

Curriculum Is More Influential Than Haptic Information During Reinforcement Learning of Object Manipulation Against Gravity

Pegah Ojaghi,¹⁺ Romina Mir,²⁺ Ali Marjaninejad,^{2,3} Andrew Erwin,^{2,4,5}
Michael Wehner,⁶ Francisco J Valero-Cuevas^{2,3,4,7,8*}

¹Computer Science and Engineering Department, University of California Santa Cruz,
Santa Cruz, California, USA

²Department of Biomedical Engineering, University of Southern California,
Los Angeles, California, USA

³Ming Hsieh Department of Electrical and Computer Engineering,
University of Southern California, Los Angeles, California, USA

⁴Division of Biokinesiology and Physical Therapy, University of Southern California,
Los Angeles, California, USA

⁵Mechanical and Materials Engineering Department, University of Cincinnati,
Cincinnati, Ohio, USA

⁶Mechanical Engineering Department, University of Wisconsin-Madison,
Madison, Wisconsin, USA

⁷Department of Aerospace & Mechanical Engineering, University of Southern California,
Los Angeles, CA, USA

⁸Department of Computer Science, University of Southern California,
Los Angeles, California, USA

*To whom correspondence should be addressed; E-mail: valero@usc.edu.

+These authors contributed equally to this work.

Learning to lift and rotate objects with the fingertips is necessary for autonomous in-hand dexterous manipulation. In our study, we explore the impact of various factors on successful learning strategies for this task. Specifically, we investigate the role of curriculum learning and haptic feedback in enabling the learning of dexterous manipulation. Using model-free Reinforcement Learning, we compare different curricula and two haptic information modalities (No-tactile vs. 3D-force sensing) for lifting and rotating a ball against gravity with a three-fingered simulated robotic hand with no visual input. Note that our best results were obtained when we used a novel curriculum-based learning rate scheduler, which adjusts the linearly-decaying learning rate when the reward is changed as it accelerates convergence to higher rewards. Our findings demonstrate that the choice of curriculum greatly biases the acquisition of different features of dexterous manipulation. Surprisingly, successful learning can be achieved even in the absence of tactile feedback, challenging conventional assumptions about the necessity of haptic information for dexterous manipulation tasks. We demonstrate the generalizability of our results to balls of different weights and sizes, underscoring the robustness of our learning approach. This work, therefore, emphasizes the importance of the choice curriculum and challenges long-held notions about the need for tactile information to autonomously learn in-hand dexterous manipulation.

Introduction

Dexterous manipulation is a triumph of biology (1–8). However, the autonomous learning of such behavior continues to remain out of reach for robots (4, 9–12). Robots have excelled at *grasping* (reaching for and statically coupling an object to the hand by applying forces with the fingertips, fingers, and palm (4, 6, 13, 14)) for decades (e.g., (15–23)), but grasp is not dexterous

manipulation (4). *Dexterous in-hand manipulation* (i.e., dynamically holding and reorienting an object with the fingertips (4, 18, 24, 25)) is critical for interaction with, and use of, objects in unstructured human environments.

To achieve this kind of manipulation with multi-fingered robotic hands, the robotics community has developed sophisticated control theoretical approaches ¹ (e.g., (4, 7, 26–33)). These control theoretical approaches, however, tend to require accurate models and state estimation, have narrow stability margins, and have difficulty compensating for friction, interpreting intermittent/deformable contact, and coordinating between multiple fingers. As an alternative approach, biorobotic, neuromechanics, and artificial intelligence communities have introduced a variety of bio-inspired and data-driven machine-learning approaches (for reviews see (4, 11, 14, 34, 35)) in simulation and hardware.

One particularly promising approach is the sub-field of Reinforcement Learning (RL), which has provided several successful examples (12, 23, 36–39). RL empowers robots to iteratively enhance their manipulation skills through trial and error (without of a need of an accurate model of the task or the environment), resulting in gradual improvements within complex environments. However, manipulation RL studies to date are usually highly computationally intensive—and have relied on vision—which limits their applicability (28, 35, 40–49). Lastly, most studies have been limited to the upward-facing hand configuration, relying on the palm as a resting platform for the object being manipulated which makes it an inherently more stable task to handle than a down-facing hand configuration (9). Adding the downward-facing hand configuration broadens the scope of solutions, delivering valuable insight to the robot manipulation community (9, 12, 50). However, it introduces additional challenges as this orientation requires the hand to counteract gravity at all times (51), and errors can lead to instabilities and failure by dropping the object. Here we use an RL based on the Proximal Policy Optimization (PPO) (52) algorithm to autonomously learn manipulation with a downward-facing hand without direct vision. We

¹Henceforth we use the shorthand *manipulation* to mean dexterous in-hand manipulation

find that the choice of curriculum biases learning manipulation toward one or another combination of skills (i.e., lifting the ball and/or rotating it) more profoundly than the availability of tactile information.

Surprisingly, the absence of tactile information did not necessarily prevent or significantly degrade learning relative to the influence of curriculum. These results reveal fundamental and previously underappreciated aspects of curricula as a powerful tool for autonomous learning of multi-objective tasks. For example, curricula commencing with both lift and rotation exhibit initial superior performance compared to those building up from simpler blocks, such as focusing solely on lift or rotation. Focusing on a single skill thereafter, however, can be additionally beneficial. Beyond assessing the impact of curricula on autonomous manipulation, our study yielded the significant revelation that, contrary to long-held notions, the absence of tactile information (and direct vision) does not inherently impede or degrade the learning process. In fact, there seems to be a functional interaction with a curriculum where available sensing capabilities bias the learning process toward combinations of dexterous manipulation skills that can leverage the available tactile information.

Results

The goal of this project was to utilize curriculum-based RL to learn in-hand manipulation of an object against gravity in a data-efficient way—even while not using visual information. We demonstrate how the choice of curriculum is more influential than tactile information when learning to lift and rotate a **ball (weighing 50 g with 35 mm radius)** with a three-finger robotic hand in simulation (Fig. 1). To do so, we systematically explored **two tactile conditions**: No-tactile (no force perception at all at the fingertip) vs. 3D-force (a 3D force vector in the direction of force at the fingertip) during **five distinct curricula** (details in Methods). We defined each curriculum as implementing a learning policy that rewards various combinations and sequences of lift (**L**) and rotation (**R**) of a ball, which can switch at the halfway point (Methods, Table 1).

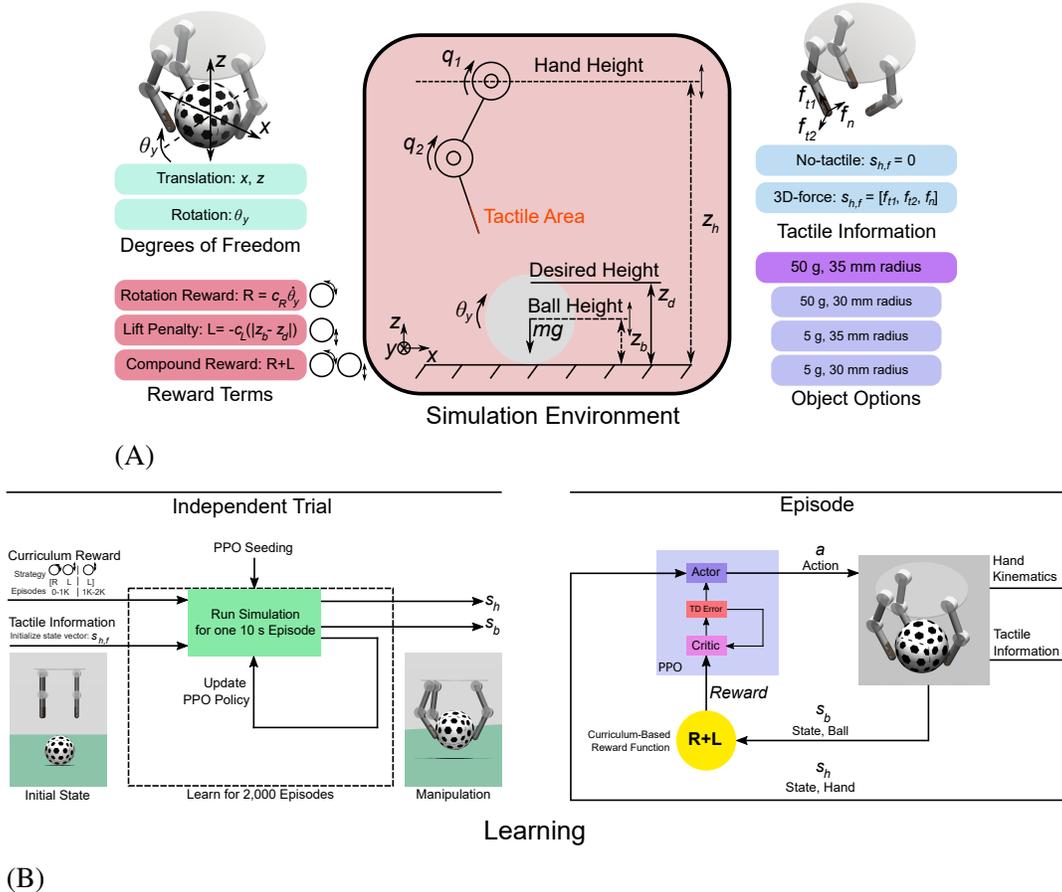


Figure 1: Overview of Simulation Environment and Learning. High-level overview of the simulation environment and learning approach to autonomous manipulation. See the Methods section for further details. **A: Simulation Environment.** A simulated three-finger robotic hand attempted to lift and rotate (i.e., dexterously manipulate) a ball. The 3D movement of the ball was lightly constrained to the X-Z plane. Changes in the ball state affect the reward, which is a function of rotation, lift, and/or a combination of the two. We tested this approach with two different tactile information conditions (No-tactile and 3D-force) available at the fingertips and four balls of different weights and sizes. **B: Learning Algorithm.** *Independent Trial, Left:* For each of the five curricula, autonomous learning was evaluated over 60 independent trials (one trial shown). Each trial in a curriculum consisted of two learning phases lasting 1,000 episodes for a total of 2,000 episodes. The reward function changed at the end of the first learning phase (with the exception of Curriculum 3, see Table 1). *Episode, Right:* Each episode lasted 10 s and began *de novo* with the ball on the ground with the hand and fingertips suspended above it. In each episode, the PPO learning algorithm dynamically updates the agent’s action (i.e., moving the fingers and hand) to increase the curriculum’s reward.

For example, Curriculum 1 (i.e., C1) only rewards lift (L) in the first half of the trial, and both lift and rotation (L+R) in the second half are described as [L|L+R]). Lastly, we confirmed the generalizability of our approach by learning to manipulate balls (objects) of different weights

and sizes (Fig. 5). We find that the order of reward (curriculum) greatly affects the progression of learning and the final performance, 3D tactile information was not consistently better than No-tactile information, and a similar trend was observed across all configurations (see the video file in Supplementary Information).

Curriculum profoundly affects the progression of learning and final performance

Each combination of curriculum and tactile information (Methods, Table 1) leads to a distinct evolution of learning and final performance. This effect of curriculum affects both the progression of learning (path) and final performance (endpoint), and can be visualized as traversing a developmental process (as ‘Waddington Landscapes’ in biology, Fig. 2, see Discussion).

Curricula, as expected, diverge in their ability to lift and rotate the ball. In fact, they had the profound effect of biasing toward one or another combination of skills (L or R) and also adapt to the available sensory input, much like experience-dependent developmental paths from an initial pluripotent state (Fig. 2C). As we describe in detail in the Discussion section, we explicitly explored *different* initial rewards with *similar* final rewards (C1 [L|L+R] vs. C2 [R|L+R]), and vice versa (C4 [L+R|R] vs. C5 [L+R|L]). In all cases, the system was able to respond to the change in reward (albeit with variable success). Note the evolution of skills for each curriculum tended to saturate quickly within the first 250 episodes of the first and second phases of learning. They tended to asymptote between the 250 and 1,000, and between the 1,250 and 2,000 episodes, respectively. Nevertheless, the final endpoints for each curriculum differed significantly, showing that curricula are more than simply a means to learn multi-objective tasks, but can actually produce different learning paths and endpoints—which can be exploited by the user to achieve different capabilities with the same naïve system (Fig. 2).

Counterintuitively, starting with a multi-objective reward can be as effective, if not more effective, than starting with simpler rewards. For example, rewarding *both lift and rotation*

during the first 1,000 episodes (C3 [L+R|L+R] , C4 [L+R|R] , and C5 [L+R|L]) improves rotating the ball at the end of learning (episode 2,000) better than when only rewarding rotation (C2 [R|L+R]) at the start.

Tactile information is not necessary but can affect learning

Most surprisingly, the absence of tactile information did not preclude learning. Moreover, learning with No-tactile information was comparable to the 3D-force information (Fig. 2). The presence or absence of 3D-force information did, however, change the learning paths and endpoints of each curriculum (Fig. 2, 3)—but the effect was not uniform. For example, 3D-force information did produced more lifting than No-tactile in C1 [L|L+R] at the end of learning. But this was reversed in C3 [L+R|L+R] ; and tactile information did not affect C4 [L+R|R] or C5 [L+R|L] much (Fig. 2). This nuanced effect of tactile information at the end of learning is also seen in Fig. 3, and on average during learning in Fig. 4. This interaction was also seen while learning with different objects (see details in the Generalizability section and Fig. 5).

Further nuance of the effect of tactile information can be seen in the different paths of learning and in the response to switching of rewards between the first and second learning phases (i.e., after episode 1,000). Note C3 [L+R|L+R] rewards both skills during the entirety of both phases, but tends to be most effective at lifting in the No-tactile condition compared to 3D-force condition, Fig. 2. Nevertheless, when switching the reward to only lift C5 [L+R|L] or only rotation C4 [L+R|R] at the end of the first learning phase, the 3D-force case makes up for lost ground and has endpoints similar to those for the No-tactile condition. This effect seems to be reversed for C1 [L|L+R] and C2 [R|L+R] where only lift or rotation were rewarded at first. In these cases, the 3D-force condition produced greater lift and rotation during both learning phases.

