varying materials across multiple tasks, including pushing granular objects [42], straightening ropes [23], and unfolding clothing [13]. Additionally, Pymunk [4] is utilized for simulating boxes that vary in shape and center of pressure.

For each material type, a dataset consisting of 1000 episodes is generated, with each episode featuring 5 random robot-object interactions. Within each episode, an object is assigned random physical properties (such as stiffness and granule size) that fall within a pre-defined range. To simulate interactions between the robot and the object, five random trajectories, involving either pushing or pulling actions, are created for every object. Throughout these interactions, data on the positions of particles and the robot's end-effector are gathered, which are then utilized for model training.

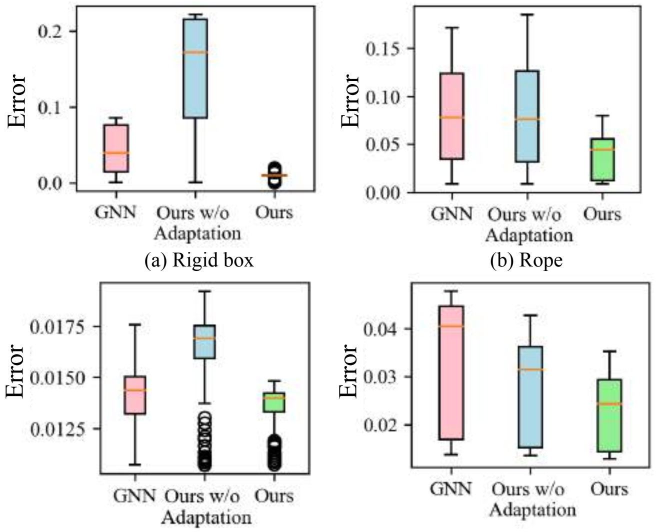
Real World. Fig. 1 presents the general setup in both the simulator and the real world. In the real-world experiments, we use a UFACTORY xArm 6 robot with 6 DoF and xArm's parallel gripper. For rigid box pushing and rope straightening tasks, we substitute the original grippers with a cylinder stick while we utilize a flat pusher for the granular pile manipulation task. These tools are 3D-printed and the same with the simulation setup to mitigate the sim-to-real gap. We fix four calibrated RealSense D455 RGBD cameras at four locations



(a) Rigid box (b) Rope

(c) Granular object (d) Cloth

Fig. 2: **Qualitative results on dynamics prediction:** We conduct qualitative comparisons to assess the performance of our method against the baseline of a GNN without adaptation, focusing on the one-step prediction of dynamics across eight objects within four distinct material categories exhibiting varying extreme physical properties. The results, delineated by red dashed boxes, demonstrate that our approach surpasses the baseline in accurately capturing the variations in dynamics that arise due to differences in the objects' physical properties.



(a) Rigid box (b) Rope

(c) Granular object (d) Cloth

Fig. 3: **Quantitative results on dynamics prediction:** We validate our model's effectiveness on a test set of 200 objects with distinct physical properties for each material type in simulation. Across all types of materials, our approach surpasses the baseline with respect to both the precision and consistency of predictions.

surrounding the workspace to capture the RGBD images at 15Hz and 1280x720 resolution. The robot manipulates objects within a 70 cm×45 cm planar workspace.

Implementation Details. To extract object point clouds from raw RGB-D inputs, we deploy the GroundingDINO [25] and Segment Anything [17] model to detect and segment the table surface and objects. For the target object, we fuse the segmented partial point cloud from 4 views and apply a farthest point sampling method to a fixed pointwise distance threshold. For the cylinder stick and gripper, we use one particle to represent the end effector position, and for the flat granular pusher, we use 5 points to represent the end effector position and geometry.

Baselines. We consider two baseline methods in our experiments: (1) *GNN* uses a graph neural network with the same architecture of our model, but without physics parameter conditioning. The model is trained on the same simulation dataset but without distinguishing the objects in the dataset with different physical properties. (2) *Ours w/o Adaptation* is an ablated version of our material-adaptive model by using only the mean physical property variable $\bar{\phi}$ as input in deployment.

