Estimating Deformable-Rigid Contact Interactions for a Deformable Tool via Learning and Model-Based Optimization

Mark Van der Merwe, Miquel Oller, Dmitry Berenson, and Nima Fazeli[1]

***Abstract*— Dexterous manipulation requires careful reasoning over extrinsic contacts. The prevalence of deforming tools in human environments, the use of deformable sensors, and the increasing number of soft robots yields a need for approaches that enable dexterous manipulation through contact reasoning where not all contacts are well characterized by classical rigid body contact models. Here, we consider the case of a deforming tool dexterously manipulating a rigid object. We propose a hybrid learning and first-principles approach to the modeling of simultaneous motion and force transfer of tools and objects. The learned module is responsible for jointly estimating the rigid object's motion and the deformable tool's imparted contact forces. We then propose a Contact Quadratic Program to recover forces between the environment and object subject to quasi-static equilibrium and Coulomb friction. The results is a system capable of modeling both intrinsic and extrinsic motions, contacts, and forces during dexterous deformable manipulation. We train our method in simulation and show that our method outperforms baselines under varying block geometries and physical properties, during pushing and pivoting manipulations, and demonstrate transfer to real world interactions.**

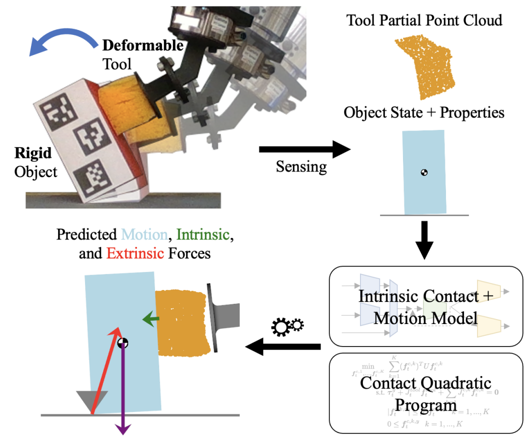


Fig. 1: We present a method that estimates motions and forces during dexterous manipulation with a deformable tool using a hybrid learning and first-principles approach.

I. Introduction

Reasoning over the contact between a tool or end-effector and a target object is crucial to enabling performant, safe, and autonomous robotic manipulation. This extends naturally to the case where the contacts are not all well represented by rigid-body contact models. Deformable tools and objects are commonly found in human environments [1] and the inherent compliance may be advantageous for both maintaining contact and preventing excessive forces while interacting with, securing and controlling objects. Additionally, the advent of deformable tactile sensors [2], [3], [4] and soft robot manipulators [5] introduce new cases of non-rigid contact into robotic manipulation.

In this work, we consider an elastically deformable tool manipulating a rigid object, supported by the environment. Manipulation of such a system presents two important challenges: First, the deformation of the tool and motion of the rigid object are inherently linked - solving for one requires reasoning about both. However, modeling the deformation of the tool is challenging, due to the high-dimensional and non-linear dynamics [6]. Second, the system is only partially observable, and we must rely on sensing to intuit deformations and contact forces.

To address these challenges, we propose a hybrid learning and first principles method for modeling the motion, contacts, and forces on the tool and extrinsic object during manipulation. Our method leverages learning to address the complexity of the paired object motion and deformable tool contacts. It then turns to first-principles and physical priors to recover the extrinsic contacts and forces, enforcing quasi-static motion and Coulomb friction. This allows our method to overcome modeling challenges for the deformable object, while exploiting physical priors for modeling efficiency. We train our method on interactions between a tool and a variety of object geometries and physical properties using simulation [7]. We benchmark our method against baselines for modeling block motion and for recovering the environment-object forces. Finally, we deploy our model on a real robot, demonstrating sim-to-real transfer.

