Active Learning of Affordances for Robot Use of Household Objects

Chang Wang, Koen V. Hindriks and Robert Babuska

Abstract— **Learning to perform household tasks is a key step towards developing cognitive service robots. This requires that robots are capable of discovering how to use human-designed products. In this paper, we propose an active learning approach for acquiring object affordances and manipulation skills in a bottom-up manner. We address affordance learning in continuous state and action spaces without manual discretization of states or exploratory motor primitives. During exploration in the action space, the robot learns a forward model to predict action effects. It simultaneously updates the active exploration policy through reinforcement learning, whereby the prediction error serves as the intrinsic reward. By using the learned forward model, motor skills are obtained to achieve goal states of an object. We demonstrate through real-world experiments that a humanoid robot NAO is able to autonomously learn how to manipulate two types of garbage cans with lids that need to be opened and closed by different motor skills.**

I. INTRODUCTION

A fundamental challenge in developing cognitive service robots is how to endow them with autonomous learning capabilities for handling household objects, i.e., to learn what objects afford what actions in a given context. The concept of affordance [9] has been introduced in robotics to address the problem of robot-object interaction [17], [19], [24]. The key benefit of learning affordances is that they can be generalized across objects for predicting action effects, e.g., based on shape features [8], [22]. Affordances can be used in various ways, such as for planning [30], imitation [19], control [12], and tool use [27], [29]. However, the affordance learning conditions in the above literature were strongly controlled by human programmers and this restricts the autonomy of the robot. Not only the amount of training data required was assumed known before affordance learning actually started, but also the motor primitives were predefined and assumed to be always effective for object manipulation. These assumptions do not guarantee that a robot can learn how to manipulate a household object which can be complex and unseen before by the robot. In this paper, we take an active learning approach where the robot decides by itself whether it has collected sufficient data to learn the underlying object affordances. Besides, a range of reusable motor skills are acquired in a bottom-up manner without manual discretization of the continuous state space or robot action space.

There has been much research on robot skill learning via human demonstration [5], [15], [16], where complex robot motions can be learned by mixing basic motor primitives. This approach is quite effective for object manipulation with human-provided training samples. However, it is still a challenge for autonomous and open-ended skill learning when such human guidance is not available or is too expensive to obtain. The same challenge exists in the aforementioned literature on affordance learning. In this paper, we address this challenge by an active learning approach.

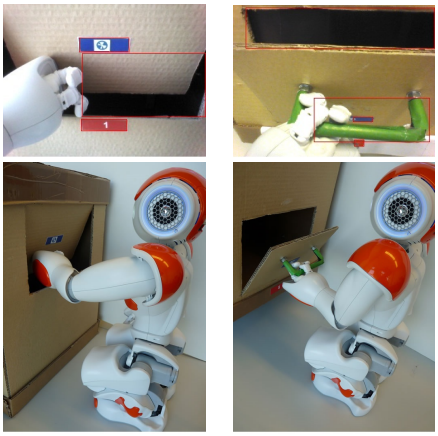
Active learning is a machine learning technique that allows active selection of training data [25]. In robotics, active learning can be used for efficient acquisition of knowledge and skills during continuous interaction with environments. For example, a robot actively generates uncertain situations [18], or queries a human teacher [6] to reduce the amount of training data for learning symbolic concepts. Without human guidance, active learning can be driven by intrinsic motivations such as artificial curiosity, surprise, or fear [1], [4], [20]. Heuristics typically direct active exploration towards the regions where uncertainty or prediction errors are maximal [7], [10]. The change of prediction errors can also be used as an intrinsic reward to optimize the learning progress [2], [21]. A relevant approach [11] proposed active learning of controllable environmental contexts for object manipulation, but again the motor skills were preprogrammed and high-level control programs were given.

In this paper, we propose active affordance learning in the framework of intrinsically motivated reinforcement learning [3]. Specifically, we use the actor-critic reinforcement learning (RL) architecture [28] to learn action exploration policies. In order to control the state changes of objects through robot actions, forward models [14] are learned through function approximation and used to predict action effects in continuous state and action spaces. The prediction error is not only a means to update the forward models, but it also serves as the intrinsic reward signal to update the critic and the actor. Then, the forward models are reused to achieve goals, during which a range of manipulation skills are acquired in a bottom-up manner. Finally, these motor skills are associated with object representation defined by perceptual proxies [13] that provide the contextual information of the learned affordances. In this way, the robot can learn to handle objects and ground the object representations in its own sensory and motor experience.

