Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Planning Multimodal Exploratory Actions for Online Robot Attribute Learning

Xiaohan Zhang
State University of New York at Binghamton
Binghamton, NY 13902, USA
Email: xzhan244@binghamton.edu

Jivko Sinapov
Tufts University
Medford, MA 02155, USA
Email: jivko.sinapov@tufts.edu

Shiqi Zhang
State University of New York at Binghamton
Binghamton, NY 13902, USA
Email: zhangs@binghamton.edu

***Abstract*—Robots frequently need to perceive object attributes, such as "red," "heavy," and "empty," using multimodal exploratory actions, such as "look," "lift," and "shake." Robot attribute learning algorithms aim to learn an observation model for each perceivable attribute given an exploratory action. Once the attribute models are learned, they can be used to identify attributes of new objects, answering questions, such as "Is this object red and empty?" Attribute learning and identification are being treated as two separate problems in the literature. In this paper, we first define a new problem called online robot attribute learning (On-RAL), where the robot works on attribute learning and attribute identification simultaneously. Then we develop an algorithm called information-theoretic reward shaping (ITRS) that actively addresses the trade-off between exploration and exploitation in On-RAL problems. ITRS was compared with competitive robot attribute learning baselines, and experimental results demonstrate ITRS' superiority in learning efficiency and identification accuracy.** [1]

I. Introduction

Intelligent robots are able to interact with objects through exploratory behaviors in real-world environments. For instance, a robot can take a $look$ behavior to figure out if an object is "red" using computer vision methods. However, vision is not sufficient to recognize if an opaque bottle is "full" or not, and behaviors that support other sensory modalities, such as $lift$ and $shake$, become necessary. Given the sensing capabilities of robots and the perceivable properties of objects, it is important to develop algorithms to enable robots to use multimodal exploratory behaviors to identify object properties, answering questions such as "*Is this object red and empty?*" In this paper, we use **attribute** to refer to a perceivable property (of an object) and use **behavior** to refer to an exploratory action that a robot can take to interact with the object.

Robot multimodal perception is a challenge for several reasons. First, exploratory behaviors can be costly, and even risky in the real world. For instance, to $shake$ a water bottle to identify the value of attribute "empty", the robot must first $grasp$ and $lift$ it. Those behaviors take time and can break the bottle in case of failed grasps. Second, those behaviors are not equally useful for recognizing different attributes. For instance, $lift$ is more useful than $look$ for "heavy," while $look$ works much better for "shiny." **Robot attribute learning** (RAL) algorithms aim to learn an observation model for each attribute given an exploratory behavior and play a key role in robot multimodal perception. Most existing RAL algorithms are considered **offline**: the robot learns the attributes by interacting with objects *without* considering data collection costs. In the evaluation phase, the robot uses the learned attributes to identify attributes of new objects (i.e., attribute identification). In this research, we are concerned with a novel **online** RAL (On-RAL) setting, where the robot needs to learn an action policy for interacting with objects toward efficient attribute learning and accurate attribute identification at the same time.

On-RAL faces the fundamental trade-off between exploration and exploitation. A trivial solution is to let the robot optimize its behaviors solely on attribute identification as if the attributes have been learned already. In doing so, the robot still learns the observation models of attribute-action pairs as it becomes more experienced, but this trivial solution lacks a mechanism for actively improving its long-term attribute identification performance. The main contribution of this paper is an algorithm, called *information-theoretic reward shaping* (**ITRS**), for On-RAL problems. ITRS, for the first time, equips a robot with the capability of optimizing its sequential action selection toward (efficiently and accurately) learning and identifying attributes at the same time, as shown in Figure 1. ITRS has been evaluated using two datasets: one dataset, called **CY101**, contains 101 objects with ten exploratory behaviors and seven types of sensory modalities [36]; and the other, called **ISPY32**, includes 32 objects with eight behaviors and six types of modalities [39]. Compared with existing methods from the RAL literature [2, 41], ITRS reduces the overall cost of exploration in the long term while reaching a higher accuracy of attribute identification.

II. Related Work

A. Multimodal Perception in Robotics

Significant advances have been achieved recently in computer vision, e.g., [18, 28] and natural language processing, e.g., [6, 7]. While language and vision are important communication channels for robotic perception, many object properties cannot be detected using vision alone [12] and people are not always available to verbally provide guidance in exploration tasks. Therefore, researchers have jointly modeled language and visual information for multimodal text-vision

[1] Additional materials are available at https://sites.google.com/view/on-ral

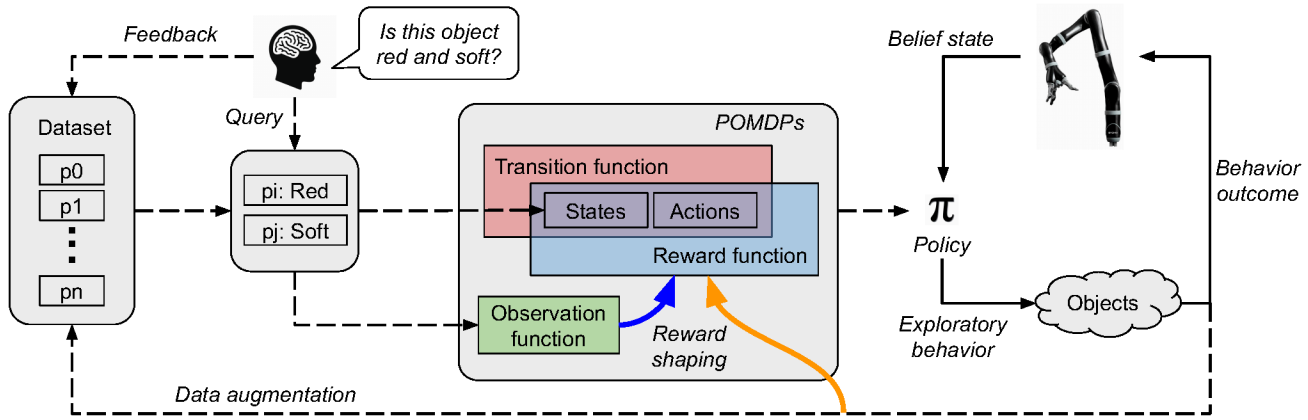


Fig. 1: An overview of the ITRS algorithm. A human user will choose an object and ask a query such as "*Is this object red and soft?*". The robot will generate a perception model on the specified attributes, i.e., "red" and "soft". Queried attributes and the corresponding perception model then will be used to construct states and the observation function of the POMDP model respectively. The reward function will be shaped by the quality of the observation function and the robot's experience. The robot uses the generated POMDP model to compute a policy π and interacts with the queried object. Newly-perceived feature data will be used to update the robot's experience and augment the dataset. Humans will give feedback to the robot's answer and attach labels to the feature data points.