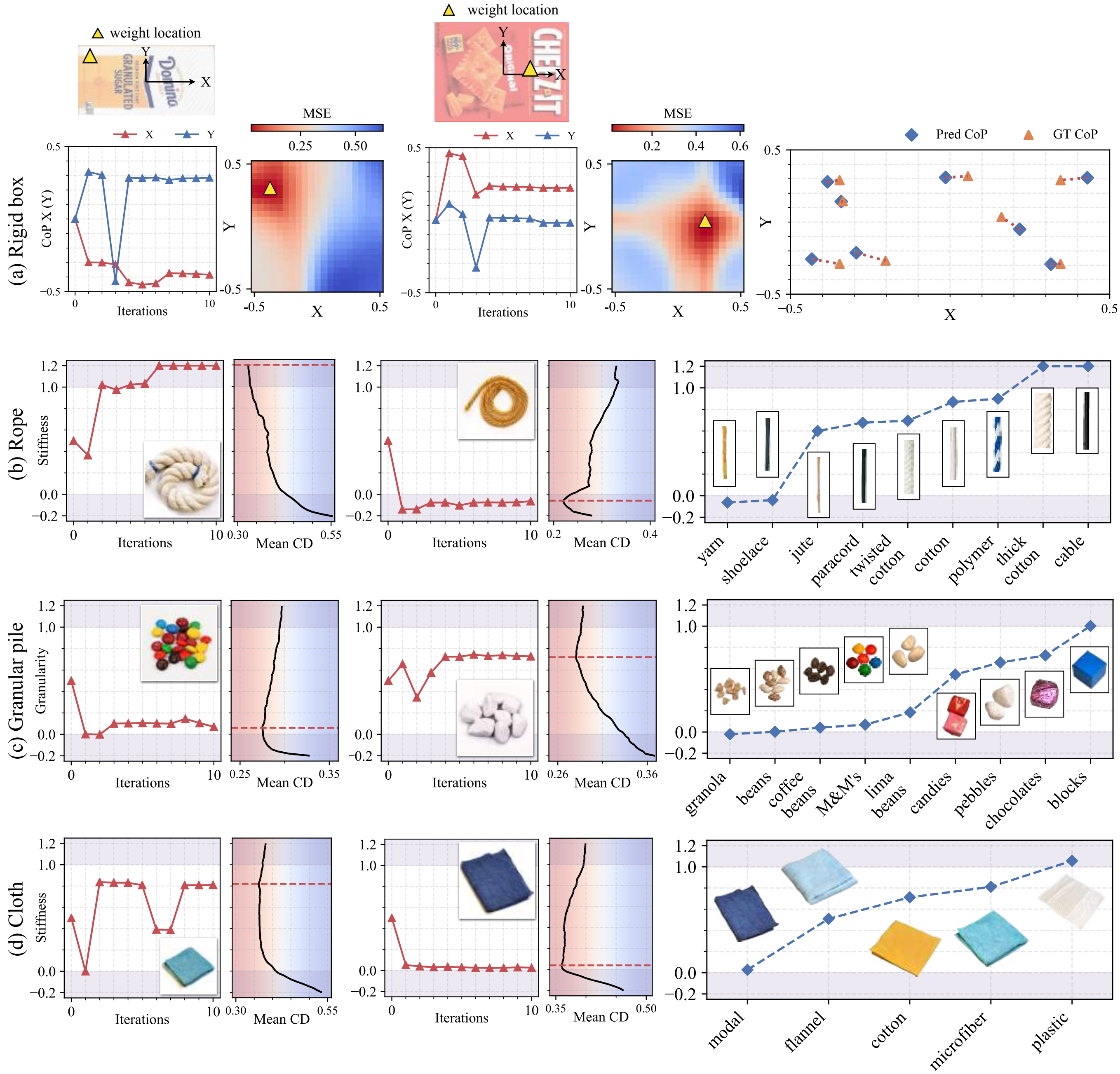


Fig. 4: **Experimental results on physical property estimation:** Through the inverse optimization process, we estimate the physical properties of real-world objects. For each material type, we display the optimization trajectories alongside their associated costs, measured by the Chamfer Distance, for two objects with notably contrasting physical attributes. The rightmost column demonstrates that our estimated values align with human perceptions regarding the perceptual order of objects based on their physical property values, such as stiffness and granularity.