Throughout the paper, we use the manipulation of garbage cans with different lids (see Fig. 1) as a running example. As a prerequisite for learning affordances [19], we assume that the robot is already equipped with appropriate sensory functions and motor skills, such that it can recognize objects and perform elementary motor tasks such as controlling its arm in Cartesian space.

All authors are with the TU Delft Robotics Institute, The Netherlands. {c.wang-2, k.v.hindriks,r.babuska}@tudelft.nl

This paper will appear in the proceedings of 2014 IEEE-RAS International Conference on Humanoid Robots.



(a) Push to open (b) Pull to open

Fig. 1. An illustration of different motor skills (push and pull) and their effects on the lid opening. The upper row shows the images by the NAO's camera. Note that these motor skills are to be learned by the robot rather than preprogrammed. A video is available at http://youtu.be/oluLDwMaVoY.

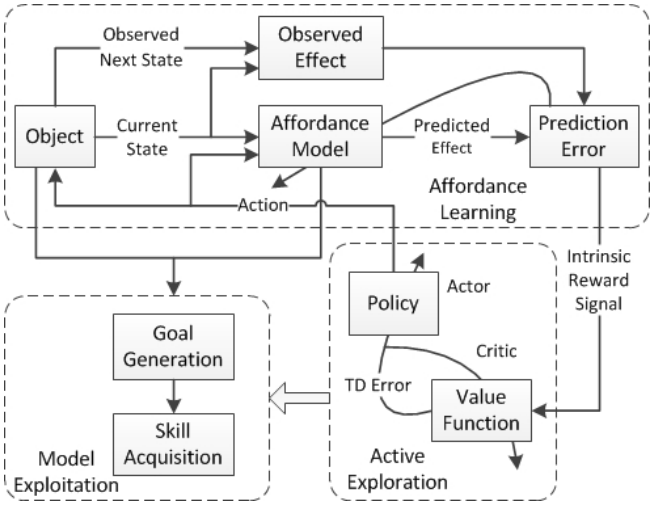


Fig. 2. An architecture of active learning of object affordances.

In our previous work, we proposed on-line affordance learning in goal-directed tasks [33], as well as transfer learning of affordances across objects [32]. However, random exploration policy was applied and the motor primitives were predefined for specific tasks. The main contributions of this paper are:

- An active learning architecture updates affordance models simultaneously with exploration policies.
- Affordance learning takes place in continuous state and action spaces without prior discretization.
- Manipulation skills are acquired in a bottom-up manner without human intervention.

The paper is organized as follows: Section II describes the active learning architecture. Section III discusses the formal affordance learning model. Section IV introduces the task environment with the experimental results. Section V concludes the paper and outlines our plans for future work.

II. Active Affordance Learning Architecture

As in previous work [33], we define an affordance as the triple:

$$(Object, Action, Effect) \tag{1}$$

Object refers to a household object and/or its part along with its state, e.g., the state of a garbage can lid that is partially open. *Action* refers to a repertoire of motor commands that can be used to interact with the object, e.g., the change of joint angles or end-effector positions. *Effect* refers to the outcome of applying the action to the object, e.g., the handle is displaced, or the lid is open (see Fig. 1).

The overall architecture we propose for active learning of affordances is illustrated in Fig. 2. It consists of three components: affordance learning, active exploration, and model exploitation.

In the affordance learning component, we define an affordance model that associates the three elements of (1). First of all, an object is recognized by extracting features from sensory input. As we take into account household objects that are composed of several parts, each part is represented by a perceptual proxy, e.g., a bounding box (see Fig. 1). In this case, we can obtain the state of each part as the size and location of the bounding box. Then, the affordance model is represented as a forward model [14] that produces a prediction of an effect, based on the state of object and the selected action. As a result, affordance learning is to learn forward models associated with perceptual proxies of an object. For example, a robot learns a forward model to predict the lid opening as a result of its arm and hand movement. Whenever an action is performed, the forward model makes a prediction about the consequent effect. After the actual next state is observed, the actual effect is measured by comparing the states of lid opening before and after the action, e.g., by subtraction of the opening size. Then, the prediction error of the effect is calculated to update the forward model. In this paper, we choose a feedforward neural network as the forward model and use back-propagation to update the model based on the prediction error.