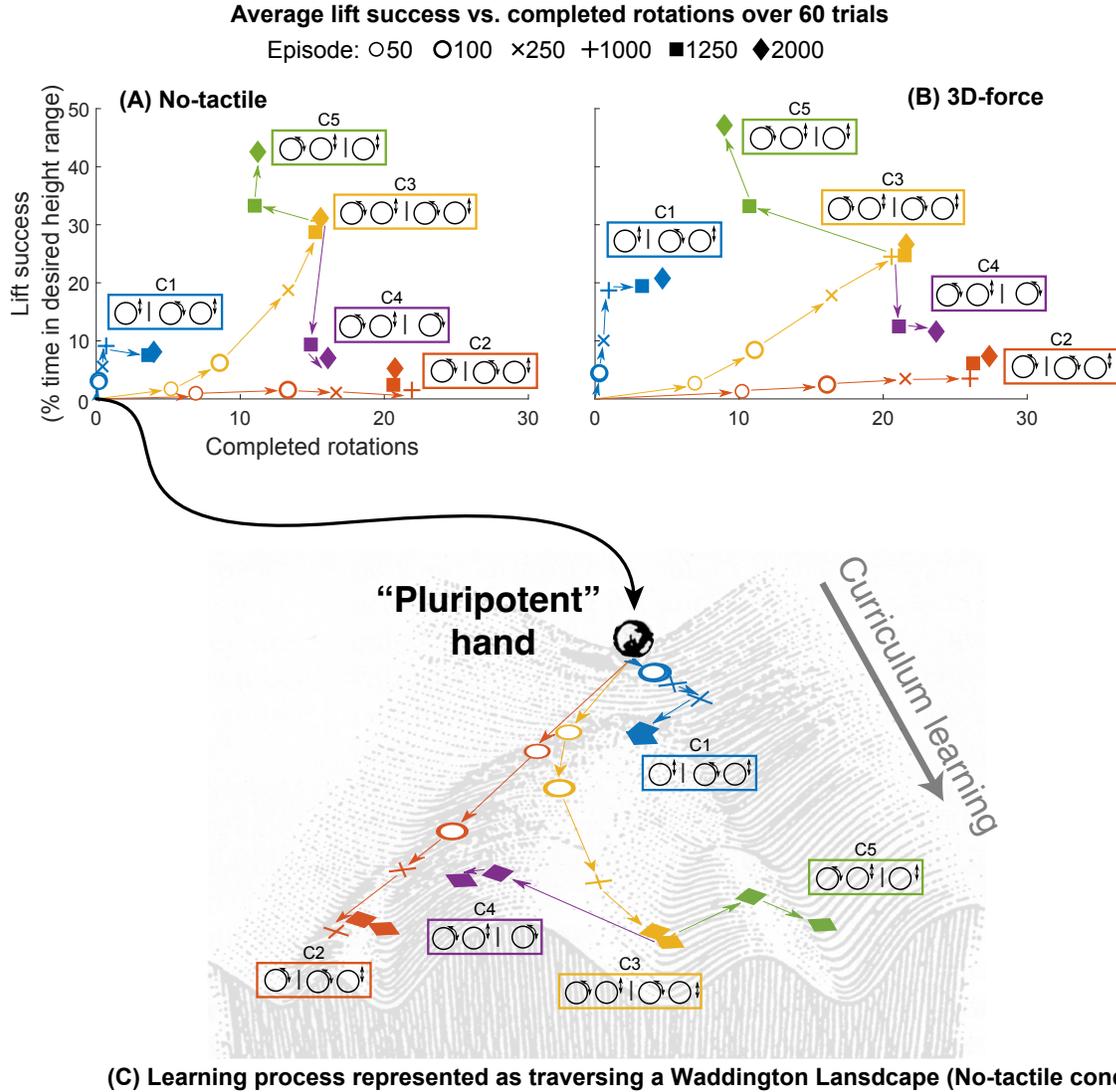


Figure 2: The evolution of learning highlights the dynamic functional interaction between curriculum and tactile information. Manipulation performance during the last 10s of each episode noted: the percent of the time the ball is within the desired height range vs. number of complete rotations. Each point is the average of 60 independent trials. Arrows point in the direction of increasing episodes. Negative rotations were set to zero. Note that the choice of curriculum had a profound effect on learning for both tactile conditions ((A) No-tactile and (B) 3D-force). Surprisingly, learning happened even in the absence of tactile information, and manipulation performance was not always better with 3D-force information. (C) An analogy of learning as a developmental trajectory from a pluripotent state based on experience (curriculum). This effect of curriculum (and tactile information, cf. A vs. B) affects both learning (path) and final performance (endpoint), and can be visualized as traversing a ‘Waddington Landscape’ (adapted from (53)).

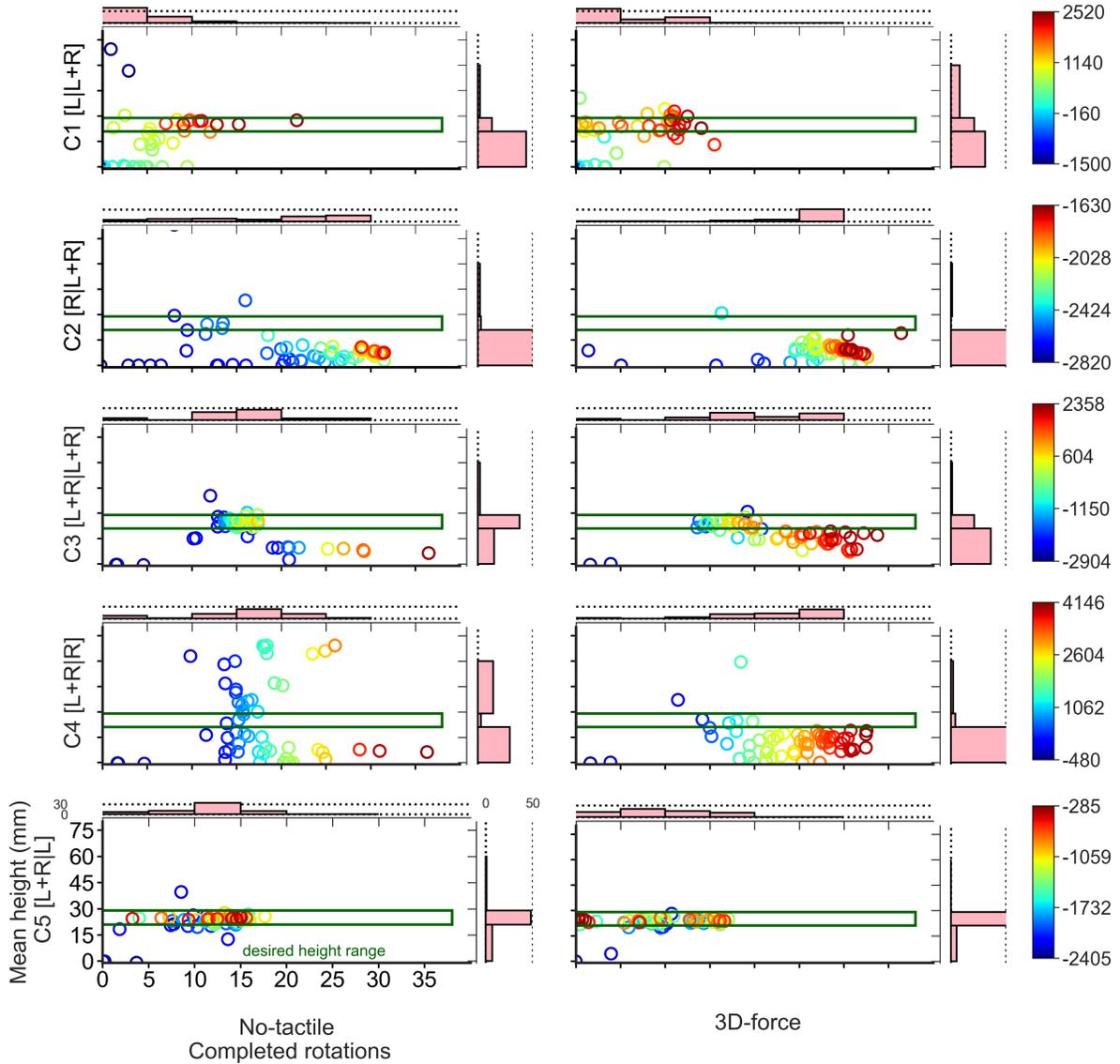


Figure 3: **Performance across all curricula and both tactile information conditions.** The joint distribution illustrates the performance during the final 10s episode of each of the 60 trial runs (showcasing the mean ball height (mm) versus the number of completed rotations). The color-coded cumulative reward for the last episode of each run (refer to equation (1)) corresponds to different curricula. Note that the final manipulation performance is represented by those points inside the green box defining the desired ball height (25 ± 4 mm).

Discussion

What did we learn about learning to manipulate?

We provide proof-of-principle that it is possible to learn the hard problem of dynamic dexterous manipulation. Putting our work in context is critical and best done by pointing to its place in the

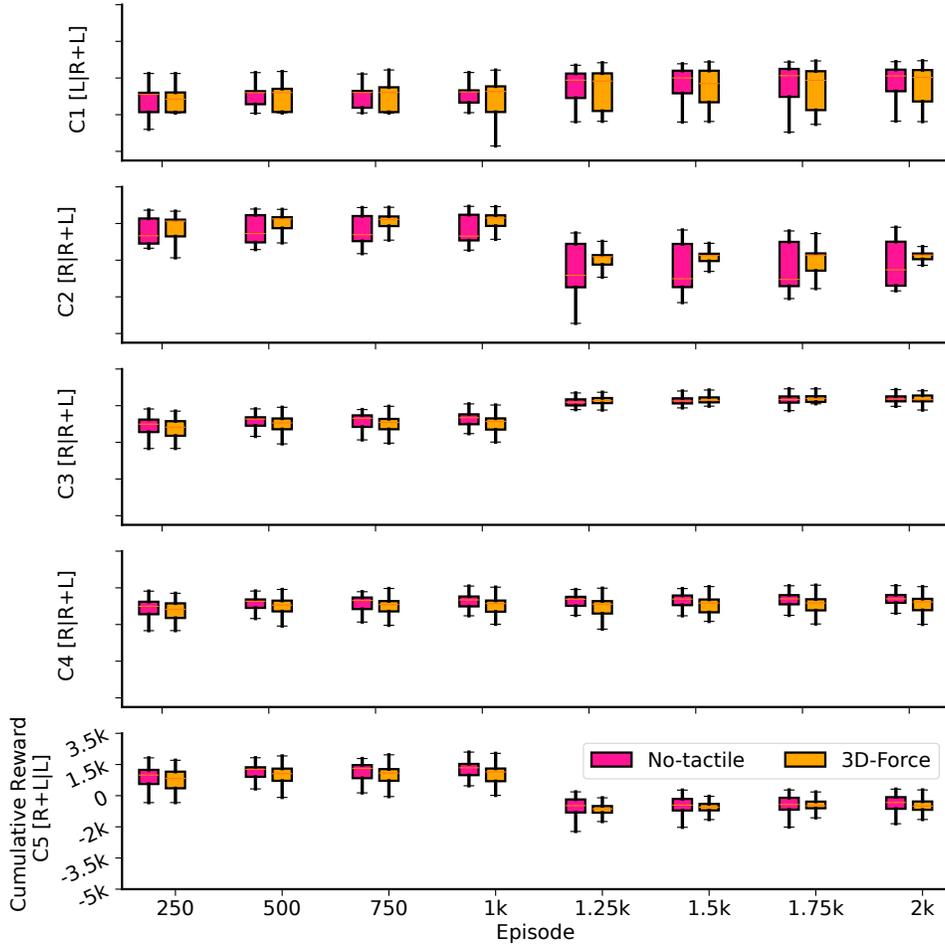


Figure 4: **Cumulative reward across all curricula and tactile information conditions.** Boxplots, with median, across tactile conditions for 60 runs, every 250 episodes. Note learning tends to saturate early.

updated taxonomy of hand function put forth by MacKenzie, Iberall, Brand, Curkosky, Dollar, and others (2, 18, 54–57). In particular we have addressed the problem of dynamic manipulation with three fingers while the ball is at risk of being dropped at any moment (see ‘Comparison to State-of-the-Art’ section). This definition emphasizes that ‘grasp’ and ‘pick-and-place manipulation’ are conceptually and mechanically distinct from ‘dynamic manipulation’ as addressed here, even though they are at times used interchangeably in the literature (58). Such dynamic manipulation is, in fact, an enviable ability that is also difficult for biology to achieve as it develops in humans late in childhood, degrades in healthy aging, and is quickly lost in