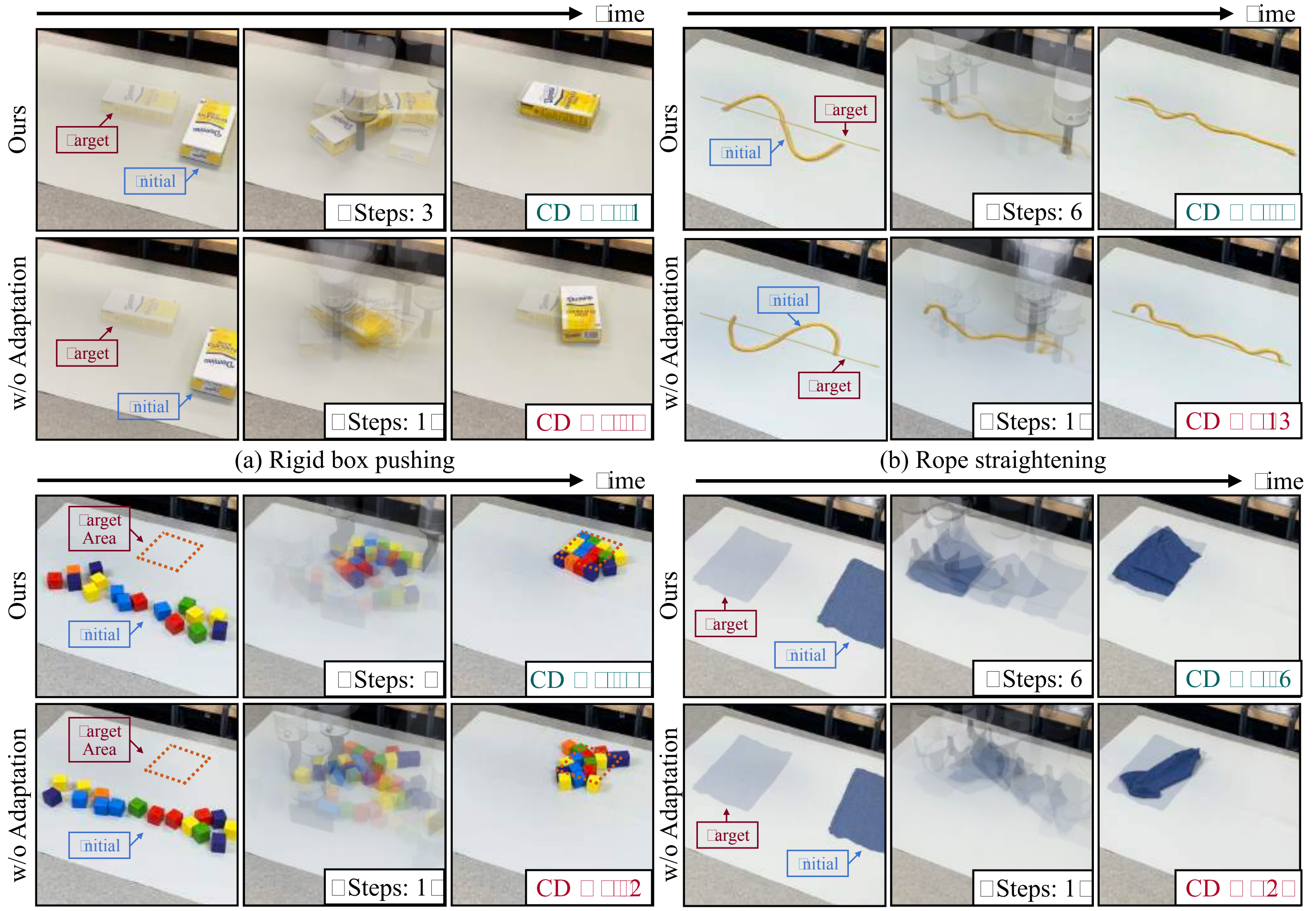
C. Forward Dynamics Prediction

Fig. 3 further validates our model's effectiveness on a simulated test set of 200 objects each with distinct physical properties. Our approach surpasses both baselines, *GNN* and *Ours w/o Adaptation*, demonstrating superior accuracy and stability for all the material types addressed in our study. Particularly, for rigid boxes, our model significantly outperforms the baselines with a near-perfect prediction accuracy.

