MANIPULATION

Learning contact-rich whole-body manipulation with example-guided reinforcement learning

Jose A. Barreiros†, Aykut Özgün Önol†, Mengchao Zhang†, Sam Creasey, Aimee Goncalves, Andrew Beaulieu, Aditya Bhat, Kate M. Tsui, Alex Alspach*

Humans use diverse skills and strategies to effectively manipulate various objects, ranging from dexterous in-hand manipulation (fine motor skills) to complex whole-body manipulation (gross motor skills). The latter involves full-body engagement and extensive contact with various body parts beyond just the hands, where the compliance of our skin and muscles plays a crucial role in increasing contact stability and mitigating uncertainty. For robots, synthesizing these contact-rich behaviors has fundamental challenges because of the rapidly growing combinatorics inherent to this amount of contact, making explicit reasoning about all contact interactions intractable. We explore the use of example-guided reinforcement learning to generate robust whole-body skills for the manipulation of large and unwieldy objects. Our method's effectiveness is demonstrated on Toyota Research Institute's Punyo robot, a humanoid upper body with highly deformable, pressure-sensing skin. Training was conducted in simulation with only a single example motion per object manipulation task, and policies were easily transferred to hardware owing to domain randomization and the robot's compliance. The resulting agent can manipulate various everyday objects, such as a water jug and large boxes, in a similar fashion to the example motion. In addition, we show blind dexterous whole-body manipulation, relying solely on proprioceptive and tactile feedback without object pose tracking. Our analysis highlights the critical role of compliance in facilitating whole-body manipulation with humanoid robots.

INTRODUCTION

Humans display an extraordinary ability to manipulate objects with wide-ranging sizes and shapes leveraging the dexterity of our end effectors (hands), full-body engagement, and interactions with the environment such as bracing (*1*, *2*). The taxonomy of human dexterity includes both fine and gross manipulation skills (*3*, *4*). Although the former (fine dexterity) has been extensively studied in robotics, gross dexterity is a less explored domain. Gross motor skills in humans and other primates involve engaging the whole body through the activation of large muscle groups including the arms, trunk, and legs. These skills enable everyday functions for humans such as carrying grocery bags, moving a couch around a living room, reorienting heavy canisters, and holding a baby (even while climbing a tree in the case of orangutans).

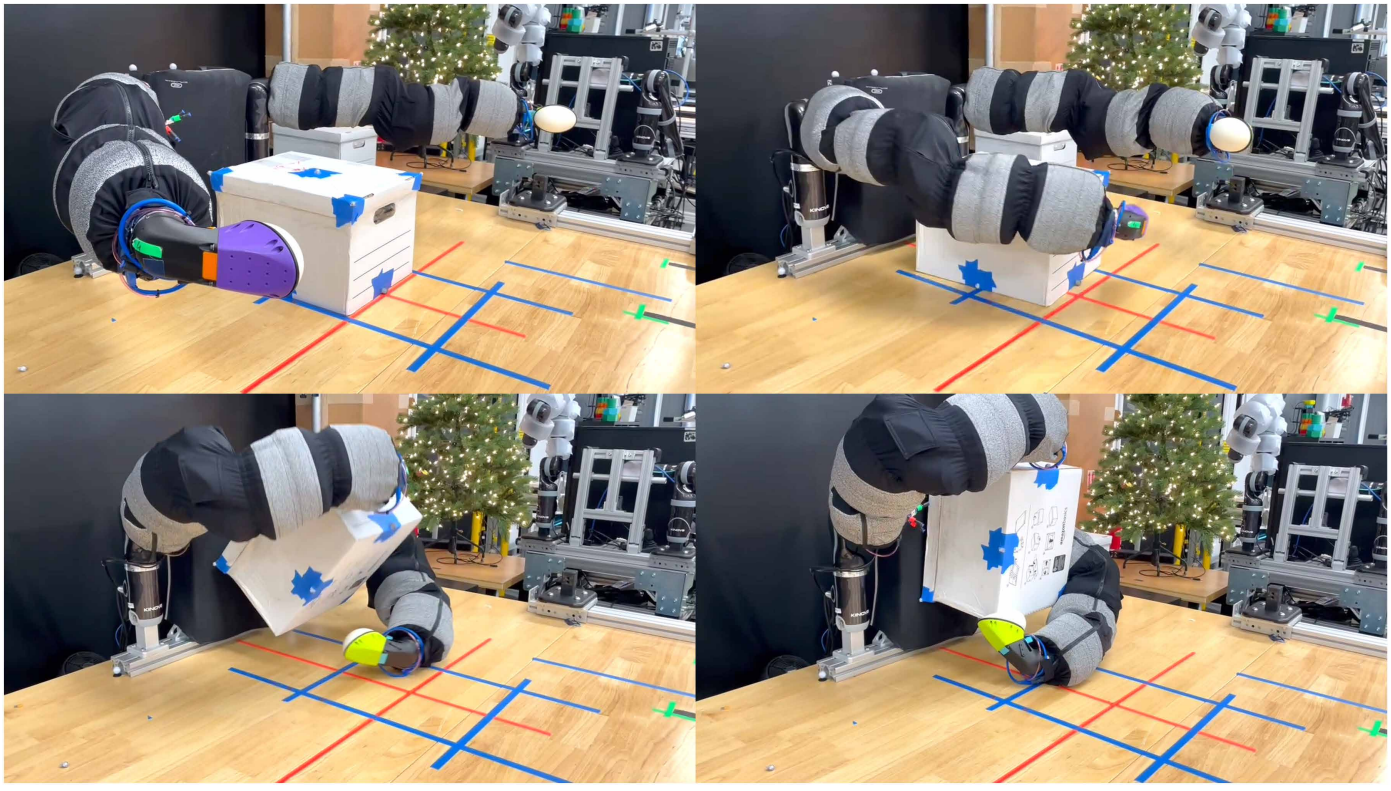
In robotics, there has been a long-standing endeavor to replicate and integrate these dexterous human skills. To enable the whole-body manipulation of big and unwieldy objects, we developed a hardware platform named Punyo (*5*). Punyo is an upper-body humanoid robot with soft "flesh" over a hard dual-arm "skeleton." Punyo's end effectors, arms, and chest are covered with highly deformable, pressure-sensing materials so that it can feel contact and react appropriately. This softness is provided by passively compliant, air-filled bladders under a fabric cover, enabling Punyo to conform to the surfaces it touches. As a result, contact stability is enhanced through increased friction and more uniform force distribution. The latest version of Punyo introduced here is also equipped with active compliance via a joint-level admittance controller that allows the robot to comply with generalized external forces.

The challenges of whole-body manipulation with Punyo are multifaceted. First, the deformable structures covering the robot are difficult to model. Second, the object properties, such as inertia and friction, are uncertain. Third, motivated by future downstream applications that include interacting with people, the Punyo robot needs to operate with human-like natural motion (*6*–*8*), also referred to as legible or interpretable motion in the related literature (*8*, *9*), to align the robot's behavior with human expectations and enhance the perceived and actual safety (*9*). Furthermore, minimal human supervision for skill acquisition is desirable for operating in open-world settings and rapidly accumulating new skills.

For instance, consider the scenario delineated in Movie 1, in which Punyo is tasked with moving a large box to a desired pose. For this purpose, the robot might first pull the box toward itself using both arms, then use one hand to pivot the box by bracing it against the torso, and lastly translate it until it reaches the desired configuration. Devising a framework to systematically plan and control such whole-body manipulation behaviors presents a formidable challenge, primarily because of the complexity of contact combinatorics.

Model-based planning methods face challenges in contact-rich domains, where contact events lead to stiff and discontinuous numerics with numerous discrete modes, resulting in a nonconvex and disconnected search space. Nevertheless, planning through contact—synthesizing complex, contact-rich behaviors from only a high-level goal—remains a compelling and actively pursued challenge. The seminal work by Mordatch *et al.* (*10*) introduced contact-invariant optimization to generate dexterous manipulation motions. Subsequent contact-implicit optimization techniques (*11*–*13*) expanded on this idea, reducing reliance on heuristics and improving generality for manipulation planning. More recently, sampling- (*14*) and graph-based (*15*) methods have demonstrated strong scalability and effective exploration in high-dimensional spaces. Despite these advances, such approaches remain computationally intensive. Planning a single motion can take from several seconds to minutes, restricting these methods' practicality to offline settings, especially when dealing

Toyota Research Institute, Cambridge, MA, USA.
*Corresponding author. Email: alex.alspach@tri.global
†These authors contributed equally to this work.



Movie 1. Hardware rollout of the cardboard-box-pivot-and-lift policy.

with complex contact sequences. Once a motion is generated, open-loop execution remains sensitive to uncertainties in model parameters and object pose (*12*). Achieving robust execution necessitates closing the loop via state feedback. Complementary to closed-loop control is the exploitation of physical intelligence enabled by body compliance, which has been shown to ease the control problem by increasing the contact stability mechanically, as demonstrated in (*16*), and offloading computation that would otherwise fall to a high-level controller, such as the brain in animals.

Imitation learning (IL) has emerged as a promising pathway to tackle contact-rich manipulation, a prospect bolstered by recent progress in gradient-field learning methods (*17*–*19*). For instance, diffusion policy (DP) (*17*) exhibits great potential for learning complex manipulation tasks. This approach has been shown to generalize well to scenarios that are difficult to model and estimate states for, such as peeling vegetables, rolling dough, and making pancakes (*20*). Yet, applying this strategy requires a considerable number of expert demonstrations, facing challenges at scale because of the limitations of teleoperation methodologies. Predominantly tailored for end-effector tracking, existing teleoperation techniques struggle to effectively demonstrate complex whole-body maneuvers, with notable recent exceptions that leverage learned human-to-humanoid retargeting models (*21*, *22*). Even when resorting to whole-body teleoperation techniques, such as conventional kinematic retargeting (*23*, *24*), performing the same task repeatedly to train a proficient policy may be tedious.

For more structured tasks that are possible to simulate, reinforcement learning (RL) has been proven to yield notable outcomes (*25*–*28*). These advancements frequently hinge on the availability of task-specific insights, either in the form of well-defined reward functions or expert guidance. As a means to streamline the process of reward design, guided RL (*29*) capitalizes on pre-existing knowledge inferred from data to improve the efficiency and efficacy of the RL process. In particular, example-guided RL (EGRL) aims to combine motion imitation with task-based rewarding (standard RL) and has shown promise in aiding exploration by instilling a desired motion style, thus accelerating learning and easing reward shaping (*30*–*40*). As an example, generative adversarial imitation learning (GAIL) (*41*) effectively integrates a generative adversarial network (*42*), with RL incorporating a discriminator that evaluates the resemblance between the policy and example motions. However, GAIL's direct applicability is limited to cases when the demonstrator's actions are observable. Addressing this, a recent work introduced adversarial motion priors (AMPs) (*43*), which leverages the GAIL framework to discern whether a state transition is a sample from the example motions or one generated by the agent. This approach does not require designing imitation objectives or motion selection mechanisms, and it can automatically synthesize a policy that completes a desired high-level task given a set of example motions and a generic reward function. Owing to its promise to impose motion style on the RL policy without the burden of reward engineering, the AMP idea (originally proposed for character animation) has caught the attention of the robotics community and has led to impressive results for quadruped locomotion (*44*–*48*). AMP has been applied to generate whole-body motions for animated characters (*49*), such as kicking and striking, where the fidelity of physics and the manipulation of objects are not critical constraints. This approach's translation to manipulation tasks in the real world, which involves unique

challenges arising from intricate contact interactions and constraints between the robot and the object, had not previously been explored in terms of its applicability and effectiveness.

In this Research Article, we demonstrate that our controller (Fig. 1A), based on AMP combined with Punyo's passive and active compliance, tackles the aforementioned challenges in whole-body manipulation and enables robust execution of complex whole-body manipulation tasks on a humanoid robot. Leveraging the versatility of AMP's formulation to accept any state-based trajectory as an example motion, we used both teleoperation data collected in simulation and, separately, motion plans from a model-based planner and evaluated the trade-offs of the choices of motion source. We show that a single teleoperated demonstration collected in simulation is sufficient to train policies for stylized, contact-rich motions. We further show that even a rough, dynamically infeasible motion plan can serve as an effective example, allowing the method to scale beyond the constraints of teleoperation. Using an asymmetric actor-critic policy, we trained policies with privileged information that is readily available in simulation while relying only on proprioceptive and tactile inputs at inference time, enabling blind manipulation without the need for object pose tracking. The resulting policies successfully complete various real-world tasks (Fig. 2), including lifting a jug over the shoulder, flipping it upside down, and reorienting a large box. Last, we analyzed how the example motion quality affects learning performance and evaluated the influence of compliance and domain randomization (DR) on the resultant policies' robustness, measured by success rate under randomized initial object pose. Movie 2 provides a visual summary of the results.

RESULTS

We conducted experiments on the Punyo robot (see the "Punyo" section in Materials and Methods) for the following manipulation tasks: plastic-box-hug-lift, plastic-box-paw-lift, cardboard-box-pivot-and-lift, jug-over-shoulder, jug-rotate, and cube-lift (see the "Manipulation tasks" section in Materials and Methods). For all experiments, task-specific policies for each manipulation scenario were trained separately, and unless explicitly stated otherwise, DR was used during policy training, and the object pose was measured via an OptiTrack (*50*) motion capture system with active markers.

Versatility in the source of example motions

The proposed EGRL formulation is agnostic to the source of example motions, which are required only to be a sequence of robot states. In this work, we used example motions obtained by teleoperating the robot in simulation and motions generated by a model-based planner, named global quasidynamic planner (GQDP) (*14*). In the following paragraphs, we discuss the details of each example motion choice and a comparison.

Teleoperation data

A direct way to generate example motions is to teleoperate the robot. However, for whole-body manipulation, controlling both arms simultaneously presents a substantial challenge because of the mapping of the many degrees of freedom of the robot to a limited number of "knobs" that a human teleoperator can effectively handle, considering the precise timing and coordination that is needed for complex whole-body manipulation. Inspired by the concept of Eigengrasp (*51*), we broke down whole-body motions into synergies (as described in the "Eigengrasp-based whole-body teleoperation" section in Materials and Methods) that were combined linearly and controlled via a standard game pad interface.

During teleoperation, we logged the robot's configuration $\mathbf{q}_t^{\mathrm{a}}$, the object's configuration $\mathbf{q}_t^{\mathrm{u}}$, and the position commands $\mathbf{a}_t$ issued to the robot. In addition, we recorded the pressure readings p_t from the robot's end effectors (paws) and all contact-pressure sensors along the robot's arms. We collected teleoperation data in simulation using the high-fidelity hydroelastic contact model (*52*, *53*) in Drake (*54*). This choice was motivated by the ease of initializing the simulated environment at any given state at reset time, which requires manual effort in the real world.

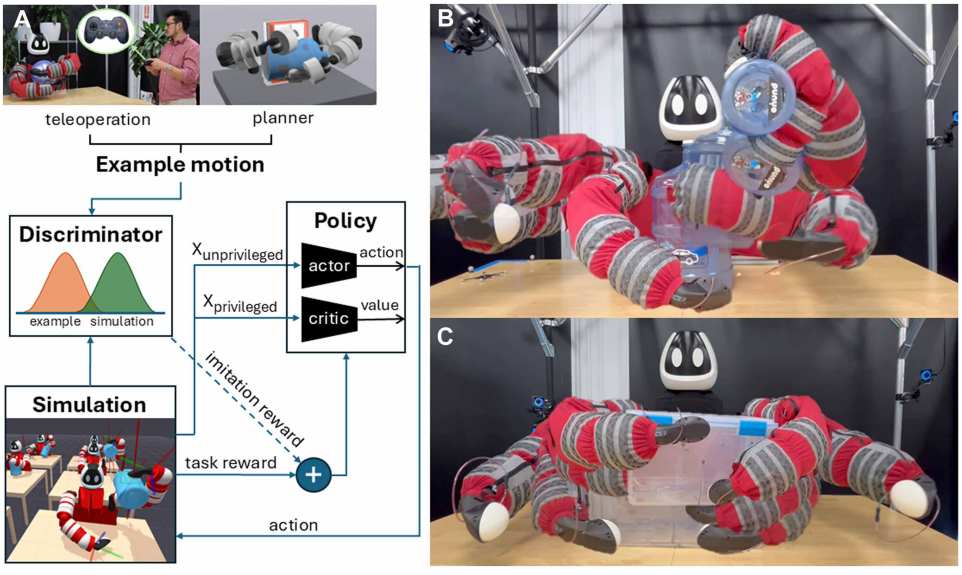


Fig. 1. Dexterous whole-body manipulation with the Punyo robot. (**A**) Given an example motion, which can be either a teleoperation demonstration or a planned trajectory using a model-based planner, a feedback control policy is trained along with a discriminator in simulation using example-guided RL to allow the robot to perform dexterous whole-body manipulation tasks that follow the prescribed motion style in the example motion. (**B** and **C**) Executions of the control policies trained using the method shown in (A) with the Punyo robot for the jug-over-shoulder task (B) and the plastic-box-hug-and-lift task (C); the policies are trained in simulation and transferred to the real world without any postprocessing.

Motion plans

Another approach to obtaining example motions is through motion planning. We selected GQDP because of its capability for synthesizing long-horizon behaviors with multiple intermittent contacts. This approach assumes quasidynamic mechanics to reduce the problem into the configuration space, and it uses a contact smoothing scheme along with the locally linearized model to derive a reachability metric. As a result, the problem of planning through contact, given the initial and desired configurations of the object, can be effectively addressed using a sampling-based planner, specifically rapidly exploring random tree (RRT) (*55*).

The coarse path generated by RRT is then refined using trajectory optimization to output a trajectory denoted as

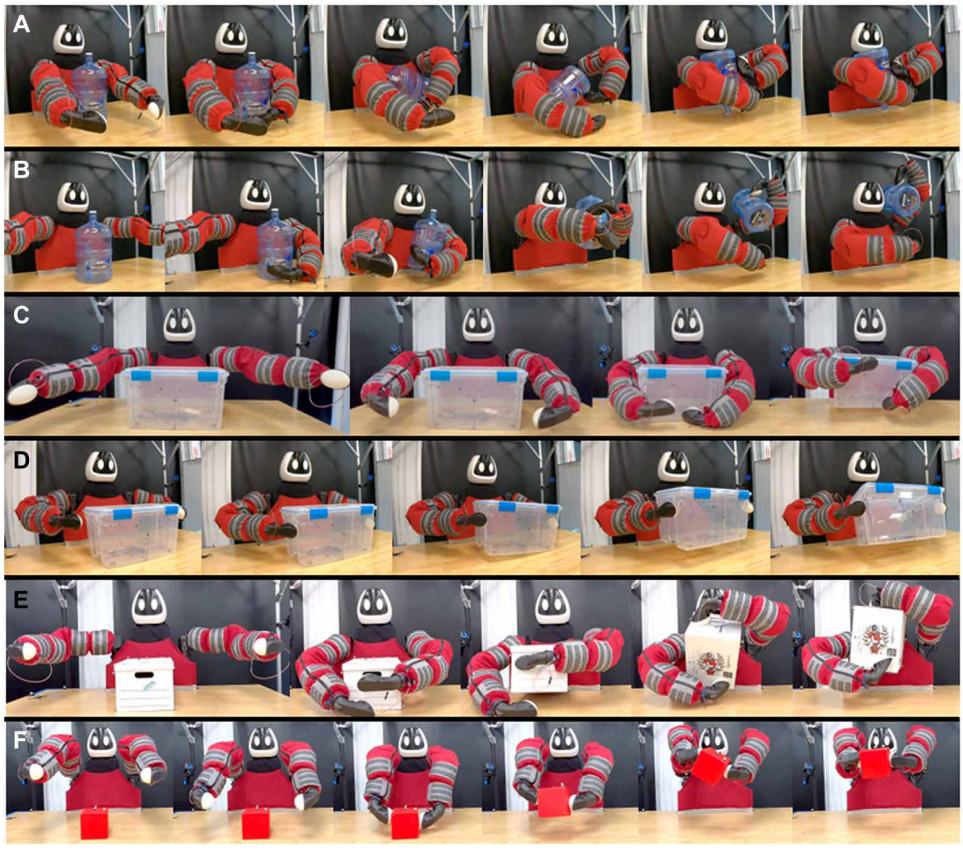


Fig. 2. Dexterous whole-body, bimanual manipulation of various large daily objects. Sequence of keyframes showing policy rollouts on the Punyo robot for (**A**) rotating a water jug 180°, jug-rotate; (**B**) lifting a water jug over its shoulder, jug-over-shoulder; (**C**) wrapping and lifting a plastic box with its arms, plastic-box-hug-lift; (**D**) lifting a plastic box using both hands, plastic-box-paw-lift; (**E**) lifting and pivoting a cardboard box, cardboard-box-pivot-and-lift; and (**F**) lifting a plastic cube with the end effectors, cube-lift.

$$\mathcal{T}' = \left\{ \left(\mathbf{q}_t^{\mathrm{a}}, \mathbf{q}_t^{\mathrm{u}}, \mathbf{q}_{\mathrm{cmd},t}^{\mathrm{a}} \right) \mid t = 0, \ldots, T \right\} \quad (1)$$

The refinement step eliminates nonphysical artifacts caused by contact smoothing and large time steps used for the search and improves the path that may be nonsmooth because of the random nature of RRT. In this context, $\mathbf{q}_t^{\mathrm{a}}$ and $\mathbf{q}_t^{\mathrm{u}}$ represent the configurations of the actuated and unactuated degrees of freedom of the system corresponding to the robot and the object, respectively. $\mathbf{q}_{\mathrm{cmd},t}^{\mathrm{a}}$ denotes the robot position commands for a joint stiffness controller at each discrete time step t.

Notably, despite the refinement, the resulting plan may exhibit physically infeasible artifacts because of the quasidynamic assumption (*14*) breaking when the system moves at non-negligible speeds and a robot teleport issue, where the robot jumps in a single time step from one contact configuration with the object to another because of contact sampling. Given the robot teleport issue, the object is not guaranteed to be stable while the robot is transitioning between contact configurations. This implies that during the rollout of the planned trajectory, if the robot adheres to the planned waypoints $\mathbf{q}_t^{\mathrm{a}}$, then the object may not remain at the corresponding $\mathbf{q}_t^{\mathrm{u}}$, particularly in tasks where the object is not in static equilibrium.

Comparison

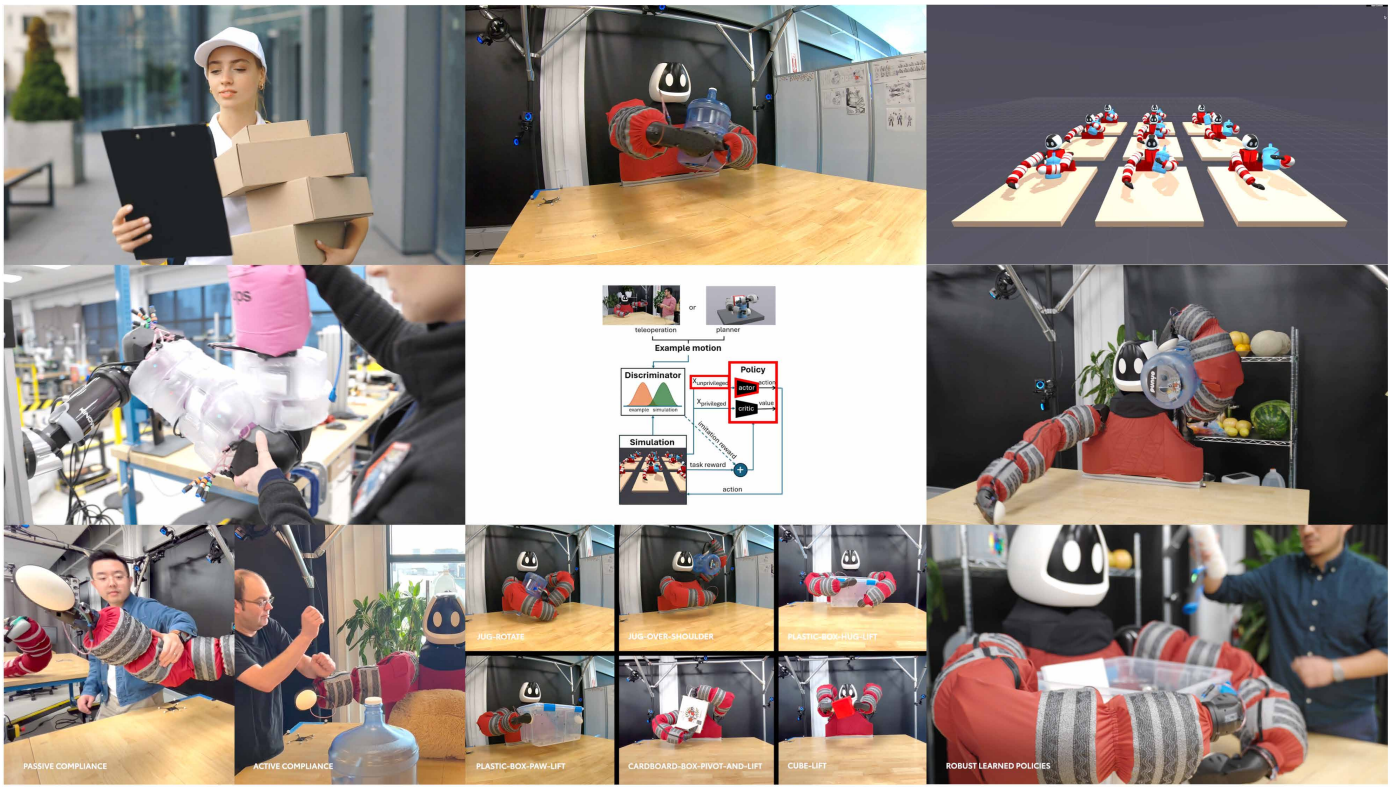
Note that there are advantages and limitations of the above two example motion sources. GQDP can generate motions that are potentially physically infeasible because of modeling simplifications. Moreover, the planning time can vary substantially, taking anywhere from several seconds to minutes depending on the task's complexity. In addition, without heuristics, these plans may not align with human expectations regarding motion style, and the user only has control over the initial and final state. Nevertheless, this planner is capable of producing complex motions that might be challenging to achieve through teleoperation because of interface limitations, and they do so requiring minimal human supervision.

On the other hand, teleoperation ensures that the desired motion style is incorporated into the generated motions, with expert operators able to create examples within a short time frame. Using our simplified teleoperation interface, an expert demonstrator can repetitively succeed at the jug-over-shoulder task shown in Fig. 3A. Such a task requires extensive fine-tuning when the motion is synthesized using GQDP. However, using the teleoperation interface may be tedious for the operator and require extensive training, especially for tasks demanding nontrivial coordination between the robot's arms, even with our simplified whole-body teleoperation interface. Figure 3B shows a task that could not be accomplished even by our most expert teleoperator.

Figure 3 (A and B) depicts a juxtaposition of the keyframes of example motion from teleoperation (Fig. 3A) and the model-based planner (Fig. 3B) and policy rollouts showing that the policy follows the style of each demonstration. Note that the overall style, such as the contact sequence, remains similar to the example motion with a few alterations that improve task performance and enable task completion because of the exploration capability of the RL process. A comparison of example motions generated by GQDP and teleoperation for the cardboard-box-pivot-and-lift task is shown in the "Comparison between GQDP and teleoperation example motions" section in the Supplementary Materials.

The influence of example motion quality

Empirical evidence from previous work (*46*) and our own suggests that rough, partial demonstrations might be enough guidance to synthesize successful policies using EGRL. To understand the extent to which the demonstration quality matters in the training of an effective policy, we perturbed the robot's configuration $\mathbf{q}_t^{\mathrm{a}}$ in the example motions in two ways. The first involved injecting noise sampled from a Gaussian distribution with zero mean and varying



Movie 2. Video summary of the methods.

SD. The second method entailed discarding intermediate configurations along the trajectory, retaining only every *k*th waypoint, and performing linear interpolation for the intervening time steps. These disruptions were intended to emulate scenarios of noisy sensor readings and signal loss. Figure 3C illustrates an example of each of these perturbations.

We tested the effect of the above perturbations on the training performance for three tasks: the jug-over-shoulder, cube-lift, and cardboard-box-pivot-and-lift tasks. The first two tasks used teleoperation demonstrations as example motions, whereas the cardboard-box-pivot-and-lift task used a GQDP plan as the example motion. We conducted this experiment for both types of motion sources because there is an inherent quality difference between a teleoperated motion and a motion plan. That is, the teleoperation data were collected by running a dynamic simulation using Drake's high-fidelity physics, whereas the model-based plan was generated assuming quasidynamics and included discontinuous robot motions, as detailed in the "Versatility in the source of example motions" section.

When introducing Gaussian noise, we used three different SD (σ_{noise}) magnitudes of 0.01, 0.1, and 0.5 radians. For discarding intermediate configurations, we retained every 10th, 20th, and 50th waypoint. For the nominal case and each of the above scenarios, we trained policies using six different random seeds. The success rates achieved by the trained policies using various perturbed example motions are depicted in Fig. 3D. Overall, the EGRL method demonstrated robustness to both types of perturbations, provided that the motion style remained consistent and was not substantially changed by the perturbations. In general, sparsely interpolating the example motion, such as every 50th waypoint, or introducing a large noise, such as $\sigma_{\text{noise}} = 0.5$ radians, to the example motion degraded the performance of the trained policy. However, small noise or dense interpolation did not result in a notable performance change. Hence, the proposed method was not highly sensitive to example motion quality. This result opens the door for alternative, potentially easier, ways of example motion collection, such as kinesthetic teaching with only a few key waypoints.

The effect of compliance on task success

We hypothesized that physical compliance helped to increase the robustness of whole-body manipulation by adding damping to contact interactions, increasing the stability of contacts at the various robot-object interfaces, and increasing the robot's overall ability to obtain and maintain form closure around an object (*56*). We leveraged two forms of compliance when performing manipulation tasks with Punyo: surface-level passive compliance from Punyo's soft outer covering (Fig. 4A) and joint-level active compliance from an optimization-based controller that altered the joint position commands according to the estimated external joint torque (Fig. 4B).

To quantify the influences of compliance on whole-body manipulation performance, first, we conducted hardware experiments comparing these two types of compliance on two different tasks. We considered the jug-over-shoulder and plastic-box-hug-lift tasks because of their contact-rich nature, where an object must be wrapped with multiple contacts across the robot's body. An important difference between these tasks is the amount of contact and internal forces. The jug-over-shoulder task uses only one arm, whereas the plastic-box-hug-lift uses both arms and the torso of the robot and hence produces larger internal forces.

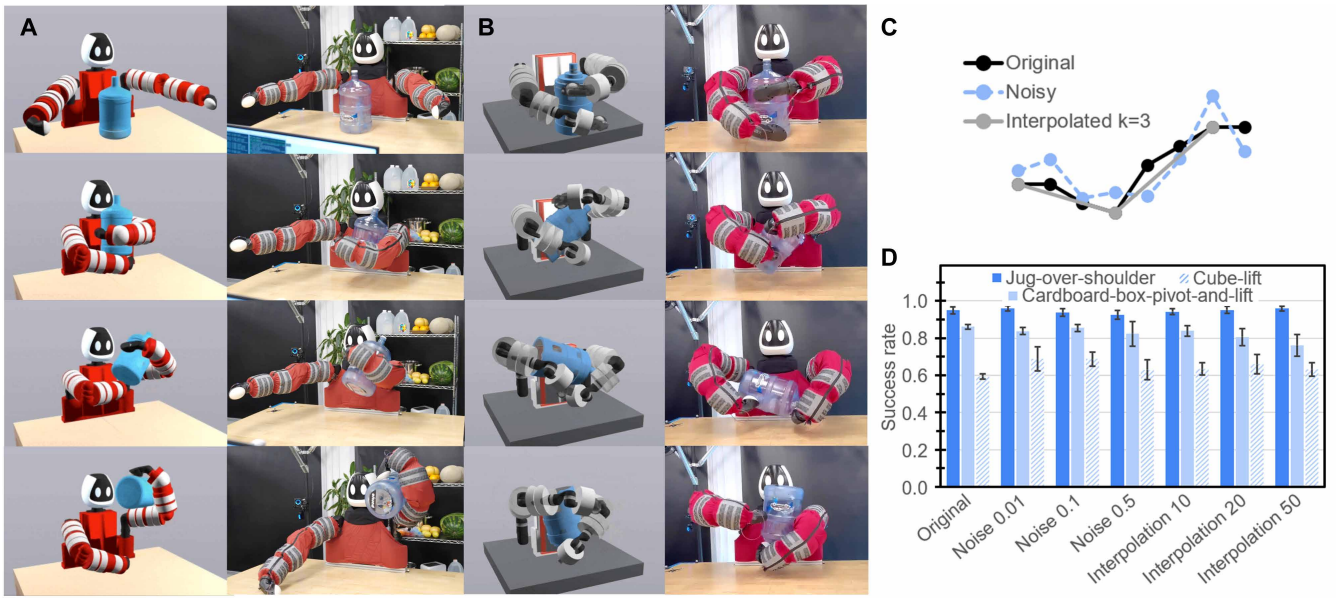


Fig. 3. Visualization of example motions and results of the demonstration quality analysis. Keyframes from the example motions and policy rollouts for the (**A**) jug-over-shoulder task using a teleoperation motion example and (**B**) jug-rotate task using a GQDP plan. (**C**) One-dimensional illustration of the noise injection and interpolation of a demonstration used to evaluate the (**D**) influence of demonstration quality on the task success for policies trained across six different random seeds. Error bars indicate the SD across six seeds (±1 SD, $n = 6$).

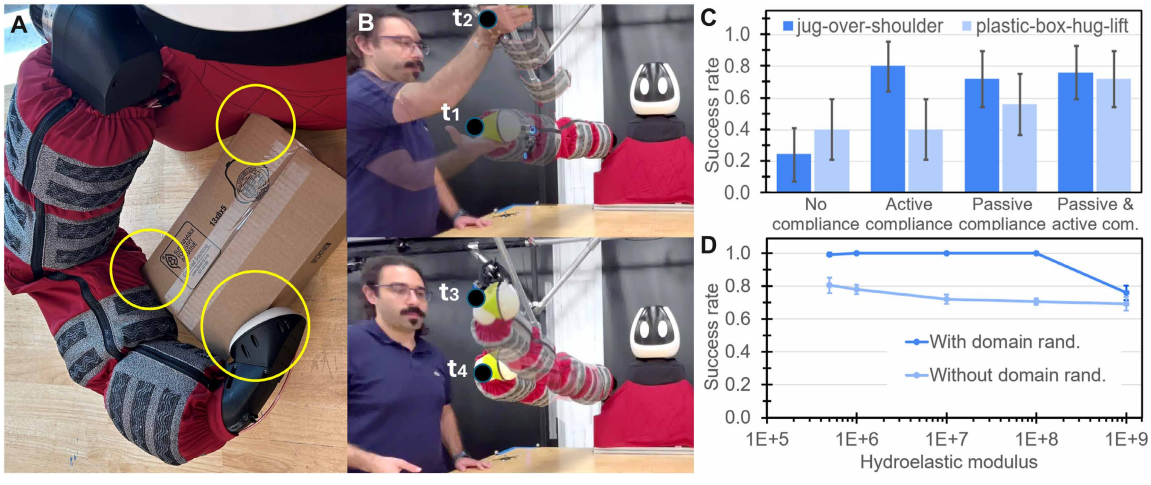


Fig. 4. Visualization of the different types of compliance of the Punyo robot and their effect on policy performance. (**A**) Depiction of passive compliance enabled by the addition of soft covers and air-filled chambers; yellow circles highlight the deformation of the body of the robot during manipulation. (**B**) Depiction of active compliance achieved through a compliant joint controller, with overlaid keyframes and waypoints (black circles) from times t1 to t4 of a video showing a robot commanded to stay still while responding to an upward push on its arm by a person. (**C**) The effect of active and passive compliance on the success of two whole-body manipulation tasks assessed by 25 hardware experiments per condition. Error bars indicate the 95% confidence intervals. (**D**) Effect of DR across varying body compliance levels on policy performance in simulation. Success rates are shown for policies trained with and without DR, which were evaluated in the Drake simulator under different compliance conditions, parametrized by hydroelastic modulus. For each domain-randomization condition, we trained three policies with different random seeds. Error bars represent ±1 SD ($n = 3$).

For the surface-level (passive) compliance evaluation, we built rigid equivalents of Punyo's deformable structures (namely, the arm air-filled chambers and soft paws; fig. S1) on the basis of geometry measured at nominal levels of inflation. We compared the success rate of a policy execution on hardware for the following conditions: only passive compliance (soft air-filled chambers, no compliant joint controller), only active compliance (rigid air-filled chamber stand-ins with the compliant joint controller), passive-and-active compliance (soft air-filled chambers with the compliant joint controller), and no compliance (rigid air-filled chamber stand-ins, no compliant joint controller). Note that for all conditions, the manufacturer's built-in joint controller was used (that is, the compliant controller was an additional component), and the robot was covered with the fabric sweater; thus, the surface friction remained the same.

Figure 4C shows the mean success rate over 25 hardware experiments for each of the four conditions and each task. Given that these hardware experiments were conducted using a policy trained with a single random seed, the error bars show the confidence intervals for a Bernoulli distribution at a 95% confidence level. The results indicate that adding any type of compliance (that is, passive, active, or both) is beneficial, showing an increase of 206% in the success rate, on average, compared with no compliance. We attributed this benefit to the aforementioned effects of compliance, namely, more damped, stable, and grippy contact interactions.

The form closure effect gained by passive compliance boosted the success rate for the plastic-box-hug-lift task substantially, from 40% (no compliance) to 56%. Moreover, an improved success rate of 72% was obtained when combined with active compliance. For this task, active compliance did not provide any benefit over the no-compliance condition. For the jug-over-shoulder task, we could not draw a conclusion between the active, passive, and active and passive compliance conditions because of highly overlapping confidence intervals. Nevertheless, the compliant modes yielded a 316% higher success rate, on average, compared with no compliance for this task.

Joints actively conforming with the external force (using the compliant controller; see the "Compliant joint controller" section in Materials and Methods) were helpful regardless of whether the robot was outfitted with soft or rigid contact bodies. Active compliance enabled the highest success rate for the jug-over-shoulder task, showing the efficacy of joint-level admittance when the contacts were contained within one arm. For the plastic-box-hug-lift task, on the other hand, active compliance alone did not help to increase the success rate. However, it provided another substantial advantage over no compliance in terms of robot safety. That is, with no compliance, robot torque-outs (joint torques exceeding rated limits) occurred in 14 of the 15 failed executions as a result of excessive contact forces being exerted by the robot. Both active and passive compliance were effective in preventing these torque-outs and, therefore, protecting the robot from potential damage due to overexerting the actuators.

Passive compliance versus DR

Another method used to improve the policy robustness during training was DR. We attributed our ability to deploy the simulation-trained policies directly on the hardware to both DR and robot compliance. However, their relative contribution in enabling the direct transfer from simulation to hardware remained unclear.

To quantify the effect of DR and compliance on the task success, we ran simulation experiments with varying stiffness for robot contact geometries using the high-fidelity hydroelastic contact model (*52*, *53*) in Drake (*54*) as a way to emulate different levels of passive compliance. We trained policies with and without DR for three different random seeds per condition (see the "Domain randomization" section in Materials and Methods) for the jug-over-shoulder task. We then evaluated the success rates of these policies for various levels of compliance by changing the hydroelastic modulus in the simulator—a parameter that serves as a proxy for Young's modulus. Higher hydroelastic values corresponded to stiffer contact interactions.

We ran 50 simulation experiments per policy (totaling 150 simulation rollouts per domain-randomization condition) with random initialization of the jug's position with respect to the table within a 10-cm square around the nominal pose. The average success rate along with the SD is shown in Fig. 4D. The results show that, for the same level of compliance, the policy trained with DR is substantially more robust to object pose uncertainty compared with the policy trained without DR. The compliance, due to a lower hydroelastic modulus, increased the tolerance of both policies to the jug placement uncertainty, which was consistent with the finding of the aforementioned hardware experiments, concluding that passive compliance enhanced the robustness. Yet, the success rate of the policy trained with DR was less sensitive to the change in compliance, achieving an almost perfect success rate except for in the highest hydroelastic modulus (stiffest) condition. This suggests that DR may be able to overcome some of the shortcomings of stiff contact interactions.

Whole-body manipulation without exteroceptive object tracking

A limiting factor in our pipeline was the use of exteroceptive sensors, which in our case was a motion capture system, to track the object pose during policy rollouts. Motion capture was restrictive for two reasons: The robot had to operate in a certain area covered by sensors, and the use of markers, such as infrared light-emitting diodes or reflective dots, was difficult to track because of heavy occlusions that occurred during whole-body manipulation.

With the motivation of eliminating the reliance on motion capture, we explored a privileged learning approach. A simple and effective option for this purpose was the choice of an asymmetric actor-critic policy architecture, which is favorable when training in a simulation environment where privileged information that may not be accessible in the real world is available (*57*, *58*), such as the object pose. We implemented a version of asymmetric actor-critic learning where the actor observes a subset of the critic's observations. The motivation is to leverage the privileged information that is readily available in simulation while training and to rely solely on the easy-to-access information during inference.

We conducted an ablation study to compare policies trained and tested using a set of privileged observations for the critic (robot and object poses, binary floatie and paw contact state, and actions from the previous time step) and various sets of unprivileged observations for the actor: previous actions (O1); robot pose (O2); robot pose and previous actions (O3); robot pose and binary contact state (O4); and robot pose, binary contact state, and previous actions (O5). For comparison, we also provided the full set of observations including object pose (privileged) in addition to those of O5 (O6). For each condition, we trained three policies with different seeds for the plastic-box-paw-lift, plastic-box-hug-lift, jug-over-shoulder, and jug-rotate tasks. For each task, success rates during training aggregated by seed are visualized in Fig. 5A.

As one may anticipate, observing only the previous actions is not sufficient to learn a meaningful policy. All of the other partial observation configurations, however, have enough information to implicitly deduce the object pose to learn a policy that can complete the task, except for the jug-rotate task. Tactile feedback, in addition to proprioception (measured joint positions), provides a substantial advantage over proprioception only for all of the tasks. Moreover, tactile-and-proprioceptive feedback (O4) is the only actor observation configuration that achieves a nonzero task success for our most complex task, jug-rotate. Including the previous actions did not necessarily improve the performance.

The influences of including the tactile feedback along with the proprioception on the training performance became more evident as the uncertainty in the object pose increased. Figure 5B shows a comparison of these two configurations averaged across three seeds for a longer training period and an enlarged range of initial object

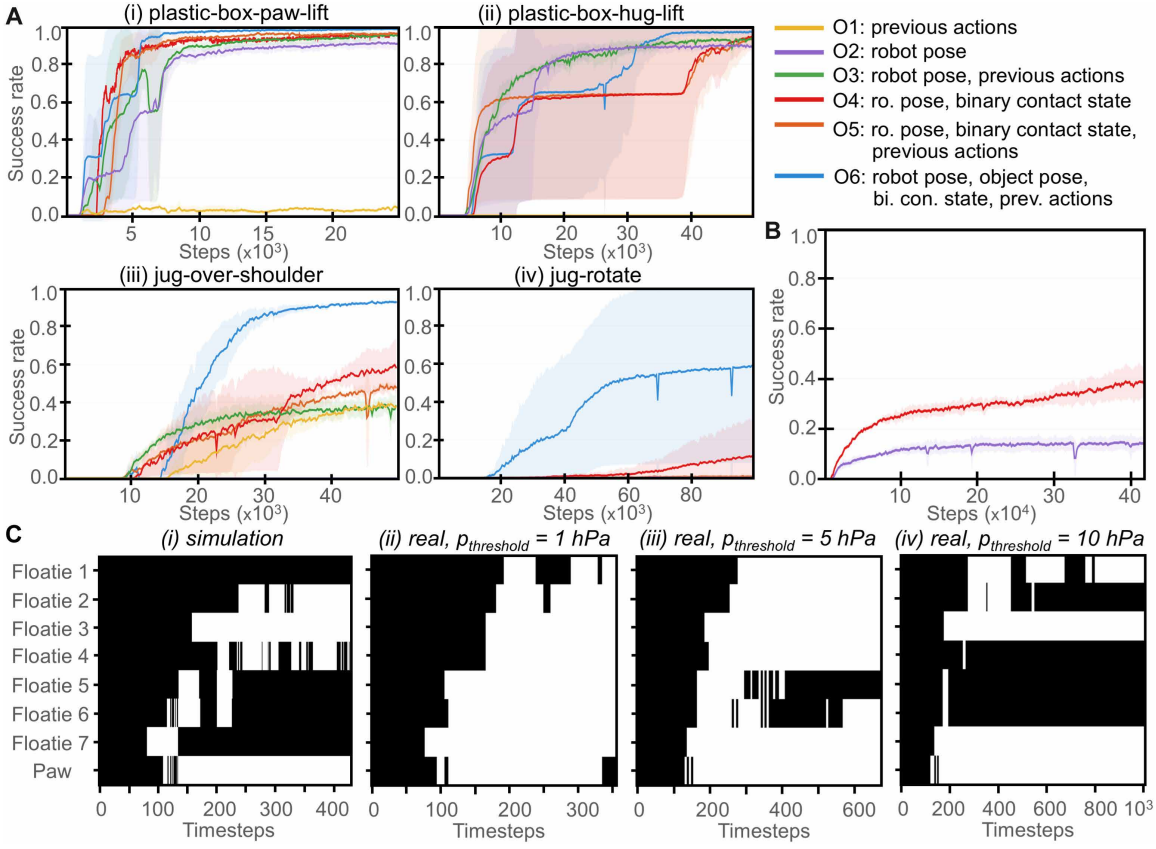


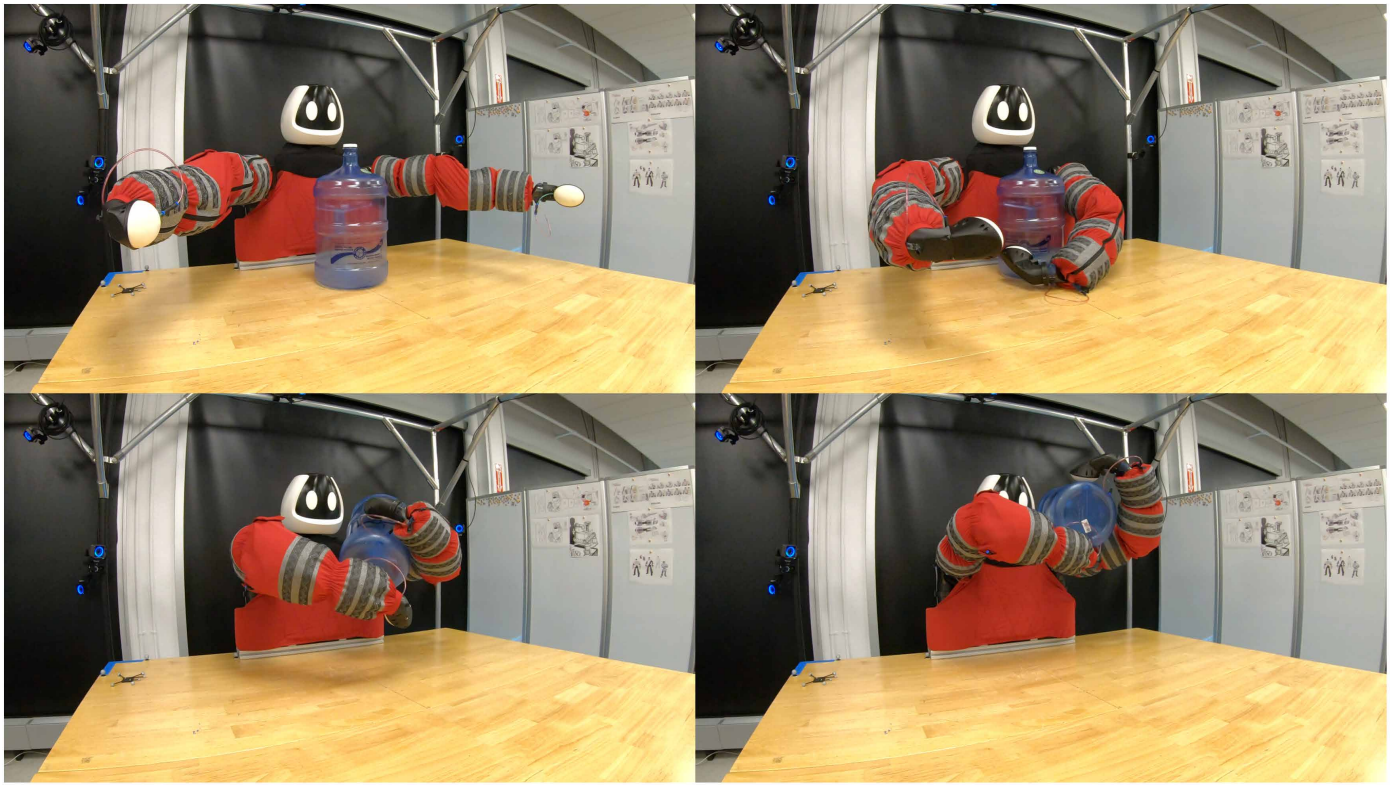
Fig. 5. Results of the analysis of the effect of privileged learning and the sim-to-real gap of low-fidelity tactile feedback. (**A**) Success rate during training for the (i) plastic-box-paw-lift, (ii) plastic-box-hug-lift, (iii) jug-over-shoulder, and (iv) jug-rotate tasks with the object initialized within a 10-cm square, averaged across three seeds with various actor observation sets (O1 to O6). The *x* axis corresponds to gradient steps. (**B**) Plot of the success rate during training for the jug-over-shoulder task with jug initialized within a 50-cm square averaged across three seeds showing a comparison of two sets of unprivileged observations: proprioception only (O2) and proprioception-and-tactile (O4). (**C**) Visualization of the binary contact states for the left arm during jug-over-shoulder policy executions in simulation (i) and on hardware (ii to iv) with pressure thresholds, $p_{threshold}$, of 1, 5, and 10 hPa above atmospheric pressure. White represents in-contact state, and black represents not-in-contact. In (A) and (B), shaded regions indicate the SD across three random seeds (±1 SD, $n = 3$).

poses (the object was initialized within a 50-cm square of the nominal pose instead of a 10-cm square) for the jug-over-shoulder task. Movie 3 shows the hardware rollout of this policy. The tactile feedback not only accelerated the learning process but also enabled substantially higher task performance with more training.

Low-fidelity tactile feedback

For ease of transferability of the tactile signals from simulation to real, we used binary contact states instead of the continuous pressure signals produced by the sensor hardware. As discussed in (*59*), it is difficult to align the contact force measurement between the simulation and real even for near-rigid tactile sensors. Transferring the exact force feedback would be even more challenging for the Punyo robot because the tactile sensors are highly deformable and difficult to simulate. Checking the existence of a contact in simulation is equivalent to a collision query, whereas continuous signals depend on the accuracy of contact modeling and, therefore, are substantially more difficult to transfer. Moreover, given that we assumed rigid-body dynamics to model our deformable robot, we relied on collision filtering to prevent artificially constraining the motions of the arms (for example, when joint-adjacent sensors inside the elbow collide with one another). Despite these aspects, for the whole-body manipulation skills targeted in this study, we found binary tactile feedback to be sufficient.

On hardware, we selected a pressure threshold over which the continuous pressure signals from our inflatable sensors would be considered a binary in-contact state. This threshold effectively filters noisy readings and zero-force-baseline fluctuations due to temperature changes (see the "Floatie characterization" section in the Supplementary Materials). To calibrate the sensors, before each set of experiments, we zeroed the pressure readings of the tactile sensors. The low-fidelity simulation of tactile sensors can cause mismatches in binary contact signals between simulation and real, as displayed in Fig. 5C. Figure 5C(i) shows the simulated binary contact states for the seven air-filled chambers and the paw on the left while executing a proprioception-and-tactile jug-over-shoulder policy, where the activation in simulation is determined on the basis of a geometric collision query. Figure 5C (ii to iv) shows the binary contact states for hardware runs with pressure thresholds of 1, 5, and 10 hPa, respectively, over the no-contact floatie pressure (roughly 1013 hPa or 1 atm). It is noteworthy that taxels that are active (shown in Fig. 5C in white) at a lower pressure threshold (1 hPa) are not necessarily active at a higher threshold (5 and 10 hPa). This relationship is not



Movie 3. Hardware rollout of the blind (no object pose tracking, only proprioceptive-and-tactile feedback), jug-over-shoulder policy.

as straightforward as in an open-loop fashion because in closed-loop control (via the policy), current changes in tactile readings due to different thresholds affect future contact sequences and hence future tactile readings. Another peculiarity is that when a floatie is in contact in hardware, its lateral expansion often triggers the longitudinally adjacent air-filled chambers, especially when the pressure threshold is low. When testing these pressure thresholds, only 5 hPa resulted in a successful task execution and in a closer tactile signature to simulation, as shown in Fig. 5C (iii). Hence, we used this value for the rest of the experiments. In the hardware experiments, we observed that the robot applied very light contact force for the lowest threshold of 5 hPa and, conversely, exerted more force as the pressure threshold was increased. This pressure threshold can therefore be used to tune contact forces over a task.

DISCUSSION

In this work, we present a holistic approach that combines software (mind) and hardware (body) solutions to enable humanoid robots to use their entire bodies to carry and manipulate large and unwieldy objects more effectively and efficiently than grasping only with outstretched hands. We showed that example-guided RL is capable of learning whole-body manipulation skills with complex contact sequences in a similar style to a human demonstrator when using teleoperation and with minimal human supervision when using global planners, even when the plan produced is physically infeasible. The ability to learn from a single, infeasible plan is noteworthy because it can enable scaling up the proposed framework through automatic generation of plans for arbitrary start and goal poses. We emphasized that our framework is independent from the source of example motions and can directly benefit from advancements in planning through contact. In addition, we demonstrated that DR as well as active (via admittance control) and passive (via soft bodies) compliance are desirable to robustly execute learned policies on hardware. Furthermore, we showed that privileged learning allows us to use more accessible (unprivileged) feedback modalities during inference. We found, for instance, tactile sensing to be sufficient as the only exteroceptive observation to perform the whole-body manipulation tasks investigated in this study.

These characteristics are a step forward in whole-body manipulation, yet our system can be further improved. For fast prototyping, we used the motion-capture system to track the manipulated object's pose. To make our system more deployable in the real world, we could rely on advancements in computer vision, particularly in object pose estimation and tracking. FoundationPose (*60*), for instance, can robustly estimate and track the six-dimensional (6D) pose for even novel objects. Such methods can be integrated with our EGRL framework to eliminate the dependency on motion tracking and facilitate its application beyond laboratory settings. A drawback of our privileged learning approach (asymmetric actor-critic) for blind manipulation is the reduced training efficiency, as shown in Fig. 5A. More sophisticated privileged learning methods, such as (*61*, *62*), could mitigate this issue. We binarized the continuous tactile signal into active and inactive contact states to minimize the gap between simulation and reality and to enable the transfer of learned policies. Although effective, we expect binary tactile signals to be restrictive for tasks that require fine and dynamic regulation of contact forces. Similar to the actuator network (*63*) that uses function approximators trained on real-world data

to represent complex actuators in simulation, we envision using a sensor network that substitutes hard-to-model aspects in our simulation, such as Punyo's soft pressure sensors. This would allow the use of higher-fidelity tactile signals. When resorting to binary tactile signals, actively changing the pressure threshold could offer a way of regulating contact forces. Compared with behavior cloning methods, such as diffusion policy, which can work directly with output feedback, for example, raw camera images, our approach is limited by the need for modeling. Transferring policies from simulation to real depends on the quality of the models of the robot, the object(s), and the environment. Hence, it is not straightforward to extend our framework to arbitrary tasks, such as more objects in terms of both quantity and complexity. Moreover, our approach hinges on the capability of parallelized simulators to model contact mechanics and deformable bodies accurately and efficiently. However, owing to the minimum human supervision required, our method could allow for policy generation pipelines that feed policy rollout data to supervised learning methods such as DP. Our demonstrations are limited to tasks that can be accomplished with a nonmobile upper torso humanoid; however, because of the generality of the formulation of the manipulation problem, we could expand the method to a mobile version by appending the extra states, such as the mobile base position, to the current states of the system (arm joint angles and object positions). Nevertheless, we hypothesized that our method would be challenged by longer-horizon tasks such as those likely to be targeted for a mobile Punyo. We defer these points to future work. Although our real-world experiments are extensive, they were conducted over a few days of continuous robot execution, and the degradation of performance due to distribution shift caused by hardware changes was not evidenced. Mitigating such distribution shift is an active area of research, with notable methods including automatic calibration, active adaptation (*64*), and online fine-tuning (*65*, *66*), among others (*67*, *68*).

We believe that systems, such as Punyo, that combine mechanical and computational intelligence could allow humanoids to go beyond existing robotic capabilities. We showed that with such systems, one can leverage well-studied methods—such as teleoperation, model-based planning, RL, soft sensing, and admittance control—to pave the way toward a future where robots and people collaborate safely, productively, and happily, side by side.

MATERIALS AND METHODS

Punyo the hardware platform

With Punyo-1 (*5*), we introduced an upper-body system that outfits off-the-shelf robot arms with soft, air-filled contact pressure sensors. This provided a low-cost and flexible solution for achieving robust whole-body maneuvers and grasps. Given Punyo-1, we improved on this system with a higher spatial density of contact sensing (see "Improvements on the Punyo-1 hardware platform" and "Punyo components and dimensions" in the Supplementary Materials). Similar to the first generation, Punyo was still built upon two Kinova Jaco Gen2 arms with inflatable bubble chambers that are used for sensing, a custom chest surface, and soft-bubble–based (*69*) pressure-sensing "paws" as end effectors. These air-filled chambers were connected pneumatically to air pressure sensors. The air pressure change was measured and used as an indicator of imparted force. With further engineering improvements, these air chambers were meant to work for a long time without the need for reinflation, as has been shown in everyday commercial products such as vehicle tires. Such improvements could include the use of a robust material with low tensile degradation and a low coefficient of thermal expansion [for example, polyimide $\alpha \sim 4 \times 10^{-5}$ mm/(mm °C)] and more leak-proof fittings.

The bubble-shaped sensors approximate an expanded boundary around the roughly cylindrical arms to create a mechanical compliance envelope around the hard parts of the robot. The sweater that wraps around the compliant bubbles and rigid robot has added friction features on the inside and outside (fig. S2). The inside features were added to prevent slipping along the underlying robot and inflatable structures. The outside features were added to improve grasp friction on the objects with which the robot interacts. The Punyo system here is a humanoid upper body and is bolted at the waist to a support surface with a workspace in front of it where it manipulates objects.

Manipulation tasks

We define "task" as a sequence of state/action pairs to accomplish a goal that is specified in terms of a desired object pose. The object dimensions and weights are shown in table S1. Figure 2 shows a depiction of a policy rollout for each task (table S2 includes the full set of observations of each policy).

Cube-lift

The robot was tasked with relocating a cube to a desired pose, which was 32.5 cm higher than the table surface, using its two end effectors. An inverse kinematics (IK)–based coupled end-effector teleoperation was used to generate the example motion for this task. Although this does not require whole-body engagement, it serves as a test for our method in a canonical end-effector bimanual task.

Plastic-box-paw-lift

Similar to the cube-lift task, the robot was tasked with lifting a plastic box to a desired pose, which was 13 cm higher than the table surface, using its two end effectors. Teleoperation data collected with the IK-based coupled end-effector teleoperation were used as the example motion for this task as well. Similar to the cube-lift task, this is a canonical end-effector bimanual task.

Plastic-box-hug-lift

Similar to the plastic-box-paw-lift task, the robot was tasked with relocating a plastic box to a desired pose, which was 13 cm higher than the table surface and 21.5 cm closer to its torso, using the robot's whole body, resulting in multiple contacts along the arms and the torso. Teleoperation data collected in simulation using our Eigengrasp-based whole-body teleoperation method were used as the example motion for this task.

Cardboard-box-pivot-and-lift

The robot was tasked with relocating a large box to a different pose, which involved a 90° clockwise rotation around the world y axis (x^+ pointing toward the front of the robot and z^+ above it) while maintaining the box at a certain height to prevent it from touching the table. A plan generated by the GQDP planner was used as the example motion for this task. In the demonstration, the robot used one paw to pivot the box against its torso then used the other paw to stabilize the box and prevent it from falling.

Jug-over-shoulder

The robot was tasked with relocating a water jug using only one of its arms from the table onto its shoulder. Teleoperation data collected with Eigengrasp in simulation were used as the example motion for this task. The task required the robot to initially wrap the arm around the water jug and push it against its torso and then use the wrapping contact to move the jug onto its shoulder.

Jug-rotate

The robot was tasked with rotating a water jug by 180° using its two arms. A plan generated by the GQDP planner was used as the example motion for this task. In the demonstration, the robot coordinated the movements of its two arms, consistently using one arm to hold the jug while using the other arm to rotate it. In addition, the torso of the robot was actively engaged throughout the entire process to stabilize the water jug.

Hardware control pipeline

The controls pipeline (fig. S3) relied on a policy that outputs the desired change in joint positions $\mathbf{a}_t$ in radians. $\mathbf{a}_t$ was then fed into an action integrator that calculates the desired joint position $\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd}}$. We used the built-in joint velocity controller of the Kinova Jaco Gen2 arms using the default control parameters along with an outer velocity proportional-derivative (PD) loop that converts desired joint positions $\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd}}$ to velocity references for the built-in joint velocity controller such that

$$\dot{\mathbf{q}}^{\mathrm{a}}_{\mathrm{cmd},t} = 3\left(\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd},t} - \mathbf{q}^{\mathrm{a}}_{t}\right) - 0.1\dot{\mathbf{q}}^{\mathrm{a}}_{t} \tag{2}$$

As a result, we virtually control the joint positions and, therefore, model the actuators as PD position–controlled degrees of freedom in our simulation such that $\tau = kp(q - q_{\mathrm{cmd}}) + kd(\dot{q})$ where τ is the output torque and kp and kd are joint stiffness and damping, respectively. We estimated the equivalent joint stiffness and damping values for the position controller in simulation as $kp = 500$ and $kd = 180$ through system identification experiments. Optionally, a compliant controller can be integrated into our control pipeline (in between the action integrator and the outer PD loop) to enable joint-space admittance control, as illustrated in fig. S3 and detailed below.

Action integration

In training, we integrated $\mathbf{a}_t$ to obtain a joint position command $\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd}}$

$$\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd},t} = \mathbf{q}^{\mathrm{a}}_{\mathrm{cmd},t-1} + \mathbf{s}_{\mathrm{a}}\dot{\mathbf{q}}^{\mathrm{a}}_{\mathrm{lim}} \odot \mathbf{a}_t \Delta t \tag{3}$$

where $\odot$ denotes element-wise product, $\mathbf{s}_{\mathrm{a}}$ is a scalar action scale, $\dot{\mathbf{q}}^{\mathrm{a}}_{\mathrm{lim}}$ is the vector of joint speed limits, t subscript indicates the time step, and Δt is the time step size. Note that the relative change is added to the previous position command as a feedforward term.

When executing a policy on hardware, this scheme often causes excessive joint torques, especially for contact-rich tasks with high internal forces. This is mainly caused by the high stiffness of the built-in joint controller that is necessary to compensate for the missing feedforward inverse dynamics terms, including the generalized gravitational forces. This could be addressed by incorporating such a feedforward term and reducing the joint stiffness when using a torque-controlled robot. However, our robot does not allow torque control; thus, we resorted to a blending scheme that emulates such a low-stiffness effect. That is, we applied the relative changes to a weighted average of the feedforward and feedback joint position terms

$$\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd},t} = \beta\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd},t-1} + (1-\beta)\mathbf{q}^{\mathrm{a}}_{t} + \mathbf{s}_{\mathrm{a}}\dot{\mathbf{q}}_{\mathrm{lim}} \odot \mathbf{a}_t \Delta t \tag{4}$$

where $\beta \in [0, 1]$ is the blending factor and $\mathbf{q}^{\mathrm{a}}$ is the measured joint pose. We nominally used $\beta = 0.95$ in our experiments. The feedback term helped dampen the robot motions and prevented commanding overpenetrating positions and the resulting high forces between the robot and objects. In the case of zero action, this ad hoc strategy was equivalent to reducing the stiffness of the robot by a factor of β. We found this to be more effective than directly lowering the joint position controller's proportional gain because it avoided compromising the motion tracking accuracy, which was an issue that would arise because of the absence of any compensation for inverse dynamics, such as gravity, in the built-in controller.

Compliant joint controller

For the active compliance experiments, we devised an optimization-based joint-space admittance controller. Our objective was to enable a position-controlled robot to comply with external forces. Hence, we formulated the following numerical program that minimized the deviation of the joint position commands from the desired joint positions, $\mathbf{q}_{\mathrm{des}}$, while respecting the joint torque limit τ_{lim} and incorporating the estimated control and external torques, $\hat{\tau}_{\mathrm{ctrl}}$ and $\hat{\tau}_{\mathrm{ext}}$

$$\begin{aligned} &\underset{\mathbf{q}^{\mathrm{a}}_{\mathrm{cmd}}}{\text{minimize}} \left\| \mathbf{q}^{\mathrm{a}}_{\mathrm{des}} - \mathbf{q}^{\mathrm{a}}_{\mathrm{cmd}} \right\|^2 \\ &\text{subject to} \left| \hat{\tau}_{\mathrm{ctrl}} + \hat{\tau}_{\mathrm{ext}} \right| \le \tau_{\mathrm{lim}} \end{aligned} \tag{5}$$

The external torque was estimated and reported by the internal torque sensors of the Jaco Gen2 arms. We estimated the control torque on the basis of the current joint state, the desired joint positions, and the estimated proportional-derivative gains of the built-in position controller, K_{p} and K_{d}

$$\hat{\tau}_{\mathrm{ctrl}} \triangleq K_{\mathrm{p}}\left(\mathbf{q}^{\mathrm{a}}_{\mathrm{des}} - \mathbf{q}^{\mathrm{a}}_{t}\right) - K_{\mathrm{d}}\dot{\mathbf{q}}^{a}_{t} \tag{6}$$

Using this formulation, the amount of admittance can be adjusted via the joint torque limit. In our implementation, we used $K_{\mathrm{p}} = 480$, $K_{\mathrm{d}} = 90$, and $\tau_{\mathrm{lim}} = [24.2, 18, 19, 23, 20, 20, 20]$ N·m. Torque limits were adjusted experimentally to values that minimized exceeding joint torque limits that compromised the robot integrity (for example, joint torque-outs or blown fuses).

Eigengrasp-based whole-body teleoperation

To generate human-guided example motions for whole-body manipulation, we developed a teleoperation strategy inspired by Eigengrasp (*51*), which was based on the concept of grasp synergies. In this approach, we grouped joints to create arm motion primitives such that whole-body motions could be controlled through a lower-dimensional interface. We used a standard gamepad interface to control the activation of these primitives, which were defined in terms of joint displacements (as detailed in the "Eigengrasp primitives" and "Teleoperators" sections in the Supplementary Materials).

GQDP

We used a version of the code provided in (*14*) to generate plans. We further refined the motion plans obtained from this framework for two reasons: to prevent undesirable robot-object collisions and to stabilize the object using a task-specific heuristic that made the idle arm support the object when moving between segments (via regrasps). For preventing undesirable robot-object collisions, the kinematic trajectory optimization method in Drake (*54*) was used with minimum distance constraints.

EGRL

Given a dataset of example motions and a task objective defined by a reward function, EGRL synthesizes a control policy that enables the

agent to achieve the specified task objective while adopting behaviors that mimic the style of the example motion dataset. The process commences with the acquisition of an example motion dataset, M, generated through the use of a model-based planner or teleoperation. The objective of EGRL was to train a policy, π, capable of completing a desired manipulation task while adhering to the motion style defined in the example motions of dataset M. Note that in this work, M contained only one example motion. For this purpose, the framework trained the policy along with a discriminator. The discriminator was designed to learn an imitation reward such that the agent used the demonstrated motion style while simultaneously learning how to accomplish the task.

In manipulation, a motion style can be considered a particular sequence of states by which a manipulation task is performed. In an example motion, the system experiences a sequence of states, denoted as $\mathrm{T}_{\text{example}} = [\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_n]$. Given a control policy $\pi(\cdot)$, which at each time step t perceives the state $\mathbf{s}_t \in \mathbf{S}$ of the system and samples an action $\mathbf{a}_t \in A$, adhering to the probabilistic distribution $\mathbf{a}_t \sim (\mathbf{a}_t|\mathbf{s}_t)$. Then starting from the same initial state $\mathbf{s}_1$ and following the actions produced by the policy, the system will experience a sequence of states, denoted $\mathrm{T}_\pi = [\mathbf{s}_1, \mathbf{s}'_1, \ldots, \mathbf{s}'_n]$. The state transitions $(\mathbf{s}_t, \mathbf{s}_{t+1})$ from both $\mathrm{T}_{\text{example}}$ and T_π can be viewed as samples from two different distributions, D_{example} and D_π. Then, the style imitation can be achieved by seeking a policy π that could minimize the difference between the distributions D_{example} and D_π.

The policy was modeled as a neural network that maps a given state to a Gaussian distribution over actions. This distribution features a mean dependent on the input and a fixed diagonal covariance matrix. The mean is determined by a fully connected network with hidden layers and a linear output layer.

The agent receives a scalar reward $r_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ after executing the action $\mathbf{a}_t$ produced by the policy at time t. As in the AMP algorithm (*43*), we formulated the reward function as a weighted average of two distinct components: the task reward r^{T} that quantifies the degree of task accomplishment, which encourages the policy to perform the task in hand, and the style reward $r^{\mathbf{S}}$ that assesses the resemblance between the robot's motion and the example motion(s), which guides the policy on how to perform the task

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \lambda r^{\mathrm{T}}(\mathbf{s}_{t+1}, \mathbf{a}_t) + (1-\lambda) r^{\mathrm{S}}(\mathbf{s}_t, \mathbf{s}_{t+1}) \tag{7}$$

where $\lambda \in [0, 1]$ determines the task reward weight with respect to the imitation reward weight. Obtaining the ideal behavior for a specific task may require tuning. A too-large λ could result in the robot focusing only on finishing the task without respecting the motion style, whereas a too-small λ could cause the robot to focus only on mimicking the motion style without accomplishing the task. We chose the ideal λ by running a parameter sweep over a grid of values for each task. We found that the behaviors were not very sensitive to this parameter and chose $\lambda = 0.7$ for all of the tasks except for the jug-rotate task, which uses a physically infeasible motion example coming from a planner.

To minimize reward shaping, we chose to use a simple and generic task reward function $r^{\mathrm{T}} = r_{\text{kp}} + p$, where r_{kp} depends on the distance between a set of keypoints on the object and their corresponding positions when the object is in the goal pose and p encompasses conventional penalty components often adopted for RL, such as penalties for actions and velocities, early termination, and a success bonus.

The detailed task reward is defined as follows

$$r_t^{\mathrm{T}} = w_{\text{kp}}\left(1/\left(\left\|d_{\text{kp}}\left(\mathbf{q}_t^{\mathrm{u}}, \mathbf{q}_{\mathrm{g}}^{\mathrm{u}}\right)\right\| + 0.1\right)\right) + w_{\mathrm{a}} \| \mathbf{a}_t \|^2 + w_{\mathrm{da}} \| \Delta\mathbf{a}_t \|^2 + w_\tau \| \tau_t \|^2 + w_{\text{lin}} \left\|\dot{\mathbf{q}}_{t,\text{lin}}^{\mathrm{u}}\right\|^2 + w_{\text{rot}} \left\|\dot{\mathbf{q}}_{t,\text{rot}}^{\mathrm{u}}\right\|^2 + w_{\text{term}} 1_{\text{term}}(\mathbf{q}_t^{\mathrm{u}}) + w_{\text{succ}} 1_{\text{succ}}(\mathbf{q}_t^{\mathrm{u}}, \dot{\mathbf{q}}_t^{\mathrm{u}}) \tag{8}$$

The first term incentivizes task completion by penalizing the distance of the keypoints from their desired positions, $d_{\text{kp}}(\cdot)$. The following terms impose penalties on the squared L2 norms of the robot's actions, $\mathbf{a}_t$; the change in actions, $\Delta\mathbf{a}_t \triangleq \mathbf{a}_t - \mathbf{a}_{t-1}$; joint torques, τ_t; and the object's translational and rotational velocities, $\dot{\mathbf{q}}_{t,\text{lin}}^{u}$ and $\dot{\mathbf{q}}_{t,\text{rot}}^{u}$. The final two terms impose a penalty upon the activation of termination conditions and a bonus upon the completion of success criteria. The weight parameters w_{kp}, w_{a}, w_{da}, w_τ, w_{lin}, w_{rot}, w_{term}, and w_{succ} determine the relative importance of these terms in the overall reward. We used the same weights across all experiments, as specified in table S3. The functions $1_{\text{term}}(\cdot)$ and $1_{\text{succ}}(\cdot)$ are activation functions based on termination and success conditions (table S4). The termination conditions are triggered if the object goes out of the workspace limits (table S5), whereas the success criteria are based on the deviation of the object pose from the goal and the linear and angular object velocity terms.

The style reward $r^{\mathbf{S}}$ is decided by the discriminator D, which discriminates whether a state transition belongs to the example motion distribution. It is trained together with the policy from scratch, and the discriminator learns to assign a score of 1 to samples from the example motion dataset M and 0 to samples in the policy's rollout buffer B. This is trained by solving the least squares regression problem with loss defined as follows

$$L = \mathbb{E}_{(\mathbf{s},\mathbf{s}')\sim M}\left[\left(D(\Phi(\mathbf{s}), \Phi(\mathbf{s}')) - 1\right)^2\right] + \mathbb{E}_{(\mathbf{s},\mathbf{s}')\sim B}\left[\left(D(\Phi(\mathbf{s}), \Phi(\mathbf{s}'))\right)^2\right] + w_{\text{gp}} \mathbb{E}_{(\mathbf{s},\mathbf{s}')\sim M}\left[\|\nabla_\Phi D(\Phi(\mathbf{s}), \Phi(\mathbf{s}'))\|^2\right] \tag{9}$$

where $\mathbb{E}$ denotes the expected value operator, $\Phi(\cdot)$ extracts desired imitation features from the system state $\mathbf{s}$, and the final term is a gradient penalty term with coefficient w_{gp}, which penalizes nonzero gradients on samples from the example motions, resulting in improved stability of the training (*70*).

At each time step during training, the latest state transition is used to calculate the style reward r^{S}, which is defined as follows

$$r^{\mathrm{S}} = -\log\left(1 - \frac{1}{1 + e^{-D(\Phi(\mathbf{s}),\Phi(\mathbf{s}'))}}\right) \tag{10}$$

As detailed in (*43*), the discriminator's role in specifying rewards for policy training underscores the importance of selecting appropriate input features for its discrimination tasks. This selection is critical to providing the policy with constructive feedback. Therefore, the observation map $\Phi(\mathbf{s}_t)$ is used to extract features that could represent example motions effectively from the system state $\mathbf{s}_t$ instead of directly feeding a state transition into the discriminator.

Simulation

We trained our policies in simulation using rigid-body dynamics parallelized over a graphical processing unit (GPU) using Isaac Gym (*71*). For policy learning, we used the proximal policy

optimization (PPO) algorithm (*72*). Particularly, we used the PPO implementation in (*73*) and built upon the AMP implementation in (*74*).

The learning networks and algorithms were implemented in PyTorch 1.8.1 with CUDA 12.0. The training procedure encompassed the collection of experiences from 4096 agents simulated in parallel. The entirety of the experimental work was executed on a desktop equipped with an NVIDIA GeForce RTX 3090 GPU. A single run comprising 2000 epochs, adhering to the aforementioned computational settings and device specifications, was accomplished within ~2 hours. The network architectures and training parameters that we used in this study are given in tables S6 and S7, respectively.

Domain randomization

We incorporated DR (*75*) to enhance robustness to uncertainties in the model parameters and the initial object pose and to enable sim-to-real transfer. This involved applying a disturbance sampled from a distribution (as specified in table S8) to the nominal value of each randomization parameter. Nominal initial and goal object positions are shown in table S9.

Statistical analysis

For all quantitative results, performance metrics such as success rate were computed by averaging across n independently trained policies initialized with different random seeds. Shaded regions and error bars in training and simulation evaluation plots represent the SD across these seeds.

For real-world experiments, policies trained with a single seed were evaluated per each condition. Error bars represent 95% confidence intervals based on the Bernoulli distribution of success rates defined as follows

$$\mathrm{CI} = p \pm z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}} \tag{11}$$

where p is the proportion of successes, n is the sample size, and z is the z score for the desired confidence level α.

Supplementary Materials

This PDF file includes:

Methods
Tables S1 to S9
Figs. S1 to S8
References (*76*–*84*)

REFERENCES AND NOTES

1. M. T. Mason, Toward robotic manipulation. *Annu. Rev. Control Robot. Auton. Syst.* **1**, 1–28 (2018).
2. N. Chavan-Dafle, R. Holladay, A. Rodriguez, Planar in-hand manipulation via motion cones. *Int. J. Robot. Res.* **39**, 163–182 (2020).
3. K. M. Newell, What are fundamental motor skills and what is fundamental about them? *J. Mot. Learn. Dev.* **8**, 280–314 (2020).
4. A. F. Bos, K. N. J. A. Van Braeckel, M. M. Hitzert, J. C. Tanis, E. Roze, Development of fine motor skills in preterm infants. *Dev. Med. Child Neurol.* **55**, 1–4 (2013).
5. A. Goncalves, N. Kuppuswamy, A. Beaulieu, A. Uttamchandani, K. M. Tsui, A. Alspach, "Punyo-1: Soft tactile-sensing upper-body robot for large object manipulation and physical human interaction," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)* (IEEE, 2022), pp. 844–851.
6. M. J. Gielniak, C. K. Liu, A. L. Thomaz, Generating human-like motion for robots. *Int. J. Robot. Res.* **32**, 1275–1301 (2013).
7. D. Hostettler, S. Mayer, C. Hildebrand, Human-like movements of industrial robots positively impact observer perception. *Int. J. Soc. Robot.* **15**, 1399–1417 (2023).
8. A. D. Dragan, K. C. T. Lee, S. S. Srinivasa, "Legibility and predictability of robot motion" in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (IEEE, 2013), pp. 301–308.
9. C. Lichtenthäler, T. Lorenzy, A. Kirsch, "Influence of legibility on perceived safety in a virtual human-robot path crossing task" in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication* (IEEE, 2012), pp. 676–681.
10. I. Mordatch, Z. Popović, E. Todorov, "Contact-invariant optimization for hand manipulation" in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (ACM, 2012), pp. 137–144.
11. A. Ö. Önol, R. Corcodel, P. Long, T. Padır, "Tuning-free contact-implicit trajectory optimization" in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 1183–1189.
12. X. Cheng, E. Huang, Y. Hou, M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3D" in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022), pp. 2730–2736.
13. M. Zhang, D. K. Jha, A. U. Raghunathan, K. Hauser, Simultaneous trajectory optimization and contact selection for multi-modal manipulation planning. arXiv:2306.06465 [cs.RO] (10 June 2023).
14. T. Pang, H. J. T. Suh, L. Yang, R. Tedrake, Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Trans. Robot.* **39**, 4691–4711 (2023).
15. R. Natarajan, G. L. Johnston, N. Simaan, M. Likhachev, H. Choset, "Torque-limited manipulation planning through contact by interleaving graph search and trajectory optimization" in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 8148–8154.
16. L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, A. M. Dollar, A compliant, underactuated hand for robust manipulation. *Int. J. Robot. Res.* **33**, 736–752 (2014).
17. C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, S. Song, Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robot. Res.*, 10.1177/02783649241273668 (2024).
18. S. Haldar, J. Pari, A. Rai, L. Pinto, Teach a robot to FISH: Versatile imitation from one minute of demonstrations. arXiv:2303.01497 (2023).
19. M. Du, S. Nair, D. Sadigh, C. Finn, Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. arXiv:2304.08742 (2023).
20. Toyota Research Institute, "TRI's robots learn new manipulation skills in an afternoon" (14 September 2023); https://medium.com/toyotaresearch/tris-robots-learn-new-skills-in-an-afternoon-here-s-how-2c30b1a8c573.
21. T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation" in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2024), pp. 8944–8951.
22. T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, G. Shi, OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. arXiv:2403.04436 (7 March 2024).
23. K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, D. Pucci, "Whole-body geometric retargeting for humanoid robots" in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2019), pp. 679–686.
24. N. S. Pollard, J. K. Hodgins, M. J. Riley, C. G. Atkeson, "Adapting human motion for the control of a humanoid robot" in *Proceedings 2002 IEEE International Conference on Robotics and Automation* (IEEE, 2002), vol. 2, pp. 1390–1397.
25. A. Nagabandi, K. Konolige, S. Levine, V. Kumar, "Deep dynamics models for learning dexterous manipulation" in *Conference on Robot Learning* (PMLR, 2020), pp. 1101–1112.
26. T. Chen, J. Xu, P. Agrawal, "A system for general in-hand object re-orientation" in *Conference on Robot Learning* (PMLR, 2022), pp. 297–307.
27. O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3–20 (2020).
28. T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, P. Agrawal, Visual dexterity: In-hand reorientation of novel and complex object shapes. *Sci. Robot.* **8**, eadc9244 (2023).
29. J. Eßer, N. Bach, C. Jestel, O. Urbann, S. Kerner, Guided reinforcement learning: A review and evaluation for efficient and effective real-world robotics. *IEEE Robot. Autom. Mag.* **30**, 67–85 (2023).
30. M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, M. Riedmiller, Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. arXiv:1707.08817 (27 July 2017).
31. A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, S. Levine, Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. arXiv:1709.10087 (28 September 2017).
32. X. B. Peng, P. Abbeel, S. Levine, M. Van de Panne, Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.* **37**, 1–14 (2018).
33. A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations" in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 6292–6299.

34. Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, N. Heess, Reinforcement and imitation learning for diverse visuomotor skills. arXiv:1802.09564 (27 May 2018).
35. H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, V. Kumar, "Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost" in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 3651–3657.
36. V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, N. R. Waytowich, Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments. arXiv:1910.04281 (9 October 2019).
37. S. Christen, S. Stevšić, O. Hilliges, "Demonstration-guided deep reinforcement learning of control policies for dexterous human-robot interaction" in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 2161–2167.
38. X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, S. Levine, Learning agile robotic locomotion skills by imitating animals. arXiv:2004.00784 (21 July 2020).
39. A. Nair, A. Gupta, M. Dalal, S. Levine, AWAC: Accelerating online reinforcement learning with offline datasets. arXiv:2006.09359 (10 September 2020).
40. S. P. Arunachalam, S. Silwal, B. Evans, L. Pinto, "Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation" in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 5954–5961.
41. J. Ho, S. Ermon, "Generative adversarial imitation learning" in *Proceedings of the 30th International Conference on Neural Information Processing Systems* (NIPS, 2016), pp. 4572–4580.
42. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
43. X. B. Peng, Z. Ma, P. Abbeel, S. Levine, A. Kanazawa, Amp: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.* **40**, 1–20 (2021).
44. E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning" in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 5120–5126.
45. A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions" in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022), pp. 25–32.
46. C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations" in *Conference on Robot Learning* (PMLR, 2023), pp. 342–352.
47. C. Li, S. Blaes, P. Kolev, M. Vlastelica, J. Frey, G. Martius, "Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions" in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 2944–2950.
48. J. Wu, G. Xin, C. Qi, Y. Xue, Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robot. Autom. Lett.* **8**, 4975–4982 (2023).
49. X. B. Peng, Y. Guo, L. Halper, S. Levine, S. Fidler, ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.* **41**, 1–17 (2022).
50. OptiTrack, Motion capture cameras (2024); https://optitrack.com/cameras/.
51. M. Ciocarlie, C. Goldfeder, P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping" in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2007), pp. 3270–3275.
52. J. Masterjohn, D. Guoy, J. Shepherd, A. Castro, Velocity level approximation of pressure field contact patches. *IEEE Robot. Autom. Lett.* **7**, 11593–11600 (2022).
53. R. Elandt, E. Drumwright, M. Sherman, A. Ruina, "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects" in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 8238–8245.
54. R. Tedrake, the Drake Development Team, "Drake: Model-based design and verification for robotics" (2019); https://drake.mit.edu.
55. S. LaValle, *Rapidly-Exploring Random Trees: A New Tool for Path Planning* (Tech. Rep., Iowa State University, 1998), pp. 98–11.
56. R. Deimel, O. Brock, A novel type of compliant and underactuated robotic hand for dexterous grasping. *Int. J. Robot. Res.* **35**, 161–185 (2016).
57. F. Jenelten, J. He, F. Farshidian, M. Hutter, DTC: Deep tracking control. *Sci. Robot.* **9**, eadh5401 (2024).
58. L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, P. Abbeel, Asymmetric actor critic for image-based robot learning. arXiv:1710.06542 (18 October 2017).
59. Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, X. Wang, Rotating without seeing: Towards in-hand dexterity through touch. arXiv:2303.10880 (27 March 2023).
60. B. Wen, W. Yang, J. Kautz, S. Birchfield, FoundationPose: Unified 6D pose estimation and tracking of novel objects. arXiv:2312.08344 (13 December 2023).
61. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, eabc5986 (2020).
62. V. Vapnik, R. Izmailov, Learning using privileged information: Similarity control and knowledge transfer. *J. Mach. Learn. Res.* **16**, 2023–2049 (2015).
63. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
64. A. Cully, J. Clune, D. Tarapore, J.-B. Mouret, Robots that can adapt like animals. *Nature* **521**, 503–507 (2015).
65. K. Lei, Z. He, C. Lu, K. Hu, Y. Gao, H. Xu, Uni-O4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization, poster presented at the 12th International Conference on Learning Representations, Vienna, Austria, 7 to 11 May 2024.
66. R. Rafailov, K. B. Hatch, V. Kolev, J. D. Martin, M. Phielipp, C. Finn, "MOTO: Offline pre-training to online fine-tuning for model-based robot learning" in *Proceedings of the 7th Conference on Robot Learning* (PMLR, 2023), pp. 3654–3671.
67. K. R. Varshney, *Trustworthy Machine Learning* (2022).
68. Y. Lee, H. Yao, C. Finn, Diversify and disambiguate: Learning from underspecified data. arXiv:2202.03418 (21 June 2022).
69. N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, R. Tedrake, "Soft-bubble grippers for robust and perceptive manipulation" in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020), pp. 9917–9924.
70. L. Mescheder, A. Geiger, S. Nowozin, "Which training methods for GANs do actually converge?" in *International Conference on Machine Learning* (PMLR, 2018), pp. 3481–3490.
71. V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, G. State, Isaac Gym: High performance GPU-based physics simulation for robot learning. arXiv:2108.10470 (25 August 2021).
72. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 (28 August 2017).
73. D. Makoviichuk, V. Makoviychuk, rl-games: A high-performance framework for reinforcement learning (GitHub, 2021); https://github.com/Denys88/rl_games.
74. J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, D. Fox, "GPU-accelerated robotic simulation for distributed reinforcement learning" in *Proceedings of the 2nd Conference on Robot Learning* (PMLR, 2018), pp. 270–282.
75. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world" in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 23–30.
76. Kinova, "Discover our Gen2 robots"; www.kinovarobotics.com/product/gen2-robots.
77. Arm air-filled chambers; www.amazon.com/dp/B0BV31YCJ9?ref=cm_sw_r_cp_ud_dp_Z5MGXMTXPDRFJF5ARF6P&ref_=cm_sw_r_cp_ud_dp_Z5MGXMTXPDRFJF5ARF6P&social_share=cm_sw_r_cp_ud_dp_Z5MGXMTXPDRFJF5ARF6P&th=1.
78. Punyo; https://punyo.tech/bubble-gripper.html.
79. Spandex World Inc.; https://spandexworld.com/c3/catalog/product/792.
80. Punyo artwork; https://punyo.tech/images/punyo-bots.jpg.
81. Cut-Tex PRO–Globally renowned cut, tear and abrasion resistant fabric; https://cut-tex.com/.
82. Silicone elastic; https://thecostumesource.com/collections/notions/products/silicone-elastic.
83. YKK #3 molded plastic jacket zippers; www.wawak.com/zippers/jacket/molded-plastic/ykk-3-molded-plastic-jacket-zipper/#sku=mz310ag.
84. W. V. Titow, *PVC Technology* (Elsevier Applied Science, 1984).

Acknowledgments: We thank A. Castro, B. Burchfiel, H. Dai, S. Feng, D. Guoy, K. Liu, P. Shah, M. Lutter, and M. Zolotas for the insights and helpful discussions. **Funding:** This work was funded by Toyota Research Institute. **Author contributions:** Algorithm development and experiments: A.Ö.Ö, J.A.B., and M.Z. Environment development: A.Ö.Ö, J.A.B., and M.Z. Robot software infrastructure: A.Ö.Ö, J.A.B., and S.C. Sensor design and hardware: A.A., A.B., and A.G. Project management: A.A., A.B., and K.M.T. **Competing interests:** All authors are employees of Toyota Research Institute. A provisional patent application (no. 63/685,881) has been filed based on this work. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and the Supplementary Materials. Code can be found in a permanent repository (DOI: 10.5061/dryad.ncjsxkt80).

Submitted 26 August 2024
Accepted 23 July 2025
Published 20 August 2025
10.1126/scirobotics.ads6790