tasks [27]. However, many of the most common nouns and adjectives (e.g., "soft", "empty") have a strong non-visual component [22] and thus, robots would need to perceive objects using additional sensory modalities to reason about and perceive such linguistic descriptors. To address this problem, several lines of research have shown that incorporating a variety of sensory modalities is the key to further enhance the robotic capabilities in recognizing multisensory object properties (see [4] and [21] for a review). For example, visual and physical interaction data yields more accurate haptic classification for objects [11], and non-visual sensory modalities (e.g., *audio*, *haptics*) coupled with exploratory actions (e.g., *touch* or *grasp*) have been shown useful for recognizing objects and their properties [5, 10, 15, 24, 30], as well as grounding natural language descriptors that people use to refer to objects [3, 39]. More recently, researchers have developed end-to-end systems to enable robots to learn to perceive the environment and perform actions at the same time [20, 42].

A major limitation of these and other existing methods is that they require large amounts of object exploration data, which is much more expensive to collect as compared to vision-only datasets. A few approaches have been proposed to actively select behaviors at test time (e.g., when recognizing an object [9, 31] or when deciding whether a set of attributes hold true for an object [2]). One recent work has also shown that robots can bias which behavior to perform at training time (i.e., when learning a model grounded in multiple sensory modalities and behaviors) but they did not learn an actual policy for doing so [41]. Different from existing work, we propose a method for learning a behavior policy for object exploration that a robot can use when learning to ground the semantics of attributes.

B. Planning under Uncertainty

Decision-theoretic methods have been developed to help agents plan behaviors and address uncertainty in non-deterministic action outcomes [26, 35]. Existing planning models such as partially observable Markov decision process (POMDP) [13], belief space planning [25] and Bayesian approaches [29] have shown great advantages for planning robot perception behaviors, because robots need to use exploratory actions to estimate the current world state. To learn semantic attributes, robots frequently need to choose multiple actions, so POMDP which is useful for long-term planning is particularly suitable. Many of the POMDP-based robot perception methods are vision-based [8, 34, 43, 45]. Compared to those methods, our robot takes advantage of non-visual sensory modalities, such as *audio* and *haptics*.

Work closest to this research plans under uncertainty to interact with objects using multimodal exploratory actions [2], where they modeled the mixed observability [23] in domains of a robot interacting with objects (we use their work as a baseline approach in experiments). The work of Amiri et al. [2] and this work share the same spirit from the planning and perception perspectives. The main difference is that their work assumed that sufficient training data and annotations are available for the robot to learn the perception models of its exploratory actions. In comparison, we consider a more challenging setting, called "Online RAL," where the data collection and task completion processes are simultaneous.

C. Robot Attribute Learning (RAL)

To select actions to identify objects' perceivable properties (e.g., "heavy," "red," "full," and "shiny"), robots need observation models for their exploratory actions. Researchers have developed algorithms to help robots determine the presence of possibly new attributes [17] and learn observation models of objects' perceivable properties (i.e., attributes) given different exploratory actions [33, 39, 40]. In the case where the object attributes refer to the object's function, they are then referred to as 0-order affordances [1]. Those methods focused on learning to improve the robots' perception capabilities. Once the learning process is complete, a robot can use the learned attributes to perform tasks, such as attribute identification (e.g., to tell if a

Fig. 2: Everyday objects used in the demonstrations and evaluations of this research [36].

bottle is "heavy" and "red"). Compared to those learning methods, we consider an online multimodal RAL setting, where the robot learns the attributes (an exploration process) and uses the learned attributes to identify object properties (an exploitation process) at the same time. The exploration-exploitation trade-off is a fundamental decision-making challenge in unknown environments. While the problem has been studied in multi-armed bandit [14] and reinforcement learning settings [35], it has not been studied in RAL contexts.

III. Theoretical Framework

In this section, we first formally define three types of robot multimodal perception problems, including On-RAL. We then describe information-theoretic reward shaping (ITRS), a novel algorithm for On-RAL problems.

A. Problem Definitions

A robot has a set of actions, such as $look$, $push$, and $lift$, that can be used for interacting with everyday objects as shown in Figure 2. Let $o \in Obj$ be an object and $a \in A^e$ be an *exploratory action*. Each action is coupled with a set of sensory modalities, e.g., $vision$, $haptics$, and $audio$. We use $m \in \mathcal{M}$ to refer to a sensory modality. This action-modality coupling is formulated using function Γ:

$$\mathcal{M}_a = \Gamma(a) \tag{1}$$

where $\mathcal{M}_a \subseteq \mathcal{M}$. For example, $\{audio, haptics, vision\} = \Gamma(push)$ means that action $push$ produces signals from $audio$, $haptics$, and $vision$ modalities.

Each action-modality pair specifies a set of *viable* combinations $\mathcal{C}$, and $c \in \mathcal{C}$ is called a sensorimotor context:

$$\mathcal{C} = A^e \otimes^{\Gamma} \mathcal{M} \tag{2}$$

where $\otimes^{\Gamma}$ is a *viable Cartesian product* operation that outputs only those viable pairs from $A^e \times \mathcal{M}$, and the viability is determined by Γ. We use c_a^m to refer to the context specified by m and a. For instance, $(look, vision)$ is a viable context, whereas $(look, audio)$ is not. When a is performed on object o, for each $m \in \mathcal{M}_a$, the robot is able to record a data instance f_a^m. We use f_a to represent the instances of all modalities that a robot receives after performing a.

