

Rapid Adaptation of Particle Dynamics for Generalized Deformable Object Mobile Manipulation

Bohan Wu, Roberto Martín-Martín*, and Li Fei-Fei*



Fig. 1: RAPiD: Rapid Adaptation of Particle Dynamics (Left) is a method for learning to perform real-world deformable object mobile manipulation tasks by inferring and adapting to the unknown dynamics of deformable objects in real-time. *Top Right:* In simulation, RAPiD first learns a visuomotor policy to achieve a deformable object mobile manipulation task, such as bowl-covering, by inferring the deformable object’s dynamics from the robot’s visual observations and actions. *Bottom Right:* RAPiD then deploys the learned visuomotor policy directly to the real world to achieve the deformable object mobile manipulation task, under **unseen object dynamics, instances, categories, and lighting conditions**, using only onboard sensor signals.

Abstract—We address the challenge of learning to manipulate deformable objects with unknown dynamics. In non-rigid objects, the dynamics parameters define how they react to interactions –how they stretch, bend, compress, and move– and they are critical to determining the optimal actions to perform a manipulation task successfully. In other robotic domains, such as legged locomotion and in-hand rigid object manipulation, state-of-the-art approaches can handle unknown dynamics using *Rapid Motor Adaptation (RMA)*. Through a supervised procedure in simulation that encodes each rigid object’s dynamics, such as mass and position, these approaches learn a policy that conditions actions on a vector of latent dynamic parameters inferred from sequences of state-actions. However, in deformable object manipulation, the object’s dynamics not only include its mass and position, but also how the shape of the object changes. Our key insight is that the recent ground-truth particle positions of a deformable object in simulation capture changes in the object’s shape, making it possible to extend RMA to deformable object manipulation. This key insight allows us to develop RAPiD, a two-phase method that learns to perform real-robot deformable object mobile manipulation by: 1) learning a visuomotor policy conditioned on the object’s dynamics embedding, which is encoded from the object’s privileged information in simulation, and 2) learning to infer this embedding using non-privileged information instead, such as robot visual observations and actions, so that the learned policy can transfer to the real world. On a 22-DOF robot, RAPiD enables 80%+ success rates across two real-world vision-based deformable object mobile manipulation tasks, under unseen object dynamics, categories, and instances. More details are at <https://sites.google.com/view/rapid-robotics>.

* Equal Advising. Authors are with Stanford University and the University of Texas at Austin, USA. {bohanwu, feifeili}@stanford.edu, robertomm@utexas.edu

I. INTRODUCTION

Deformable objects, such as cables and clothes, can change in shape. As a result, the optimal strategies for deformable object manipulation often rely heavily on the object’s dynamics, such as material properties, which are unknown a priori. For example, consider covering the top of a bowl with a 2D deformable object, such as a hat or a piece of cloth. For a hat, thanks to its high rigidity, a simple pick-and-place motion would be sufficient. But for a cloth, because of its softness, a fling followed by a sweep-and-cover motion is necessary. However, the dynamics of deformable objects are not fully observable a priori, and we as humans adapt to their dynamics in real-time *during* our interaction with them. As we grasp and lift a hat or a cloth, we swiftly infer its material properties and adjust our placing motion or flinging speed to cover the bowl entirely. This rapid adaptation of unknown dynamics enables humans to efficiently manipulate deformable objects across diverse dynamics, categories, and instances, and we hope to endow robots with similar capabilities.

While many existing solutions can perform various deformable object manipulation tasks, generalizing these solutions to in-the-wild settings across diverse deformable object dynamics, categories, and instances remains challenging. For example, state estimation methods model deformable objects using state representations such as particles [1, 2], edges [3, 4], and graphs [5, 6], but they require full object observability and therefore cannot handle object occlusions. In parallel, a