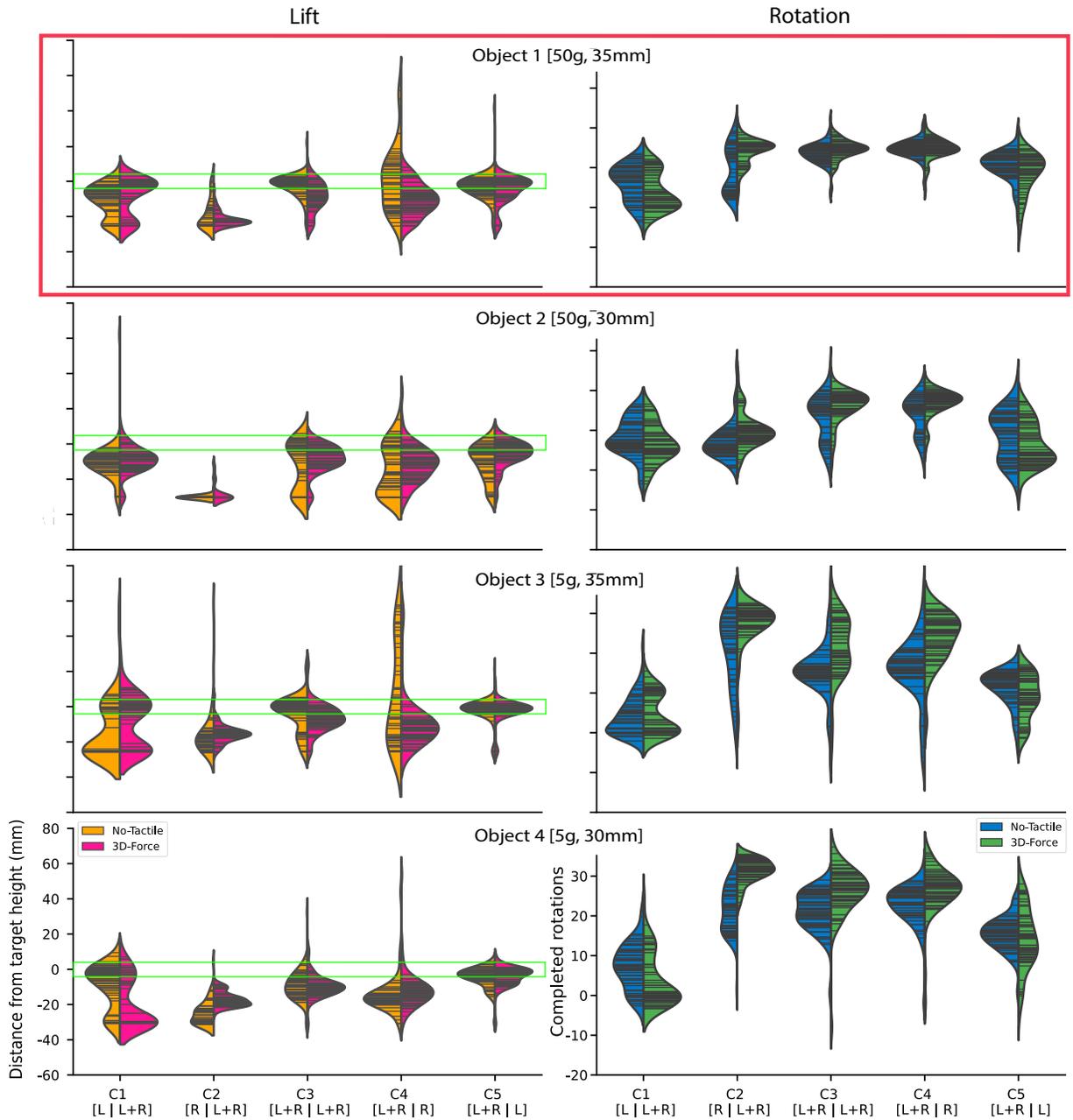


Figure 5: Final Performance for Lift (Left) and Rotation (Right) for both tactile conditions for four objects. The top row corresponds to the baseline object described in the main results. Violin plots show the distribution of Lift and Rotation at the end of learning (i.e., the last 10 seconds of the 2,000th episode) for all 60 trials. Lift is described as a distance from the desired height (the green box shows the distance from the desired height range ± 4 mm) and Rotation as the number of completed rotations for both tactile conditions, No-tactile and 3D-force.

even mild/initial forms of neurological conditions such as peripheral neuropathies, stroke, and Parkinson’s disease (e.g., (58, 59)). In our work, the fingertips induce dynamic translation and rotation of the ball while making and breaking contact. As such, the hand function we achieved merits the description of dynamic dexterous manipulation.

Curriculum learning can be seen as a developmental process from a pluripotent state. We use the analogy of the Waddington Landscape (Fig. 2C) for curriculum learning of manipulation because of its similarity to epigenetic transformation from a pluripotent state in biological development (53, 60). Curriculum learning, in fact, produces a developmental trajectory from a naïve (i.e., pluripotent) state based on resources (tactile information) and experience (curriculum) (Fig. 2A&B). Each curriculum affects both the progression of the learning (path) and its final performance (endpoint), and can thus be thought of as traversing a Waddington Landscape. Importantly, the evolution of skills for each curriculum was (unlike cell differentiation) not strictly irreversible, but remained adaptable. Specifically, the change of reward after the first learning phase did not preclude the system from emphasizing the improvement of the new skill. This is visually represented by 90° shifts in the paths (see C1 [L|L+R] and C2 [R|L+R] in Fig. 2). In some cases, the response to a switch in reward even reversed a learned skill for the first 250 episodes in the second phase of learning, and only then increased the new skill (see C4 [L+R|R] and C5 [L+R|L] in Fig. 2). In one case, C3 [L+R|L+R], there was no change in reward after the end of the first learning phase, and the system was saturated already. In others, the system did respond like an ‘irreversible’ system that learned little of the new skill, of at all, when the reward function was switched (e.g., C2 [R|L+R] in the 3D-tactile case in Fig. 2). See the next Discussion Section.

The role of sensory information

Manipulation can be achieved without tactile information or vision. Tactile information has long been thought as necessary for human—and by extension robotic—manipulation (4, 61).

This idea was reinforced by the work of Johansson and Westling (62, 63) demonstrating that numbing the fingerpads with temporary anesthetic greatly impairs fine manipulation. Our results in Fig. 2 provide a counter-example to this longstanding notion. Interestingly, we found that our system was able to learn even in the absence of tactile information (the No-tactile condition in Figs. 2A and 3). In fact, having 3D-tactile information not always produced better performance (cf. C3 [L+R|L+R] in Fig. 2A&B).

How is it possible to learn to manipulate without vision or tactile information? The answer, we believe, comes from the nature of reinforcement itself. As described in Fig. 1B, PPO—as a reinforcement learning algorithm—condition its actions (next-step finger joint angles, angular velocities, palm position, and velocity) based on the system state, ultimately optimizing for increased reward. In the No-tactile case, the hand’s state comprises finger joint angles, angular velocities, palm position, and palm velocity—which seem to suffice to learn the task. Therefore, lift and rotation of the ball was a product of guided hand kinematics that properly affect ball dynamics to increase the reward in the No-tactile case, such as in our previous work to learn locomotor movements without the need to sense the ground (39).

As such, a main contribution of our work is to provide an existence proof that an agent using reinforcement learning is able to learn a sophisticated manipulation behavior even in the absence of tactile information. Note that direct vision was not necessary either, as in other prior work (39). Our important result about dynamic manipulation provides impetus to revise our thinking about, and use of, tactile and visual information to allow freer thinking for engineers (and bio-roboticists) creating the next generation of dexterous hands and robots.

The presence or absence of tactile information did, however, alter the progression of learning. Figures 2A&B (and the Supplementary Information Fig. S1) show that the type of sensory information did affect learning—but the general features of the path and endpoint of each curriculum remained similar in both tactile conditions. Importantly, the effect of tactile information was not

systematic. The 3D-tactile cases were not consistently or necessarily better than the No-tactile cases, or vice versa. Thus, curriculum is a dominant factor compared to tactile information.

From the computational perspective, one could have expected that when learning with a fixed number of episodes, 3D-tactile information would perform systematically worse because of the computational demands associated with extending the length of the hand state vector s_h by 9 elements (3 forces per finger) for the same PPO algorithm architecture which now has to tune more weights (Fig. 1). But, 3D-tactile cases at times outperformed the No-tactile cases (e.g., C1 [L|L+R] in Fig. 1A&B), which strongly suggests that our comparisons across curricula and tactile conditions are not the result of an imbalance in computational demands for a fixed number of learning episodes (1,000 per learning phase for a total of 2,000). This is additionally supported by the fact that 3D-tactile cases also saturated their learning by the 250th episode (like the No-tactile cases).

Also, it is important to note that this study does not undermine the effectiveness of tactile information in many everyday tasks. It merely provides a proof-of-principle that it is possible to learn a specific task (i.e., the manipulation task of interest in this paper) without using tactile information; and with performance comparable to when tactile information is available. It is clear that many tasks exist for which sensory signals would either be crucial to perform, or would greatly enhance, either the learning speed for the task, the final performance, error correction, and/or their robustness and repeatability. These are beyond the scope of our work.

What did we learn about learning?

Our system exhibits some important features of lifelong learning. As defined in (11), our system shows *transfer and adaptation* because it reuses knowledge to improve performance and rapidly adapts to novel skills as in C1 [L|L+R] C2 [R|L+R] , C4 [L+R|R] , and C5 [L+R|L] (in Figs. 23, and 4). Similarly, our system did not suffer from *catastrophic forgetting* as it was able to retain varying amounts of previously learned knowledge on a case-by-case basis (Fig. 2 and

Supplementary Information, Fig. S1). For example C4 [L+R|R] and C5 [L+R|L] did not entirely forget to lift or rotate when they were no longer rewarded, respectively.

Curriculum learning does not necessarily have to advance gradually from single-objective to multi-objective rewards. In many applications such as locomotion, investigators have found that curriculum learning is indispensable to advance gradually from single-objective (i.e., ‘simpler’) to multi-objective (i.e., ‘more complex’) rewards (64). This has led to curriculum learning becoming the standard approach in the field. From the traditional definitions of Vanilla or Progressive curriculum learning (65, 66), one might assume that first learning to lift the ball (a form of grasp) is ‘easier’ than rotating it, which involves a dynamic behavior (4, 27) and a curriculum strategy in which rotation is learned only after lift is going to be a significantly more successful one. However, rewarding lift *and* rotation from the start does not hinder learning, as demonstrated by C3 [L+R|L+R]. In fact, it allowed transfer and adaptation for C4 [L+R|R] and C5 [L+R|L] to subsequently refine the single skill rewarded during the second phase of learning—albeit at the expense of some reduction of the non-rewarded skill. However, it is noteworthy that curricula that rewarded only one skill from the start (C1 [L|L+R] and C2 [R|L+R]) were not able to learn the second skill as efficiently during the second learning phase (rotation and lift, respectively).

Another aspect of lifelong learning involves the saturation of capacity causing learning to slow down (67, 68). Capacity saturation arises due to the fixed representational capacity of parametric models, including the PPO algorithm (68). We see this in our implementation of PPO—which increasingly fails to absorb additional knowledge from successive episodes. This is most evident in C3 [L+R|L+R] for the entire second phase of learning, as shown in Fig. 2. A learning model with more free parameters would theoretically be able to absorb additional knowledge from successive episodes.

The curriculum-based learning rate scheduler enhances the efficiency of learning which accelerates convergence to higher reward

We sought to align the implementation of learning rates in PPO with the nature of curriculum learning. To do this, we defined our curriculum-based learning rate scheduler to adjust the linearly-decaying learning rate when the reward changed (Fig. 6). We find this improved learning and allowed a more fair comparison across curricula as it reduced heuristic tuning efforts. This curriculum-based learning rate scheduler offers an effective approach tailored to curriculum learning for autonomous systems by modifying the learning rate only when changing task complexities and rewards. Empowering curriculum learning to adapt learning rates in a way compatible with changing rewards enables autonomous systems to learn complex and dynamic environments more systematically, autonomously, and effectively. Thus, integrating curriculum-based learning and reward scheduling into a ‘curriculum-based learning rate scheduler’ for autonomous systems is vital to enhance their learning capabilities and performance in manipulation tasks.

Lastly, we demonstrate our results generalize to balls of different weights and size. As shown in the Supplementary Information in Figs. S1–S7, our results were consistent across the four objects we studied (i.e., of two masses, 50 and 5 g, and two sizes, 35 and 30 mm in radius, Fig. 1A). Namely, our system was successful at learning to manipulate, but in a way that curriculum had a greater impact than tactile information. There were minor differences across the endpoint performance for each object (note the difference is the scales of the axis). But the learning paths for each curriculum, and the effect of switching the reward, remained consistent (Fig. S1). This can also be seen in the detailed depiction of the distribution of rewards as learning progressed, Fig. 5). This further shows that the effect of ball size or weight (like that of tactile information as mentioned above) was not substantial nor systematic.

Comparison to the state of the art

It is critical to note that, as we have stated in the past (4), *grasp* or *pick-and-place tasks* are not *dexterous manipulation* in the rigorous sense of grasp taxonomies. Even reorienting a cube resting on an upward-facing palm (9, 16, 50, 69–72) is not prehensile manipulation. Likewise, Prior work has used extensive vision with the upright palm to hold an object being reoriented (12, 73), other than one demonstration of learning under increasing force of gravity (74), we know of no published demonstration of dynamic manipulation task against full gravity utilizing curriculum learning either directly in simulation or hardware. We employ a novel curriculum-based learning rate scheduler for PPO, which significantly enhances the success performance across all scenarios. We now discuss how our novel approach to manipulation compares and contrasts with other studies in robotics and RL. The state-of-the-art of *autonomous learning* for in-hand manipulation is limited. Although important advances have been made using computationally intensive approaches in simulation and hardware (e.g., (12, 28, 35, 41–43)), these tend to be impractical for autonomous learning at the edge. Augmenting RL for manipulation with imitation learning has shown some successes (12, 36–38), but collecting task-specific expert demonstrations from humans are often limited to specific objects or tasks, might not always be practical, require specialized equipment and can be time-consuming.

In contrast, we used a model-free data-driven approach because precise prior knowledge of the system, objects and the environment is not always available, especially in unstructured environments. Although some other studies also use model-free RL methods for rotating objects with simulated fingers or a robotic hand (9, 75–77), we have overcome some of their drawbacks. In (9, 76), the orientation of an object was controlled while resting on an upward-facing palm. Thus, it did not have to be held against gravity as it was not at risk of being dropped at any time.