II. Related Work

To predict the motion of a rigid body, existing work has investigated directly predicting SE3 transforms of objects [8], [9] or predicting point cloud or mesh vertex motion [10], [11]. Pfrommer et al. [12] propose to learn rigid body motion by parameterizing inter-body signed distances and contact Jacobians, enabling analytical physical simulation that can reflect rigid contacts. Calandra et al. [13] recovers contact forces and incorporates them into dynamics for a robot arm; we recover contact forces and incorporate them into quasi-static equilibrium for a rigid body. Other work seeks to directly predict the motion of heterogenous materials by applying Graph Neural Networks [14] or implicit models [15], [6]. In contrast, we avoid directly modeling the deformation of the deforming tool and only focus on the contact forces and rigid body motion.

[1]Robotics Department, University of Michigan, Ann Arbor, MI, USA 48105 {markvdm, oller, dmitryb, nfz}@umich.edu

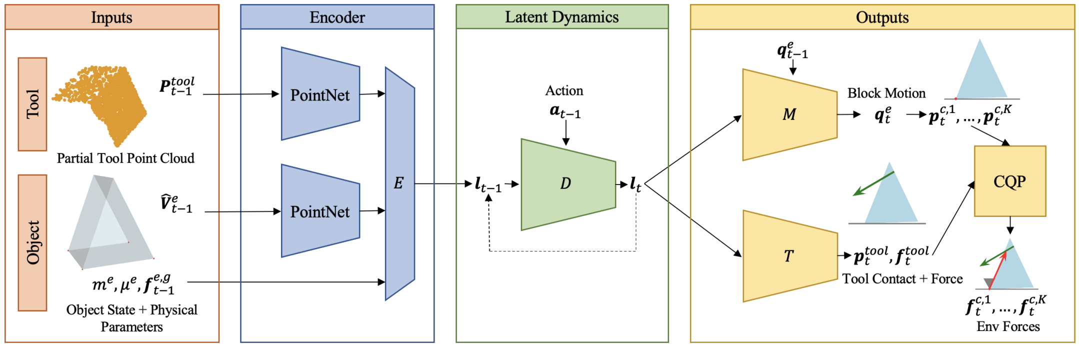


Fig. 2: Architecture of our proposed method.

III. Problem Formulation

We assume the following information from each part of the system:

1) **Object and Support Surface:** We assume knowledge of the mass m^e, the center of mass $\boldsymbol{p}^{e,CoM}$, the friction between object and rigid supporting surface μ^e, the geometry of the object, provided as a mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, and the geometry of the support surface. Furthermore, we can track the motion of the rigid object to receive the current object pose $\boldsymbol{q}_t^e \in SE(2)$. By our assumed knowledge of the environment geometry, we additionally can use the known object pose and geometry to recover the current contact locations $\boldsymbol{p}_t^{c,1}, ..., \boldsymbol{p}_t^{c,K}$.
2) **Tool:** We assume access to a *segmented* partial pointcloud of the deformable tool $\boldsymbol{P}^{tool}$, provided by our sensors.

Given these inputs and assumptions, and the actions to be taken $\boldsymbol{a}_{t:t+H}$ for some horizon H, where each action is a planar translation and rotation, we wish to recover the future poses of the object $\boldsymbol{q}_{t:t+H}^e$ and all forces acting on the object. We consider a simple contact representation for the deformable-rigid contact: a summary contact point $\boldsymbol{p}_t^{tool}$ and force $\boldsymbol{f}_t^{tool}$, whose resulting wrench on the block is equivalent to the actual and potentially extended contact. We use the centroid of the contact between the tool and object as $\boldsymbol{p}_t^{tool}$, which prevents ambiguity in the representation. Finally, we wish to also recover the set of contact points $\boldsymbol{p}_t^{c,1}, ..., \boldsymbol{p}_t^{c,K}$ and forces $\boldsymbol{f}_t^{c,1}, ..., \boldsymbol{f}_t^{c,K}$ between the rigid object and the environment.

IV. Method

Our proposed method has two main components - a learned module for predicting a) the object motion and b) the contact location and forces between the deforming tool and object, and a model-based optimization problem for recovering the resulting frictional contacts with the environment. An overview of our approach can be found in Fig. 2.