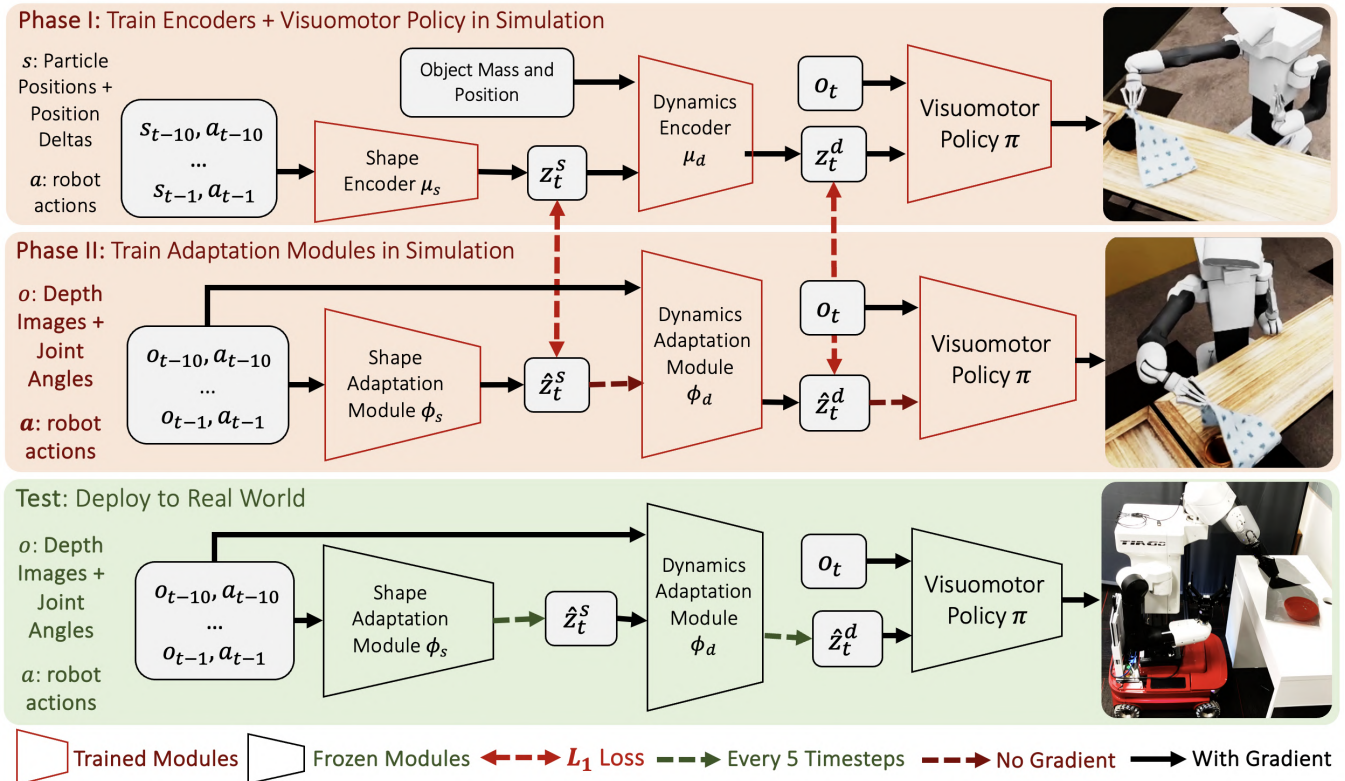


Fig. 2: The RAPiD Method. Below we explain training (*top and middle*) in simulation and deploying in the real world (*bottom*) a deformable object mobile manipulation solution with RAPiD. Training with RAPiD (*top and middle*) is a two-step procedure. In the first step (*top*), the agent is asked to solve a simulated version of the task by learning a visuomotor policy and two encoders: a Shape Encoder, μ_s , and a Dynamics Encoder, μ_d , using privileged state information of the object from the simulator, such as mass and particle positions. In the second step (*middle*), RAPiD frees itself from privileged simulator information by learning a new Shape Adaptation module, ϕ_s , and Dynamics Adaptation module ϕ_d , using L1-losses to replace the Shape Encoder and the Dynamics Encoder, respectively. At test time (*bottom*), both the Shape and Dynamics adaptation modules and the visuomotor policy are directly deployed to the real world, with the adaptation modules updating the Shape Embedding \hat{z}_t^s and Dynamics Embeddings \hat{z}_t^d to the visuomotor policy once every 5 timesteps.

system identification method such as [7] tries to estimate the deformable object’s physics parameters from real-robot trajectories and then fine-tune the manipulation strategy in simulation, but this method requires multiple trajectories and therefore cannot adapt to unknown dynamics in real-time. Prior methods have also attempted to learn generalizable deformable object manipulation from large-scale cross-embodiment robotics datasets [8, 9], but due to their limited scale, zero-shot generalization to a new robot embodiment remains challenging. Finally, other methods have also applied Rapid Motor Adaptation [10] to other robotics tasks such as legged locomotion [11], in-hand manipulation [12], and simulated robotic manipulation [13]. However, applying RMA to deformable object mobile manipulation presents an added complexity: the dynamics of the deformable object includes not only its mass and position, but also how the shape of the object changes.

In this work, our key insight is that the recent ground-truth particle positions of a deformable object in simulation capture changes in the object’s shape, making it possible to extend the RMA framework to learn to manipulate deformable objects with unknown dynamics. This key insight allows us to create a novel two-phase method for learning to manipulate real-world deformable objects by inferring and

adapting to their unknown dynamics: First, in a simulated version of the task, we train a visuomotor policy that is conditioned on a dynamics embedding of privileged simulator information, such as the recent ground-truth particle positions as well as the mass and position of the deformable object. Second, we learn a dynamics adaptation module in simulation to predict the dynamics embedding learned in the first step, using only non-privileged information, such as the robot’s recent visual observations and actions. At test time, since the dynamics adaptation module and the visuomotor policy are learned in simulation using only onboard visual sensor signals, we can directly transfer them to a real robot to accomplish a deformable object mobile manipulation task. We call this method **RAPiD** from **R**apid **A**daptation of **P**article **D**ynamics (Fig. 1).

We evaluated RAPiD in two real-robot deformable object mobile manipulation tasks involving 1D and 2D deformable objects with unseen dynamics, categories, and instances: 1D_Inserting and 2D_Covering. RAPiD significantly outperforms existing state-of-the-art methods by 65%+ success rates in both tasks based only on onboard visual inputs. This indicates that RAPiD’s online adaptation algorithm provides a general and versatile mechanism for learning to manipulate deformable objects with unknown dynamics in