Let $\mathcal{P}$ specify a set of attributes that are used to describe objects in a domain. Given object o, v^p is either *true* or *false*, depending on if p applies to o, where we use $ID(p, o)$ to refer to this attribute identification function. Here we "override" function ID to use it to process a set of attributes:

$$\mathbf{v} = ID(\mathbf{p}, o) \tag{3}$$

where $\mathbf{v} = [v^{p_0}, v^{p_1}, \cdots]$, $\mathbf{p} = [p_0, p_1, \cdots]$, and v^{p_i} is the value of attribute p_i of object o. For instance, given a red empty object (i.e., o) and two attributes of $blue$ and $empty$ (i.e., $\mathbf{p}$), the $ID(\mathbf{p}, o)$ outputs $[false, true]$.

Definition 1 (Off-RAL). *At training time, the input includes a set of labeled sensory data instances, each in the form of $(f_a, p) : v^p$. Solving an **offline RAL** (Off-RAL) problem produces a binary classifier:*

$$\Psi(f_a, p), \text{ for each pair of } a \in A^e \text{ and } p \in \mathcal{P}$$

At testing time, given object o, a robot collects data instances f_a after performing action a and $\Psi(f_a, p)$ outputs *true* or *false* estimating if attribute p applies to o.

Definition 2 (RAI). *Solving a **robot attribute identification** (RAI) problem produces policy π that sequentially selects action $a \in A^e$ to identify the value of:*

$$ID(\boldsymbol{p}, o), \text{ given } \Psi$$

where the objective is to identify the attribute(s) in each identification task while minimizing the total action cost.

Definition 3 (On-RAL). *Solving an **online RAL** (On-RAL) problem produces policy π that sequentially selects action $a \in A^e$ to identify the value of:*

$$ID(\boldsymbol{p}, o)$$

At execution time, after performing a to identify p, the robot collects data in the form of (f_a, p). After each identification task, the robot receives $\boldsymbol{v}$, the values of attributes $\boldsymbol{p}$. The objective is to minimize the discounted cumulative action cost and maximize the success rate of attribute identification.

Remarks: Rational RAI agents treat individual attribute identification tasks independently, whereas rational On-RAL agents learn from the data collected in early tasks, trading off early-phase performance for long-term performance. Although existing RAI methods used different action policies, they all require an Off-RAL algorithm (for computing Ψ) running as a *preprocessing* step [2, 33, 41]. For instance, Thomason et al. [41] used a random strategy, and Amiri et al. [2] used a planning under uncertainty approach.

B. Dynamically Constructed POMDPs

Aiming at computing an action policy toward solving On-RAL problems, we dynamically construct partially observable MDPs (POMDPs) [13]. POMDPs can be defined in the form of $(S,\ A,\ T,\ R,\ Z,\ O,\ \gamma)$, where S, A, and Z are the sets of states, actions, and observations respectively; T, R, and O are functions of transition, reward, and observation respectively. γ is a discount factor.

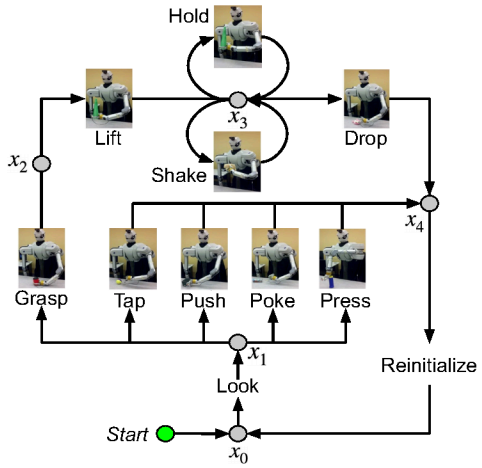


Fig. 3: Transition diagram among the $\mathcal{X}$ states led by exploratory actions A^e.

The state space S of the POMDP is a Cartesian product of two components, $\mathcal{X}$ and $\mathcal{Y}$:

$$S = (x, y)|x \in \mathcal{X}, y \in \mathcal{Y}$$

where $\mathcal{X}$ is the state set specified by fully observable domain variables. In our case, $\mathcal{X}$ includes a set of five states $x_0, ..., x_4$, as shown in Figure 3, and a terminal state $term \in \mathcal{X}$ that identifies the end of an episode. The states in $\mathcal{X}$ are fully observable, meaning that the robot knows the current state of the robot-object system, e.g., whether $grasp$ and $drop$ are successful or not. $\mathcal{Y}$ is the state set specified by partially observable domain variables (N queried attributes), $p_0, p_1, ..., p_{N-1}$. Thus, $|\mathcal{Y}| = 2^N$. For instance, given an object description that includes three attributes (e.g., a "red" "empty" "container"), $\mathcal{Y}$ includes 2^3 states.

Let $A : A^e \cup A^r$ be the action set that is available to the robot. A^e includes the object *exploration* actions (Figure 3), and A^r includes a set of *reporting* actions that deterministically lead the state transition to *term* and are used for attribute identification. Continuing the "red empty container" example, there are three binary variables and $|A^r| = 8$, each A^r corresponding to $y \in \mathcal{Y}$.

Let $T_{\mathcal{X}} : \mathcal{X} \times A \times \mathcal{X} \rightarrow [0, 1]$ be the state transition function in the fully observable component of the current state. Most exploration actions may be unreliable to some degree. For instance, $p(x_3,\ drop,\ x_4) = 0.95$ in our case, indicating there is a small probability the object is stuck in the robot's hand. $T_{\mathcal{Y}} : \mathcal{Y} \times A \times \mathcal{Y} \rightarrow [0, 1]$ is an identity matrix, because object attributes do not change over time.

Let Z be a set of observations and let the observation function $O(s, a, z)$ specify the probability of observing $z \in Z$ after taking action a in state s. We calculate the probability using confusion matrix $\Theta^a_p \in R^{2\times 2}$:

$$O(s, a, z) = \mathbf{Pr}(\mathbf{p}^z|\mathbf{p}^s, a) = \prod_{i=0}^{N-1} \Theta^a_{p_i}(p^s_i, p^z_i) \quad (4)$$

where $\Theta^a_p \in R^{2\times 2}$ is a confusion matrix for attribute p and action a; $\mathbf{p}^s$ and $\mathbf{p}^z$ are the vectors of true and observed values (0 or 1) of the attributes; p^s_i (or p^z_i) is the true (or observed) value of the i^{th} attribute; and N is the total number of attributes in the query. The transition system and the computation of $\Theta^a_p \in R^{2\times 2}$ are adapted from those of [2]. To alleviate the scalability issue of POMDPs, we use a strategy of dynamically constructing minimal POMDPs to model only those attributes necessary to the current query [44].