A. Jointly Learning Object Motion and Deformable-Rigid Contact Interactions

An overview of our proposed learning architecture is shown in Fig. 2. We encode information from the object and the tool into a learned latent space. A latent dynamics backbone then propagates the latent information provided with the actions. Finally, we have multiple output heads that decode the object motion and tool contact information from the latent.

We input the partial pointcloud of the tool $\boldsymbol{P}_t^{tool}$ using a PointNet encoder [16]. We encode the geometry and current pose of the extrinsic object by extracting the mesh vertices V and transforming them to the tracked object location $\boldsymbol{q}_t^e$. We use the environment geometry to detect which vertices are contacting the surrounding environment and append these binary contact indicators to each vertex location. We use another PointNet encoder [16] to encode the resulting unstructured point set $\hat{\boldsymbol{V}}_t^e \in \mathbb{R}^{|V| \times 4}$. We fuse the resulting latent embeddings from the tool and rigid object, along with the object mass m^e, object friction μ^e, and the gravity vector $\boldsymbol{f}_t^{e,g}$. Given the input actions $\boldsymbol{a}_{t:t+T}$, we rollout the dynamics in the latent space [17].

The final learned components are two output networks which take the latent state at a given time step and estimate the motion. One MLP M is used to estimate the motion of the object, given the current latent state, expressed as a delta motion. A final MLP T is used to directly regress the tool contact point $\boldsymbol{p}_t^{tool}$ and force $\boldsymbol{f}_t^{tool}$. We assume access to a labeled dataset with the labeled tool contact and force as well as the delta extrinsic object motion, and train our method in a supervised fashion.

B. Model-Based Estimation of Frictional Object-Environment Contact

We can recover the set of K active contact points at a given time step $\boldsymbol{p}_t^{c,1}, ..., \boldsymbol{p}_t^{c,K}$ given our estimated object poses and the known object and environment geometries. We now aim to find the contact forces at those points $\boldsymbol{f}_t^{c,1}, ..., \boldsymbol{f}_t^{c,K}$. Assuming the system is in quasi-static equilibrium, and given the contact force applied by the tool $\boldsymbol{f}_t^{tool}$, we know the set of environment forces must satisfy force balance and all forces must satisfy Coloumb friction. To find our forces, we thus propose the following Quadratic Program (QP) to find our forces.

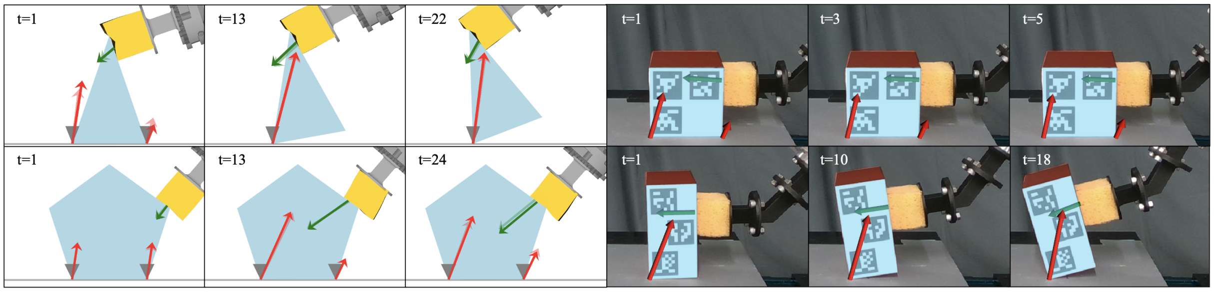
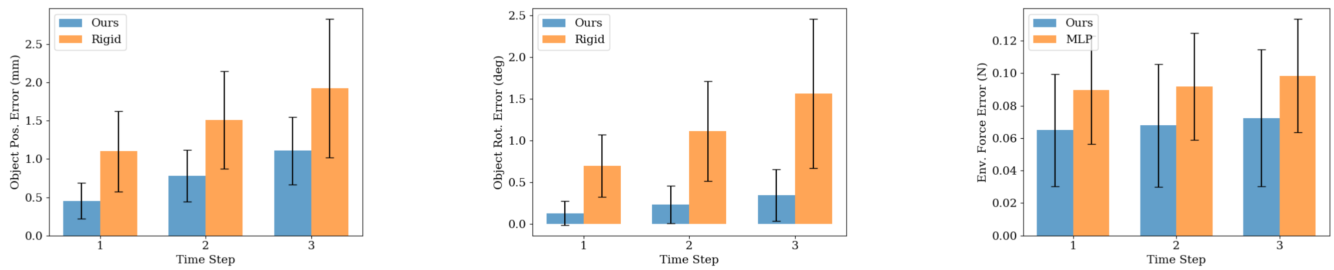