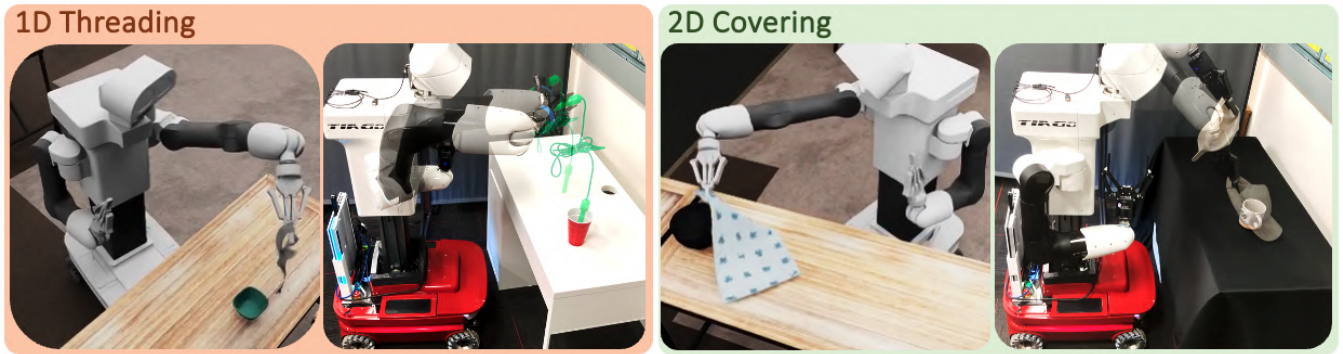


Fig. 3: Simulated and Real-World Version of the Two Tasks: 1D.Inserting (*Left Half*) and 2D.Covering (*Right Half*). Each task is trained in simulation (*left image of each half*) and then deployed directly to the real world (*right image of each half*) under unseen objects, environments, and lighting conditions. By interacting with deformable objects across a broad range of instances, geometries, and categories in simulation, RAPiD successfully learns a visuomotor policy that adapts to and excels at manipulating real-world deformable objects of unseen dynamics, instances, and categories on a real bimanual mobile manipulator.

simulation and then transferring the learned policies to the real world across a wide range of unseen deformable object dynamics, categories, and instances.

II. RELATED WORK

Model-Based Deformable Objects Manipulation. To learn deformable object manipulation with unknown dynamics, previous works have modeled deformable objects using various state representations, such as particle systems [1, 2, 14], points [4, 15–17], graphs [3–6], and differentiable simulators [2, 18–21]. They then learn or plan manipulation strategies using demonstrations [18, 22], annotations [23], or simulation [2, 24]. However, because state estimation can fail under hand-eye coordination or object occlusions [18], generalizing this representation to handle unseen object dynamics, categories, and instances in the real world remains challenging. In contrast, RAPiD is a model-free method that learns to infer a dynamics embedding from recent robot visual observations and rapidly adapts to unknown deformable object dynamics in the real world without requiring state estimation. This allows RAPiD to perform real-robot mobile manipulation tasks that require hand-eye coordination and robustness against object occlusions, such as the 1D.Inserting and 2D.Covering mobile manipulation tasks in Sec. IV.

Rapid Motor Adaptation. Our goal is to achieve rapid adaptation to unknown dynamics of deformable objects, therefore we revise previous works that apply Rapid Motor Adaptation [10] to other robotic tasks, such as quadruped locomotion [25–28], bipedal locomotion [11, 29], in-hand manipulation [12], and simulated robotic manipulation [13]. These methods first identify the task-relevant dynamics properties, such as mass and friction, obtain and encode the values of these properties from the simulator into a dynamics embedding, and learn a rapid adaptation module to infer this embedding from robot observations and actions. However, effective deformable object manipulation requires not only rapid inference and adaptation to the object’s mass and position, but also how the shape of the objects changes, an element missing from these methods. To address this

challenge, we revise RAPiD to infer a dynamics embedding of both the deformable object’s mass and position and the shape changes of the object, from its ground-truth physics parameters and recent particle positions in simulation, respectively, and then learns to infer this dynamics embedding from robot visual observations and actions instead. This allows RAPiD to learn a visuomotor policy in simulation, transfer it to a real-robot mobile manipulator, and manipulate real-world deformable objects using only onboard sensors.

Learning Deformable Object Manipulation from Real-World Data. To learn generalizable deformable object manipulation, prior methods have collected real-world data such as random trajectories [30], human demonstrations [31–33], human annotations [34], and large-scale teleoperation data [35] for both policy pre-training and fine-tuning [8, 9, 36, 37]. In particular, the last family first bootstraps learning from internet-scale vision-language pre-training and crowd-sourced cross-embodiment robotics datasets [38], before fine-tuning on robot-specific datasets to improve target-embodiment performance. While such large-scale real-world data collection methods have demonstrated impressive task complexity, they currently do not zero-shot generalize to new robot embodiments such as the TIAGo mobile manipulator used in our experiments, and whether it can collect enough data to achieve such zero-shot generalization remains to be seen. As a less labor-intensive alternative to this family of algorithms, RAPiD learns deformable object mobile manipulation purely in simulation instead and zero-shot transfers to a vision-based mobile manipulator while generalizing across unseen object dynamics, categories, and instances.

Learning Deformable Object Manipulation from Simulation. To learn deformable object manipulation from simulation and transfer learned policies to the real world, previous works have adopted various strategies such as domain randomization [39–42], domain adaptation [7], real2sim [43], and dense object descriptors [44, 45]. However, these methods do not include multiple visual observations and robot actions as input into their manipulation policy and therefore cannot infer and adapt to deformable object dynamics online. In particular, although Chebotar et al. [7] proposes to adjust



Fig. 4: All unseen real-world deformable objects used for the 1D.Inserting (left) and 2D.Covering (right) mobile manipulation tasks, respectively. **1D categories (left):** belt, hammer, marker, banana, HDMI adapter, GPU connector, jumping rope, ethernet cable, zip tie, rake, mouse, screwdriver, stick, rubber, velvet rope, VGA adapter, anchor rope, garden rope, ribbon, fan. **2D categories (right):** polyester bag, pants, towel, plastic bag, face mask, paper bag, glove, shorts, cap, lid, sock, soft cuff, cardboard, envelope, pouch, wallet, eye-mask, t-shirt, sponge, wool hat. Learning from deformable objects across a wide range of instances in simulation enables RAPiD to zero-shot transfer to real-world objects under unseen dynamics, instances, and categories.

simulation parameters based on real-world experience, this method requires multiple real-world trajectories for adaptation training. It therefore cannot rapidly adapt to unknown dynamics in real-time. In comparison, RAPiD leverages privileged simulator information to infer deformable object dynamics but then learns to infer them using robot visual observations and actions instead. This allows RAPiD to rapidly adapt to a wide range of dynamics across real-world deformable objects, from soft objects (e.g., ribbons, cloths) to stiff items (e.g., cables, wallets).