C. Algorithm Description

We first introduce our information-theoretic reward function and then present a novel algorithm for On-RAL problems.

In On-RAL problems, the robot needs to optimize its behaviors toward not only improving the accuracy of attribute identification but also minimizing the cost of exploratory actions. We introduce the two factors of *perception quality* and *interaction experience* into the reward design of POMDPs to achieve the trade-off between exploration (actively collecting data for attribute learning) and exploitation (using the learned attributes for identification tasks).

Let $Ent(s, a)$ be the entropy of the distribution over Z, given s and a, and is used for indicating the **perception quality** of exploratory action a over the y component of s:

$$Ent(s, a) = -\sum_{z_i \in Z} O(z_i|s, a) \log_2 O(z_i|s, a) \quad (5)$$

where z_i is the i^{th} observation and $O(z_i|s, a)$ is the probability of observing z_i in state s after taking action a. $O(z_i|s, a)$ is computed using the data instances gathered in the On-RAL process – see Definition 3.

Let $IE(p, a)$ be the **interaction experience** of applying action a to identify attribute p, and is in the form of:

$$IE(p, a) = \frac{1}{\delta} \cdot |\{f \in \mathbf{f}^m_a, labeled(f, p) = true\}| \quad (6)$$

where $\mathbf{f}^m_a$ is a set of instances in context c^m_a that a robot has collected so far, and $labeled(f, p)$ returns *true* if f has been labeled w.r.t. p, where the value v^p is *true* or *false*. δ is a sufficiently large integer to ensure $IE(p, a)$ is in the range of [0,1). A lower value of $IE(p, a)$ reflects a higher need of further exploring (p, a).

Information-theoretic Reward: We use $R^{real}(s, a, s')$ to refer to the standard reward function that rewards (or penalizes) successful (or unsuccessful) attribute identifications [2]. Building on the concepts of perception quality and interaction experience, our information-theoretic reward function is defined as follows:

$$R(s, a, s') = R^{real}(s, a, s') + \alpha \cdot Ent(s, a) - \beta \cdot IE(p, a) \quad (7)$$

where α and β are natural numbers and used for adjusting how much perception quality and interaction experience are considered in reward shaping. Informally, when $O(z|s, a)$ is close to being uniform, the perception model of (s, a) is poor, and the value of $Ent(s, a)$ is high. As a result, our new reward function will encourage the robot to take action a by offering extra reward $\alpha \cdot Ent(s, a)$. When the robot is

experienced in applying a to identify attribute p, $IE(p,a)$ will be high. In this case, an extra penalty of $\beta \cdot IE(p,a)$ will discourage the robot from taking those well-explored actions. In comparison to standard POMDPs, where reward and observation functions are independently developed, ITRS enables the reward function to adapt to the changes of the observation function.

ITRS Algorithm: Algorithm 1 presents ITRS for active On-RAL problems. The inputs of ITRS include attribute set $\mathcal{P}$, transition function $T_{\mathcal{X}}$, action set A^e, POMDP solver Sol, parameters α and β, naive reward function $R^{real}(s,a,s')$, and dataset $\mathcal{D}^{pre}$ for pretraining. ITRS does not have a termination condition.

ITRS starts with initializing the interaction experience function with zeros for all (p,a) pairs, and then initializes dataset $\mathcal{D}$ that will be later augmented as the robot interacts with objects (Lines 1 and 2). In each iteration of the main loop (Lines 3-24), ITRS takes an identification query from people (Line 4), constructs a POMDP model (Lines 5-10), computes its policy, uses the policy to interact with an object (Lines 13-18), and augments its dataset for improving the POMDP model in the next iteration (Lines 20-23).

In the first inner loop (Lines 13-18), the robot interacts with an object based on the generated policy. π suggests an action at each state b. The robot then executes the action and makes an observation. Based on the action and observation, the robot updates its belief using the Bayesian rule. After selecting each action, ITRS records this action along with its collected feature data (Lines 14 and 15). In Line 19, we ask people to provide the label y for the collected data of $\{f^{c_0}, f^{c_1}, \cdots\}$. The final step is to iterate over all selected actions to update function Λ, augment $\mathcal{D}$, and calculate the new interaction experience (Lines 20-23).

Intuitively, we aim to encourage the robot to select exploratory action $a \in A^e$ from those actions, where the perception model of (s,a) is of poor quality, and there is relatively limited experience of applying a to attribute p, i.e., the experience of (p,a) is limited.

Algorithm 1 ITRS algorithm

Require: $\mathcal{P}$; $T_{\mathcal{X}}$; A^e; Sol; α; β; $R^{real}(s,a,s')$; $\mathcal{D}^{pre}$
1: Initialize $IE(p,a) = 0$ for each action $a \in A$ and $p \in \mathcal{P}$
2: Let online training dataset $\mathcal{D} = \mathcal{D}^{pre}$
3: **repeat**
4: Take queried attribute(s) $\mathbf{p}$ from human, where $\mathbf{p} \subseteq \mathcal{P}$
5: Generate S, $T_{\mathcal{Y}}$ and Z using $\mathbf{p}$
6: Compute confusion matrix Θ_p^a using $\mathcal{D}$ where $p \in \mathbf{p}$, $a \in A$
7: Generate $O(z|s,a)$ with Θ_p^a for $p \in \mathbf{p}$ using Eqn. (4)
8: Compute $Ent(s,a)$ using Eqn. (5)
9: Generate $R(s,a,s')$ with $R^{real}(s,a,s')$ using Eqn. (7)
10: Compute policy π using Sol for $(S,\ A,\ T,\ R,\ Z,\ O,\ \gamma)$
11: Initialize action set A^{select} and feature set $\mathcal{F}$ with $\emptyset$
12: Uniformly initialize belief b
13: **while** Current state s is not $term$ **do**
14: Select action a with π based on b, append a to A^{select}, and execute a
15: Record data instances f_a, and $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_a\}$
16: Make an observation z where $z \in Z$
17: Update b with z and a using Bayesian rule
18: **end while**
19: Ask human to provide v^p for each $p \in \mathbf{p}$ as label(s) for $\mathcal{F}$
20: **for** each a in A^{select} **do**
21: Update $\mathcal{D}$ using $\mathcal{F}$ and v^p for each $p \in \mathbf{p}$
22: Update $IE(p,a)$ for $a \in A$ and $p \in \mathbf{p}$ using Eqn. (6)
23: **end for**
24: **until** end of interactions