Some of these limitations were addressed by Chen *et al.* (77) in simulation by manipulating the object with the palm facing downward like we did, but gravity was introduced slowly as part of the curriculum. Moreover, to successfully manipulate the object the authors found it

important to ‘initialize the object in a stable configuration’—which we did not need.

The way our work went beyond the state-of-the-art, therefore, is by demonstrating for the first time a method with the ability to autonomously learn to manipulate an object against gravity while revealing the role of curriculum learning and tactile information in in-hand manipulation. The impact of learning rate scheduling on stochastic optimizer performance has been extensively investigated in recent research (78, 79). In our study, we specifically explore the effects of a constant and linear piecewise learning rate for PPO on the success of our architecture. After careful consideration, we have decided to proceed with the piecewise learning rate. This adaptive approach adjusts rates dynamically throughout training, speeding up the process with higher initial rates and ensuring stable convergence with lower rates later on.

Lastly, our work underlines the importance of curricula in manipulation and shows how the right choice of a curriculum can enhance performance and robustness across multiple tasks by exhibiting some important features of lifelong learning.

Limitations, opportunities and future directions

While our work pushes the field of autonomous manipulation forward, it naturally has some limitations. First, our work is done in simulation. But, as with many other studies looking to bridge the sim2real divide (39, 80), we used realistic physical constraints within our state-of-art physics engine (MuJoCo) that handles dynamic contacts and impacts well. This is a foundation that will enable future implementations in hardware. As to the geometry of our hand, it is common for useful robotic hands to have three fingers (28, 75). Curriculum learning has multiple varieties (66) that can adapt as learning progresses such as Self-Paced curriculum learning. In our case, our learning phases were of fixed duration even though the system tended to plateau. Thus, it could benefit from future implementations that adapt reward changes to minimize training time. Lastly, our manipulation tasks serve as a foundation for—but do not yet address—traditional use cases for activities of everyday life.

Our choices regarding PPO, curriculum design, hand and object structure, reward function, and other parameters were specifically tailored to address the scientific questions of interest within the scope of this paper and to establish a proof of concept. It’s important to emphasize that our selections were not intended as universally applicable solutions. That is to say, to address a different need, a similar pipeline to this paper can be utilized but different tasks, environments, or robotic structures might need to be used. Also, different learning blocks (different than the RL technique or the adaptive curriculum-based learning rate scheduler function used in this paper) can be used that might serve best for another specific task or purpose.

Methods

In this section, we first describe the simulation environment and the task used in this study. Then, we elaborate on the learning policy that enabled autonomous manipulation.

Simulation environment

The manipulation and machine learning communities have used the advanced physics simulation environment MuJoCo (81) for tasks involving autonomous manipulation. MuJoCo allowed us to implement reinforcement learning algorithms on a robotic hand in a realistic environment that includes contact dynamics (including penetration) and gravitational acceleration (81, 82).

To demonstrate the adaptability and robustness of our proposed methodologies, we assessed the performance using four different objects. Our evaluations encompassed systematic exploration, considering two different weight combinations (50 g and 5 g), as well as varying ball radii (35 mm and 30 mm). The work presented herein focuses on a ball of 50 g with a radius of 35 mm with the other configurations presented in the section Generalizability in Supplementary material.

Robotic Hand Design.

We simulated a bio-inspired, three-fingered robotic hand with a palm and three identical servo-driven fingers: two adjacent fingers, analogous to the 'index' and 'middle' fingers, and one opposing them, analogous to the 'thumb'. In contrast to our prior efforts (83), where we showcased the reach-to-manipulate capability with a downward-facing orientation using distinct curricula, we modified the hand design. Each finger consisted of two joints that could rotate about the y -axis (q_1 and q_2 in (Fig. 1A)), similar to the flexion or extension seen in human fingers. The size of the palm and length of each 'phalanx' was based on an average human hand (75, 84). An additional servo motor was included at the base of the hand, which provides translational motion in the vertical direction (z_h).

Fingertip Tactile Sensors.

This work incorporated tactile information and RL, sometimes referred to as touch-augmented RL, as we covered the internal side (i.e., the 'pads' of the fingertips) of the distal phalanx of each finger with tactile sensors. Contact regions were configured near the tips of each finger (Tactile Area, Fig. 1). Objects contacting the finger outside of these tactile areas (sites, in MuJoCo) are not perceived as tactile information by the learning algorithm (76, 81).

We used MuJoCo's built-in features to record the 3D-force sensor on the fingertips of all three fingers. The 3D-force sensor sites provide a 3D array of 3 orthogonal forces (one normal and two tangential to the sensor site for each sensor) of scalar values representing the 3D-force vector. Moreover, we have considered an additional case: No-tactile. In the No-tactile case, the state vector for the tactile information $s_{h,f}$ is null (we do not consider the tactile information in learning). As shown in Fig. 1A, the possible contact tactile information at each fingertip is indicated by $\mathbf{s}_h, \mathbf{f} = [f_{t,1}, f_{t,2}, f_n]$ and it depends on tactile sensing available at fingertips. See Supplementary Table S3 for more details on the tactile information.

Task Description

The robotic hand attempts to manipulate a 50 g, 35 mm radius ball, which starts each episode on the ground with the palm of the robotic hand at a height of 200 mm above the ground. The ball height z_b is defined at the center of the ball, and we specified a desired height for the ball z_d to be 25 mm above z_b . In other words, the desired height z_d is 60 mm above the ground. Through simulation constraints, the ball is limited to 2 translational DOFs (moving vertically z and horizontally x) and 1 rotational DOF (rotation about the θ_y direction; see (Fig. 1)). We included viscous damping in the translational and rotational DOFs of the ball to stabilize the simulation and prevent numerical instabilities for the simulation of the rigid fingers.

We further limited the ball’s movement in the x direction by adding stiffness to the ball. The details of the simulation parameters, including the robotic hand and the ball) are shown in Supplementary Table S1.

Observation and Action Space

The system’s state vector includes the hand state vector (\mathbf{s}_h), consisting of fourteen kinematic degrees of freedom (DOFs), along with the position and velocities of the hand’s palm (\mathbf{s}_p) (2 DOFs), and the position and velocity of the ball (\mathbf{s}_b) (6 DOFs). This 20-dimensional vector encapsulates joint angles (q_1 to q_6) and their derivatives, as well as the vertical height of the hand (z_h) and its derivative, collectively describing the dynamic state of the system.

Additionally, the ball state vector comprises vertical (z) and horizontal (x) translation, and its rotation about the y -axis (θ_y). No other translations or rotations are permitted (see Supplementary Table S2). The height of the hand, z_h , is actuated for the hand to reach for and manipulate the state of the ball (\mathbf{s}_b) by rotating (θ_y) and lifting (z_b) it to a desired height (z_d).

It’s important to note that not all state variables are utilized in our reinforcement learning policy (observation state). Specifically, the observation state omits details about the ball’s velocity and position, as explained in the following subsection. Furthermore, it’s worth mentioning

that the action space aligns with the observation state. When a 3D-force is introduced, the state of the system dynamically changes, augmenting the hand state with an additional 9 data points.

Algorithm 1 Simulation with PPO

```
1: procedure RUNSIMULATION
2:   Initialize simulation environment
3:   Set random seed for reproducibility
4:   Initialize policy and value networks
5:   Initialize optimizers
6:   Set hyper-parameters and simulation parameters
7:   Initialize replay buffer
8:   Set training iterations and mini-batches
9:   Set PPO-specific parameters
10:  for  $episode = 1$  to 2,000 do
11:    Initialize episode
12:    for  $t = 1$  to 1,000 do
13:      Sample and execute actions
14:      Store transition in the replay buffer
15:    end for
16:    Update the policy using PPO
17:  end for
18: end procedure
```

Autonomous Learning Approach

To autonomously learn in-hand manipulation of a ball against gravity through utilizing tactile information, we used a model-free RL algorithm to learn the policy. We used Proximal Policy Optimization (PPO) as our main algorithm as it presented a balance between the ease of implementation, sample complexity, and ease of adjustment, trying to update at each step to minimize the cost function while assuring that the new policies are not too far from last policies (52, 85). PPO has also been adopted as one of the default methods of OpenAI owing to its excellent performance (86, 87).

Reward Function

The reward engineering concept (a subset of RL) focuses on finding the most appropriate reward to maximize successful learning via reward shaping (88). Reward shaping involves carefully designing reward functions that provide the agent with rewards for progress toward the goal.

In our work, we defined two goals, lift and rotation. **Lift:** Our desired height (center of the ball above the ground) is $z_d = 25$ mm, shown in (Fig. 1). In our algorithm, the goal is reached when the agent supports the ball against gravity within a desired height range of $[21, 29]$ mm, indicated with a green box in Figs. 3 (and Supplementary Figs. S2, S4, S6). A range is used to accommodate height variation during rotation and manipulation tasks. For results metrics, we report the mean height of the ball and lift success as a percent time within an episode where the ball is in the desired height range. **Rotation:** For rotation, we calculated completed rotations as our performance measurement (as opposed to rotation reward or rotation in degrees). Since we care about manipulating the ball against gravity at the desired height range, we used a combination of primary (positive) reward and punishment (penalty proportional to the distance between the current height and the desired height as a negative reward) at every time step.

In our reward function, the angular velocity of the ball $\dot{\theta}_y$ was the primary reward, and the absolute distance of the state from the reference state of having the ball at the fixed desired position ($z_d = 25$ mm, (Fig. 1)) was the punishment. The reward function is described by

$$Reward_t = c_R \dot{\theta}_{y,t} - c_L |z_{h,t} - z_d|, \quad (1)$$

where $c_R = 0.51$ and $c_L = 0.49$.

We investigated learning strategies (here, curriculum) in which lift and rotation are both rewarded (L+R), strategies in which only lift is rewarded with rotation coefficient set to zero (c_R), and strategies in which lift coefficient is set to zero (c_L) and only rotation is rewarded (R). This is described in detail in the following section (see Table 1).

Curriculum Learning

A learning trial consisted of 2,000 episodes, where each episode lasted 10 seconds. This resulted in a total simulated time of 5 hours and 33 minutes per trial. Each learning trial was split into two equal halves where the reward function changed between the two halves of the trial. We considered five distinct curricula that differed in the behavior (rotation and lift) rewarded in two halves of the trail. This is illustrated by a circle with a curved arrow (rotation) and a vertical arrow (lift) throughout the paper and pictured in the second column of Table 1). As shown in the last column, by changing c_R and c_L variables in equation (1), we update the reward function in two equal halves of the learning trial in each curriculum. The final column of the table, gives the values of c_R and c_L used in equation (1), to update the reward function in the two halves of the learning trial in each curriculum.

Learning was evaluated over 60 trials for each of the 5 curricula. Each of these 60 trials was independent by varying the seed parameters of the PPO algorithm for our reinforcement learning policy. This was repeated for the two tactile conditions (No-tactile and 3D-force). For each tactile condition, the initial seed for the random number generator was held constant across different curricula. For example, the first trial run seed was exactly the same for all curricula and both tactile conditions. Overall, we used independent trials to evaluate the effectiveness of our approach to autonomous manipulation.

Reinforcement Learning Policy

We adopted a Proximal Policy Optimization (PPO) policy to control the robotic hand to achieve autonomous manipulation. PPO is a set of policy gradient methods that optimize a surrogate objective function using multiple minibatch updates per data sample (52, 89). The objective function to optimize is the sum of several loss functions and is given by

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \quad (2)$$

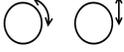
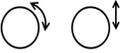
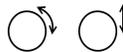
	Reward During First Half of the Trial	Reward During Second Half of the Trial	Coefficients of equation (1) [first second] halves of each trial
Curriculum 1			[L R+L] [$c_R = 0, c_L = 0.49$ $c_R = 0.51, c_L = 0.49$]
Curriculum 2			[R R+L] [$c_R = 0.51, c_L = 0$ $c_R = 0.51, c_L = 0.49$]
Curriculum 3			[R+L R+L] [$c_R = 0.51, c_L = 0.49$ $c_R = 0.51, c_L = 0.49$]
Curriculum 4			[R+L R] [$c_R = 0.51, c_L = 0.49$ $c_R = 0.51, c_L = 0$]
Curriculum 5			[R+L L] [$c_R = 0.51, c_L = 0.49$ $c_R = 0, c_L = 0.49$]

Table 1: We used five curricula that rewarded different combinations of rotation and lift during each half of the independent trials. These changes in the coefficients of the reward function define a progression of goals (i.e., curriculum learning) over the two halves of each run.

The $L_t^{CLIP}(\theta)$ is the clipped surrogate loss function and ensures that the policy updates will not be too large. While the L_t^{VF} is a squared-error loss, it ensures that the loss from both policy and value functions of the neural networks are accounted for. The S denotes the entropy bonus term, which encourages a more random policy (i.e., more exploration), so a larger entropy coefficient c_2 will encourage more exploration (89).

To implement PPO, we use the PPO1 implementation from OpenAI’s stable baselines repository (87) with MultiLayer Perceptron (MLP) Artificial Neural Network (ANN) for the actor-critic mapping. The Proximal Policy Optimization (PPO) algorithm, outlined in Algorithm 18, describes the iterative process through which the policy and value functions are updated to maximize cumulative rewards.

At every time step t , the robotic hand observes its own state $s_{h,t}$ and the state of the ball $s_{b,t}$, predicts the optimized action, executes it a_t , and a reward is used r_t . The state $s_{h,t}$ contains the angle and angular velocity q_t, \dot{q}_t of each finger and the position and linear velocity of the palm

at every time step t . The overview diagram of the Proximal Policy Optimization algorithm in this work is shown in (Fig. 1B).

To attain optimal performance, we fine-tuned the hyper-parameters and meta-parameters of the PPO algorithm during the training of our RL model. The clipped surrogate loss in the PPO algorithm serves to prevent divergence, as discussed in (52). However, it introduces a challenge by potentially prematurely reducing exploration variance across multiple iterations. Strategic tuning of the loss parameters in equation (2) becomes essential to avoid issues such as divergence or settling on a local minimum. PPO addresses this challenge by including an entropy loss term that penalizes low variance, mitigating the risk of premature convergence. It has been observed that a higher entropy loss weight minimizes the risk of getting trapped in local optima. Nevertheless, if the entropy loss weight is excessively large, it can lead to a noisy policy and a decline in average performance. Therefore, careful adjustment of the entropy loss term for PPO is necessary. Building on the findings regarding different entropy loss weights for the policy’s standard deviation in (90), we optimized the entropy loss term to strike a balance between variance and average performance.

PPO employs the Generalized Advantage Estimator (GAE) to reduce the variance of policy gradient estimates at the expense of some tolerable bias. GAE is parameterized by $\lambda \in [0, 1]$, which enables the PPO agent with a mechanism to control policy updates according to the significance of each sampled state and, therefore, enhance learning reliability (91). Changing this hyper-parameter enables PPO to find a balance between variance and bias of policy gradient estimates (92). In our work, this trade-off was achieved by changing the lambda meta-parameter to relatively demote rewards achieved later in the episode (when the ball may have been dropped) and instead emphasizing immediate rewards at every point in time (as is the case in real life).

The number of optimization epochs, GAE parameter λ , and the entropy coefficient are set to values shown in Table S4. All other parameters are kept at their default values per PPO implementation.

Adaptive curriculum-based learning rate scheduler

The impact of learning rate scheduling on the performance of stochastic optimizers has garnered considerable attention in recent research (78, 79). Traditional approaches, employing a fixed and static learning rate throughout training, often struggle to attain optimal model performance. To address this limitation, diverse scheduling algorithms, such as polynomial decay, cosine decay, and warm-up, have been proposed, each tailored with distinctive forms (93). Current methodologies often rely on predefined principles, assuming specific scheduling rules based on empirical studies and domain knowledge. These approaches may not rigidly adhere to any existing rule to find the optimal learning rate scheduling for a particular problem.

In our exploration of PPO, we aim to transcend the constraints associated with a constant learning rate. Initially, we opted for a constant learning rate and linear learning rate, commonly used approaches in reinforcement learning algorithms (94). But implementing a constant learning rate in dynamic contexts, where sensitivity to the initial rate choice can result in unstable training or sub-optimal solutions, highlights the necessity for adaptive approaches. We proposed a new method to tackle challenges with fixed learning rates, especially in dynamic environments like our manipulation tasks. This is addressed by an adaptive curriculum-based learning rate scheduler, bringing multiple advantages. This adaptive strategy dynamically adjusts rates throughout training, expediting the learning process with higher initial rates and ensuring stable convergence through decrementing rates during later stages.

Curriculum-based Learning Rate Scheduler Strategy

Instead of utilizing a fixed or decreasing learning rate, our method embraces a curriculum-adaptive learning strategy. The adaptive curriculum-based learning rate scheduler (piecewise linear learning rate) strategy is described as follows:

$$Lr = \begin{cases} \phi \cdot \left(1 - \frac{\text{sample number}}{1,000,000}\right), & \text{sample number} \leq 1,000,000 \\ \eta \cdot \left(1 - \frac{\text{sample number}}{2,000,000}\right), & \text{sample number} > 1,000,000 \end{cases} \quad (3)$$

The selection of optimal values for ϕ and η was determined empirically, ensuring adaptability across all five curricula. The curriculum-based learning rate scheduler (Lr) is established and adjusted through trial and error to emphasize the significance of curriculum learning. These coefficients are then integrated into the PPO linear scheduler according to the following equation. Our curriculum dynamically changes at 1,000 episodes (1,000,000 samples), compelling the learning rate to be piecewise linear to accommodate the variations in the dynamics of the reward and tasks. This adaptive strategy effectively responds to changes in the environment, contributing to the model’s success.

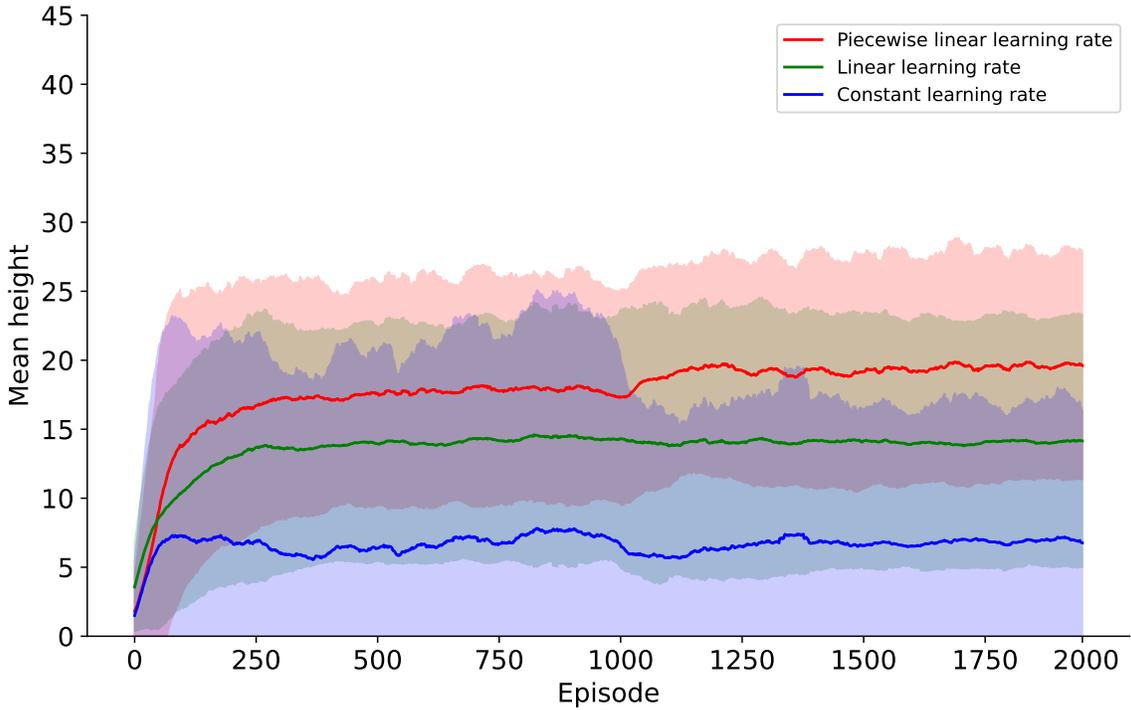


Figure 6: **Effect of PPO curriculum-based learning rate scheduler by comparison of Mean Height.** Data presented for mean height in C5 [L+RIL] throughout the whole learning period. The desired height for all cases is 25 mm. Solid lines represent the mean across all 60 trials for the specified learning rate methods. Shaded areas represent ± 1 standard deviation. The solid red line follows the PPO implementation per equation (3).

To validate the effectiveness of our approach, we explore the impact of constant, linear, and adaptive curriculum-based learning rate scheduler (Piecewise Linear Learning Rate) in C5

[L+RIL] in Fig. 6, comparing Mean height over 2,000 learning episodes. Piecewise linear learning rate was far closer to this target height than either Linear or Constant learning rate. Thus piecewise linear learning rate was used as the curriculum-based learning rate scheduler throughout this work. Our results in different curricula consistently support the superior performance of PPO with well-designed scheduling mechanisms, surpassing those utilizing a constant and linear learning rate in both convergence rate and final performance metrics (93, 95, 96). One key advantage is the reduced sensitivity to the initial rate choice, minimizing the risk of divergence. The piecewise linear learning rate promotes efficient exploration in the early stages and exploitation for optimal performance during convergence. Its curriculum-based adaptive nature contributes to faster convergence, effectively navigating both exploratory and exploitative learning phases. Moreover, the piecewise linear schedule imparts robustness against variations in task difficulty or environmental changes, automatically adjusting to maintain training stability.

To evaluate the effectiveness of different learning scheduler methods in reducing convergence time, we conducted an analysis on the average number of episodes needed after switching the reward during the second phase of learning in C5 [L+RIL]. We compared three learning schedulers: constant learning rate, linear rate, and piecewise linear rate.

Our findings reveal that the average number of episodes for convergence in successful trials (defined as trials where the hand can maintain the ball within the target height range) after the reward switch varied significantly across the different schedulers. Specifically, when focusing only on the successful trials (not shown), we observed that it took 1,000 episodes for convergence with a constant learning rate, 450 episodes with a linear rate, and only 250 episodes with a piecewise linear rate (see episodes 1,250 in Fig. 6).

Figure 6 illustrates the performance of each scheduler in reaching the target height. Remarkably, the piecewise linear learning rate outperformed both the linear and constant rates by a substantial margin. Additionally, it achieved a higher cumulative reward across all 60 trials,

indicating its superior effectiveness in learning and adaptation.

These results highlight the significant advantages of using a piecewise linear learning rate scheduler in enhancing convergence speed and overall performance in C5 [L+RIL] simulations.

In summary, our adaptive curriculum-based learning rate scheduler strategy in the PPO implementation aims to enhance training stability, expedite convergence, and improve adaptability in dynamic environments. This aligns with our goal of efficiently training the agent for effective in-hand manipulation and contributes to the exploration of learning rate scheduling strategies on a curriculum-based approach.

The complete code for learning is available at the following GitHub repository.

References

1. Roger N Lemon, RS Johansson, and G Westling. Corticospinal control during reach, grasp, and precision lift in man. *Journal of Neuroscience*, 15(9):6145–6156, 1995.
2. Christine L MacKenzie and Thea Iberall. *The grasping hand*. Elsevier, 1994.
3. Francisco J Valero-Cuevas. Why the hand? *Progress in Motor Control: A Multidisciplinary Perspective*, pages 553–557, 2009.
4. Francisco J Valero-Cuevas and Marco Santello. On neuromechanical approaches for the study of biological and robotic grasp and manipulation. *Journal of neuroengineering and rehabilitation*, 14(1):1–20, 2017.
5. Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446), 2019.
6. Valerio Ortenzi, Marco Controzzi, Francesca Cini, Juxi Leitner, Matteo Bianchi, Maximo A Roa, and Peter Corke. Robotic manipulation and the role of the task in the metric of success. *Nature Machine Intelligence*, 1(8):340–346, 2019.

7. Minas V Liarokapis and Aaron M Dollar. Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2534–2541. IEEE, 2016.
8. Amirul Syafiq Sadun, Jamaludin Jalani, and Faizal Jamil. Grasping analysis for a 3-finger adaptive robot gripper. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6. IEEE, 2016.
9. OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.
10. Abhishek Gupta, Justin Yu, Tony Z Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6664–6671. IEEE, 2021.
11. Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022.
12. Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand dexterous manipulation from depth. In *Icml workshop on new frontiers in learning, control, and dynamical systems*, 2023.

13. Sunny Katyara, Fanny Ficuciello, Darwin Caldwell, Bruno Siciliano, and Fei Chen. Leveraging kernelized synergies on shared subspace for precision grasp and dexterous manipulation. *arXiv preprint arXiv:2008.11574*, 2020.
14. Antonio Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000.
15. Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, pages 348–353. IEEE, 2000.
16. Raphael Deimel and Oliver Brock. A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*, 35(1-3):161–185, 2016.
17. Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814, 2010.
18. RM Murray, Z Li, and SS Sastry. A mathematical introduction to robotic manipulation crc press. *Boca Raton, FL*, 1994.
19. Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.
20. Aaron M Dollar and Robert D Howe. The SDM hand as a prosthetic terminal device: a feasibility study. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pages 978–983. IEEE, 2007.

21. Matthew T Mason, Kenneth Y Goldberg, and Russell H Taylor. *Planning sequences of squeeze-grasps to orient and grasp polygonal objects*. Carnegie-Mellon University. Department of Computer Science, 1988.
22. Katharina Zeissler. A robotic hand gets a grip. *Nature Electronics*, 5(1):18–18, 2022.
23. Eleftherios Triantafyllidis, Fernando Acero, Zhaocheng Liu, and Zhibin Li. Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network roman. *Nature Machine Intelligence*, 5(9):991–1005, 2023.
24. Mark R Cutkosky and Robert D Howe. Human grasp choice and robotic grasp analysis. In *Dextrous Robot Hands*, pages 5–31. Springer, 1990.
25. Vittorio Caggiano, Guillaume Durandau, Huwawei Wang, Alberto Chiappa, Alexander Mathis, Pablo Tano, Nisheet Patel, Alexandre Pouget, Pierre Schumacher, Georg Martius, et al. Myochallenge 2022: Learning contact-rich manipulation using a musculoskeletal hand. In *NeurIPS 2022 Competition Track*, pages 233–250. PMLR, 2023.
26. Di Guo, Fuchun Sun, Bin Fang, Chao Yang, and Ning Xi. Robotic grasping using visual and tactile sensing. *Information Sciences*, 417:274–286, 2017.
27. Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.
28. Andrew Morgan, Kaiyu Hang, Bowen Wen, Kostas E Bekris, and Aaron Dollar. Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics and Automation Letters*, 2022.
29. Robert O Ambrose, Hal Aldridge, R Scott Askew, Robert R Burridge, William Bluethmann, Myron Diftler, Chris Lovchik, Darby Magruder, and Fredrik Rehnmark. Robonaut:

- NASA's space humanoid. *IEEE Intelligent Systems and Their Applications*, 15(4):57–63, 2000.
30. Manuel G Catalano, Giorgio Grioli, Edoardo Farnioli, Alessandro Serio, Cristina Piazza, and Antonio Bicchi. Adaptive synergies for the design and control of the pisa/iit soft hand. *The International Journal of Robotics Research*, 33(5):768–782, 2014.
 31. M Bridges, J Beaty, F Tenore, M Para, M Mashner, V Aggarwal, S Acharya, G Singhal, and N Thakor. Revolutionizing prosthetics 2009: dexterous control of an upper-limb neuroprosthesis. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 28(3):210–211, 2010.
 32. Claudio Castellini and Patrick Van Der Smagt. Surface EMG in advanced hand prosthetics. *Biological cybernetics*, 100(1):35–47, 2009.
 33. Manfred Huber and Roderic A Grupen. Robust finger gaits from closed-loop controllers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1578–1584. IEEE, 2002.
 34. Francisco J Valero-Cuevas, Heiko Hoffmann, Manish U Kurse, Jason J Kutch, and Evangelos A Theodorou. Computational models for neuromuscular function. *IEEE Reviews in Biomedical Engineering*, 2:110–135, 2009.
 35. Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.
 36. Rae Jeong, Jost Tobias Springenberg, Jackie Kay, Daniel Zheng, Yuxiang Zhou, Alexandre Galashov, Nicolas Heess, and Francesco Nori. Learning dexterous manipulation from suboptimal experts. *arXiv preprint arXiv:2010.08587*, 2020.

37. Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.
38. Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3786–3793. IEEE, 2016.
39. Ali Marjaninejad, Darío Urbina-Meléndez, Brian A Cohn, and Francisco J Valero-Cuevas. Autonomous functional movements in a tendon-driven limb via limited experience. *Nature Machine Intelligence*, 1(3):144–154, 2019.
40. Shijing Zhang, Yingxiang Liu, Jie Deng, Xiang Gao, Jing Li, Weiyi Wang, Mingxin Xun, Xuefeng Ma, Qingbing Chang, Junkao Liu, et al. Piezo robotic hand for motion manipulation from micro to macro. *Nature Communications*, 14(1):500, 2023.
41. Vikash Kumar and Emanuel Todorov. MuJoCo HAPTIX: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 657–663. IEEE, 2015.
42. Niklas Funk, Charles Schaff, Rishabh Madan, Takuma Yoneda, Julen Uraín De Jesus, Joe Watson, Ethan K Gordon, Felix Widmaier, Stefan Bauer, Siddhartha S Srinivasa, et al. Benchmarking structured policies and policy optimization for real-world dexterous object manipulation. *IEEE Robotics and Automation Letters*, 7(1):478–485, 2021.
43. Silvia Cruciani, Balakumar Sundaralingam, Kaiyu Hang, Vikash Kumar, Tucker Hermans, and Danica Kragic. Benchmarking in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(2):588–595, 2020.

44. Evangelos Theodorou, Emanuel Todorov, and Francisco J Valero-Cuevas. Neuromuscular stochastic optimal control of a tendon driven index finger model. In *Proceedings of the 2011 American Control Conference*, pages 348–355. IEEE, 2011.
45. Huanbo Sun, Katherine J Kuchenbecker, and Georg Martius. A soft thumb-sized vision-based sensor with accurate all-round force perception. *Nature Machine Intelligence*, 4(2):135–145, 2022.
46. Mathias Thor and Poramate Manoonpong. Versatile modular neural locomotion control with fast learning. *Nature Machine Intelligence*, 4(2):169–179, 2022.
47. Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
48. Malte Schilling, Kai Konen, and Timo Korthals. Modular deep reinforcement learning for emergent locomotion on a six-legged robot. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 946–953. IEEE, 2020.
49. Jeff Clune, Kenneth O Stanley, Robert T Pennock, and Charles Ofria. On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(3):346–367, 2011.
50. Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.
51. Wenbin Hu, Bidan Huang, Wang Wei Lee, Sicheng Yang, Yu Zheng, and Zhibin Li. Dexterous in-hand manipulation of slender cylindrical objects through deep reinforcement learning with tactile sensing. *arXiv preprint arXiv:2304.05141*, 2023.

52. John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
53. Conrad Hal Waddington. Evolutionary adaptation. *Perspectives in Biology and Medicine*, 2(4):379–401, 1959.
54. Ian M Bullock and Aaron M Dollar. Classifying human manipulation behavior. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–6. IEEE, 2011.
55. Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2015.
56. Mark R Cutkosky et al. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on robotics and automation*, 5(3):269–279, 1989.
57. Paul W Brand. Clinical mechanics of the hand. In *Hand Rehabilitation in Occupational Therapy*, pages 183–184. Routledge, 2012.
58. Francisco J Valero-Cuevas. Why the hand? In *Progress in Motor Control*, pages 553–557. Springer, 2009.
59. M.T. Duruöz. *Hand Function: A Practical Guide to Assessment*. Springer International Publishing, 2020.
60. Carlos Guerrero-Bosagna, John Lees, Daniel Núñez-León, and João F Botelho. Epigenetics, evolution and development of birds. In *Epigenetics, Development, Ecology and Evolution*, pages 149–176. Springer, 2022.
61. Nicholas Wettels, Veronica J Santos, Roland S Johansson, and Gerald E Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 22(8):829–849, 2008.

62. Roland S Johansson and Goran Westling. Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Experimental Brain Research*, 56(3):550–564, 1984.
63. Roland S Johansson, Charlotte Häger, and Ronald Riso. Somatosensory control of precision grip during unpredictable pulling loads. *Experimental Brain Research*, 89(1):192–203, 1992.
64. Pavan Ramdya and Auke Jan Ijspeert. The neuromechanics of animal locomotion: From biology to robotics and back. *Science Robotics*, 8(78):eadg0279, 2023.
65. Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
66. Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.
67. Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.
68. Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. Toward training recurrent neural networks for lifelong learning. *Neural computation*, 32(1):1–35, 2020.
69. Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

70. Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. *arXiv preprint arXiv:2309.09979*, 2023.
71. Idan Shenfeld, Zhang-Wei Hong, Aviv Tamar, and Pulkit Agrawal. Tgrl: Teacher guided reinforcement learning algorithm for pomdps. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.
72. 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 using hierarchical graph neural network. *IEEE Robotics and Automation Letters*, 8(6):3605–3612, 2023.
73. Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.
74. Leon Sievers, Johannes Pitz, and Berthold Bäuml. Learning purely tactile in-hand manipulation with a torque-controlled hand. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2745–2751. IEEE, 2022.
75. Herke Van Hoof, Tucker Hermans, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 121–127. IEEE, 2015.
76. Andrew Melnik, Luca Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter. Tactile sensing and deep reinforcement learning for in-hand manipulation tasks. In *IROS Workshop on Autonomous Object Manipulation*, 2019.
77. Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. *arXiv preprint arXiv:2111.03043*, 2021.

78. A Feder Cooper, Yucheng Lu, Jessica Forde, and Christopher M De Sa. Hyperparameter optimization is deceiving us, and how to stop it. *Advances in Neural Information Processing Systems*, 34:3081–3095, 2021.
79. Zhen Xu, Andrew M Dai, Jonas Kemp, and Luke Metz. Learning an adaptive learning rate schedule. *arXiv preprint arXiv:1909.09712*, 2019.
80. Felix Ruppert and Alexander Badri-Spröwitz. Learning plastic matching of robot dynamics in closed-loop central pattern generators. *Nature Machine Intelligence*, pages 1–9, 2022.
81. Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.
82. Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6808–6815. IEEE, 2014.
83. Romina Mir, Pegah Ojaghi, Ali Marjaninejad, Michael Wehner, and Francisco Valero-Cuevas. Active sensing in a bioinspired hand as an enabler of implicit curriculum learning for manipulation. *AMAM (international symposium on Adaptive Motion of Animals and Machines)*, 2021.
84. Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383. IEEE, 2016.
85. Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6202–6209, 2020.

86. Cheng-Yen Tang, Chien-Hung Liu, Woei-Kae Chen, and Shingchern D You. Implementing action mask in proximal policy optimization (ppo) algorithm. *ICT Express*, 6(3):200–203, 2020.
87. Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
88. Kyle Mills, Pooya Ronagh, and Isaac Tamblyn. Finding the ground state of spin Hamiltonians with reinforcement learning. *Nature Machine Intelligence*, 2(9):509–517, 2020.
89. Jeppe Theiss Kristensen and Paolo Burrelli. Strategies for using proximal policy optimization in mobile puzzle games. In *International Conference on the Foundations of Digital Games*, pages 1–10, 2020.
90. Perttu Hämmäläinen, Amin Babadi, Xiaoxiao Ma, and Jaakko Lehtinen. Ppo-cma: Proximal policy optimization with covariance matrix adaptation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.
91. Gang Chen, Yiming Peng, and Mengjie Zhang. An adaptive clipping approach for proximal policy optimization. *arXiv preprint arXiv:1804.06461*, 2018.
92. John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.
93. Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018.

94. Yuanhao Xiong, Li-Cheng Lan, Xiangning Chen, Ruochen Wang, and Cho-Jui Hsieh. Learning to schedule learning rate with graph neural networks. In *International Conference on Learning Representation (ICLR)*, 2022.
95. Andrew Senior, Georg Heigold, Marc’ Aurelio Ranzato, and Ke Yang. An empirical study of learning rates in deep neural networks for speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6724–6728. IEEE, 2013.
96. Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Acknowledgements

This work is supported in part by the NIH R21 NS113613-01A1, NSF 2113096 CRCNS US-Japan, DOD CDMRP Grant MR150091, DARPA-L2M W911NF1820264 awarded to FV-C, and the USC Viterbi School of Engineering Fellowships to AM and RM. This work does not necessarily represent the views of the NIH, NSF, DoD, or DARPA.

This work was supported in part by the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education funded by the Wisconsin Alumni Research Foundation.

Author contributions

P.O., R.M., A.M., M.W., and F.J.V.-C. contributed to the conception and design of the work. P.O., R.M., A.M., and A.E. contributed to the data analysis. F.J.V.-C., M.W., and A.M. provided general direction for the project. All authors approved the final version of the manuscript and agree to be accountable for all aspects of the work. All persons designated as authors qualify for authorship, and all those who qualify for authorship are listed.

Competing Interests

The authors declare no competing interests.

Supplementary Information

Supplementary Methods

We simulate a three-fingered robotic hand using MuJoCo, an advanced physics simulation environment that enables us to model physical systems and their dynamic interactions. We also utilize the OpenAI baselines library and the MuJoCo-py interface to implement reinforcement learning algorithms developed in Python. The robotic hand attempts to autonomously learn to manipulate a ball (Fig. 1A). Our data-driven learning approach does not necessitate an explicit model of the hand, ball, or their interactions, which is advantageous in unstructured environments compared to model-based approaches (43, 82). Furthermore, our approach does not rely on visual information at any stage of the learning process.

Our aim is to achieve autonomous acquisition of manipulation skills by the robotic hand, with upcoming sections detailing the simulation, learning algorithm, and thoroughly exploring the assessment of generalizability and robustness.

Generalizability

To showcase the versatility and resilience of our proposed methodologies, we conduct thorough evaluations across four distinct objects. These evaluations involve systematic exploration, incorporating various weight combinations such as 5g and 50g weights, along with different ball radii (35 mm and 30 mm). Physical simulation parameters for the three-fingered robotic hand and parameters of the ball for different objects are given in Tables S1 and S2.

Within the scope of our investigation, we carry out a series of exhaustive experiments for each of the four distinct objects, denoted as O1 through O4. These experiments are carefully designed, incorporating five diverse curricula identified as C1 through C5. A rigorous evaluation, comprising a total of 60 trials for each curriculum, provides valuable insights into the adaptability and performance of our methods. In addition, we systematically explored two tactile conditions (No-tactile vs. 3D-force) across the five distinct curricula. The subsequent section

Parameter	Value
Palm Mass	100 g
Finger Mass	68 g
Link length	50 mm
Phalanx diameter	10 mm
Palm width	20 mm
Palm diameter	120 mm
Initial hand height ($z_h(0)$)	200 mm
Maximum translation (z_h)	130 mm
Joint damping	5.5×10^{-6} N·s/mm
Joint limits (q_1)	$[-45^\circ, 45^\circ]$
Joint limits (q_2)	$[-90^\circ, 0^\circ]$

Table S1: Physical simulation parameters for the three-fingered robotic hand for all objects. The overall mass of the hand comprises the combined masses of the three fingers and the palm is 304 g.

provides detailed simulation parameters for all four distinct objects.

Simulation parameters

Physical parameters for all entities in the simulation must be specified (either directly or indirectly) including size, mass, stiffness, and damping. Relevant simulation parameters for the hand and ball for four objects are provided in Table S1 and S2.

Tactile information.

Tactile information is provided to the learning policy via the tactile force state vector for the simulated robotic hand ($s_{h,f}$). Contact force is measured at the pad of the fingers (where the tactile area is defined, see (Fig. 1A)), and directed from the finger towards the ball (δI), which at most contains the full touch force vector $\mathbf{f} = [f_{t1}, f_{t2}, f_n]$. Each finger-pad outputs the tactile information independent from other pads. The learning policy has two options for tactile information, which are added to the state vector for the hand (s_h). These are No-tactile or 3D-force information. Table S3 indicates the detailed tactile information for the two tactile

Object	Parameter	Value
Object 1	Mass	50 g
	Radius	35 mm
	Desired height (z_d)	60 mm
	Height (z_b)	35 mm
	Stiffness in x direction	5×10^{-3} N/mm
	Damping in x direction	3.5×10^{-4} N·s/mm
	Damping in z direction	5×10^{-4} N·s/mm
	Damping about y direction	5×10^{-3} N·s/rad
Object 2	Mass	50 g
	Radius	30 mm
	Desired height (z_d)	60 mm
	Height (z_b)	30 mm
	Stiffness in x direction	5×10^{-3} N/mm
	Damping in x direction	3.5×10^{-4} N·s/mm
	Damping in z direction	2×10^{-4} N·s/mm
	Damping about y direction	5×10^{-3} N·s/rad
Object 3	Mass	5 g
	Radius	35 mm
	Desired height (z_d)	60 mm
	Height (z_b)	35 mm
	Stiffness in x direction	1×10^{-3} N/mm
	Damping in x direction	7×10^{-5} N·s/mm
	Damping in z direction	2×10^{-4} N·s/mm
	Damping about y direction	1×10^{-3} N·s/rad
Object 4	Mass	5 g
	Radius	30 mm
	Desired height (z_d)	60 mm
	Height (z_b)	30 mm
	Stiffness in x direction	1×10^{-3} N/mm
	Damping in x direction	7×10^{-5} N·s/mm
	Damping in z direction	2×10^{-4} N·s/mm
	Damping about y direction	1×10^{-3} N·s/rad

Table S2: Simulation parameters for the ball across all objects, detailing size, mass, stiffness, and damping.

options.

Tactile Information in State Variable	
No-tactile	3D-force
$s_{h,f} = 0$	$s_{h,f} = [f_{t1}, f_{t2}, f_n]$

Table S3: Tactile information options available to the learning policy.

Learning manipulation through PPO

Proximal Policy Optimization (PPO) comprises a collection of policy gradient methods designed to optimize a surrogate objective function through multiple minibatch updates per data sample (52, 89). PPO leverages the Actor-Critic Model, which consists of two Deep Neural Networks. One network is responsible for action selection (the actor), while the other handles reward estimation (the critic). We conducted a series of experiments to assess the performance of the PPO algorithm in our environment. The following section provides an in-depth examination of the hyperparameters for PPO, which were determined through trial and error, and a comprehensive analysis of the resulting performances.

PPO Hyper-parameter

The robotic agent learns manipulation using OpenAI Baselines’ PPO1 implementation as the RL algorithm. We meticulously select hyperparameters to emphasize rewards at each simulation time step over those at the end of episodes. Additionally, we employ a learning scheduler customized for the established proximal-policy optimization (PPO) algorithm (52). The hyperparameters for the PPO algorithm are listed in Table S4. Non-default hyper-parameters are chosen empirically through trial and error and careful examination of resulting performances.

Linear Rate Scheduler

Employing a fixed and unchanging learning rate throughout the entirety of training often falls short of achieving an optimal model. Recognizing the critical role of learning rate schedules

hyper-parameter	Value
Adam stepsize	1×10^{-5}
Number of epochs	8
Discount (γ)	0.99
Entropy coefficient	0.02
Advantage estimation (λ)	0.85
Minibatch size	64

Table S4: Proximal-policy optimization (PPO) hyper-parameters

in model performance, researchers have extensively investigated methods for effectively and automatically tuning the learning rate for stochastic optimizers. This is because stochastic optimizers are highly sensitive to the learning rate scheduling (78, 79).

The Constant Learning Rate Dilemma

Implementing a constant learning rate, while straightforward in stationary or slowly changing environments, presents challenges in dynamic contexts. Sensitivity to the initial rate choice is a primary limitation, with high rates risking unstable training and low rates leading to protracted convergence or suboptimal solutions. Achieving the right balance becomes a nuanced trial-and-error process, demanding significant time and effort. Furthermore, the fixed learning rate lacks adaptability to evolving learning dynamics, resulting in suboptimal training efficiency and performance.

The Appeal of Linearly Changing Learning Rates.

In response to these challenges, we transitioned to a linearly changing learning rate schedule, offering several advantages. This adaptive approach dynamically adjusts rates throughout training, expediting the learning process with higher initial rates and facilitating stable convergence through decrementing rates during the later stages.

One significant advantage is the reduced sensitivity to the initial rate choice, minimizing the risk of divergence. The linearly changing learning rate promotes efficient exploration by encouraging policy discovery in the early stages and exploitation for optimal performance dur-

ing convergence. Its adaptive nature contributes to faster convergence compared to a fixed rate, effectively navigating both exploratory and exploitative learning phases.

Moreover, the linear schedule imparts robustness against variations in task difficulty or environmental changes, automatically adjusting to maintain training stability. In summary, our transition to a linearly changing learning rate in the PPO implementation aims to enhance training stability, expedite convergence, and improve adaptability in dynamic environments. This strategy aligns with our goal of efficiently training the agent for effective in-hand manipulation and contributes to the exploration of learning rate scheduling strategies in stochastic optimization. Additionally, we present results for Curriculum 3 [L+R | L+R], comparing training dynamics under both linear and constant learning rate scenarios to gain insights into the effectiveness of different strategies in our experimental setup.

Supplementary Results

Caption for Supplementary Video

Video of the robotic hand interacting with the ball after undergoing learning The ability of the robotic hand to manipulate was evaluated after 2,000 training episodes. Learning strategy influences manipulation performance. How we started the learning had a direct effect on the same end goal. Here we showcase an example of learning in C3 [L+R|L+R] and C5 [L+R|L]. Performance representative of each Curriculum while using 3D-force tactile information are shown in the video. Interesting insights were gleaned from each approach to autonomous learning of manipulation.

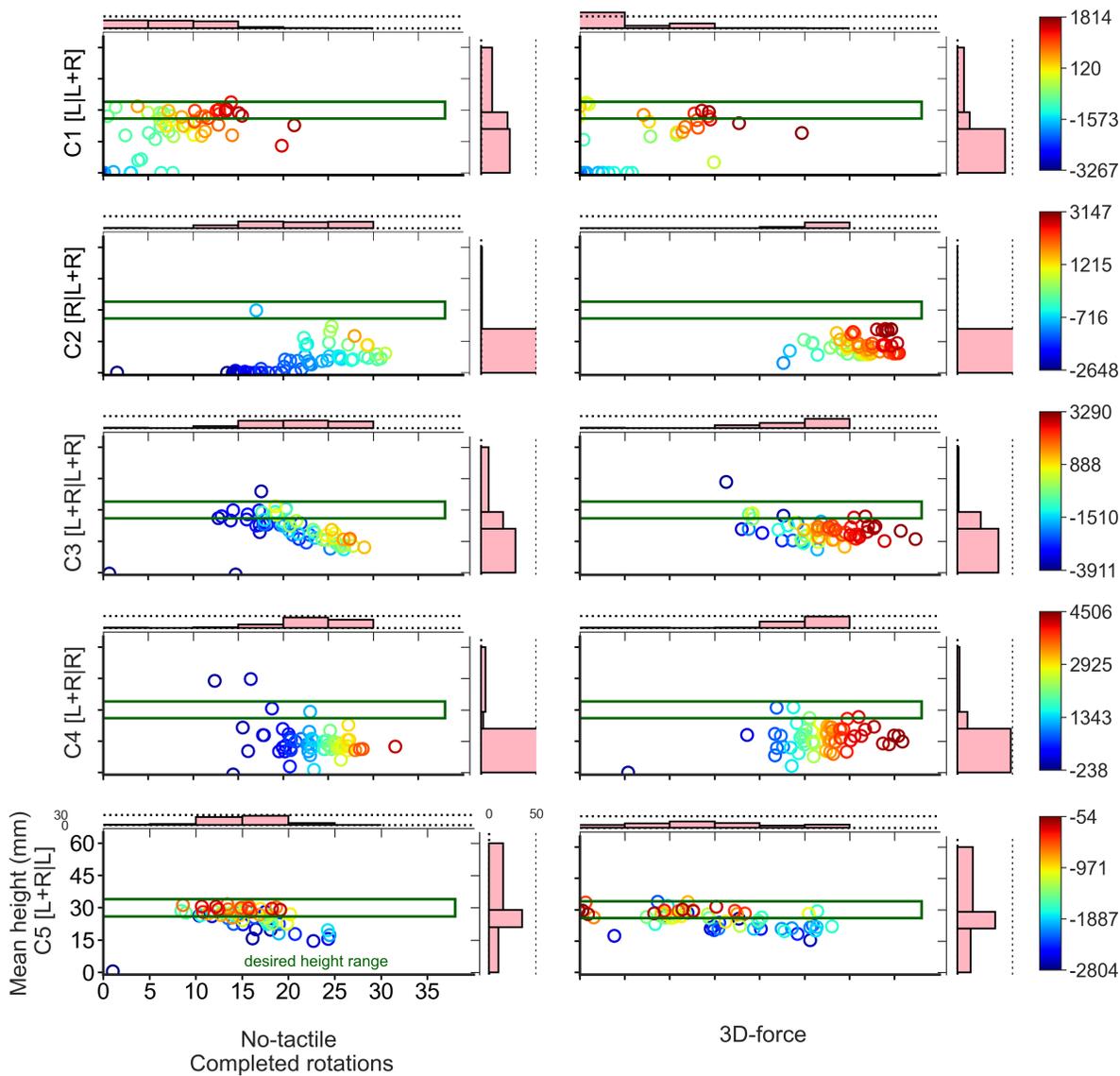


Figure S2: **Evaluation of performance across all curricula and both tactile information options for Object 2: 50g, 30 mm.** The joint distribution illustrates the performance during the final episode of 60 trial runs (showcasing the mean ball height (mm) versus the number of completed rotations). The color-coded cumulative reward for the last episode of each independent run (refer to equation 1) corresponds to different curricula. Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height (30 ± 4 mm).

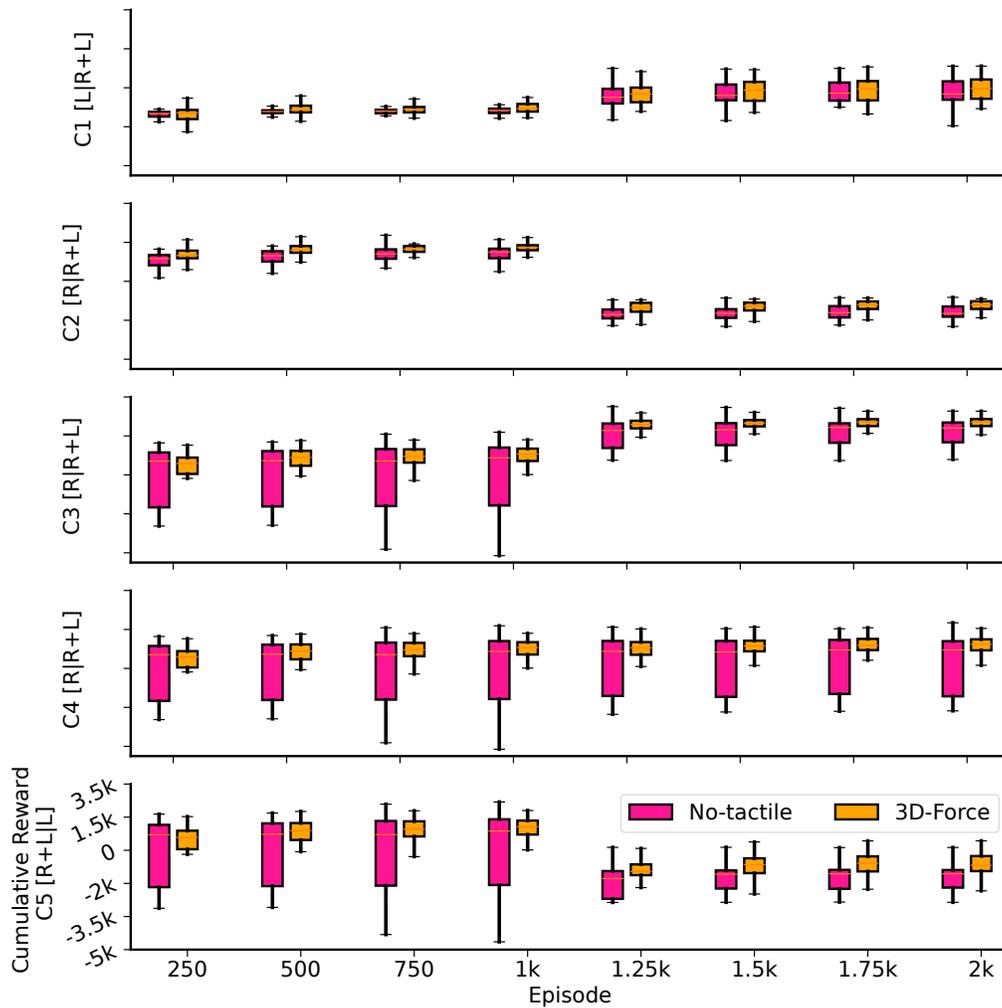


Figure S3: Cumulative reward for each representative episode across all curricula and both tactile information options for Object 2: 50 g, 30 mm. Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart.