III. RAPiD:

RAPiD ADAPTATION OF PARTICLE DYNAMICS

As shown in Fig. 2, RAPiD is a two-phase versatile method for learning to manipulate deformable objects in the real world by adapting to their unknown dynamics using raw visual inputs. In the first phase, RAPiD learns a visuomotor mobile manipulation policy conditioned on two dynamics encoders using privileged ground-truth information obtained from simulation, such as ground-truth particle positions and position deltas of the deformable object (Fig. 2, top). In the second phase, RAPiD replaces the shape and dynamics encoders with two respective adaptation modules that only accept recent depth images and robot actions as input, making the entire algorithm free of privileged simulator information (Fig. 2, middle). At test time, RAPiD freezes both the visuomotor policy and the two adaptation modules and deploys them directly to the real world (Fig. 2, bottom).

RAPiD addresses both training phases as reinforcement learning (RL) problems which are modeled as Markov decision processes defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathbb{T}, \mathcal{R}, \gamma, \rho_0, \mathcal{H} \rangle$. \mathcal{S} is the state space (observable only in simulation). \mathcal{O} is the space of observations. \mathcal{A} is the action space. \mathbb{T} is the dynamics model that governs the state transitions, $\mathbb{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. \mathcal{R} is a reward function. $\gamma \in [0, 1)$ is a discount factor. \mathcal{H} is a finite horizon. ρ_0 is the distribution of initial states. Reinforcement learning aims to train a policy, π , that takes optimal actions to maximize

the expected cumulative future rewards across horizon \mathcal{H} . Below, we describe each training and test phase in detail.

Train Phase I: Learning a visuomotor policy in simulation conditioned on two ground-truth encoders.

In the first phase, RAPiD trains a visuomotor policy, π , that generates full-DOF robot actions to achieve the deformable object mobile manipulation task in simulation (Fig. 2, top). Concretely, the policy accepts a single depth image o_t and a Dynamics Embedding z_t^d as input. The Dynamics Embedding is encoded by a Dynamics Encoder μ_d , which accepts the masses and positions of objects in simulation, and a Shape Embedding z_t^s as input. As such, this Dynamics Encoder encodes dynamics information crucial to achieving the task. The Shape Embedding is, in turn, encoded by a Shape Encoder μ_s , which accepts ground-truth object shape information as input, such as recent positions of the deformable object’s particles and corresponding robot actions. As such, this Shape Encoder encodes information about how the shape of the object changes, which is crucial for achieving the task. We learn to encode dynamics and shape information in two separate encoders to quantify the dynamics vs. shape adaptation module’s training losses we will encounter in the second phase. Because the dynamics and shape information are represented by low-dimensional privileged information from the simulator instead of high-dimensional visual observations, we train both the policy and the dynamics and shape encoders end-to-end using RL.

Once RAPiD has trained the visuomotor policy and the shape and dynamics encoders end-to-end, it learns two new adaptation modules—a shape adaptation module and a dynamics adaptation module—to replace the shape and dynamics encoders, respectively, which we discuss next.

Train Phase II: Learning shape and dynamics adaptation modules in simulation.

For RAPiD to work in the real world, its policy, shape encoder, and dynamics encoder cannot accept any privileged ground-truth simulator information as input. Therefore, in this second training phase, the shape and dynamics encoders are replaced with their respective