D. Physical Property Estimation

For physical property estimation, we randomly initialize the object location on the tabletop and perform 10 interactions.

Rigid Box. We use two boxes with different sizes: the sugar box (175mm×89mm) and the cracker box (210mm×158mm). We initialize the center of pressure (CoP) to be at 4 different locations for each box by putting weights at different locations inside the box. A visualization of all CoPs' normalized positions and our predicted CoP positions is shown in Fig. 4a. From the figure, we can observe that for all 8 data points, the predicted CoP positions are close to the ground truth CoP position. Moreover, the heatmap error shows that the low-error region for the CoP location forms a single global minima, and the predicted CoP positions converge to around the minimum value after around 5 interaction steps.



(a) Rigid box pushing

(b) Rope straightening

(c) Granular pile gathering

(d) Cloth relocating

Fig. 5: **Qualitative results on closed-loop feedback planning:** We present a qualitative comparison of MPC performance by contrasting our method across four tasks with the baseline model that does not employ physical property adaptation. Visualizations shown here demonstrate that our method effectively achieves the target configuration, whereas the baseline, even with more action steps, still exhibits a noticeable discrepancy compared to the target.

Rope. We test our model on 9 different types of ropes. As shown in Fig. 4b, the model can extrapolate beyond the training data range [0.0, 1.0] and estimate out-of-range values for ropes with extreme stiffness/softness. The mean CD on the interaction observations gives clear and unique minimum points, and the stiffness ranking of the different types of ropes is consistent with the actual stiffness from human perception.

Granular. As shown in Fig. 4c, we test our model on 9 different types of granular objects by selecting representative objects of each granularity level, ranging from approximately 1cm to 3cm. Results show that the predicted granularity ranking is consistent with the actual granular size. The model correctly predicts granola as the smallest grains and the toy blocks as the largest grains.

Cloth. As shown in Fig. 4d, we test our model on 5 different cloth instances, each with a different fabric material. The model correctly identifies the modal as the softest cloth (lowest stiffness). As another soft material, the flannel cloth is also estimated to be softer than cotton and microfiber cloths. While the training dataset does not contain any plastic-like materials, the model generalizes to a piece of plastic sheet and correctly predicts that it is very stiff.

E. Model-Based Planning

We further demonstrate that our material-conditioned GBND model and physical property adaptation can be integrated into an MPC framework to achieve a series of robotic manipulation tasks. Our experiments cover 4 distinct tasks, with a maximum limit of 10 planning steps imposed. Across all material types, our approach consistently meets the objectives within the allotted planning steps, unlike the baseline approach *Ours w/o Adaptation*, which fails to achieve the goals due to its disregard for physical properties. For instance, in the rigid box pushing task, the baseline method incorrectly assumes the geometric center as the center of pressure, leading to inaccurate predictions of the box's straightforward movement post-push. Conversely, our method dynamically adjusts the center of pressure estimations during the interactions, thereby reaching the desired configuration in just three steps. Furthermore, the dynamics of pushing granular objects of different sizes vary significantly - larger granules push forward while smaller

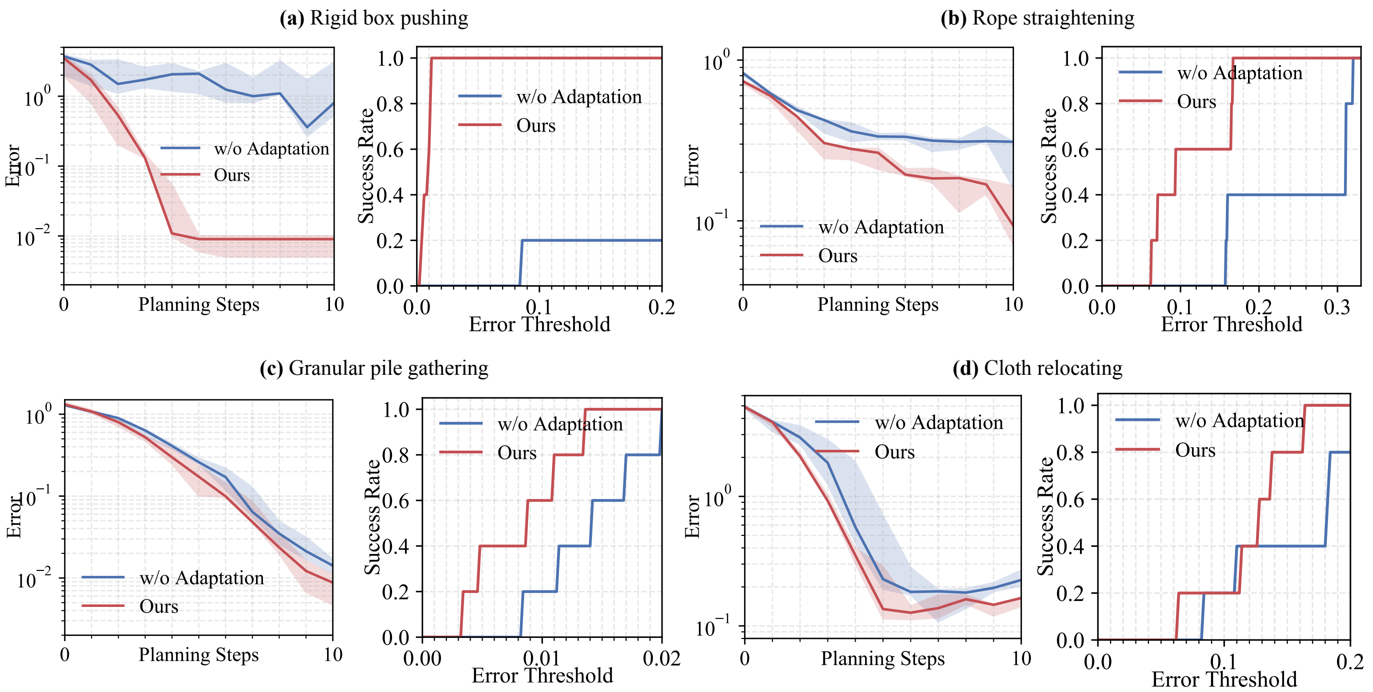


Fig. 6: **Quantitative results on planning:** For each task, we use the same target configuration and initial configuration for the baseline method and our approach. We repeat each experiment-model pair 5 times and visualize (i) the median error curve w.r.t. planning steps (area between 25 and 75 percentiles are shaded) and (ii) the success rate curve w.r.t error thresholds. Our approach consistently outperforms the baseline method by being more accurate and using fewer action steps.