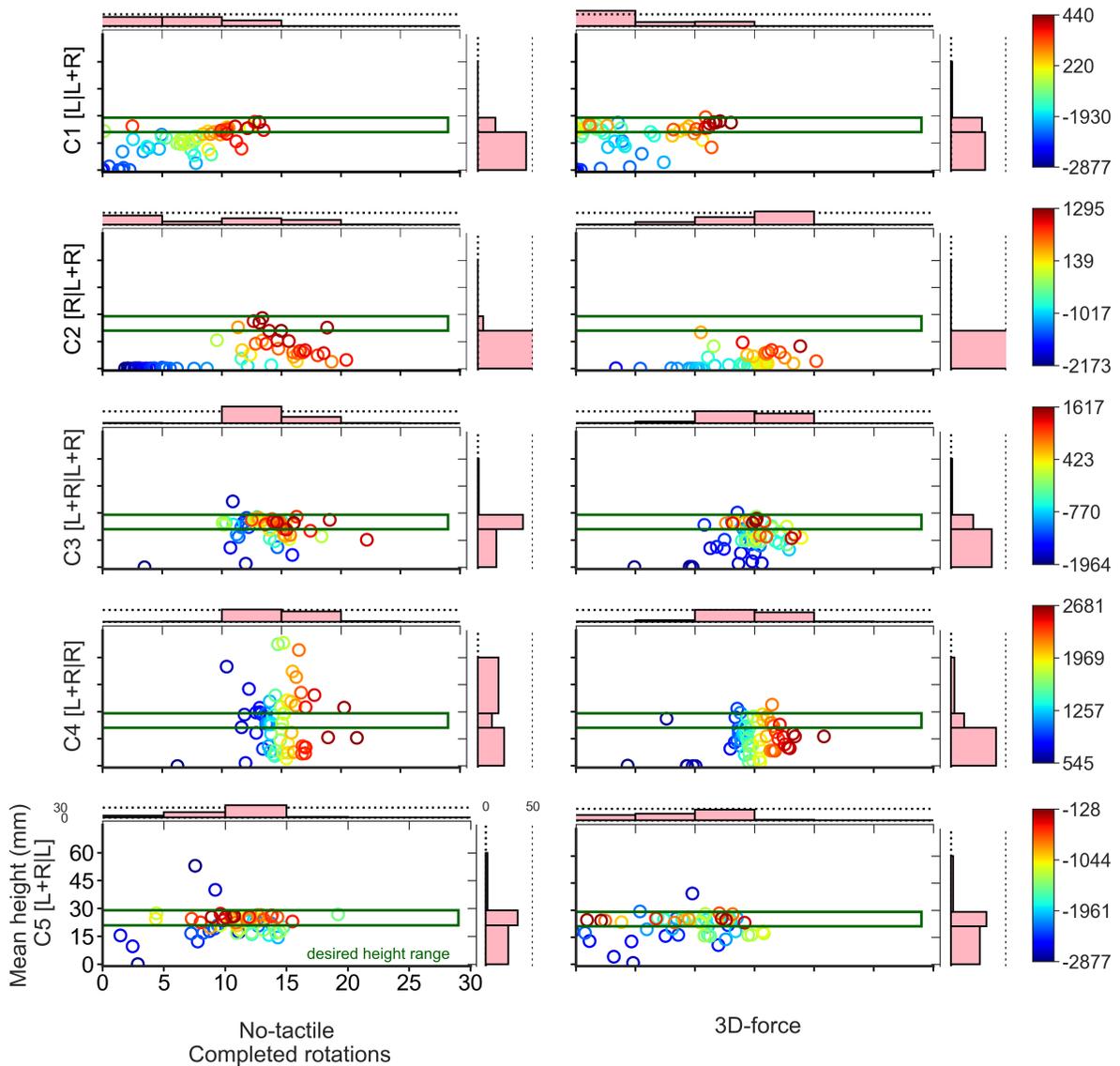


Figure S4: **Evaluation of performance across all curricula and both tactile information options for Object 3: 5 g, 35 mm.** The joint distribution illustrates the performance during the final episode of 60 trial runs (showcasing the mean ball height (mm) versus the number of completed rotations). The color-coded cumulative reward for the last episode of each independent run (refer to equation 1) corresponds to different curricula. Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height (25 ± 4 mm).

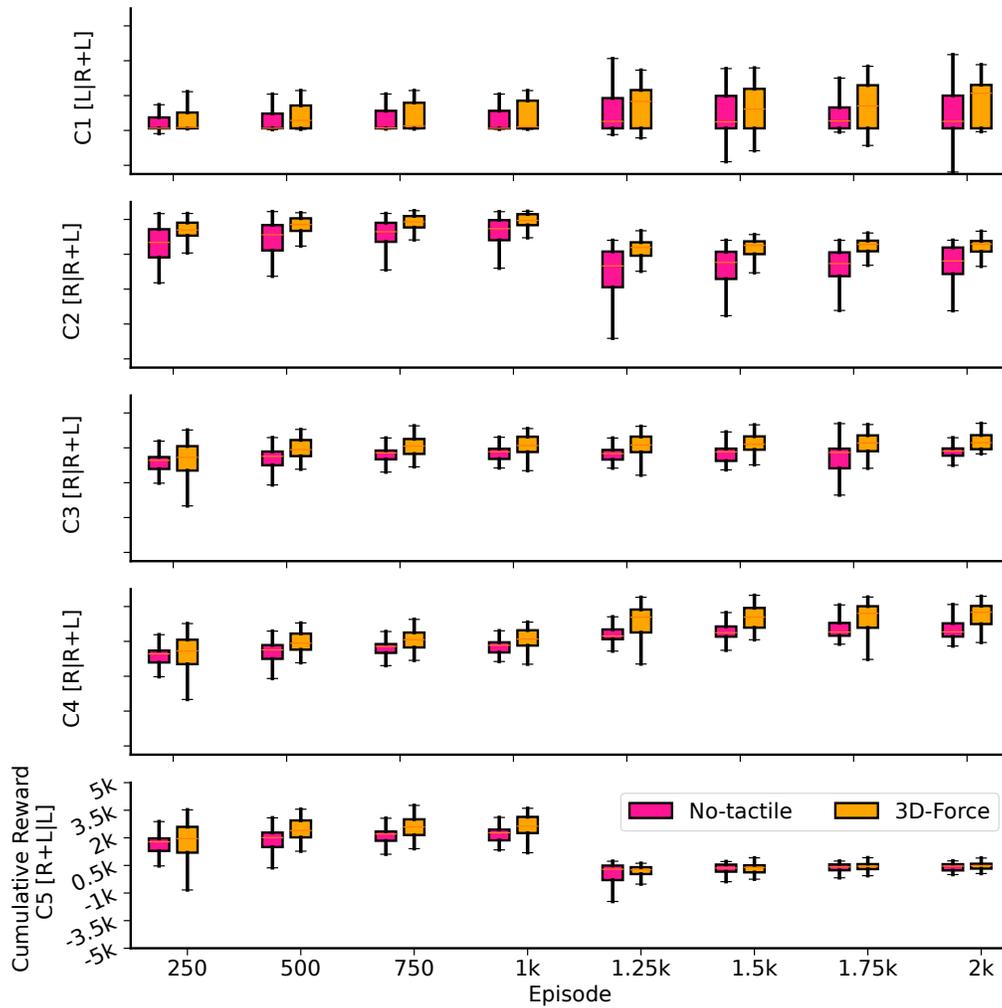


Figure S5: **Cumulative reward for each representative episode across all curricula and both tactile information options for Object 3: 5 g, 35 mm.** Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart.

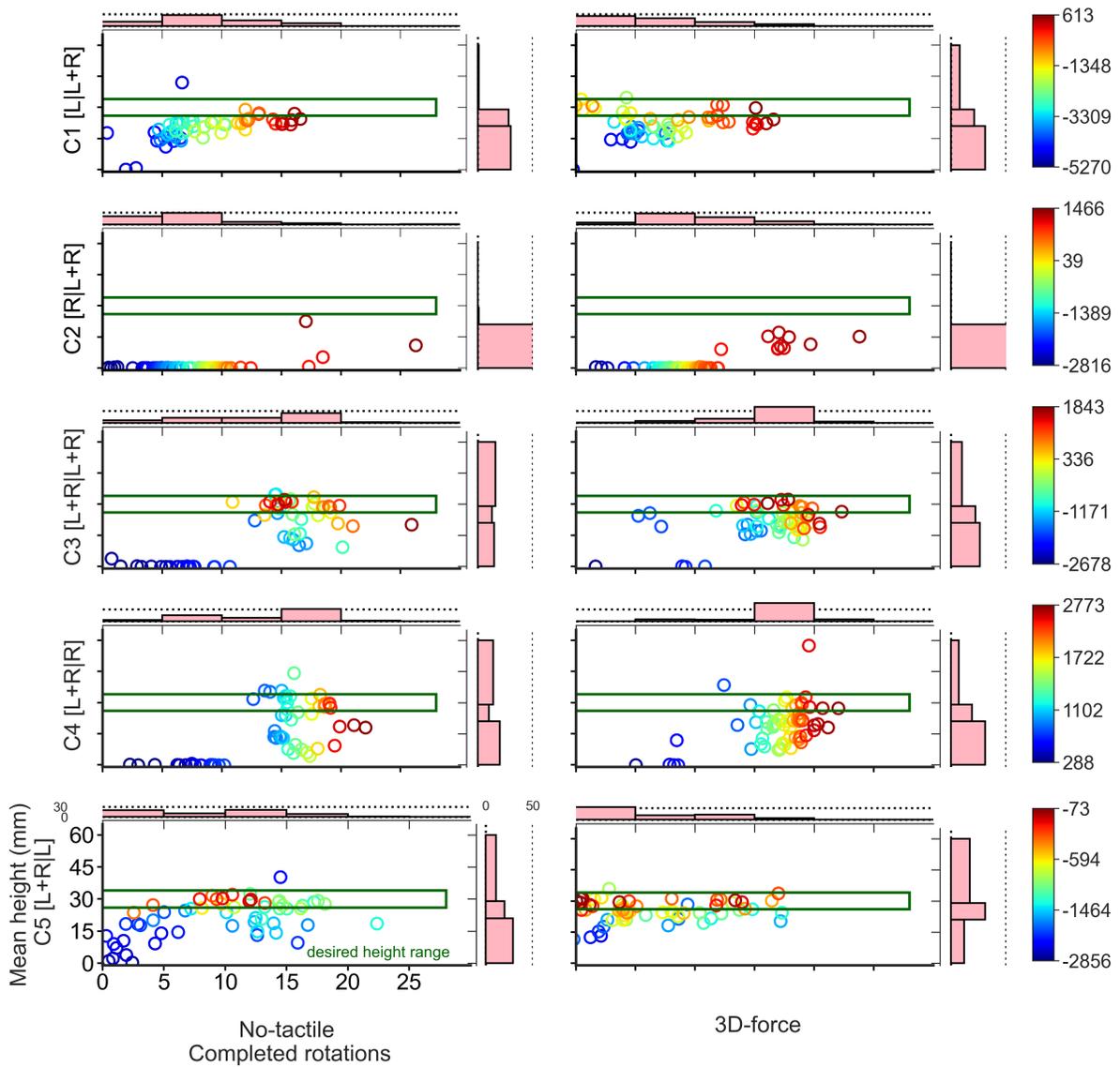


Figure S6: **Evaluation of performance across all curricula and both tactile information options for Object 4: 5g, 30mm.** The joint distribution illustrates the performance during the final episode of 60 trial runs (showcasing the mean ball height (mm) versus the number of completed rotations). The color-coded cumulative reward for the last episode of each independent run (refer to equation 1) corresponds to different curricula. Note that the desired manipulation performance is represented by those points inside the green box defining the desired ball height (30 ± 4 mm).

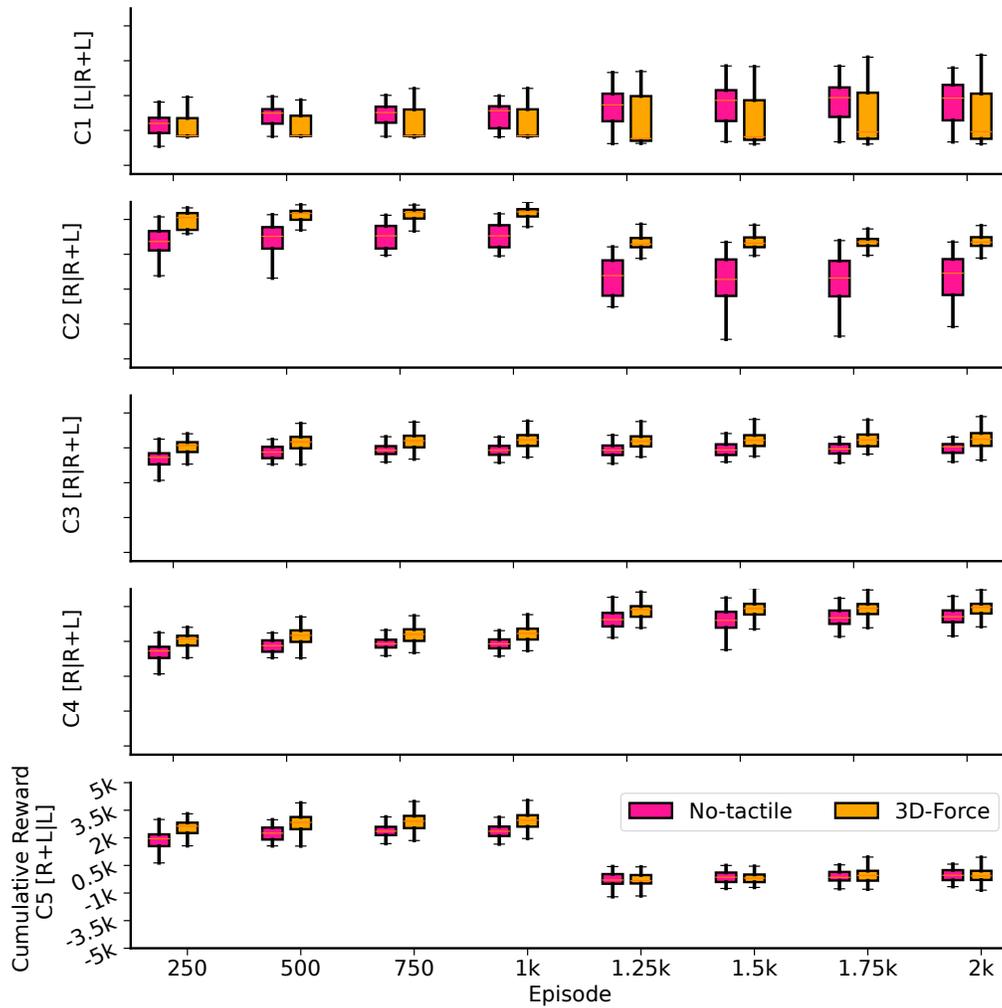


Figure S7: **Cumulative reward for each representative episode across all curricula and both tactile information options for Object 4: 5g, 30mm.** Boxplots, with median, across tactile information options for 60 MC runs at eight representative episodes, 250 episodes apart.