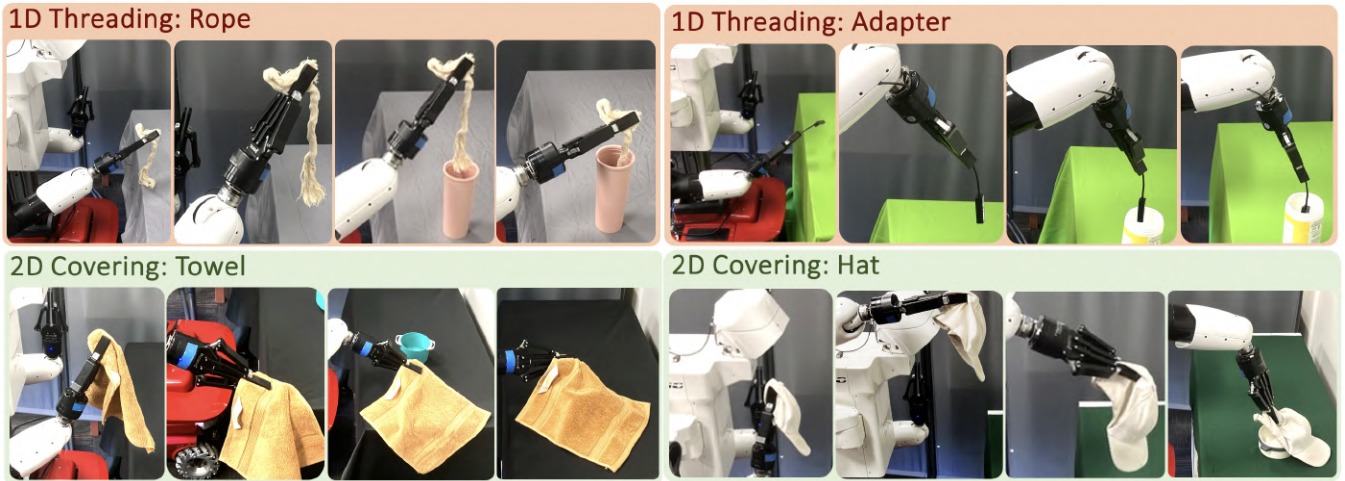


Fig. 5: RAPiD’s Dynamics Adaptive Behavior in 1D-Inserting (Top) and 2D-Covering (Bottom) tasks. *Top Left:* RAPiD hangs the rope over the top of the container before lowering it. *Top Right:* RAPiD flips the gripper upside down before performing a direct adapter insertion. *Bottom Left:* RAPiD moves the robot arm sideways and sweeps a towel horizontally from right to center to cover the bowl. *Bottom Right:* RAPiD moves the arm vertically and places a hat directly on top of the bowl. This shows RAPiD’s ability to infer different object dynamics and adapt actions accordingly to achieve the deformable object mobile manipulation tasks.

shape and dynamics adaptation modules that can be trained to infer the same dynamics embedding using only visual observations (Fig. 2, *middle*). Concretely, we first replace the Shape Encoder with a new Shape Adaptation module, ϕ_s , which accepts recent depth images and robot joint angles and actions instead of ground-truth particle positions of the object as input. To train this Shape Adaptation module, we generate the ground-truth Shape Embedding, z_t^s , using the Shape Encoder and the inferred Shape Embedding, \hat{z}_t^s , from the Shape Adaptation module, and use L1-Loss to regress the inferred embedding to the ground-truth embedding. Similarly, we replace the Dynamics Encoder by training a new Dynamics Adaptation module, ϕ_d , which accepts recent depth images and robot joint angles and actions instead of ground-truth dynamics information as input, using L1-loss between the ground-truth Dynamics Embedding z_t^d and the inferred Dynamics Embedding \hat{z}_t^d . To prevent the Shape Adaptation module from encoding redundant dynamics information, we stop upstream gradients of training the Dynamics Adaptation module from flowing into the Shape Adaptation module. Finally, we fine-tune the visuomotor policy obtained during the first phase using the two new embeddings generated by the two new adaptation modules using RL. We stop upstream RL gradients from flowing into both adaptation modules so that the two modules separately encode the object dynamics and shape information— even though they observe the same input— that cannot be directly observed by the depth image o_t . The dynamics and shape encoders from Phase I are discarded before test time deployment, as discuss below.

Test Time Deployment. At test time, the robot is randomly placed in the real world with previously unseen deformable and rigid-body objects arranged randomly on a table. It needs to perform a task that requires using a deformable object with unknown dynamics. For each timestep, the robot interacts with objects in the environment using RAPiD’s visuomotor policy, which accepts the current

robot observation (i.e., a depth image and joint angles) and the Dynamics Embedding \hat{z}_t^d as input and outputs full-DOF robot actions (Fig. 2, *bottom*). To generate the Dynamics Embedding, the robot stores a running buffer of the 10 most recent depth-image-observation-action pairs as input into both the shape and dynamics adaptation modules. To maintain short-term temporal behavioral consistency of the visuomotor policy, RAPiD only periodically updates the shape and dynamics embeddings every 5 timesteps. This cycle repeats until the robot completes the task successfully.

IV. EXPERIMENTS

In our experiments, we use a TIAGo robot (Fig. 3), which is a 22-DOF bimanual mobile manipulator with 3-DOF for the omnidirectional base, 7-DOF for each of the two arms, 1-DOF torso, 2-DOF head, and 1-DOF for each of the two Robotiq parallel-jaw grippers. The robot has an RGB-D camera mounted on its head. We use depth images of 224×224 resolution captured at 3 Hz. In both simulation and the real world, RAPiD’s input is the robot’s depth image and joint angles.

We use OmniGibson [46, 47] as the training simulator for RAPiD. In OmniGibson, A model of the same TIAGo robot is placed in front of a simulated version of the task (Fig. 3). To train the visuomotor mobile manipulation policy, we use a binary reward of 1 for task success and a small reward negatively correlated with the distance between the deformable object and the target rigid-body object on the table. In the real world, the robot is placed in front of an unseen table with unseen deformable and rigid-body objects (Fig. 3). RAPiD then controls the robot’s bimanual mobile manipulation capabilities (base, torso, head, grippers, and arms) to grasp and manipulate the deformable object to achieve the task, using all DOFs simultaneously. All experiments are conducted autonomously without human intervention from raw visual inputs with no QR codes. We