IV. Experimental Evaluation

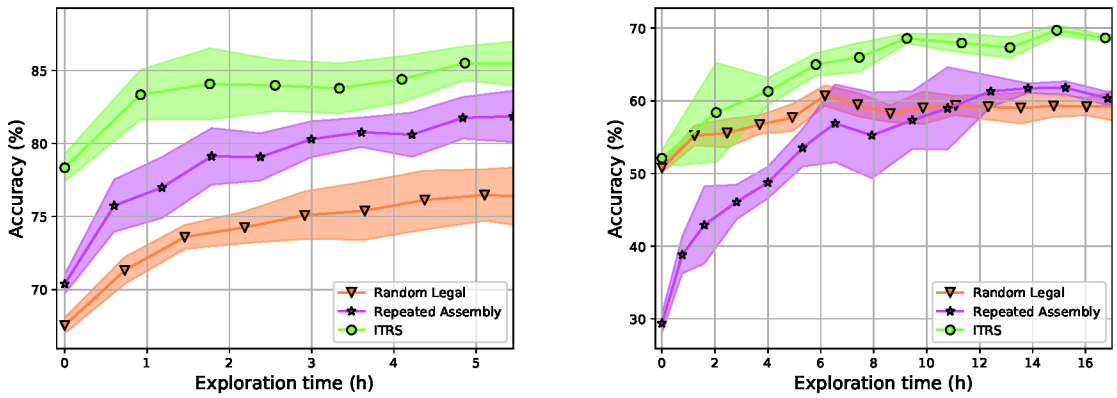
The key hypothesis is that ITRS outperforms existing RAL algorithms in learning efficiency and task completion accuracy (there does not exist an On-RAL algorithm in the literature). The first baseline is referred to as "**Random Legal**," which corresponds to the RAL approach of [41] (we did not use their linguistic component). It first used a supervised learning approach to compute the perception models of all attribute-action pairs (i.e., solving an Off-RAL problem) [32]. The robot considers only the "legal" actions (e.g., $lift$ is legal only after a successful $grasp$ action), and then randomly selects one from the legal actions. With an exploration budget for each trial (50 seconds and 80 seconds for one-attribute and two-attribute trials respectively), the robot is forced to report $y \in \mathcal{Y}$ of the highest belief.

The second baseline is referred to as "**Repeated Assembly**," which first solves an Off-RAL problem to compute Ψ using all data collected so far, and then solves an RAI problem to compute policy π for object exploration. This process is repeated after each batch. Repeated Assembly enables On-RAL behaviors by repeatedly assembling Off-RAL and RAI methods. Each iteration of Repeated Assembly corresponds to the work of [2].

A. Experiment Setup

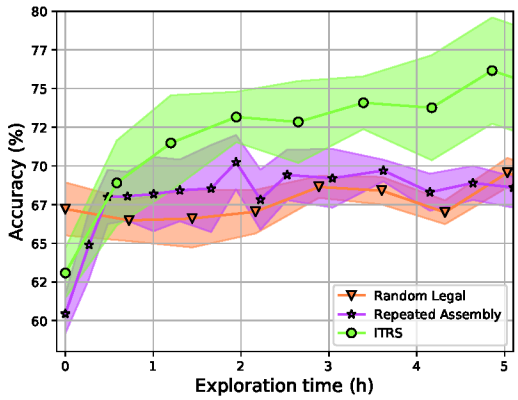
Dataset Description: Two public datasets of **CY101** [36] and **ISPY32** [39] are used in our experiments where **CY101** (an updated version of dataset [31]) contains many more household objects and attributes. In the **CY101** dataset, an uppertorso humanoid robot with 7-DOF arm explored 101 objects (Figure 2) belonging to 20 different categories using 10 exploratory behaviors: $look$, $grasp$, $lift$, $hold$, $shake$, $drop$, $push$, tap, $poke$, and $press$ (Figure 3). For **ISPY32**, a robot from the Building-Wide Intelligence project [16] explored 32 objects using 8 exploratory behaviors: $look$, $grasp$, $lift$, $hold$, $lower$, $drop$, $push$, and $press$. Each behavior was performed 5 times on each object in both datasets.

In order to select attributes that are learnable given the robot's exploratory behaviors, evaluations of all attributes in the two datasets were performed prior to the experiments. We set $|\mathcal{P}|$ to 10 and picked the attributes that are most learnable and that have enough positive and negative examples for training. Seven different types of features including auditory, vibrotactile, finger, color, optical flow, SURF, and haptics (i.e., joint forces) were considered in **CY101**; and features of VGG, color, SURF, auditory, finger, and haptics were recorded in **ISPY32**. For the $look$ behavior in dataset **ISPY32**, color (512-dimensional color histogram in RGB space), SURF features

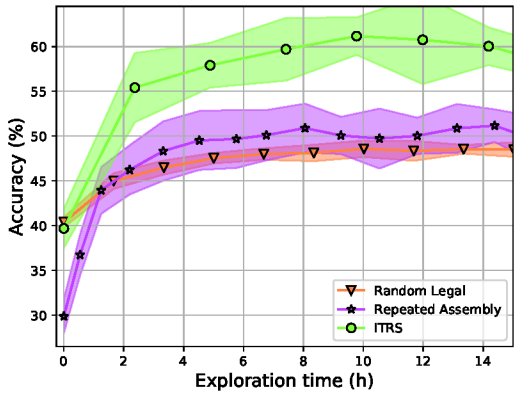


(a) One-attribute identification on CY101 dataset.

(b) Two-attribute identification on CY101 dataset.



(c) One-attribute identification on ISPY32 dataset.



(d) Two-attribute identification on ISPY32 dataset.

Fig. 4: Time length of conducting exploratory actions in hours, and identification accuracy of On-RAL tasks, where we compared ITRS (ours) to two baseline strategies including *Random Legal*, and *Repeated Assembly*.

were extracted from the image of the object, and VGG was also extracted. For the other behaviors, audio, vibrotactile, flow, SURF, and haptic features were recorded via the interaction with the object. Additionally, for the $grasp$ behavior, finger position features were recorded in the meantime.

In both datasets, we randomly split the objects into three subsets of equal sizes. The subsets are used for pretraining (Obj^{pre}), training (Obj^{train}), and testing (Obj^{test}) respectively. In the pretraining phase, the robot started with a handcrafted policy where each behavior is forced to be applied on the queried object once. We collected feature instances with labels from those interactions and built a pretraining dataset $\mathcal{D}^{pre}$ that represents the robot's prior knowledge.

Action Costs and Action Failures: Each exploratory action a has a cost (planning and execution) in the range of [0.5, 22.0] that came with the datasets, and is modeled in $R^{real}(s, a, s')$. For instance, the cost of action $press$ (22.0) is much higher than the cost of action $look$ (0.5). Additionally, the reward of the reporting action was +300.0 (or -300.0) when the robot's attribute identification is correct (or incorrect). Most actions are considered unreliable to some degree in our POMDP model and we uniformly set the failure probability to 0.05 which we did not refine in the On-RAL process. For instance, an unsuccessful $drop$ action models the situation that the object is stuck in the robot's hand. We used an off-the-shelf system for solving POMDPs [19].

Queries: At the beginning of each trial, an N-attribute identification task is assigned, i.e., $|\mathbf{p}| = N$. In our case, N was either 1 or 2. At the end of each trial, the robot is told if the identification was correct. In the case of $N = 1$, the robot could learn the attribute's ground-truth value from the human's feedback. In the case of $N = 2$, the robot could do so, only if the 2D identification was correct.

Batch-based Learning: Each batch contained 40 trials in one-attribute identification tasks, and 50 trials in two-attribute tasks, where we used objects from Obj^{train}. After each batch, we recomputed a POMDP model, including the shaped reward function, and computed an action policy for the next batch. The generated POMDP policy from each batch was then evaluated using objects from Obj^{test} in 1000 trials.

B. Illustrative Trial

From the robot's many trials of the learning experience, we selected two trials (T_1 and T_2), where the robot faced the same object (a Coke can that has attributes "metal," "empty," and "container") and needed to answer the same question "*Is this object soft?*" From the dataset, we know that the correct answer should be "no" (the robot did not know it). T_1 appeared

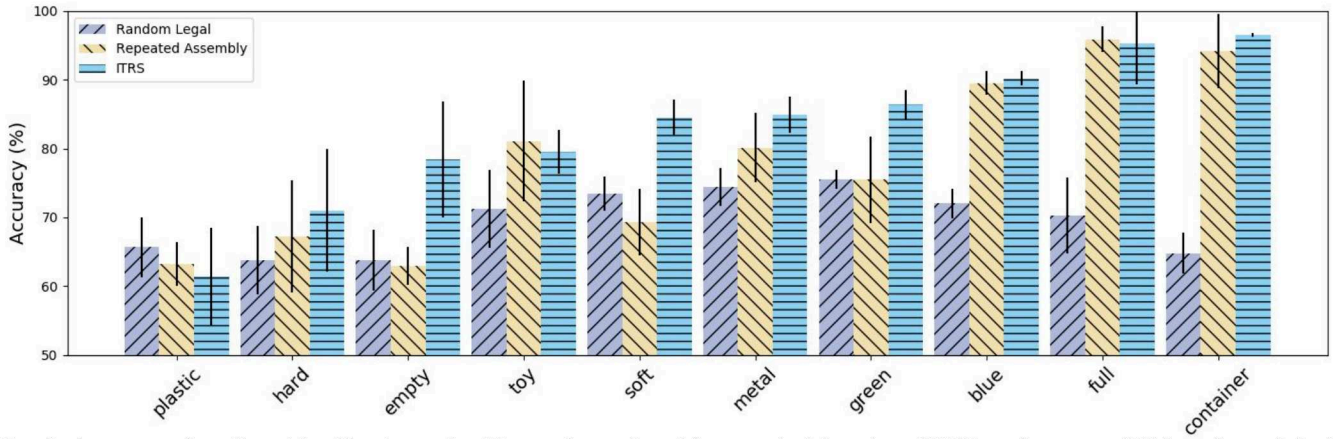


Fig. 5: Accuracy of attribute identification tasks. The attributes (x-axis) are ranked based on ITRS' performance. ITRS performed the best on seven out of the ten attributes.

TABLE I: Early and late observation models for action $press$

	Early phase		Late phase	
Not soft (Ground Truth)	0.68	0.32	0.82	0.17
Soft (Ground Truth)	0.50	0.50	0.20	0.80
	Not soft (Observed)	Soft (Observed)	Not soft (Observed)	Soft (Observed)

at the second batch of training, and T_2 appeared at the ninth. We present both trials and explain how the robot performed better in T_2.

In T_1 (early learning phase), the robot first performed the $look$ action. Then, the robot had the following options: $grasp$, tap, $push$, $poke$ and $press$ according to Figure 3. Specifically, for $press$, the confusion matrix Θ_{soft}^{press} (shown in Table I) was nearly uniform, which is typical in the early learning phase. Among those "less useful" actions, the robot chose $grasp$. The distribution over $\mathcal{Y}$ was changed from [0.37, 0.63] to [0.46, 0.54], where the entries represent "not soft" and "soft" respectively. After $press$, ITRS sequentially suggested $grasp$, $lift$, $hold$ and $hold$. Finally, the robot reported "positive" that resulted in a failed trial with a total cost of 55.5 seconds.

In T_2 (late learning phase), Action $press$ became more useful for identifying attribute "soft" compared to T_1, as shown in Table I. For $grasp$, $IE(soft, grasp) = 0.67$ and Θ_{soft}^{grasp} was [0.66, 0,33, 0.61, 0.38] (TN, FN, FP, TP), which meant that the robot was experienced with action $grasp$ and considered $grasp$ was not as useful as $press$. Accordingly, ITRS suggested $press$ instead of $grasp$ after taking $look$. The belief over $\mathcal{Y}$ changed from [0.57, 0.43] to [0.67, 0.33]. After only $look$ and $press$, the robot was able to quickly report "negative", resulting in a successful trial with a total cost of 22.5 seconds.

From the above two trials (same query and object in different learning phases), we see how the improved perception model of $(press, soft)$ helped the robot correctly identify "soft" with a low cost.

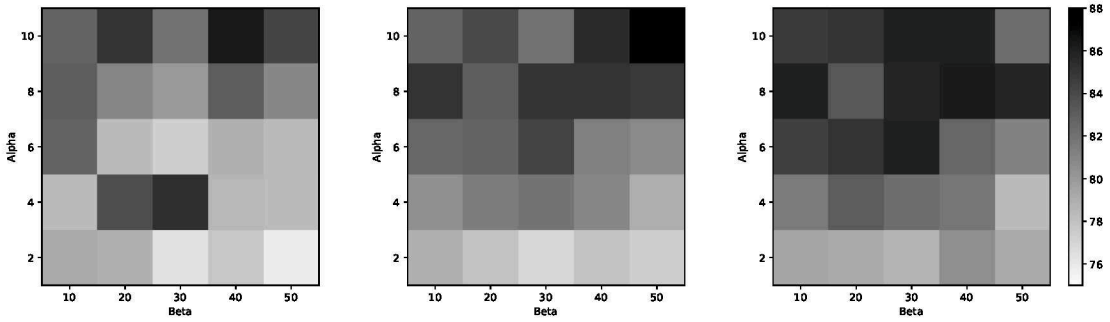
C. Experimental Results

Exploration Cost and Success Rate: Figure 4 shows the learning curve for one-attribute and two-attribute identification query evaluated on **CY101** and **ISPY32**, where we conducted experiments over three different strategies (two baselines and ITRS). x-axis is the accumulative cost over all trials at the training phase. Since the cost is determined by the time required to complete each action, we can regard the x-axis as training time. y-axis reflects the identification accuracy at the testing phase. The proposed method consistently performs better in task-completion accuracy along the whole training process and achieves higher accuracy than baselines.

Although we provided the same pretraining data, three curves in the two subfigures (for each of the two datasets) do not start from the same point. That is because pretraining data only affects the observation model for the robot, but it is not directly related to the policy for attribute identification. Three strategies have the same observation model but they use different methods to select exploratory behaviors. As a result, the task-completion accuracy is not the same for them at the starting point. ITRS assigns extra rewards for exploration at the very beginning of the training phase. It resulted in not only a bigger cost but also a higher accuracy.

Individual Attributes: At an exploration cost budget of 2 hours, we further evaluated the performance of each individual attribute on **CY101** using the three strategies we mentioned, as shown in Figure 5, where 10 attributes are ranked by the identification accuracy of our method, i.e. ITRS. The robot has a higher identification accuracy for most of the attributes using ITRS, while the random legal baseline produces a relatively weak result compared to the other two strategies. Attributes such as "plastic", "hard", and "empty" are more difficult to learn since the accuracy is no more than 80% for all three methods. And attributes such as "blue", "full" and "container" are easier, where repeated assembly and ITRS both offer pretty good results.

Parameters: In Eqn. (7), we have two parameters α and β. We conducted experiments on **CY101** with different α and β combinations, as shown in Figure 6. Results show that overall, a small α value leads to a higher identification accuracy in the beginning, but the accuracy does not improve much when it reaches the middle or late learning phase. A lower β



(a) Learning phase: early (b) Learning phase: middle (c) Learning phase: late

Fig. 6: We empirically evaluated the identification accuracies of ITRS using different values of α and β, which are two parameters of our reward shaping approach (Eqn. (7)). One observation is that the overall accuracy becomes higher in the middle and late learning phases, which is expected and verifies the robustness of ITRS to α and β selections. Another observation is that the selections of α and β affect the performance of ITRS. We leave the auto-learning of the parameters to future work.

TABLE II: Actions, observations and belief updates in the demonstration trial.

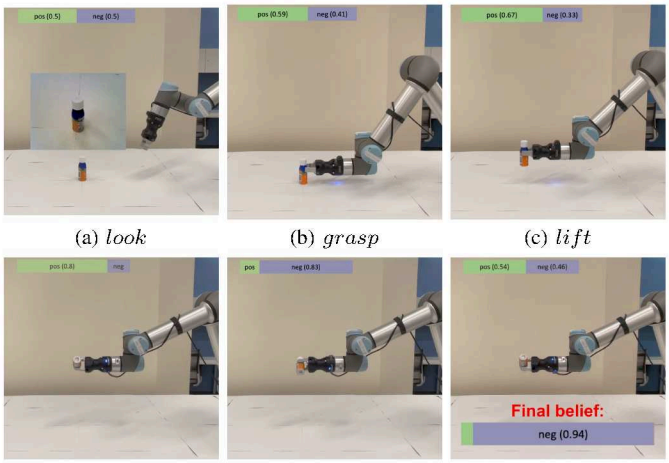
Step	Action	Observation	Belief (Initial belief: [0.5, 0.5])
1	Look	Positive	[0.41, 0.59]
2	Grasp	Positive	[0.33, 0.67]
3	Lift	Positive	[0.20, 0.80]
4	Shake	Negative	[0.83, 0.17]
5	Shake	Positive	[0.46, 0.54]
6	Shake	Negative	[0.94, 0.06]

value encourages the robot to explore no matter whether it is experienced or not, while a higher β value affects the robot to compute the optimal policy. Thus, when β is within the middle range, the robot has the best identification performance.

D. Real Robot Demonstration

We have demonstrated the learned action policy using a real robot (UR5e arm from Universal Robots). It should be noted that the two datasets we used in this research were collected on robots that are different from the robot in the demonstration. It is a major challenge in robotics of transferring skills learned from one robot to another. To alleviate the effect caused by the heterogeneity of robot platforms, after performing each action, we sampled a data instance from **CY101** according to $x \in \mathcal{X}$, the fully observable component of the current state.