Fig. 3: Qualitative Results. **Left:** We compare our methods one-step predictions (solid) to the ground truth (semi-transparent) object motion, tool force, and environment forces across several manipulation trajectories. **Right:** We show one-step predictions for real robot executions.



(a) Object Position Error (↓) (b) Object Rotation Error (↓) (c) Extrinsic Force Error (↓)

Fig. 4: Baseline comparison of our method on test simulated interactions. For (a) and (b) we compare to a Rigid baseline that predicts block motion as if rigidly attached to the tool. For (c) we compare our model-based optimization for extrinsic contact recovery to directly predicting it from an MLP. Error bars indicate one half standard deviation.

$$
\begin{aligned}
\min_{\hat{\boldsymbol{f}}_t^{tool}, \boldsymbol{f}_t^{c,1}, \ldots, \boldsymbol{f}_t^{c,K}} \quad & (\Delta \boldsymbol{f}_t^{tool})^T U (\Delta \boldsymbol{f}_t^{tool}) + \rho \sum_{k=1}^{K} (\boldsymbol{f}_t^{c,k})^T U \boldsymbol{f}_t^{c,k} \\
\text{s.t.} \quad & \boldsymbol{\tau}_t^g + J_t^{tool} \hat{\boldsymbol{f}}_t^{tool} + \sum_{k=1}^{K} J_t^{c,k} \boldsymbol{f}_t^{c,k} = \boldsymbol{0} \\
& |\boldsymbol{f}_t^{c,k,x}| \leq \mu^e \boldsymbol{f}_t^{c,k,y} \quad k = 1, \ldots, K \\
& 0 \leq \boldsymbol{f}_t^{c,k,y} \quad k = 1, \ldots, K
\end{aligned}
\tag{1}
$$

Here, $\Delta \boldsymbol{f}_t^{tool} = \boldsymbol{f}_t^{tool} - \hat{\boldsymbol{f}}_t^{tool}$. We introduce a new decision variable $\hat{\boldsymbol{f}}_t^{tool}$ which we use to achieve force balance. We introduce a cost to minimize the difference between the new decision variable and our original estimated tool force $\boldsymbol{f}_t^{tool}$. We then solve for force balance (first constraint) and Coloumb friction (second and third) on the environment contacts. By using a new decision variable $\hat{\boldsymbol{f}}_t^{tool}$, we ensure the QP is feasible, while finding forces that yield quasi-static balance.

V. Experiments

A. Training

We use the Drake simulator [18] to generate a labeled dataset, as it supports deformable-rigid contacts [7]. We attach a simple deformable tool to the end of a gripper. We use a 46mm cube to match our real tool (Fig. 1) and utilize Drake's Finite Element Method simulation, setting

TABLE I: Real World 1-Step Object and Force Error (↓)

Metric		Pos. Error (mm)	Rot. Error (deg)	Force Error (N)
Blk. 1	Pivot	0.364 (0.128)	0.262 (0.125)	0.745 (0.292)
	Push	1.163 (0.420)	1.144 (0.613)	0.977 (0.445)
Blk. 2	Pivot	0.466 (0.196)	0.071 (0.054)	1.033 (0.382)
	Push	2.044 (0.211)	0.069 (0.032)	1.180 (0.160)

Youngs Modulus to 1.1e4, Poisson's ratio to 0.1, and density to $30 kg/m^3$. For the extrinsic object, we use three object primitives from which we generate geometric variations: a rectangle, triangle and pentagon. We train on 5000 pushing and 5000 pivoting trajectories for each object type, performed by a heuristic policy. We implement our network in PyTorch and solve our CQP using CvxPy, and train over a horizon of length 4.