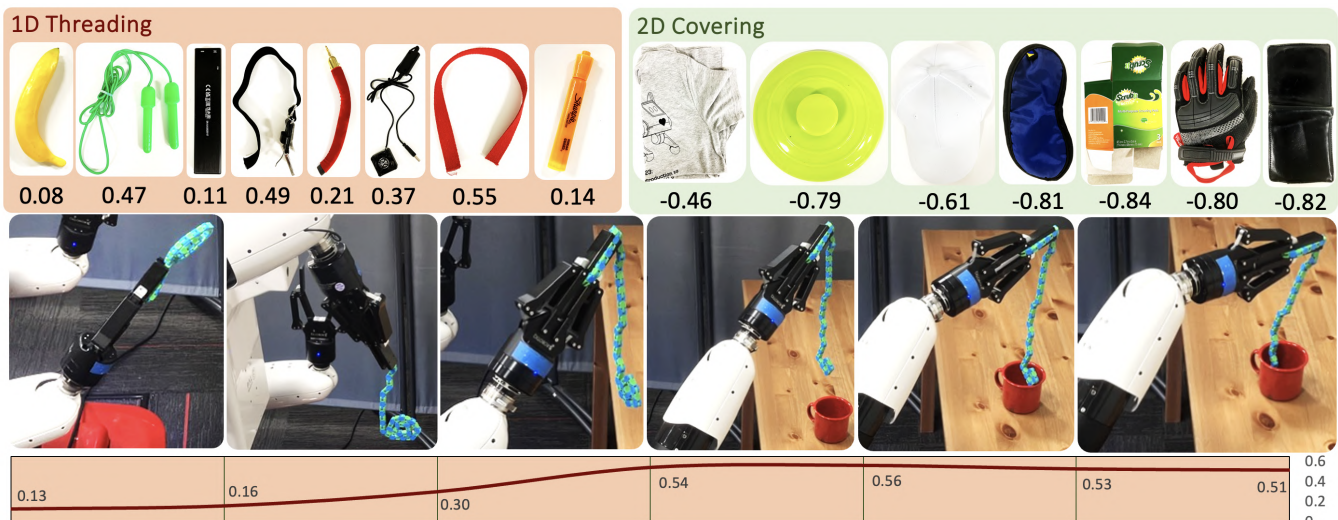


Fig. 6: Visualization of RAPiD’s dynamics embedding in 1D_Inserting (*Top-Left*) and 2D_Covering (*Top-Right*) tasks. The number below each object photo is the value of a single dimension of the dynamics embedding. *Bottom*: the rollout of the policy given a special object whose rigidity changes with rotation. The line charts below the rollout visualize the value of a single dimension of the dynamics embedding during robot execution. We see that certain dimensions of the dynamics embedding are strongly correlated with the softness of the objects, demonstrating the adaptation module’s ability to infer the softness of the objects.

use real-world objects (Fig. 4), environments (Fig. 3), and lighting conditions (Fig. 3) never seen during simulation training. We repeat each task 20 times with different initial locations and objects.

Tasks: We evaluate RAPiD across two challenging deformable-object mobile manipulation tasks that require online adaptation to the deformable object’s unknown dynamics a priori: 1D_Inserting and 2D_Covering. In 1D_Inserting (Fig. 3, *left*), the goal is to pick up a 1D deformable object on the table and insert either one of its two ends into a container. In the real-world version of the task, we place a 1D deformable object randomly on the table, selected across 20 categories such as rope, cables, and belts (Fig. 4, *left*), and a container selected across 20 object categories such as cups, bowls, and pans. Success is defined by whether the robot inserts one end of the deformable object into the container within 300 seconds. In 2D_Covering (Fig. 3, *right*), the goal is to pick up a 2D deformable object on the table and cover the opening of a container on the table—a representative task in household scenarios. In the real-world version of the task, we place a 2D deformable object randomly on the table, sampled across 20 categories such as lids, napkins, towels, and plastic films (Fig. 4, *right*). There is no parameter randomization applied to object pose, material stiffness, or friction, since such randomization is achieved implicitly via object instance randomization. Success is defined by whether the robot successfully covers 90% of the container’s opening within 300 seconds. Achieving both tasks requires closed-loop hand-eye coordination, robustness against object deformation and occlusions, and online adaptation to the unknown dynamics of the deformable object during real-time interaction, as shown in Fig. 5 and 6.

Baselines: We compare RAPiD to two baselines for learning deformable object manipulation: DMfD [43] (Table I, *column 3*) and DDOD [44, 45] (Table I, *column 4*). The first baseline, DMfD, learns deformable object stationary manipulation of 1D and 2D object categories from simulated expert demonstrations. The second baseline, DDOD, learns dense object descriptor representations for 1D [44] and 2D [45] deformable objects and learns deformable object stationary manipulation from a single human demonstration using this representation. To make DDOD comparable to RAPiD, which requires no demonstrations, we script a policy based on the dense object descriptors learned from DDOD to perform deformable object manipulation, instead of providing human demonstrations. To help both baselines overcome failures in grasping the deformable object and acquire navigation capabilities comparable to RAPiD, we use RAPiD to execute the grasping and navigation of the tasks before activating the baseline for the deformable object manipulation part of the tasks. Learning deformable object mobile manipulation with unseen deformable object dynamics, categories, and instances using real-robot imitation learning [8, 9, 36, 37, 48–52] requires collecting an infeasible number of real-world demonstrations on the TIAGO robot, which makes them incomparable to RAPiD, since RAPiD requires no real-robot demonstrations. Extending RAPiD to fine-tune on real-world data to compare to these methods is left for future work.