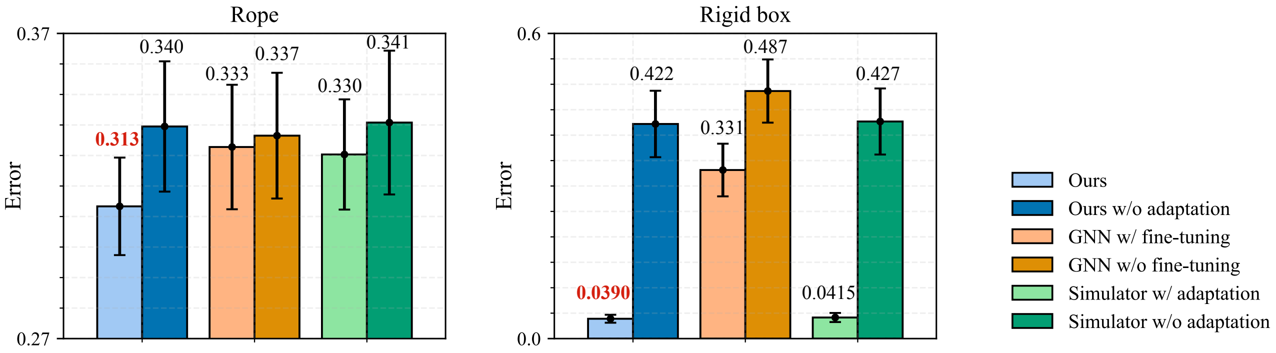


Fig. 7: **Quantitative comparison with baselines adaptation approaches:** We report the mean and standard error of the dynamics prediction errors after online adaptation on 5 interactions. The numbers denote the mean value of each bar. For both material categories, our method achieves the lowest error across all methods after adaptation. Error metrics: Rope - CD, rigid box - MSE.

ones tend to stack and leave a trail. The baseline method, treating the motion of toy blocks and average granular piles similarly, fails to accumulate them in the target zone. Our method, however, identifies and adapts to the varied dynamics of granular materials, successfully completing the task.

Fig. 6 offers quantitative results comparing the performance of our method against the baseline method *Ours w/o Adaptation*, focusing on efficiency and error tolerance. Across four distinct tasks, our approach demonstrates superior performance, achieving lower errors within a constrained number of planning steps and attaining a higher success rate under a stringent error margin.

F. Additional Comparison with Baselines

In this section, we further compare our model to (1) simulators incorporating physical property adaptation and (2) fine-tuning unconditional GNN. We evaluate their dynamics prediction error after adaptation in few-shot real-world interactions.

We evaluate on two material categories: rope and rigid boxes. For ropes, we apply mesh reconstruction based on alpha shapes [9] to derive the rope mesh in the FleX simulator. For boxes, we extract the 4 corners of the box from the top view and create an identical 2D box in Pymunk. Due to the state complexity of clothes and granular objects, both sampling- and learning-based perception models fail to accurately recover the objects' state, leading to unstable simulation results. Hence, we exclude them from the comparison.

From Fig. 7, we can observe that our method, with online adaptation, exhibits the lowest dynamics prediction error. It achieves an error reduction of 7.9% for ropes and 90.8% for rigid boxes when compared to our model without adaptation. Notably, the error reduction ratio surpasses that achieved by fine-tuning an unconditional GNN-based dynamics model. In comparison with simulator-based physical property adaptation, our model demonstrates a 5.2% lower dynamics prediction error for ropes and a 6.0% lower error for rigid boxes. We attribute this improvement to the inherent system identification error and the instability of the simulator. Using a learning-based dynamics model directly on point clouds enhances our model's robustness to noisy visual inputs.

Moreover, our model is significantly faster than simulators. Running the Bayesian optimization algorithm for 50 iterations takes approximately 7 seconds for our model on a desktop computer equipped with an i9-13900K CPU and an NVIDIA GeForce RTX 4090 GPU, whereas it takes approximately 900 seconds for the FleX simulator.

References

[1] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016. 1

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 1

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018. 1

[4] Victor Blomqvist. Pymunk. https://pymunk.org, November 2022. 3

[5] Eduardo F. Camacho and Carlos Bordons Alba. *Model Predictive Control*. Springer Science & Business Media, 2013. 2

[6] Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B. Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos, 2022. 1

[7] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018. 1

[8] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Joshua B. Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language, 2021. 1

[9] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions On Graphics (TOG)*, 13(1):43–72, 1994. 7

[10] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 1

[11] Barbara Frank, Rüdiger Schmedding, Cyrill Stachniss, Matthias Teschner, and Wolfram Burgard. Learning the elasticity parameters of deformable objects with a manipulation robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1877–1883. IEEE, 2010. 1

[12] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *arXiv preprint arXiv:2309.02561*, 2023. 1

[13] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robotic Learning (CoRL)*, 2021. 3

[14] François Robert Hogan and Alberto Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. *arXiv preprint arXiv:1611.08268*, 2016. 1

[15] Ryan Hoque, Daniel Seita, Ashwin Balakrishna, Aditya Ganapathi, Ajay Kumar Tanwani, Nawid Jamali, Katsu Yamane, Soshi Iba, and Ken Goldberg. Visuospatial foresight for multi-step, multi-task fabric manipulation. *arXiv preprint arXiv:2003.09044*, 2020. 1