B. Simulated Results

We demonstrate the performance of our model on a test simulation dataset of 500 push and 500 pivot trajectories for each object type.

We report our method's performance over a prediction horizon of three steps for tool contact point and contact force accuracy across all our simulated test trajectories in Fig. 5. For all interactions and across our prediction horizon, our method showed the ability to accurately estimate future contact points between the deforming tool and extrinsic block, to within 6mm on average (against a tool of size

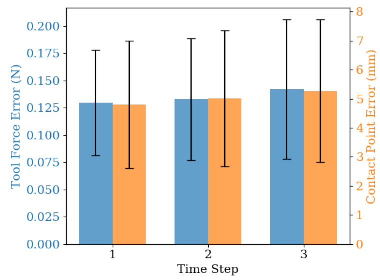


Fig. 5: Tool Contact Force (blue) and Contact Point Location (orange) error for our proposed model, plotted by prediction horizon time step. Error bars indicate one half standard deviation.

46mm), and showed the ability to estimate future contact forces to within 0.2N on average.

To benchmark our method's object motion performance, we introduce a *Rigid* baseline which assumes that the block moves "rigidly" with respect to the tool. In Fig. 4a and 4b we show that our method outperformed the *Rigid* baseline, with sub-millimeter position accuracy and less than 0.5 degrees of rotational error on average across the full prediction horizon.

To benchmark our extrinsic force modeling, we create a variation which replaces the CQP with a learned MLP following the form of our other network output heads. In particular, it takes in the latent state $\boldsymbol{l}_t$ and directly predicts the contact forces $\boldsymbol{f}_t^{c,1}, ..., \boldsymbol{f}_t^{c,K}$. In Fig. 4c, we see that our proposed method outperforms the baseline, yielding the best average force prediction across all scenarios. Our method consistently performed to within 0.08N of error on average, across the prediction horizon. Our results suggests that our proposed CQP method for recovering environment forces outperforms direct learning.

Finally, Fig. 3 (Left) shows qualitative examples of our methods predictions on simulated data with different objects and actions (pivoting and pushing).

C. Real Robot Results

We demonstrate our methods performance on a Franka Emika Panda robot and interact with two rectangular objects. We collect 5 push and 5 pivot trajectories with each object. We mount an ATI Gamma Force/Torque sensor under the tabletop. This allows us to compare the cumulative force experienced by the table via the extrinsic contacts, which we compare to our predicted extrinsic contacts.

In Table I, we show our one-step object motion prediction and extrinsic force prediction on the real data. Despite noisy sensor information, we are able to predict object motion to within roughly 2mm and 2 degrees error. Our extrinsic cumulative contact prediction is within roughly 1N on average. In Fig. 3 (Right), we show full qualitative predictions.

D. Force Tracking

Regulating force on the extrinsic object can be important in cases where the object or environment are fragile. Here, we demonstrate force tracking using our proposed extrinsic

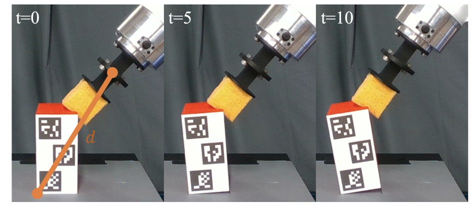


Fig. 6: Our pivot force tracking task, with the decision variable d shown in leftmost frame.