We also conduct three ablation experiments of RAPiD: 1) RAPiD-*No-Adapt* (Table I, *column 5*), which removes both the Shape and Dynamics Adaptation Modules and trains RAPiD’s visuomotor policy only; 2) *aRAPiD-No-Shape* (Table I, *column 6*), which deletes the Shape Adaptation Module and trains only the visuomotor policy and the Dynamics Adaptation module; and 3) RAPiD-*E2E* (Table I,

TABLE I: Successful Trials Out of 20 and Success Rates for RAPiD, Baselines and Ablations

	RAPiD	DMfD	DDOD	RAPiD-No-Adapt	aRAPiD-No-Shape	RAPiD-E2E
1D_Inserting	17 (85%)	3 (15%)	2 (10%)	7 (35%)	7 (35%)	5 (25%)
2D_Covering	16 (80%)	1 (5%)	5 (25%)	5 (25%)	9 (45%)	4 (20%)
Total	33 / 40 (82.5%)	4 / 40 (10%)	7 / 40 (17.5%)	12 / 40 (30%)	16 / 40 (40%)	9 / 40 (22.5%)

column 7), which ignores both Shape and Dynamics Encoders, skips the first of the two training phases and trains RAPiD with end-to-end RL without the two L1 losses.

In our experiments, we answer four main questions:

Q1: How does RAPiD compare to state-of-the-art sim2real methods in real-robot deformable object mobile manipulation? In Table I (columns 2-4), RAPiD outperforms both baselines significantly across both tasks. Qualitatively, DMfD fails in both 1D_Inserting and 2D_Covering because after grasping, the deformable object suffers from severe deformation and occlusions by the robot gripper during the hand-eye-coordinated motions of inserting and covering, preventing DMfD from generating a sufficiently accurate segmentation mask for the deformable object. In some cases where the segmentation mask is accurate enough, DMfD is still unable to adapt to the unknown dynamics of the deformable object. DDOD fails in 1D_Inserting and 2D_Covering because the deformable object suffers from in-the-wild, non-top-down camera viewpoints, severe deformation, and robot gripper occlusions, making DDOD incapable of predicting accurate dense object descriptors for the deformable object during manipulation. In some cases where the dense object descriptor predictor performs well, DDOD is still unable to adapt to the unknown dynamics of the deformable object. In contrast, RAPiD performs closed-loop 6-DOF control and hand-eye coordination directly from non-top-down depth images and outperforms both DMfD and DDOD.

Q2: How important are RAPiD’s Shape and Dynamics Adaptation Modules to deformable object mobile manipulation? We compare RAPiD (Table I, column 2) to RAPiD-No-Adapt (Table I, column 5), which ablates Shape and Dynamics Adaptation Modules. We observe a 52.5% drop in performance for RAPiD-No-Adapt. Qualitatively, in both 1D_Inserting and 2D_Covering, RAPiD-No-Adapt treats deformable objects as rigid objects, which fails when the object is made of soft materials. Therefore, RAPiD’s Adaptation Modules are crucial.

Q3: How important is RAPiD’s ability to infer the object’s shape changes? We compare RAPiD to aRAPiD-No-Shape (Table I, column 6), which removes the Shape Adaptation Module. We observe that RAPiD encounters a 42.5% success rate degradation compared to our full variant using both the Shape and Dynamics Adaptation Modules. In both 1D_Inserting and 2D_Covering, we observe that aRAPiD-No-Shape is successful at grasping the deformable object and moving it closer to the target container to insert or cover, but unable to execute the precise dynamic motion

needed to insert or cover the container, which requires inferring the deformable object’s shape changes. Thus, the ability to infer how the object’s shape changes is crucial.

Q4: Is optimizing both adaptation modules to mimic their corresponding ground-truth encoders necessary, or is end-to-end RL sufficient for task success? We compare RAPiD to RAPiD-E2E (Table I, column 7), which trains the entire architecture with end-to-end RL. We observe that RAPiD-E2E fails to converge optimally on its RL policy gradients during training and experiences a 60% success rate reduction: due to the high-dimensionality of the 10 most recent depth image observations and unstable RL training gradient signals, RAPiD relies solely on current depth observations to perform the task, ignoring and failing to learn a useful dynamics embedding. Thus, optimizing both adaptation modules using L1-loss in a two-phase training process is crucial.

V. CONCLUSION

We proposed RAPiD, a method for learning to perform real-world deformable object mobile manipulation tasks by inferring and adapting to the unknown dynamics of deformable objects in real-time. RAPiD results from a two-phase training procedure that first infers a deformable object dynamics embedding from the object’s recent ground-truth particle positions, and then infers this same embedding using recent robot visual observations and actions instead. RAPiD achieved success in two deformable object mobile manipulation tasks across unknown deformable object dynamics, categories, and instances.

REFERENCES

- [1] F. Jaillet, B. Shariat, and D. Vandorpe, “Deformable object reconstruction with particle systems,” *Computers & Graphics*, vol. 22, no. 2-3, pp. 189–194, 1998.
- [2] 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.
- [3] D. Triantafyllou, I. Mariolis, A. Kargakos, S. Malassiotis, and N. Aspragathos, “A geometric approach to robotic unfolding of garments,” *Robotics and Autonomous Systems*, vol. 75, pp. 233–243, 2016.
- [4] H. Yuba, S. Arnold, and K. Yamazaki, “Unfolding of a rectangular cloth from unarranged starting shapes by a dual-armed robot with a mechanism for managing recognition error and uncertainty,” *Advanced Robotics*, vol. 31, no. 10, pp. 544–556, 2017.
- [5] A. Longhini *et al.*, “Edo-net: Learning elastic properties of deformable objects from graph dynamics,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 3875–3881.
- [6] M. Lippi *et al.*, “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.