In the active exploration component, the prediction error of the affordance model provides an intrinsic reward signal to optimize the action selection policy which outputs an action for the affordance learning component. This is an active learning approach in the sense that the training data collection is controlled by the robot itself based on its own observation of the environment without human intervention. The underlying heuristic is straightforward: the affordance model is maximally corrected when the sampled state and action spaces have the highest prediction error. In the actor-critic RL architecture, the actor plays the key role that determines the new samples for learning the affordance model. It outputs an action which is to be performed on the object, and the action is also the input of the affordance model for effect prediction. The critic learns to predict the value of each state and computes the Temporal Difference (TD) error [28], which is used by the actor to output optimal actions that will maximize the accumulated future rewards.

Finally, active model learning stops when the TD errors become stable.

In the model exploitation component, the robot evaluates the quality of the learned models by itself. It generates goals in the effect space and selects actions to achieve them. A range of manipulation skills can be acquired in various object states for solving these goal-directed tasks. The underlying assumption is that the learned model is good enough for use, which is guaranteed by the convergence of the actor-critic structure in the active exploration component. In this way, the robot can develop object manipulation skills autonomously when no task is specified by human. In the case of garbage can manipulation, a robot may use the learned affordance model to open, close or move a garbage can in a given object configuration.

III. Formal Affordance Learning Model for Handling Household Objects

We now discuss in detail each of the three elements of (1) as well as the three components in Fig. 2.

A. Affordance model

1) Perception of object and parts: In this paper, a robot perceives its environment and extracts visual features from its camera image. We assume that the robot can identify object parts based on known features (markers in our experiments) and color segmentation. The robot then recognizes a household object as a combination of all observed parts. Refer to [26] for a method to recognize object parts with a RGB-D camera. As our focus is on active learning, such a method is beyond the scope of this paper.

Denote by Ψ the set of all known object parts (body, lid, handle, pedal, etc.). As not all objects necessarily contain the same parts, denote by $\Psi_o \subseteq \Psi$ the set of parts that an object o is composed of. We use $s_o \in S$ to denote the state of the object o. The state changes with time and is continuously measured by robot's sensors. For example, s_o can be the current size of the lid opening. We note that s_o may also include the states of other parts in Ψ_o if necessary.

2) Robot actions: Robot actions can be defined in the constrained joint space as well as in the Cartesian space. In this paper, we control a robot arm in 3D Cartesian space with available inverse kinematics. This does not exclude other action representations in our active learning architecture.

Denote by $s_r = (x, y, z)^T \in \mathbb{R}^3$ the current state of a robot's end-effector in the 3D space. Denote by $a = (\Delta x, \Delta y, \Delta z)^T \in \mathcal{A} \subset \mathbb{R}^3$ a bounded action that changes the position of the end-effector. In our case, the robot interacts with only one part of the object using one end-effector at a time. The robot always approaches the vicinity of the chosen object part before interacting with it. The reaching and grasping behaviors are assumed available in the robot's motor skill repertoire.

3) Effects of actions: The effect of action a on object o is denoted by $e_o \in \mathcal{E}_o$. It is measured by

$$e_o = m(s_o, s'_o) \tag{2}$$

where s'_o is the state of o after a was applied, and m is a suitable metric, e.g., subtracting s_o from s'_o.

4) Forward models: A forward model is an internal model that produces a predicted output based on a given input [14]. In our case, the input is the current object state and the applied action, and the output is the predicted effect. Then, object affordances are encoded in the following forward models $\mathcal{F}_\psi$:

$$e_o = \mathcal{F}_\psi(s_o, a, w) \text{ for } \psi \in \Psi_o \tag{3}$$

where ψ indicates that the robot interacts with a specific part $\psi \in \Psi_o$ by performing a. In our case, $\mathcal{F}_\psi$ is a neural network and w is the weight vector. Other function approximation approaches are also applicable.

B. Affordance learning

Affordance learning is to learn the forward models (3) by updating the model parameters based on prediction errors. We use an on-line version of neural networks. Denote by $(s_o^k, a^k, e_o^k), k \in \mathbb{N}$ the collected data after applying an action a^k, where s_o^k and e_o^k are the corresponding object state and consequent effect. The decision of data sampling and its termination will be discussed in Section III-C.

When learning a forward model $\mathcal{F}_\psi$, denote by $\hat{e}_o^k$ the predicted effect of a^k in the state s_o^k, i.e.,

$$\hat{e}_o^k = \mathcal{F}_\psi(s_o^k, a^k, w^k) \tag{4}$$

where w^k is the current weight vector. The prediction error η_k is obtained as follows:

$$\eta_k = e_o^k - \hat{e}_o^k \tag{5}$$

where $e_o^k = m(s_o^k, s_o^{k+1})$ is obtained from (2). Then, the new model parameter w^{k+1} is updated as follows:

$$w^{k+1} = w^k + \alpha\, \eta_k \nabla \mathcal{F}_\psi(s_o^k, a^k, w^k) \tag{6}$$

where $0 \leq \alpha \leq 1$ is the step size parameter, $\nabla \mathcal{F}_\psi$ is the gradient of the output of the network to the weight vector.

C. Active learning with intrinsic motivation

The goal of active affordance learning is to autonomously learn the relations between objects, actions and effects in an efficient manner. In our approach, this means that the robot needs to learn forward models. Meanwhile, the policy of selecting exploratory actions should also be learned to optimize the affordance learning process. A baseline to be compared with is the random action selection policy.

In order to learn the exploration policy, we integrate an RL component in the affordance learning loop (see Fig. 2). A conventional RL scheme requires manual definition of a reward function to develop goal-directed exploration behaviors for a specific goal. In our architecture, the reward signal is generated intrinsically by using the prediction error of a forward model, whose maximization is expected to result in an optimal action selection policy. The underlying heuristic is that sampling in state and action spaces with higher prediction error is more rewarding rather than sampling in the already well-predicted area.

We have chosen to use Continuous Actor-Critic Learning Automation (CACLA) because it has been proved to have good performance for RL problems in continuous action spaces [31]. Like other actor-critic algorithms, CACLA is based on the simultaneous online approximation of two structures, the actor and the critic. The actor corresponds to an action selection policy, mapping states to actions in a probabilistic manner. The critic corresponds to a value function, mapping states to expected cumulative future reward.

An actor is represented as a function approximator Act_k that approximates the function $Act^* : S \rightarrow \mathcal{A}$, where $Act^*(s_o^k)$ denotes the optimal action for state s_o^k. A critic is also represented as a function approximator V_k that approximates a state value function $V : S \rightarrow \mathbb{R}$ which stores the expected sum of discounted rewards for states. The strategy of active exploration is learned as follows.

During exploration, an action a^k is selected stochastically from the Gaussian probability function $G(x, \mu, \sigma)$ centered around the output of the current actor $Act_k(s_o^k)$:

$$G(x, Act_k(s_o^k), \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x - Act_k(s_o^k))^2/2\sigma^2} \quad (7)$$

where σ is an exploration parameter. If $\mathcal{A}$ is more than one dimension, σ could be chosen separately for each dimension.

After action a^k is applied on the object, the new state s_o^{k+1} is observed, and the actual effect is compared with the predicted effect to get the prediction error by (5). The current reward is given as the absolute value of this prediction error:

$$r = |\eta_k| \quad (8)$$

Then, the current TD error [28] is obtained as follows:

$$\delta^k = r + \gamma V_k(s_o^{k+1}) - V_k(s_o^k) \quad (9)$$

where $0 \leq \gamma \leq 1$ discounts future rewards.

The current actor Act_k is updated only if $\delta^k > 0$, which means that the performed action a^k is better than expected and should therefore be enforced. The actor Act_k is then updated towards this action:

$$Act_{k+1}(s_o^k) = Act_k(s_o^k) + \zeta(a^k - Act_k(s_o^k)) \quad (10)$$

where $0 \leq \zeta \leq 1$ is a step size parameter.

The critic is always updated with the TD error:

$$V_{k+1}(s_o^k) = V_k(s_o^k) + \beta\, \delta^k \quad (11)$$

where $0 \leq \beta \leq 1$ is a step size parameter.

The action exploration process terminates when the RL almost stops, i.e., there is not much change in the actor-critic RL component. This is measured by the convergence of the TD error, when the following condition is satisfied:

$$|\bar{\delta}^{k+1} - \bar{\delta}^k| < \epsilon \quad (12)$$

where ϵ is a small positive threshold, and

$$\bar{\delta}^k = \frac{1}{N} \Sigma_{i=k-N+1}^{k} |\delta^i| \quad (13)$$

is the averaged absolute TD error of recent N actions.

The whole loop of active affordance learning is summarized in Algorithm 1. In case of endless exploration, the loop terminates anyway after a maximal number N_s of actions.

Algorithm 1 Active affordance learning of an object.

Input: An object o; Maximal action steps N_s;
Output: Forward models $\mathcal{F}_\psi, \psi \in \Psi_o$ as in (3);
Iteration:

```
for all ψ ∈ Ψ_o do
  Initialize k = 1;
  while k ≤ N_s and (12) is not satisfied do
    Observe the object state s_o^k;
    Select an exploratory action a^k using (7);
    Predict the action effect ê_o^k using (4);
    Apply a^k and observe the resulted object state s_o^{k+1};
    Calculate the prediction error η_k using (5);
    Update the parameter w^k of F_ψ using (6);
    Calculate the intrinsic reward r using (8);
    Calculate the TD error δ^k using (9);
    if δ^k > 0 then
      Update Act_k using (10);
    end if
    Update the critic V_k using (11);
    k ← k + 1;
  end while
end for
```

D. Model exploitation for skill acquisition

In order to evaluate the learned models (3), goals are generated in the effect space $\mathcal{E}_o$ to show whether useful manipulation skills can be acquired. In each dimension of $\mathcal{E}_o$, the robot rehearses internally and selects an action to maximize or minimize an effect. For example, the maximized goal effect e_o^g is:

$$e_o^g = \underset{\psi \in \Psi_o, a \in A_M}{\arg\max} \; \mathcal{F}_\psi(s_o, a, w) \quad (14)$$

where s_o is the current object state, and $A_M \subset \mathcal{A}$ is a set of M samples, e.g., evenly sampled in $\mathcal{A}$.

The acquisition of a motor skill starts in an initial object state s_o^i and terminates when no more effect is observed. Then, this object state is the termination state s_o^t. The learned skills are represented as a sequence of primitive actions with initial and termination conditions. They are similar to options [23] that can be transfered across tasks. The whole skill learning process terminates when the current termination state is similar to an initial state of learned skills. In the garbage can example, the robot would first choose to open it and then close it if it is initially closed, vice versa.

IV. A CASE STUDY: GARBAGE CAN MANIPULATION

We used a humanoid robot NAO and two garbage cans to test our active affordance learning model (see Fig. 1).

A. Task Setting

In our experiment, the garbage cans were presented to NAO separately. One had a pushable lid (Fig. 1(a)), and the other had a pullable handle (Fig. 1(b)). In each learning trial, a garbage can was positioned approximately 10 to 12 cm in front of NAO and the area to be explored was about 25 to 45

cm high. These values agreed with the capabilities of NAO due to its height and the length of its arms. Only the left arm of NAO was used to interact with the garbage cans. The garbage cans were reachable and manipulatable by NAO.

The bottom camera on NAO's head was used as the main sensory input, with a resolution of $W \times H$ (e.g., 320×240). For each garbage can, the same blue marker ($5\,\text{cm} \times 2\,\text{cm}$) was used for the recognition of lid (with a NAO marker at its center), and a green marker for the recognition of the handle ($10\,\text{cm} \times 1\,\text{cm}$), if there was one. The markers were recognized based on color segmentation. As a result, the set of object parts was $\Psi = \{\psi_l, \psi_h\}$ where ψ_l denoted a lid, ψ_h denoted a green handle. Each $\psi \in \Psi$ was located by a bounding box (see the top row in Fig. 1).

In this paper, the state of a garbage can was described by its openness. To detect the opened area, we put a black plastic bag in each garbage can and calculated the area of the dark part in an captured image. The opened area was also located by a bounding box with a size of $w \times h$ in pixels. Then, s_o was the percentage of opened area in an image:

$$s_o = \frac{w \times h}{W \times H} \tag{15}$$

where $0 \leq s_o \leq 1$.

At each time step, a robot action a was selected from $\mathcal{A} = \{(x, y, z)^T \in \mathbb{R}^3 | -0.01 \leq x, y, z \leq 0.01\}$ (in meters)[1]. After an action was performed, the robot captured another image and obtained the new state of object s'_o in the same way as (15). The effect was obtained as follows:

$$e_o = s'_o - s_o \tag{16}$$

To approximate each forward models in (3), we used a feed-forward neural network with four input neurons (one neuron for s_o and three neurons for a), one hidden layer with 10 neurons and one output neuron for e_o. We also used two neural networks to approximate the actor and critic. We normalized the action values to $[-1, 1]$ in each dimension. For the three layers of all neural networks, we used linear, hyperbolic tangent and linear transfer functions, respectively. All weights of neural networks were initialized randomly in $[-0.3, 0.3]$. The learning rates in (6), (10) and (11) were $\alpha = 0.3$, $\zeta = 0.3$, $\beta = 0.3$. The Gaussian exploration parameter in (7) was $\delta = 0.2$ for each action dimension. The discount factor in (9) was $\gamma = 0.9$. The TD errors were averaged over $N = 20$ actions in (12) and $\epsilon = 1 \times 10^{-4}$.

We tested the active exploration approach against the baseline of random exploration. In the case of random exploration, we used a random actor and its output was a random number in $\mathcal{A}$. We ran experiments in both 1D action space (X axis of NAO space) and 3D Cartesian space for the two garbage cans. In all experiments, NAO performed the first 20 actions randomly, then it continued random exploration or switched to the active learning mode. The maximal allowed exploration steps were $N_s = 100$ and $N_s = 300$, respectively.

[1]In the Cartesian space of NAO, the X axis is positive toward NAO's front, the Y from right to left and the Z is vertical. For more details, refer to http://www.aldebaran-robotics.com.

B. Results

1) Learned forward models: The result of a learned forward model is shown in Fig. 3 (active exploration of the push-lid in 1D state space and 1D action space). The state space was $[0, 0.2]$ and the action space was $[-0.01, 0.01]$ (in meters). They were meshed into 10×10 grids for plotting the surface of predicted effects. For example, action $= 0$ corresponded to $a = (-0.01, 0, 0)^T$ and action $= 10$ corresponded to $a = (0.01, 0, 0)^T$.

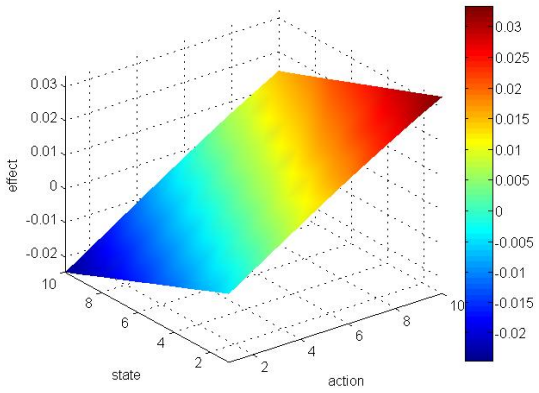


Fig. 3. The learned forward model with the push-lid using active exploration in 1D action space ($0 \leq$ state ≤ 0.2, $-0.01 \leq$ action ≤ 0.01).

Fig. 3 shows the linear relations between states, actions and effects. Stretching the arm (action > 5) would result in the opening effect (effect > 0), and stretching further would result in more opening. Besides, the maximal opening effect decreases when the current state of opening increases. This prediction agrees with the hinged design of the push-lid. Similarly, the closing effect (effect < 0) was predicted by contracting the arm (action < 5).

2) Convergence of affordance learning: The averaged TD errors in (13) during action exploration in both 1D and 3D action spaces are shown in Fig. 4 - Fig. 7. In all experiments, they converged for active exploration while the random exploration failed to converge within allowed number of action steps. Besides, the active exploration in 1D action space converged faster than in 3D action space.

In the active learning mode, NAO intended to explore the most uncertain spaces in an organized way. It usually ended up being blocked by the boundaries of garbage cans, i.e., when a lid was maximally opened or tightly closed. In this case, the object state became stable and no more effect was observed, which gave the TD errors a good chance to converge. In contrast, the random exploration was less efficient because it wasted time on exploring in well predicted action space which contributed little to improving the model prediction accuracy. Besides, it occasionally ran into situations with high prediction errors so that the TD errors would take longer time to converge.

3) Skill acquisition: The initial and termination object states are shown along with the acquired motor skills in TABLE I. Due to page limitation, we note that not all actions

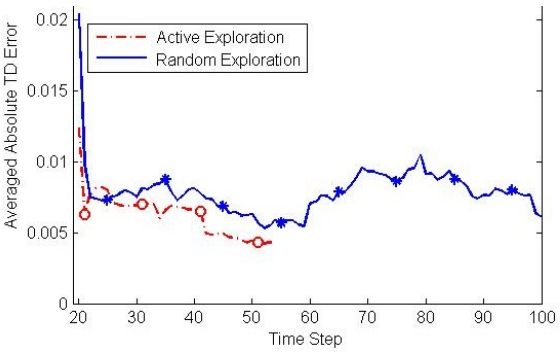


Fig. 4. Experimental result with the push-lid in 1D action space.

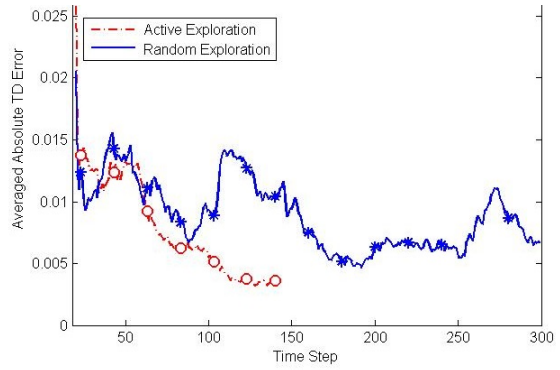


Fig. 5. Experimental result with the push-lid in 3D action space.

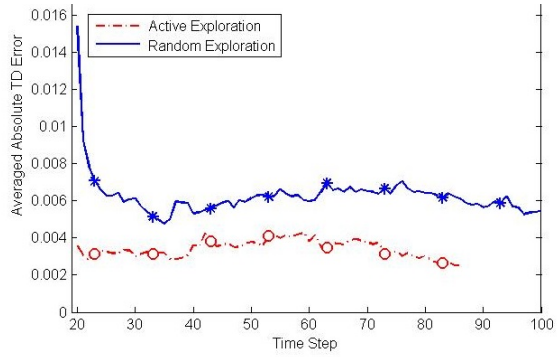


Fig. 6. Experimental result with the pull-handle in 1D action space.

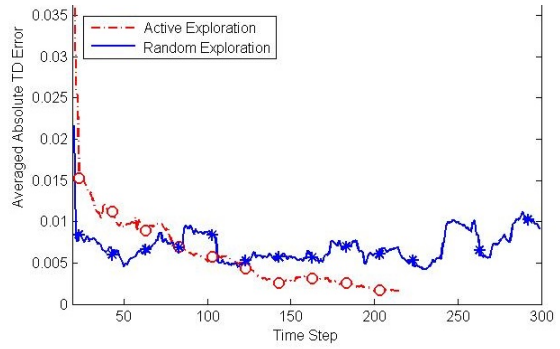


Fig. 7. Experimental result with the pull-handle in 3D action space.

are listed, and the values are rounded to three decimal points. In all cases, NAO started skill learning with a closed lid and succeeded to choose correct action sequences to maximize the opening effect first, then maximize the closing effect.

TABLE I

THE ACQUIRED MOTOR SKILLS BY SELF-GENERATED GOALS (e_o^g) IN DIFFERENT STATES (s_o) USING THE EIGHT FORWARD MODELS (TWO PARTS, TWO EXPLORATION POLICIES, AND TWO ACTION SPACES).

ψ	policy	s_o	a (1D or 3D)	e_o^g	error
ψ_l	random	0	0.008	0.018	0.009
		0.114	0.008	0.003	-0.0005
		0.117	-0.010	-0.031	-0.023
		0.018	-0.010	-0.020	-0.020
ψ_l	active	0	0.008	0.033	0.008
		0.124	0.008	0.020	0.004
		0.135	-0.010	-0.019	-0.002
		0.035	-0.010	-0.008	-0.008
ψ_l	random	0	(0.008,0.008,0.002)	0.034	0.018
		0.072	(0.008, 0.008, 0)	0.024	0.024
		0.071	(-0.010, 0.008, -0.010)	-0.012	0.009
		0.005	(-0.010, 0.008, -0.010)	-0.005	-0.004
ψ_l	active	0	(0.008, 0.010, 0.010)	0.031	0.012
		0.112	(0.008, 0.010, 0.010)	0.021	0.021
		0.094	(-0.010, -0.010, -0.010)	-0.044	-0.009
		0.024	(-0.010, -0.010, -0.010)	-0.039	-0.039
ψ_h	random	0	-0.010	0.008	0.001
		0.124	0.008	0.020	0.004
		0.135	-0.010	-0.019	-0.002
		0.035	-0.010	-0.008	-0.008
ψ_h	active	0	-0.010	0.017	0.006
		0.101	-0.010	0.014	0.014
		0.101	0.008	-0.006	0.038
		0	0.008	-0.004	-0.005
ψ_h	random	0	(-0.010, 0, -0.010)	0.017	0.024
		0.119	(-0.010, 0.008, -0.010)	0.010	0.009
		0.123	(0.008, -0.010, 0.008)	-0.019	0.019
		0	(0.008, -0.010, 0.008)	-0.012	-0.012
ψ_h	active	0.012	(-0.010, -0.008, -0.008)	0.009	-0.003
		0.094	(-0.010, -0.008, -0.008)	0.012	0.011
		0.095	(0.008, 0.008, 0.008)	-0.003	-0.004
		0	(0.008, 0.008, 0.008)	-0.003	-0.003

In the push-lid case with 1D action space, NAO first performed 5 or 6 pushing actions ($a = 0.008$ m) until the lid was maximally opened ($s_o = 0.114$). Then, it contracted the arm ($a = -0.01$ m) until the object state was not changed anymore ($a = 0.018$ m). The push-lid could not be closed completely by NAO due to the friction between the lid and the body part (see the attached video). In the 3D cases, the direction of the optimal action for opening was slightly different with the 1D cases. For example, NAO pushed forward the push-lid while moving the arm left and up, i.e., $a = (0.008, 0.010, 0.010)^T$. NAO also pulled the handle while moving the arm downwards, i.e., $a = (-0.01, 0, -0.010)^T$. These results agreed with the design of hinges on the lids.

V. CONCLUSIONS

In this paper, we investigated an approach for active learning of affordances in continuous state and action spaces for robot use of household products. Affordances were learned on-line to predict action effects meanwhile the prediction

error served as intrinsic reward to update the action exploration policy using an actor-critic RL structure. We have demonstrated that a humanoid robot is able to actively learn affordances and efficiently acquire manipulation skills to handle garbage cans. In the future, we will consider the scale of model complexity and the speedup of model convergence, along with the transfer of learned exploration policies for learning novel objects.

REFERENCES

[1] G. Baldassarre and M. Mirolli. Intrinsically motivated learning systems: an overview. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 1–14. Springer, 2013.

[2] A. Baranès and P.-Y. Oudeyer. R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169, 2009.

[3] A. G. Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer, 2013.

[4] A. G. Barto, S. Singh, and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning (ICDL 2004)*, pages 112–19, 2004.

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. *Handbook of robotics*, 1, 2008.

[6] M. Cakmak, C. Chao, and A. L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.

[7] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *arXiv preprint cs/9603104*, 1996.

[8] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about objects through action - initial steps towards artificial cognition. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 3140 – 3145 vol.3, sept. 2003.

[9] J. J. Gibson. *The ecological approach to visual perception*. Houghton Mifflin, 1979.

[10] G. Gordon, E. Fonio, and E. Ahissar. Learning and control of exploration primitives. *Journal of computational neuroscience*, pages 1–22, 2014.

[11] S. Hart and R. Grupen. Intrinsically motivated affordance discovery and modeling. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 279–300. Springer, 2013.

[12] T. Hermans, J. M. Rehg, and A. Bobick. Guided pushing for object singulation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4783–4790. IEEE, 2012.

[13] T. Hermans, J. M. Rehg, and A. F. Bobick. Decoupling behavior, perception, and control for autonomous learning of affordances. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4989–4996. IEEE, 2013.

[14] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354, 1992.

[15] E. Klingbeil, A. Saxena, and A. Y. Ng. Learning to open new doors. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2751–2757. IEEE, 2010.

[16] J. Kober, A. Wilhelm, E. Oztop, and J. Peters. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, 33(4):361–379, 2012.

[17] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, et al. Object–action complexes: Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems*, 59(10):740–757, 2011.

[18] J. Kulick, M. Toussaint, T. Lang, and M. Lopes. Active learning for teaching a robot grounded relational symbols. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence(IJCAI)*. AAAI Press, 2013.

[19] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15 –26, feb. 2008.

[20] P.-Y. Oudeyer, F. Kaplan, et al. How can we define intrinsic motivation? In *proceedings of the 8th international conference on epigenetic robotics: modeling cognitive development in robotic systems*, 2008.

[21] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286, 2007.

[22] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner. Learning to perceive affordances in a framework of developmental embodied cognition. In *Proceedings of IEEE 6th International Conference on Development and Learning*, pages 110 –115, july 2007.

[23] T. J. Perkins, D. Precup, et al. Using options for knowledge transfer in reinforcement learning. *University of Massachusetts, Amherst, MA, USA, Tech. Rep*, 1999.

[24] E. Sahin, M. Cakmak, M. R. Dogar, E. Ugur, and G. Ucoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.

[25] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[26] S. C. Stein, F. Wörgötter, M. Schoeler, J. Papon, and T. Kulvicius. Convexity based object partitioning for robot applications. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[27] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3060 – 3065, April 2005.

[28] R. Sutton and A. Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[29] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta. Exploring affordances and tool use on the icub. 2013.

[30] E. Ugur, E. Oztop, and E. Sahin. Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems*, 59(7–8):580–595, 2011.

[31] H. van Hasselt and M. A. Wiering. Using continuous action spaces to solve discrete problems. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1149–1156. IEEE, 2009.

[32] C. Wang, K. Hindriks, and R. Babuska. Effective transfer learning of affordances for household robots. In *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2014.

[33] C. Wang, K. V. Hindriks, and R. Babuska. Robot learning and use of affordances in goal-directed tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.