[16] Isabella Huang, Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, Tucker Hermans, and Dieter Fox. Defgraspnets: Grasp planning on 3d fields with graph neural nets, 2023. 1

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4

[18] 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. 1

[19] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019. 1, 3

[20] Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *ICRA*, 2019. 1

[21] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel Yamins, Jiajun Wu, Joshua Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *International conference on machine learning*, pages 5927–5936. PMLR, 2020. 1

[22] Junbang Liang, Ming Lin, and Vladlen Koltun.

Differentiable cloth simulation for inverse problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/28f0b864598a1291557bed248a998d4e-Paper.pdf. 1

[23] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, 2020. 3

[24] Xingyu Lin, Yufei Wang, Zixuan Huang, and David Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022. 1

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[26] Ziang Liu, Genggeng Zhou, Jeff He, Tobia Marcucci, Li Fei-Fei, Jiajun Wu, and Yunzhu Li. Model-based control with sparse neural dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ymBG2xs9Zf. 1

[27] Alberta Longhini, Marco Moletta, Alfredo Reichlin, Michael C Welle, David Held, Zackory Erickson, and Danica Kragic. Edo-net: Learning elastic properties of deformable objects from graph dynamics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3875–3881. IEEE, 2023. 1

[28] 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. 3

[29] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 1

[30] Anusha Nagabandi, Kurt Konoglie, Sergey Levine, and Vikash Kumar. Deep Dynamics Models for Learning Dexterous Manipulation. In *Conference on Robot Learning (CoRL)*, 2019. 1

[31] Tao Pang, HJ Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 2023. 1

[32] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020. 1

[33] Kavya Puthuveetil, Sasha Wald, Atharva Pusalkar, Pratyusha Karnati, and Zackory Erickson. Robust body exposure (robe): A graph-based dynamics modeling approach to manipulating blankets over people. *IEEE Robotics and Automation Letters*, 2023. 1

[34] 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. 1

[35] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper, 2020. 1

[36] Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *arXiv preprint arXiv:2205.02909*, 2022. 1

[37] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023. 1

[38] Priya Sundaresan, Rika Antonova, and Jeannette Bohgl. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10828–10835. IEEE, 2022. 1

[39] Fish Tung, Mingyu Ding, Zhenfang Chen, Daniel M. Bear, Chuang Gan, Joshua B. Tenenbaum, Daniel L. K. Yamins, Judith Fan, and Kevin A. Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *arXiv*, 2023. 1

[40] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward H Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 1

[41] Yi Ru Wang, Jiafei Duan, Dieter Fox, and Siddhartha Srinivasa. Newton: Are large language models capable of physical reasoning? *arXiv preprint arXiv:2310.07018*, 2023. 1

[42] Yixuan Wang, Yunzhu Li, Katherine Driggs-Campbell, Li Fei-Fei, and Jiajun Wu. Dynamic-Resolution Model Learning for Object Pile Manipulation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.047. 1, 3

[43] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017. 2

[44] Jiajun Wu, Ilker Yildirim, Joseph J Lim, William T Freeman, and Joshua B Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems*, pages 127–135, 2015. 1

[45] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In *Robotics: Science and Systems (RSS)*, 2019. 1

[46] Linhan Yang, Bidan Huang, Qingbiao Li, Ya-Yen Tsai, Wang Wei Lee, Chaoyang Song, and Jia Pan. Tacgnn:learning tactile-based in-hand manipulation with a blind robot, 2023. 1

[47] Shaoxiong Yao and Kris Hauser. Estimating tactile models of heterogeneous deformable objects in real time. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12583–12589, 2023. doi: 10.1109/ICRA48891.2023.10160731. 1

[48] Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 30–37. IEEE, 2016. 1

[49] Jiaji Zhou, Yifan Hou, and Matthew T Mason. Pushing revisited: Differential flatness, trajectory planning, and stabilization. *The International Journal of Robotics Research*, 38(12-13):1477–1489, 2019. 3