- [7] Y. Chebotar *et al.*, “Closing the sim-to-real loop: Adapting simulation randomization with real world experience,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8973–8979.
- [8] K. Black *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [9] M. J. Kim *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [10] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “Rma: Rapid motor adaptation for legged robots,” *arXiv preprint arXiv:2107.04034*, 2021.
- [11] A. Kumar, Z. Li, J. Zeng, D. Pathak, K. Sreenath, and J. Malik, “Adapting rapid motor adaptation for bipedal robots,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 1161–1168.
- [12] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, “In-hand object rotation via rapid motor adaptation,” in *Conference on Robot Learning*, PMLR, 2023, pp. 1722–1732.
- [13] Y. Liang, K. Ellis, and J. Henriques, “Rapid motor adaptation for robotic manipulator arms,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16404–16413.
- [14] F. Liu, E. Su, J. Lu, M. Li, and M. C. Yip, “Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics,” *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3964–3971, 2023.
- [15] L. Sun, G. Aragon-Camarasa, P. Cockshott, S. Rogers, and J. P. Siebert, “A heuristic-based approach for flattening wrinkled clothes,” in *Towards Autonomous Robotic Systems: 14th Annual Conference, TAROS 2013, Oxford, UK, August 28–30, 2013, Revised Selected Papers 14*, Springer, 2014, pp. 148–160.
- [16] 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.
- [17] R. Jangir, G. Alenya, and C. Torras, “Dynamic cloth manipulation with deep reinforcement learning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 4630–4636.
- [18] S. Chen, Y. Xu, C. Yu, L. Li, and D. Hsu, “Differentiable particles for general-purpose deformable object manipulation,” *arXiv preprint arXiv:2405.01044*, 2024.
- [19] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos, “Disect: A differentiable simulation engine for autonomous robotic cutting,” *arXiv preprint arXiv:2105.12244*, 2021.
- [20] C. Schenck and D. Fox, “Spnets: Differentiable fluid dynamics for deep neural networks,” in *Conference on Robot Learning*, PMLR, 2018, pp. 317–335.
- [21] Z. Xian *et al.*, “Fluidlab: A differentiable environment for benchmarking complex fluid manipulation,” *arXiv preprint arXiv:2303.02346*, 2023.
- [22] B. Balaguer and S. Carpin, “Combining imitation and reinforcement learning to fold deformable planar objects,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 1405–1412.
- [23] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, “Iterative residual policy: For goal-conditioned dynamic manipulation of deformable objects,” *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 389–404, 2024.
- [24] 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.
- [25] Z. Fu, X. Cheng, and D. Pathak, “Deep whole-body control: Learning a unified policy for manipulation and locomotion,” in *Conference on Robot Learning (CoRL)*, 2022.
- [26] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, “Coupling vision and proprioception for navigation of legged robots,” in *CVPR*, 2022.
- [27] Z. Fu, A. Kumar, J. Malik, and D. Pathak, “Minimizing energy consumption leads to the emergence of gaits in legged robots,” in *Conference on Robot Learning (CoRL)*, 2021.
- [28] Z. Zhuang *et al.*, “Robot parkour learning,” in *Conference on Robot Learning (CoRL)*, 2023.
- [29] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science robotics*, vol. 5, no. 47, eabc5986, 2020.
- [30] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, “Learning robotic manipulation through visual planning and acting,” *arXiv preprint arXiv:1905.04411*, 2019.
- [31] D. Seita *et al.*, “Deep transfer learning of pick points on fabric for robot bed-making,” in *The International Symposium of Robotics Research*, Springer, 2019, pp. 275–290.
- [32] D. Seita *et al.*, “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.
- [33] R. P. Joshi, N. Koganti, and T. Shibata, “Robotic cloth manipulation for clothing assistance task using dynamic movement primitives,” in *Proceedings of the 2017 3rd International Conference on Advances in Robotics*, 2017, pp. 1–6.
- [34] W. H. Lui and A. Saxena, “Tangled: Learning to untangle ropes with rgb-d perception,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 837–844.
- [35] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, “Learning force-based manipulation of deformable objects from multiple demonstrations,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 177–184.
- [36] O. M. Team *et al.*, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.
- [37] T. Z. Zhao *et al.*, “Aloha unleashed: A simple recipe for robot dexterity,” *arXiv preprint arXiv:2410.13126*, 2024.
- [38] A. O’Neill *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 6892–6903.
- [39] J. Matas, S. James, and A. J. Davison, “Sim-to-real reinforcement learning for deformable object manipulation,” in *Conference on Robot Learning*, PMLR, 2018, pp. 734–743.
- [40] J. Hietala, D. Blanco-Mulero, G. Alcan, and V. Kyrki, “Closing the sim2real gap in dynamic cloth manipulation,” *arXiv preprint arXiv:2109.04771*, vol. 2, 2021.
- [41] 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.
- [42] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, “Learning to manipulate deformable objects without demonstrations,” *arXiv preprint arXiv:1910.13439*, 2019.
- [43] 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.
- [44] P. Sundaresan *et al.*, “Learning rope manipulation policies using dense object descriptors trained on synthetic depth data,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 9411–9418.
- [45] A. Ganapathi *et al.*, “Learning dense visual correspondences in simulation to smooth and fold real fabrics,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11515–11522.
- [46] C. Li *et al.*, “Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation,” in *Conference on Robot Learning*, PMLR, 2023, pp. 80–93.
- [47] C. Li *et al.*, “Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation,” *arXiv preprint arXiv:2403.09227*, 2024.
- [48] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.
- [49] A. Nair *et al.*, “Combining self-supervised learning and imitation for vision-based rope manipulation,” in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 2146–2153.
- [50] A. Brohan *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023.
- [51] A. Brohan *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [52] S. Liu *et al.*, “Rdt-1b: A diffusion foundation model for bimanual manipulation,” *arXiv preprint arXiv:2410.07864*, 2024.