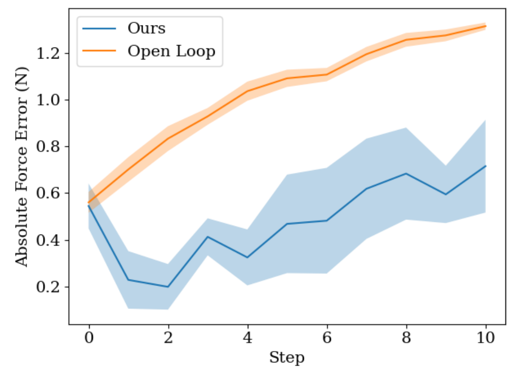


Fig. 7: We plot, over five trials, the evolution of the absolute force magnitude error as a function of the trajectory step. Our method is able to better track the desired force, compared to a method without force feedback executing a similar pivot.

contact estimates. We execute a pivoting motion, pivoting around the contact point furthest from the robot. We setup a simple greedy controller which can adjust the arc length d for the next step, thereby allowing it to press harder or release as it pivots. We use our proposed model and predict the motion and forces for the sampled actions (with a single-action horizon). We then select the action with the resulting cumulative force magnitude closest to the desired force setting ν. We set the target force magnitude ν to be the gravitation force given the mass of the block plus 1.5N. We use the F/T sensor mounted on the environment to determine approximately how well we tracked the desired force. An example execution is shown in Fig. 6. We show tracking performance over five trials against an open loop method in Fig. 7.

VI. Conclusion

We present a method capable of jointly recovering object motion and intrinsic/extrinsic contact information during dexterous deformable manipulations. We found that our hybrid learning and first-principles approach outperformed purely learning methods when recovering environment forces. In future work, we hope to leverage this work for the planning of dexterous deformable motion, where we aim to utilize force estimates to reason about contact modes and force targets.

References

[1] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science*

Robotics, vol. 6, no. 54, p. eabd8803, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abd8803

[2] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1927–1934.

[3] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake, "Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 597–604.

[4] H. Sun, K. J. Kuchenbecker, and G. Martius, "A soft thumb-sized vision-based sensor with accurate all-round force perception," *Nature Machine Intelligence*, vol. 4, no. 2, pp. 135–145, 2022.

[5] U. Yoo, N. Dennler, M. Mataric, S. Nikolaidis, J. Oh, and J. Ichnowski, "Moe-hair: Toward soft and compliant contact-rich hair manipulation and care," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 1163–1167.

[6] Y. Wi, P. Florence, A. Zeng, and N. Fazeli, "Virdo: Visio-tactile implicit representations of deformable objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3583–3590.

[7] X. Han, J. Masterjohn, and A. Castro, "A convex formulation of frictional contact between rigid and deformable bodies," *IEEE Robotics and Automation Letters*, 2023.

[8] A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 173–180.

[9] A. Byravan, F. Leeb, F. Meier, and D. Fox, "Se3-pose-nets: Structured deep dynamics models for visuomotor control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3339–3346.

[10] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, "Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 1038–1049. [Online]. Available: https://proceedings.mlr.press/v205/seita23a.html

[11] K. R. Allen, T. L. Guevara, Y. Rubanova, K. Stachenfeld, A. Sanchez-Gonzalez, P. Battaglia, and T. Pfaff, "Graph network simulators can learn discontinuous, rigid contact dynamics," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 1157–1167. [Online]. Available: https://proceedings.mlr.press/v205/allen23a.html

[12] S. Pfrommer, M. Halm, and M. Posa, "Contactnets: Learning discontinuous contact dynamics with smooth, implicit representations," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 2279–2291. [Online]. Available: https://proceedings.mlr.press/v155/pfrommer21a.html

[13] R. Calandra, S. Ivaldi, M. P. Deisenroth, E. Rueckert, and J. Peters, "Learning inverse dynamics models with contacts," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3186–3191.

[14] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *International Conference on Learning Representations*, 2019.

[15] M. J. V. der Merwe, Y. Wi, D. Berenson, and N. Fazeli, "Integrated Object Deformation and Contact Patch Estimation from Visuo-Tactile Feedback," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[17] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[18] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu