Learning Gentle Object Manipulation with Curiosity-Driven Deep Reinforcement Learning

Sandy H. Huang*[1], Martina Zambelli*[2], Jackie Kay[2], Murilo F. Martins[2],
Yuval Tassa[2], Patrick M. Pilarski[2], Raia Hadsell[2]
[1]University of California, Berkeley
[2]DeepMind

shhuang@cs.berkeley.edu, {zambellim,tassa,kayj,murilomartins,ppilarski,raia}@google.com

***Abstract*—Robots must know how to be *gentle* when they need to interact with fragile objects, or when the robot itself is prone to wear and tear. We propose an approach that enables deep reinforcement learning to train policies that are gentle, both during exploration and task execution. In a reward-based learning environment, a natural approach involves augmenting the (task) reward with a penalty for non-gentleness, which can be defined as excessive impact force. However, augmenting with only this penalty impairs learning: policies get stuck in a local optimum which avoids all contact with the environment. Prior research has shown that combining auxiliary tasks or intrinsic rewards can be beneficial for stabilizing and accelerating learning in sparse-reward domains, and indeed we find that introducing a surprise-based intrinsic reward does avoid the no-contact failure case. However, we show that a simple dynamics-based surprise is not as effective as *penalty-based surprise*. Penalty-based surprise, based on predicting forceful contacts, has a further benefit: it encourages exploration which is contact-rich yet gentle. We demonstrate the effectiveness of the approach using a complex, tendon-powered robot hand with tactile sensors. Videos are available at http://sites.google.com/view/gentlemanipulation.**

I. Introduction

Deep reinforcement learning (RL) can be used to train policies that achieve superhuman performance on Atari games [27] and Go [45], learn locomotion tasks [25, 43], and perform complex robotic manipulation skills [24]. However, deploying deep RL on real-world robots often leads to a considerable amount of wear and tear over time, on both the robot itself and the environment, because existing approaches require many trials to learn and often rely on simple stochastic exploration. If robots were able to explore and learn safely, minimizing excessive forces and impacts, they would last longer before needing repairs, and the objects they interact with would not need to be replaced as often.

Moreover, destructive or risky behaviors by robots trained with RL go beyond the exploration phase. Agents trained solely to maximize a reward signal will often converge on policies which are high velocity and thus high impact, resulting in so-called "bang-bang control," which may be optimal in terms of returns, but is potentially damaging or dangerous to the robot and the environment. As a further motivation, we might care about gentleness in terms of task execution itself, for instance if the robot needs to pick-and-place an object, but either the object and/or goal location is fragile. In particular, when robot manipulation involves humans (e.g., feeding a patient), being gentle is important [19]. In these situations, we would like robots to accomplish the given task in a reasonable amount of time, while minimizing applied force and impact as much as possible.

Thus, in order to broadly deploy deep RL on real robots, we need an approach for training policies that are gentle, both during exploration and task execution. A naïve approach is to constrain the maximum torques that a robot's motors can exert. However, many manipulation tasks require occasional, momentary, or variable high force (e.g., hammering a nail or turning a lever); the torque limit cannot be any lower than this, otherwise the robot will not be able to complete the task. But we do not want the robot to freely exert this much force along its entire trajectory. Alternatively, one could constrain the total amount of force or impact allowed, but this requires knowing *a priori* the minimum total amount necessary for accomplishing the task [3, 4, 50].

Instead, our approach is to give the robot negative rewards for actions that are not gentle, for instance those that result in high impact forces. Incorporating this in the reward function is a natural approach for encoding preferences about *how* robots should perform a task (e.g., driving style [1]), and can be seen as an intrinsic "pain" signal that encourages learning safer policies. However, perhaps unsurprisingly, adding this penalty often makes it much harder for an agent to learn a successful policy; instead, it gets stuck in a local optimum of avoiding contact altogether, because it encounters the penalties before ever obtaining the task reward, and thus learns a policy that is dominated by aversion to pain.

To motivate agents to interact with the environment *and* do it gently, we propose balancing this "pain" signal by adding another intrinsic signal, this one positive, for curiosity. In particular, we reward the agent for *surprising* experiences—those that contradict the agent's current understanding of the world. A concrete example of this is giving intrinsic rewards for transitions that have low probability under a learned dynamics model [2, 15, 35, 47]. However, we find that using this kind of *dynamics-based surprise* is not as effective as using a *penalty-based surprise*, that leads robots to be explicitly curious about the non-gentleness penalty itself. In this formulation, the agent makes predictions about the pain penalty that will result from a given state and action, and erroneous predictions deliver a small positive reward.

Although this curiosity about pain may seem somewhat counterintuitive, it has been shown that humans are specifically curious about painful or unpleasant experiences [16]; moreover, it is well known that children engage in physical risk-seeking behavior [29, 40, 41]. This is also observed in other species in the form of play fighting [46], and seems to have an evolutionary benefit. Research in developmental psychology suggests that risky play is essential to the development of children, by allowing them to test their physical limits, improve hand-eye coordination, and learn to avoid or adapt in dangerous environments [10, 20].

Motivated by this specialized form of curiosity, our work takes a step toward using deep RL to train policies for gentle, object- and contact-centric manipulation. In this work, gentleness is defined as minimizing impact forces. We demonstrate that our proposed approach, which introduces both a penalty for excessive impact forces and a curiosity reward focused on this penalty, enables efficient and safe exploration, precise task execution, and successful manipulation of fragile objects.

II. Related Work

Our goal of gentle manipulation is closely related to impact minimization in classical control, but existing approaches typically rely on having accurate dynamical contact models [17, 18, 52]. Another related domain is safe reinforcement learning [14, 36]: safety may refer to either physical safety or handling environment stochasticity. Typically, the former is concerned with avoiding catastrophic situations (e.g., crashing into another car or falling off a cliff), whereas in our work, we take a broader view of physical safety, in terms of reducing wear-and-tear in order to delay *eventual* damage.

In the context of developmental robotics, curiosity and intrinsic motivation take inspiration from developmental psychology, where they are believed to be essential for achieving autonomous mental development. This is realized by using an intrinsic curiosity drive that encourages a robot to focus on situations that are neither too simple (or predictable) nor too hard (or unpredictable) [7, 32, 33, 34]. These approaches, which often rely on engineered features and action scripts or primitives, have claimed that curiosity can drive a self-guided curriculum to train a robot arm for block manipulation and stacking [31]. Our approach follows in this vein, using curiosity-based intrinsic rewards to encourage agents to (gently) explore their environment, in the presence of impact penalties. Curiosity-based intrinsic rewards can also be viewed as providing reward shaping [30]: they make tasks easier to learn, by making it less likely for agents to get stuck in undesirable local optima.

In the context of deep RL, curiosity-based intrinsic rewards typically reward agents for either encountering novel states [9, 13, 49] or encountering surprising experiences [2, 15, 35, 47]. Recently, researchers have shown that intrinsic motivation can be used to learn human-like social skills in a human-robot interaction domain [37]. Our work investigates surprise-based intrinsic rewards, but we find that the usual method of computing this with respect to a learned dynamics model is not effective in our setting. Instead, we formulate surprise with respect to a penalty-prediction model.

In our work, the reward comes from a combination of several signals: extrinsic rewards from the task (positive), and intrinsic rewards from impacts (negative) and curiosity (positive). This falls under multi-objective reinforcement learning (MORL). Typically MORL approaches either train a single policy by finding the right balance of rewards, or learn a set of policies that approximate the Pareto optimal frontier [26, 38]. Some authors have used the Constrained MDP framework [4] to develop policy optimization methods which ensure that constraints are satisfied at all points throughout learning [3, 11]. While such methods could potentially augment our approach, we believe a simpler mechanism might suffice. We assume the balance between "pain" and task achievement should be encoded in the reward structure and naturally found by the agent; thus if it is very important to achieve the task, then a higher task reward conveys that being less gentle (e.g., experiencing higher impact forces) is acceptable. From this standpoint, we directly add these rewards together, rather than searching for a weighting that leads to the correct behavior.

An important aspect of our method is learning with tactile sensors. In [5] touch was used, along with proprioception and vestibular information, to learn awareness models that predict proprioceptive information about the agent's body and also represent objects in the external world. Touch has also been integrated in robotic exploration and model learning; it has been shown to improve performance in discriminating objects and in solving tasks involving multiple sensory modalities [5, 21, 22, 42, 53]. However, these latter approaches do not leverage intrinsic motivation to improve exploration strategies.

III. Preliminaries

A. Markov Decision Process

A Markov Decision Process (MDP) is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : S \times A \times S \mapsto \mathbb{R}$ specifies the transition probabilities, $\mathcal{R} : S \times A \times S \mapsto \mathbb{R}$ specifies the reward function, and $\gamma \in [0, 1]$ is the discount factor.

A policy π is a function that maps each state to a distribution over actions ($\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, where $\Delta_{\mathcal{A}}$ is the probability simplex on $\mathcal{A}$). Reinforcement learning optimizes policies to maximize expected returns (i.e., cumulative discounted future rewards):

$$\mathbb{E}_{a_t \sim \pi(s_t), s_{t+1} \sim \mathcal{P}(s_t, a_t)}[\sum_t \gamma^t \mathcal{R}(s_t, a_t, s_{t+1})].$$

A policy π's action-value function is

$$Q^\pi(s, a) = \int_{s'} P(s, a, s') \left(\mathcal{R}(s, a, s') + \gamma \mathbb{E}_{a' \sim \pi(s')}[Q^\pi(s', a')]\right).$$

Typically $\mathcal{R}$ specifies how well the policy is doing in terms of accomplishing a task. In this work, we augment $\mathcal{R}$ with several types of auxiliary rewards, in order to train policies for contract-centric, low-impact manipulation.

B. Deep Reinforcement Learning

The policy π can be represented by a function parameterized by θ; for instance, θ may be a weighting on predefined features of the state [1]. In deep RL, θ is the parameterization of a neural network. We use Distributed Distributional Deterministic Policy Gradients (D4PG) [8] to train our policies, but in principle our proposed approach is algorithm-agnostic.

D4PG is an actor-critic algorithm used to train policies for continuous control, where both the actor and critic are parameterized by neural networks. The critic is a distributional action-value function: it takes the current state s_t and action a_t as input, and outputs a categorical distribution over the predicted $Q(s_t, a_t)$. It is trained with off-policy evaluation, on batches of transitions (s_t, a_t, s_{t+1}) sampled from a replay buffer. The actor is a deterministic policy: it takes in the current state s_t as input, and outputs an action a_t. During training, the actor's policy is updated using gradients that are computed only with respect to the critic, such that actions are adjusted in the direction of increased Q-values.

C. Formalizing gentleness

In this work, we define being gentle as minimizing impact. This is closely related to the notion of impact force in physics, which is the maximum amount of force experienced during a collision. However, we consider a more general definition of "impact," that does not only apply to cases when the initial applied force is zero. Instead, assuming a discrete time step, we define impact m_t as

$$m_t = \max(0, f_{t+1} - f_t), \tag{1}$$

where f_t is the sensed force at time step t. In other words, for a robot to be gentle, it should minimize *increases in sensed force*. To illustrate, consider a robot that needs to push a heavy object with a force of 20N. If the robot increases the applied force from zero to 20N in a fraction of a second, the action is more likely to cause damage compared to amortizing the increase in force over several seconds.

IV. Proposed Approach

In order to train policies that exhibit gentle manipulation, we propose to augment the original reward (r_t) with an impact force penalty (r_t^f) and an intrinsic reward based on surprise (r_t^s). Agents are trained to maximize the total expected return,

$$r'_t = r_t + r_t^f + r_t^s. \tag{2}$$

A. Impact penalty

The impact force penalty acts as an intrinsic pain signal to encourage agents to accomplish manipulation tasks in a more gentle way. Of course, in order to accomplish any manipulation task, small impacts are necessary—at some point the robot needs to go from zero to non-zero applied force on an object, in order to manipulate it. So, the impact penalty should scale non-linearly with the level of impact, by taking into account the *acceptability* of a particular amount of impact.

Let $a_\lambda(m) \in [0, 1]$, parametrized by λ, be the acceptability of an amount of impact m. This is a monotonically increasing function, that should be designed according to how resilient the robot and environment are to impacts. For instance, if the robot is interacting with very fragile objects, then the range of acceptable impacts should be smaller. Ideally, this function should express the probability of damage (to either robot or environment) from a given amount of impact, and could be learned from experience of actual damage. In our experiments, since we do not have enough data on damage to estimate likelihoods, we use a sigmoid function for acceptability:

$$a_\lambda(m) = \text{sigmoid}(\lambda_1(-m + \lambda_2)) = \frac{1}{1 + e^{\lambda_1(m-\lambda_2)}} \tag{3}$$

The impact penalty at time step t is then:

$$r_t^f = -\sum_i (1 - a_\lambda(m_t^i)) m_t^i, \tag{4}$$

where the sum is over force sensors at different locations on the robot (e.g., the fingers of a robot hand). In our experiments, we set $\lambda = [2, 2]^\top$.

However, if the environment reward merely combines the task reward and the impact penalty, that is, $r'_t = r_t + r_t^f$, we find that policies get reliably stuck in a local optimum of not making contact with anything in the environment—the agent learns to be afraid of contact, since it encounters the impact penalty before the sparse task reward, hindering exploration.

B. Dynamics-based surprise

The purpose of adding surprise-based intrinsic rewards is to encourage policies to make *contact* with objects in the environment but still in a *gentle* way. For an agent to be "surprised," it must have some predictor of future states, i.e., a model. In the case of dynamics-based surprise, this model is a learned dynamics model that takes in the current state and action, and predicts the mean and variance of the next state. We train an ensemble of neural networks for the dynamics model, in order to have predictive uncertainty [23]. Predictive uncertainty is useful for capturing novelty: in the case of environments with deterministic dynamics, if the networks in the ensemble either individually have high variance in their predictions, or have high variance across the ensemble, then this indicates a novel area that should be explored further.

Each of the M networks in the ensemble outputs the mean and variance of a Gaussian for each dimension d of the prediction. The ensemble's combined output is a mixture of Gaussians for each output dimension d:

$$\frac{1}{M} \sum_i \mathcal{N}(\mu_{\theta_i}(\mathbf{x})_d, \sigma^2_{\theta_i}(\mathbf{x})_d),$$

where $\mathbf{x}$ denotes the input and θ_i are the parameters of the ith network in the ensemble. During training, each network is randomly initialized, and they are trained on different batches of transitions. We choose $M = 5$, as recommended by related work [23].

To compute dynamics-based surprise intrinsic reward r_t^s, we approximate the dynamics model's predicted distribution over next states with a single Gaussian per output dimension d, to

measure how much variance there is *across* networks in the ensemble. The mean and variance of this is

$$\mu_*(\mathbf{x})_d = \frac{1}{M}\sum_i \mu_{\theta_i}(\mathbf{x})_d$$
$$\sigma^2_*(\mathbf{x})_d = \frac{1}{M}\sum_i (\sigma^2_{\theta_i}(\mathbf{x})_d + \mu^2_{\theta_i}(\mathbf{x})_d) - \mu^2_*(\mathbf{x})_d.$$

Then the intrinsic reward is the negative log-likelihood of the true next state under this predicted distribution over next states:

$$r^s_t = -\sum_d \log \mathcal{N}(s_{t+1,d} \,|\, \mu_*(s_t,a_t)_d, \sigma^2_*(s_t,a_t)_d). \quad (5)$$

This intrinsic reward is computed with respect to a target dynamics model, which is updated every 5000 iterations; this makes training more stable, so that the agent is not trying to surprise a model that is constantly changing. In addition, we wait for the dynamics model to become more accurate before providing intrinsic rewards to the agent: after 20,000 training steps for experiments in simulation, and after 8,000 training steps on the real robot.

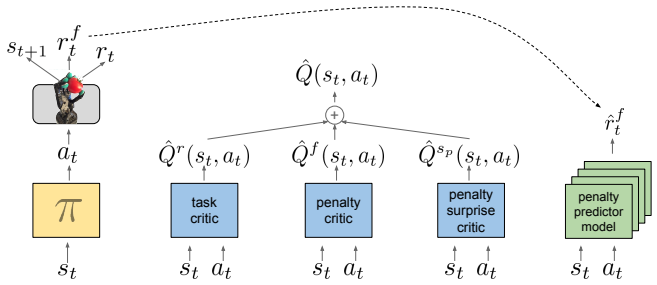


Fig. 1. Illustration of the architecture of the agent which uses penalty-based surprise. The policy, shown in yellow, takes actions on the environment, producing the next state, the task reward, and the penalty. There are three identical critics (a task reward critic, a penalty-surprise critic, and a penalty critic), in blue, which estimate action-values with respect to the different reward components. The penalty surprise reward is derived from the prediction error of the penalty predictor model, which is an ensemble of (five) differently-seeded networks.

C. Penalty-based surprise

Motivated by the role of risk-seeking behavior in childhood learning, we propose to reward the agent for being curious about the impact penalty itself. In other words, we add a reward to focus the learning and exploration of the agent on the intrinsic pain signal, with two motivations: to enable better prediction of pain, and to encourage contacts by mitigating some of the penalty. To compute the penalty-based surprise reward $r^{s_p}_t$, we train an impact penalty predictor in parallel with the agent, with the same implementation as the general dynamics model (an ensemble of five neural networks).

To compute the intrinsic reward, we do not use the negative log-likelihood directly, as was done with the dynamics reward, because it would make high impact forces acceptable as long as the prediction likelihood in those areas was low enough, leading to very non-gentle actions by the agent. Rather, we would like agents to focus on learning about areas with low penalty (i.e., areas where impacts occur but are small), so that they learn how to be gentle. For areas of high penalty, it is enough for the agent to just know that the penalty is high, not necessarily *exactly* how high it is. To enforce this preference for exploring areas with low penalty while avoiding ones with high penalty, a natural approach is to augment the task reward with a convex combination between the negative log-likelihood and the impact penalty:

$$\begin{aligned} &a_{\lambda'}(r^f_t) * -\log \mathcal{N}(r^f_t \,|\, \mu_*(s_t,a_t), \sigma^2_*(s_t,a_t))) \\ &+ (1 - a_{\lambda'}(r^f_t)) * r^f_t, \end{aligned} \quad (6)$$

where $a_{\lambda'}(r^f_t) \in [0,1]$ is the acceptability of a particular penalty r^f_t. This is a monotonically increasing function, that should be chosen based on how much penalty the robot may experience for the sake of exploration or task completion. We use a sigmoid function for this acceptability, as we did for impact $a_\lambda(m)$ (Sec. IV-A).

Note that λ and λ' play different roles: λ modulates the mapping of impact forces to pain penalties, whereas λ' controls the trade-off between these pain penalties and the agent's penalty-focused curiosity. The choice of λ' may be adjusted dynamically during learning; the higher λ'_2 is, the easier it is for the robot to learn the task, at the cost of higher impact forces on average.

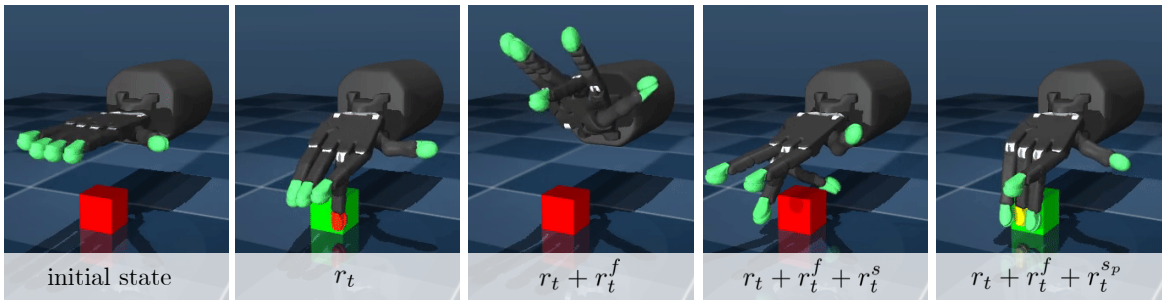
In order to augment the reward with this convex combination, we need to choose $r^{s_p}_t$ such that $r^{s_p}_t + r^f_t$ is equal to (6). In addition, we only provide this intrinsic reward if the penalty is non-zero, because the purpose is to encourage the agent to (cautiously) learn more about the penalty. Based on this, we set the penalty-based surprise intrinsic reward to be:

$$r^{s_p}_t = \begin{cases} a_{\lambda'}(r^f_t)\left[-\log \mathcal{N}(r^f_t \,|\, \mu_*(s_t,a_t), \sigma^2_*(s_t,a_t)) - r^f_t\right] \\ \quad \text{if } r^f_t < 0 \\ 0 \text{ otherwise.} \end{cases} \quad (7)$$

In the same way as dynamics-based surprise, this penalty-based surprise intrinsic reward is computed with respect to a target impact penalty predictor model, which is updated every 1000 iterations, and we do not provide intrinsic rewards to the agent until after 20,000 training steps for simulation experiments, and after 8,000 training steps for real robot experiments.

D. Agent architecture and implementation details

As mentioned, we use D4PG to train an actor (e.g., policy) and critic (e.g., action-value function). We use a separate critic for each of the reward components: the task reward critic encodes $\hat{Q}^t(s_t,a_t)$, the penalty-based surprise critic encodes $\hat{Q}^{s_p}(s_t,a_t)$, the dynamics-based surprise critic encodes $\hat{Q}^s(s_t,a_t)$, and the impact penalty critic encodes $\hat{Q}^f(s_t,a_t)$. The separation of the critics supports more stable learning [39]. The components of the agent with penalty-based surprise are illustrated in Figure 1.



(10) How can we have both a green block and a red fingertip?

Fig. 2. The task is to apply at least 5N of force to the block; green block indicates success. Fingertip color shows the amount of impact force, from yellow (near-zero) to red (10N). Policies were trained for 500k iterations. Policies trained on only task reward, r_t, learn how to do the task, but do it a high-impact way. In contrast, policies trained on the combination of task reward, an impact penalty, and penalty-based surprise intrinsic reward ($r_t + r_t^f + r_t^{s_p}$) learn to achieve the task in a *gentle* (i.e., low-impact) way, by gradually increasing force applied to the block. Without the penalty-based intrinsic reward, policies get trapped in a local optimum of avoiding contact with the environment ($r_t + r_t^f$ and $r_t + r_t^f + r_t^s$). Corresponding videos are available at http://sites.google.com/view/gentlemanipulation

The output of the actor is passed through a tanh, so that it is between -1 and +1. This output specifies delta position: it is added to the current position and then clipped based on the minimum and maximum joint angle per action dimension, to obtain the action. The actor network consists of two fully-connected layers of 300 and 200 hidden units each. Each of the critic networks consists of two fully-connected layers of 400 and 300 hidden units each. The distributional output of the critic has support $(-100, 100)$ and 101 bins. For both the actor and critic networks, the first hidden layer is followed by layer normalization [6] and a tanh, and all other hidden layers are followed by exponential linear unit (ELU) activations. For D4PG, we used a batch size of 256 and a replay buffer of 1 million transitions.

(11) (12)

The dynamics model consists of three ensembles, one each for predicting the three types of state features: joint position, joint velocity, and touch. The non-gentleness predictor model consists of a single ensemble. Each of these ensembles consists of five neural networks, with three fully-connected layers of 128 hidden units each. All hidden layers are followed by rectified linear unit (ReLU) activations.

V. Experiments

Our goal is to learn policies that are safer, with less forceful impacts, while also improving sample efficiency and overall task performance. The following experiments compare three approaches for achieving this; these approaches differ in terms of what the task reward is augmented with:

- an impact penalty (r_t^f)
- an impact penalty and a dynamics-based surprise intrinsic reward ($r_t^f + r_t^s$)
- an impact penalty and a penalty-based surprise intrinsic reward ($r_t^f + r_t^{s_p}$).

A. Experimental domain

We run experiments both in simulation with MuJoCo [51] and on a physical robot. The robot platform we use is the Shadow Dexterous Hand [44], with five fingers and a total of 24 degrees of freedom, actuated by 20 motors. We use this platform for several reasons: because it is actuated by antagonistic tendons, it is more susceptible to wear-and-tear (and thus gentle exploration and manipulation has greater potential benefit); it can be equipped with high-fidelity tactile sensors; and it is anthropomorphic and well suited for handling fragile objects.

Great band name!

In simulation, each fingertip has a spatial touch sensor attached, with three channels and a spatial resolution of 4×4: one for normal force and two for tangential forces. We simplify this by taking the absolute value and then summing across the spatial dimensions, to obtain a 3D force vector for each fingertip. The impact force m_t^i is then the sum over the increase in force per channel for fingertip i.

On our real-world Shadow Hand, BioTac® sensors [12, 48] provide a more complex array of tactile signals. To compute the forces exerted by each finger, readings from the pressure channel of each tactile sensor were acquired and then normalized to match the range of the simulated tactile sensors. In this way, it was possible to directly compare results in simulation and on the real robot, without having to change the parameters of the task or learning algorithm.

The state consists of proprioception (joint position and joint velocity) and touch. The action space is 20-dimensional. We use position control and a control rate of 20 Hz, both in simulation and on the physical robot.

The environment consists of the Shadow Hand and a single block (Fig. 2, Fig. 7); the task reward r_t depends on the experiment. Focusing on this simple environment enables us to clearly characterize the effectiveness of our three approaches for training low-impact policies. We find that even in this simple environment, learning policies for gentle manipulation is challenging for most approaches. Results from the simulated environment are presented first.

B. Exploration with impact penalty

First, we are interested in whether these approaches enable training policies that are gentle during exploration. We investigate this in a no-reward setting, where the policy receives intrinsic rewards (either from dynamics-based surprise or penalty-based surprise) and the intrinsic pain penalty, but

no task reward. The goal is for policies to be gentle (i.e., experience low impact) while still exploring effectively, in terms of interacting with objects in the environment.

As a baseline, we trained policies with only dynamics-based surprise intrinsic rewards, without an impact penalty. As expected, these policies experience a large amount of impact while exploring: the maximum amount of impact experienced per rollout is in the 5 to 15N range (Fig. 3, left). This suggests that this form of curiosity is not practical for running on real-world robots, if either the robot or the objects it interacts with are susceptible to wear and tear.

When we add the impact penalty, we do observe more gentle exploration: there is a significant decrease in the maximum amount of impact experienced per rollout (now in the 0 to 5N range), for both kinds of intrinsic reward. However, having penalty-based surprise intrinsic rewards leads to more gentle touching, whereas dynamics-based surprise leads to the policy exploring interesting configurations of the hand, but with limited touching (Fig. 3, center and right).

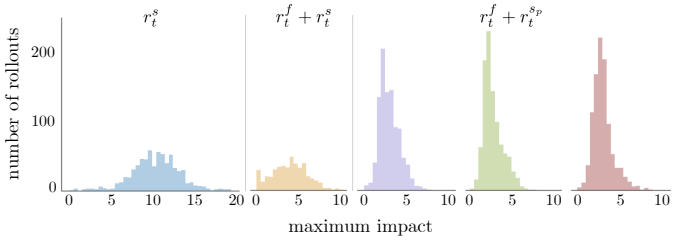


Fig. 3. We train policies in a no-reward setting, with either dynamics-based surprise (r_t^s), left; dynamics-based surprise plus impact penalty ($r_t^f + r_t^s$), center; or penalty-based surprise plus impact penalty ($r_t^f + r_t^{s_p}$), right. The histograms show the maximum amount of impact (in Newtons) per rollout; rollouts are collected regularly throughout 500k training steps, and rollouts with no impact at all are ignored. $\lambda_2' = 2$, 3 , or 4 . $\lambda_1' = 2$ for all.

C. Manipulation with impact penalty

Next, we are interested in whether these approaches enable training policies that learn how to perform a task gently, while still being relatively sample-efficient. In the task, the episode terminates with a reward of +1 if the hand presses the block with any fingerpad (thus activating the touch sensor) with a force greater than 5N. A non-gentle way of achieving this is to go from no contact to 5N of applied force in a single timestep; in contrast, policies trained to be gentle should more gradually increase to 5N of applied force.

This task is simple: without an impact penalty, agents learn this task quickly (Fig. 4, top), although with a significant amount of impact (Fig. 2, top center). However, once impact penalties are added, if there is no form of surprise-based intrinsic rewards to counteract them, then agents fail to learn the task—they get trapped in a local optimum of avoiding contact with the environment, since they experience penalties from contact before discovering how to perform the task (Fig. 2, center).

In line with the results from our previous experiment, in which we observed that dynamics-based surprise leads to only

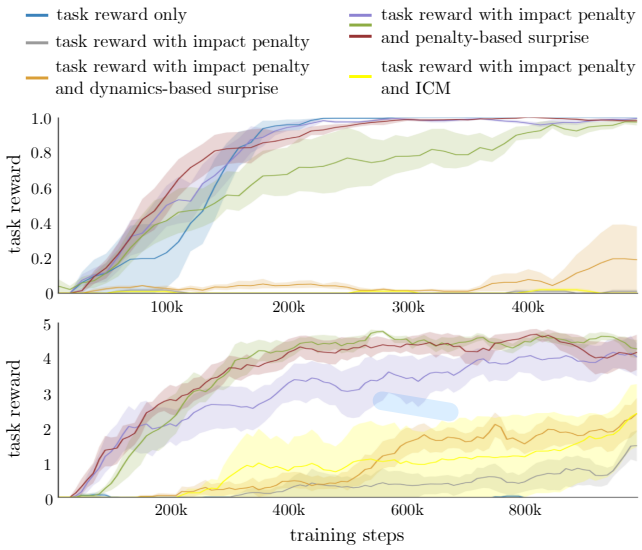


Fig. 4. Learning curves for training policies on different reward augmentations (with five random seeds each), for two tasks: pressing a block (**top**) or a *fragile* block (**bottom**) with greater than 5N of force. When the block is fragile, the episode terminates with a negative reward if the impact is greater than 3N. Our approach of training policies with a combination of task reward, impact penalty, and penalty-based surprise intrinsic reward is the only one that learns effectively for both tasks. Policies are trained on: task reward only ; task reward with impact penalty and no intrinsic rewards , dynamics-based surprise , or penalty-based surprise. The parameterization λ' for acceptability of penalties varies: $\lambda_2' = 1.5$, 2 , or 3 (top) and $\lambda_2' = 1$, 1.5 , or 2 (bottom). $\lambda_1' = 2$ for all. Policies trained on only task reward are unable to learn the fragile-block task at all (bottom). ICM agents also do not learn the tasks successfully.

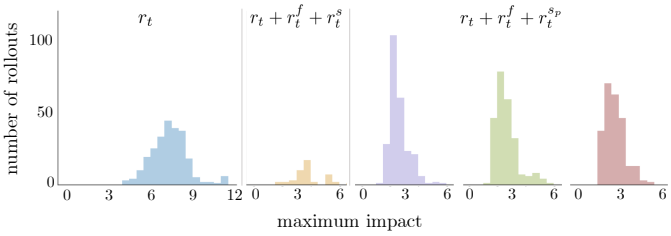


Fig. 5. We train policies to press a block with greater than 5N of force. These histograms show the maximum impact (in Newtons) experienced per rollout, when the agent performs the task successfully. Rollouts are collected after 500k training steps. $\lambda_2' = 1.5$, 2 , or 3 . $\lambda_1' = 2$ for all. (Note: with only an impact penalty, agents never succeed in performing the task, so that histogram is not shown.)

limited gentle touching in the presence of impact penalties, we saw that penalty-based surprise was much more effective in terms of agents learning how to perform the task gently, with low impacts (Fig. 5). Even more, these agents learned as quickly as ones trained without the impact penalty (Fig. 4, top). This may be because this task is particularly contact-focused (in general manipulation tasks are contact-focused, but to varying degrees), so it is a setting in which contact-focused exploration is especially helpful. We also compare our approach with agents trained using the Intrinsic Curiosity Module (ICM) proposed in [35]. Similarly to the agents trained with dynamics-based surprise, these agents do not successfully learn the task.

D. Manipulation of fragile objects

Finally, we made the task more difficult by introducing a 'fragile block'. This is analogous to a manipulation task involving fragile objects, such as picking ripe fruit or assisting humans. This fragile block 'breaks' if the impact force at any point is greater than 3N, and the episode terminates with a negative reward of -0.5. The reward for completing the task is +5. Now, policies trained with only the task reward are unable to learn the task at all, because they accidentally break the block a few times, and learn that any contact with the block is undesirable. There is no reward shaping that incentivizes these policies to try interacting with the block in a gentle way.

In contrast, policies trained with the impact penalty are better able to learn the task. As before, penalty-based surprise intrinsic rewards are more effective than dynamics-based ones in terms of how quickly policies are able to learn the task (Fig. 4, bottom).

Fig. 6 plots the evolution of reward components over time obtained by the agent trained on task reward, impact penalty and penalty-based surprise intrinsic reward. Each reward component in this plot is the average over batches sampled from the agent's replay buffer. The total reward (blue) is the combination of the pain surprise reward, the task reward, and the pain penalty. The dashed vertical line indicates the timestep when the intrinsic reward starts to be applied (i.e. after 20,000 timesteps). Before this point, the total reward is only affected by the task reward and the impact penalty.

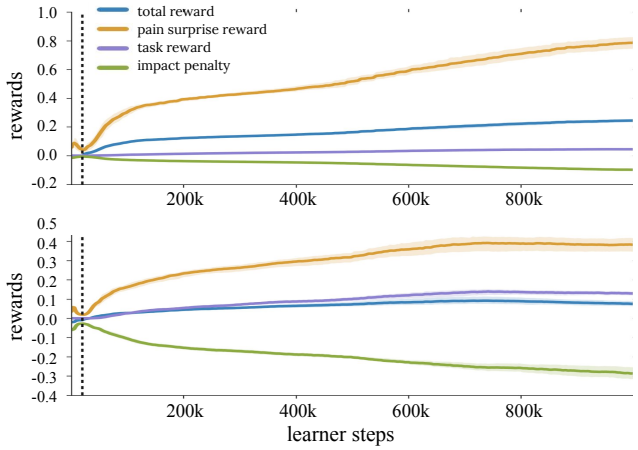


Fig. 6. Reward components relative to the agent trained on task reward, impact penalty and penalty-based surprise intrinsic reward, for the two tasks: pressing a block (**top**) or a *fragile* block (**bottom**). Each point in time represents the average reward (for each component) over a batch of samples.

E. Real robot experiments

We also conducted experiments for the two manipulation tasks on a real Shadow Dexterous Hand [44]. The setup used for these experiments is shown in Fig. 7. A force/torque sensor is attached to a foam block, and is used to measure the force on the block.

As in the experiments in simulation, for the first task, the episode terminates with a reward of +1 if the hand presses the block with any fingerpad with a force greater than 5N. For the fragile objects case, the episodes terminate with a negative reward if the impact force at any point is greater than 3N, and the reward for completing the task is +5.

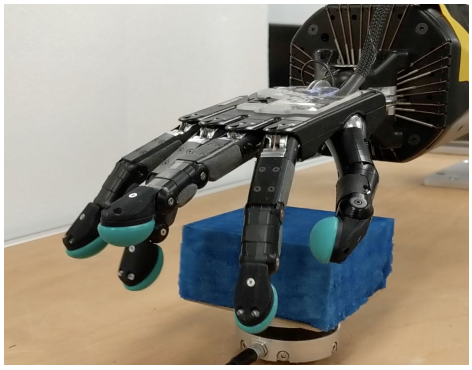


Fig. 7. Experimental setup on the real Shadow Dexterous Hand [44]. A force/torque sensor is attached to a foam block, to measure the force applied to the block. BioTac® tactile sensors [48] were used to compute the forces exerted by the corresponding fingers.

As baselines, we also ran a random agent, and agents trained using the Intrinsic Curiosity Module (ICM) proposed in [35]. Although these agents sometimes randomly hit the block, they do not consistently solve the tasks, and their interaction with the block often includes high-impact forces that exceed 5N.

We evaluated the different agents over 25,000 training steps. Since this is a shorter time window compared to that of the experiments executed in simulation, we increased the learning rate by one order of magnitude (from 0.0001 to 0.001). In line with the results obtained in simulation, we saw that penalty-based surprise was much more effective in learning how to perform the task gently, both in terms of learning speed (Fig. 8) and minimizing impacts (Fig. 9). The agent trained with dynamics-based surprise continues exploring the (complex) dynamics of the system on the real robot, and thus struggles to learn the simple manipulation task. On the other hand, the agent trained with penalty-based surprise is not only successful on the simple manipulation task, but notably it is the one that learns the task of manipulation of fragile objects more consistently.

VI. Discussion and Future Work

Our work takes a step toward using deep RL to train policies for gentle, contact-rich manipulation. Although curiosity has long been established as a source of endogenous motivation for artificial agents exploring the world, it may be too broad and general to drive an agent towards contact-rich policies, especially when penalties are used to discourage high-impact interactions. We found that in this scenario, choosing the appropriate focus of curiosity is important for incentivizing agents to interact gently with the environment. This enables efficient and safe exploration, precise task execution, and successful manipulation of fragile objects. Although the proposed approach is demonstrated on relatively simple tasks, we believe

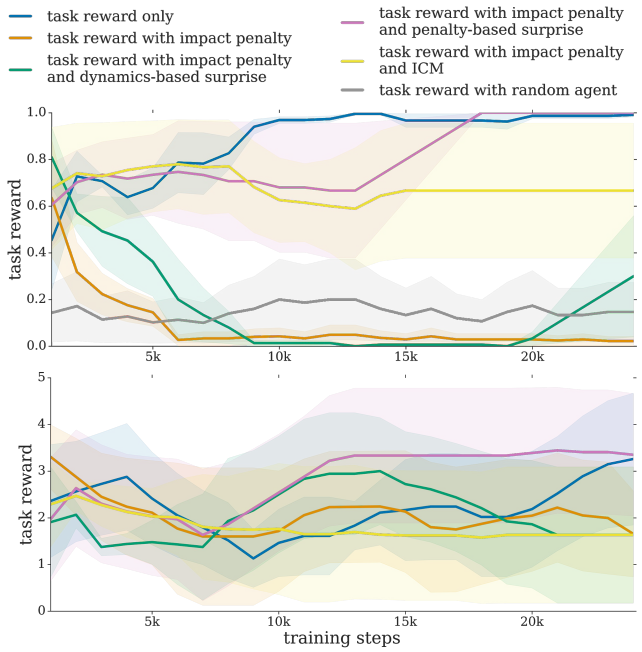


Fig. 8. Learning curves for training policies on different reward augmentations (with three seeds each), on the real Shadow hand for two tasks: pressing a block (**top**) or a *fragile* block (**bottom**) with greater than 5N of force. When the block is fragile, the episode terminates with a negative reward if the impact is greater than 3N. Our approach of training policies with a combination of task reward, impact penalty, and penalty-based surprise intrinsic reward is the only one that learns effectively for both tasks. Policies are trained on: task reward only, task reward with impact penalty and no intrinsic rewards, dynamics-based surprise, or penalty-based surprise. The parameterization λ' for acceptability of penalties is $\lambda'_1 = 2$ and $\lambda'_2 = 1.5$. As baselines, we also report a random agent on the touch task, and ICM agents.

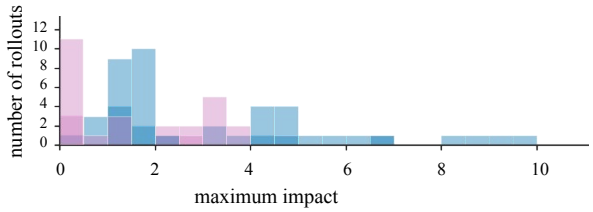


Fig. 9. We train policies to press a block with greater than 5N of force. These histograms show the maximum impact (in Newtons) experienced per rollout, when the agent performs the task successfully. Colors refer to agents trained on task reward only and penalty-based surprise. $\lambda'_1 = 2$ and $\lambda'_2 = 1.5$. (Note that agents trained on task reward with impact penalty and no intrinsic rewards and dynamics-based surprise, the random agent and the ICM agents don't learn the task, hence the corresponding histograms are not shown.)

that it paves the way towards a new direction in curiosity research, one that identifies more nuanced types of curiosity and intrinsic motivation for deep RL agents.

A main direction of future work is to apply this approach to more complex tasks, for instance in dynamic environments. In addition, this work only considers one aspect of being gentle—impact. Our approach could be used to train policies while minimizing other sources of wear and tear, for instance total force (rather than the increase in force), or the torques exerted by a robot's motors (which would reduce energy consumption as well [28]). Additionally, we note that although we use a multimodal robot environment—integrating tactile and proprioceptive sensors—we have not incorporated vision, which would provide an additional observation to support tactile and force predictions. Future work will seek to establish the value of contact-focused curiosity across this broader multimodal landscape.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, 2004. ISBN 1-58113-838-5.

[2] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

[3] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning (ICML)*, 2017.

[4] Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, 1999.

[5] Brandon Amos, Laurent Dinh, Serkan Cabi, Thomas Rothörl, Alistair Muldal, Tom Erez, Yuval Tassa, Nando de Freitas, and Misha Denil. Learning awareness models. In *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, 2018.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[7] Adrien F Baranes, Pierre-Yves Oudeyer, and Jacqueline Gottlieb. The effects of task difficulty, novelty and the size of the search space on intrinsically motivated exploration. *Frontiers in neuroscience*, 8:317, 2014.

[8] Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy P. Lillicrap. Distributed distributional deterministic policy gradients. In *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, 2018.

[9] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the Twenty-Ninth Advances in Neural Information Processing Systems (NIPS)*, 2016.

[10] Mariana Brussoni, Lise L. Olsen, Ian Pike, and David A. Sleet. Risky play and children's safety: Balancing priorities for optimal child development. *International Journal of Environmental Research and Public Health*, 9 (9):3134–3148, 09 2012.

[11] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[12] Jeremy A. Fishel and Gerald E. Loeb. Sensing tactile microvibrations with the biotac—comparison with human

sensitivity. In *Proceedings of the Fourth IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012.

[13] Justin Fu, John D. Co-Reyes, and Sergey Levine. EX2: exploration with exemplar models for deep reinforcement learning. In *Proceedings of the Thirtieth Advances in Neural Information Processing Systems (NIPS)*, 2017.

[14] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

[15] Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Proceedings of the Twenty-Ninth Advances In Neural Information Processing Systems (NIPS)*, 2016.

[16] C. K. Hsee and B. Ruan. The pandora effect: The power and peril of curiosity. *Psychological Science*, 2016.

[17] Jingchen Hu and Tianshu Wang. Pre-impact configuration designing of a robot manipulator for impact minimization. *Journal of Mechanisms and Robotics*, 9(3), 2017.

[18] Panfeng Huang, Wenfu Xu, Bin Liang, and Yangsheng Xu. Configuration control of space robots for impact minimization. In *IEEE International Conference on Robotics and Biomimetics*, pages 357–362. IEEE, 2006.

[19] Koji Ikuta, Hideki Ishii, and Makoto Nokata. Safety evaluation method of design and control for human-care robots. *The International Journal of Robotics Research*, 22(5):281–297, 2003.

[20] Tom Jambor. Risk-taking needs in children: An accommodating play environment. *Children's Environments Quarterly*, 3(4):22–25, 1986.

[21] Mohsen Kaboli, Di Feng, Kunpeng Yao, Pablo Lanillos, and Gordon Cheng. A tactile-based framework for active object learning and discrimination using multimodal robotic skin. *IEEE Robotics and Automation Letters*, 2 (4):2143–2150, 2017.

[22] Oliver Kroemer, Christoph H Lampert, and Jan Peters. Learning dynamic tactile sensing with robust vision-based training. *IEEE Transactions on Robotics*, 27(3):545–557, 2011.

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the Thirtieth Advances in Neural Information Processing Systems (NIPS)*, 2017.

[24] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[25] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the Fourth International Conference on Learning Representations (ICLR)*, 2016.

[26] Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3), 2015.

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. In *Neural Information Processing Systems (NIPS) Workshop on Deep Learning*, 2013.

[28] Abdullah Mohammed, Bernard Schmidt, Lihui Wang, and Liang Gao. Minimizing energy consumption for robot arm movement. *Procedia CIRP*, 25:400–405, 2014.

[29] Barbara A. Morrongiello and Shawn Matheis. Understanding children's injury-risk behaviors: The independent contributions of cognitions and emotions. *Journal of Pediatric Psychology*, 32(8):926–937, 05 2007.

[30] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, 1999.

[31] Hung Quoc Ngo, Matthew D. Luciw, Alexander Förster, and Jürgen Schmidhuber. Learning skills from play: Artificial curiosity on a Katana robot arm. In *International Joint Conference on Neural Networks (IJCN)*, pages 1–8, 2012.

[32] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? A typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

[33] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

[34] Pierre-Yves Oudeyer, Adrien Baranes, and Frédéric Kaplan. Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 303–365. Springer, 2013.

[35] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the Thirty-Fourth International Conference on Machine Learning (ICML)*, 2017.

[36] Martin Pecka and Tomas Svoboda. Safe exploration techniques for reinforcement learning – an overview. In *Modelling and Simulation for Autonomous Systems*, 2014.

[37] Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. Intrinsically motivated reinforcement learning for human–robot interaction in the real-world. *Neural Networks*, 2018.

[38] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48(1):67–113, 2013.

[39] Stuart Russell and Andrew L. Zimdars. Q-decomposition for reinforcement learning agents. In *Proceedings of the Twentieth International Conference on Machine Learning*

(ICML), 2003.

[40] Ellen B. H. Sandseter. Categorising risky play—how can we identify risk-taking in children's play? *European Early Childhood Education Research Journal*, 15(2):237–252, 2007.

[41] Peter C. Scheidt, Yossi Harel, Ann C. Trumble, Diane H. Jones, Mary D. Overpeck, and Polly E. Bijur. The epidemiology of nonfatal injuries among US children and youth. *American Journal of Public Health*, 85(7): 932–938, 07 1995.

[42] Alexander Schmitz, Yusuke Bansho, Kuniaki Noda, Hiroyasu Iwata, Tetsuya Ogata, and Shigeki Sugano. Tactile object recognition using deep learning and dropout. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 1044–1050. IEEE, 2014.

[43] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the Thirty-Second International Conference on Machine Learning (ICML)*, 2015.

[44] ShadowRobot. ShadowRobot Dexterous Hand. https://www.shadowrobot.com/products/dexterous-hand/.

[45] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[46] Marek Spinka, Ruth C. Newberry, and Marc Bekoff. Mammalian play: Training for the unexpected. *The Quarterly Review of Biology*, 76(2):141–168, 06 2001.

[47] Bradly C. Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[48] Inc. SynTouch. BioTac Technologies. https://www.syntouchinc.com/en/sensor-technology/.

[49] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Proceedings of the Thirtieth Advances in Neural Information Processing Systems (NIPS)*, 2017.

[50] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

[51] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[52] Liang-Boon Wee and M. W. Walker. On the dynamics of contact between space robots and configuration control for impact minimization. *IEEE Transactions on Robotics and Automation*, 9(5):581–591, 1993.

[53] Martina Zambelli and Yiannis Demiris. Multimodal imitation using self-learned sensorimotor representations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3953–3958. IEEE, 2016.

Questions and (Sometimes) Answers

1. How do you "surprise" a robot?
   The authors define a "surprising" experience as one "that contradicts the agent's current understanding of the world."

2. Positive reward for *incorrect* predictions?

   "dynamics-based surprise": "Giving intrinsic rewards for transitions that have low probability under a learned dynamics model." ← Less effective

   "penalty-based surprise": similar to observational research of children at play, the policies are trained to be curious about the pain penalty itself by being rewarded for encountering it. Together, the reward for completing the task + penalty-based surprise reward – pain penalty is the agent's score. ← More effective

3. Propioception: "Perception or awareness of the position of the body."

4. Vestibular information: information about objects in the external world (so just like it sounds!)

5. Probability simplex: In probability theory, the points of the standard n-simplex in $(n + 1)$-space are the space of possible parameters (probabilities) of the categorical distribution on $n + 1$ possible outcomes. (Wikipedia)

6. Replay buffer: sounds a bit like this video game feature (how very DeepMind of them!): "The replay buffer allows you to save the last x seconds of video and audio to your disk." https://obsproject.com/forum/resources/obs-classic-how-to-use-the-replay-buffer.103/

7. Q-value: In statistical hypothesis testing, specifically multiple hypothesis testing, the ***q*-value** provides a means to control the positive false discovery rate (pFDR).[1] Just as the *p*-value gives the expected false positive rate obtained by rejecting the null hypothesis for any result with an equal or smaller *p*-value, the *q*-value gives the expected pFDR obtained by rejecting the null hypothesis for any result with an equal or smaller *q*-value. (Wikipedia)

8. Why $\lambda = [2, 2]^T$, instead of just $\lambda = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$?

   Must allow greater flexibility for dimensions in other areas?

9. Convex combination: In convex geometry, a **convex combination** is a linear combination of points (which can be vectors, scalars, or more generally points in an affine space) where all coefficients are non-negative and sum to 1. As a particular example, every convex combination of two points lies on the line segment between the points.

10. If a green block indicates success and a red fingertip indicates failure, how can both appear simultaneously as in the second image of Fig. 2?

    Ah. Block color indicates success in terms of contact vs non-contact, and fingertip color indicates success in terms of amount of force. Different kinds of success.

11. "This output specifies delta position: it is added to the current position and then clipped based on the minimum and maximum joint angle per action dimension to obtain the action."

    What's a "joint angle?" If it's the angle of the joint of the robot's "finger," does that mean we'd have to figure out a different approach for, say, an octopus robot (no joints!)?

12. "The distributional output of the critic has support $(-100, 100)$ and 100 bins."

    What do "support" and "bin" mean in this context?

    "… probabilities over categorical variables representing different reward values ranges. They've tested different amounts of bins splitting the possible value ranges: 5, 11, 21, 51, and 51 bins outperformed others with a large margin. Values that are beyond this range were clipped into the last bin." https://towardsdatascience.com/learning-distributions-over-rewards-leads-to-state-of-the-art-in-rl-5afbf70672e