In the demonstration trial, our robot was given an object – a pill bottle half-full of beans. The one-attribute query was "*Is this object empty?*" The robot performed a sequence of exploratory actions, as shown in Table II, where we also listed the observation and the belief after each action. For instance, a "Positive" observation means that the robot perceives that the object is "empty." Figure 7 shows a sequence of screenshots of the UR5e robot completing the task using a learned ITRS policy.



(a) *look* (b) *grasp* (c) *lift*

(d) *shake* (e) *shake* (f) *shake*

Fig. 7: A demonstration of the learned action policy using ITRS algorithm. The robot performed six actions in a row. At the beginning, the robot started with a uniform distribution (it evenly believed the object can be empty or not). After completing the six actions, the belief converged to "negative" (0.94 probability). Finally the robot selected a reporting action to report that the object is "*not empty*."

V. Conclusions and Future Work

In this work, we focus on a new On-RAL problem where the robot is required to complete attribute identification tasks and, at the same time, learn its observation model for each attribute. We propose an algorithm called ITRS that selects exploratory actions toward simultaneous attribute learning and attribute identification. The proposed method and baseline methods are evaluated using two real-world datasets. Experimental results show that ITRS enables the robot to complete attribute identification tasks at a higher accuracy using the same amount of training time compared to baselines.

One limitation of our existing framework is that the attribute recognition models are learned by a single robot and cannot directly be used by another robot that has different behaviors, morphology, and sensory modalities. We plan to use sensorimotor transfer learning (e.g., [37, 38]) to scale up our framework to allow multiple different robots to learn such models and share their knowledge as to further speed up learning. In addition, considering correlations between

attributes and handling fuzzy attributes can potentially improve the performance of On-RAL. Another direction for the future is to learn the world dynamics through the task completion process (currently the transition function is provided and the observation function is learned), where we can potentially use reinforcement learning methods. Finally, we plan to evaluate the learned policy using a robot platform that is different from the robot used for data collection, and with real human-robot dialogue to acquire attribute labels for objects.

ACKNOWLEDGMENTS

The authors thank Sinem Ozden, Yan Ding, Kishan Chandan, Kexin Li, and anonymous reviewers for helpful comments and discussions. A portion of this research has taken place at the Autonomous Intelligent Robotics (AIR) Group, SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (NRI-1925044), Ford Motor Company (URP Awards 2019-2021), OPPO (Faculty Research Award 2020), and SUNY Research Foundation.

REFERENCES

[1] Aitor Aldoma, Federico Tombari, and Markus Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In *2012 IEEE International Conference on Robotics and Automation*, pages 1732–1739. IEEE, 2012.

[2] Saeid Amiri, Suhua Wei, Shiqi Zhang, Jivko Sinapov, Jesse Thomason, and Peter Stone. Multi-modal predicate identification using dynamically learned robot controllers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018.

[3] Jacob Arkin, Daehyung Park, Subhro Roy, Matthew R Walter, Nicholas Roy, Thomas M Howard, and Rohan Paul. Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions. *The International Journal of Robotics Research*, 39(10-11):1279–1304, 2020.

[4] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[5] Raphaël Braud, Alexandros Giagkos, Patricia Shaw, Mark Lee, and Qiang Shen. Robot multi-modal object perception and recognition: synthetic maturation of sensorimotor learning in embodied systems. *IEEE Trans. on Cognitive and Developmental Systems*, 2020.

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Robert Eidenberger and Josef Scharinger. Active perception and scene modeling by planning with probabilistic 6d object poses. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1036–1043. IEEE, 2010.

[9] Jeremy A Fishel and Gerald E Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in neurorobotics*, 6:4, 2012.

[10] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. Swoosh! Rattle! Thump! - Actions that Sound. In *Proceedings of Robotics: Science and Systems*, 2020.

[11] Yang Gao, Lisa Anne Hendricks, Katherine J Kuchenbecker, and Trevor Darrell. Deep learning for tactile understanding from visual and haptic data. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 536–543. IEEE, 2016.

[12] Eleanor J Gibson. Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual review of psychology*, 39(1):1–42, 1988.

[13] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[14] Michael N Katehakis and Arthur F Veinott Jr. The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research*, 12(2):262–268, 1987.

[15] Matthias Kerzel, Erik Strahl, Connor Gaede, Emil Gasanov, and Stefan Wermter. Neuro-robotic haptic object classification by active exploration on a novel dataset. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[16] Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.

[17] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *J. of Artificial Intelligence Research*, 61:215–289, 2018.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[19] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Citeseer, 2008.

[20] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations

for contact-rich tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.

[21] Qiang Li, Oliver Kroemer, Zhe Su, Filipe Fernandes Veiga, Mohsen Kaboli, and Helge Joachim Ritter. A review of tactile information: Perception and action through touch. *IEEE Transactions on Robotics*, 36(6): 1619–1634, 2020.

[22] Dermot Lynott and Louise Connell. Modality exclusivity norms for 423 object properties. *Behavior Research Methods*, 41(2):558–564, 2009.

[23] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[24] Francisco Pastor, Jorge García-González, Juan M Gandarias, Daniel Medina, Pau Closas, Alfonso J García-Cerezo, and Jesús M Gómez-de Gabriel. Bayesian and neural inference on LSTM-based object recognition from tactile and kinesthetic information. *IEEE Robotics and Automation Letters*, 6(1):231–238, 2020.

[25] Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[26] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[29] Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *J. of Machine Learning Research*, 12(5), 2011.

[30] Amrita Sawhney, Steven Lee, Kevin Zhang, Manuela Veloso, and Oliver Kroemer. Playing with food: Learning food item representations through interactive exploration. In *Proceedings of 17th International Symposium on Experimental Robotics (ISER '20)*. Springer Proceedings in Advanced Robotics (SPAR), November 2021.

[31] Jivko Sinapov, Connor Schenck, Kerrick Staley, Vladimir Sukhoy, and Alexander Stoytchev. Grounding semantic categories in behavioral interactions: Experiments with 100 objects. *Robotics and Autonomous Systems*, 62(5): 632–645, 2014.

[32] Jivko Sinapov, Connor Schenck, and Alexander Stoytchev. Learning relational object categories using behavioral exploration and multimodal perception. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5691–5698. IEEE, 2014.

[33] Jivko Sinapov, Priyanka Khante, Maxwell Svetlik, and Peter Stone. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *IJCAI*, pages 3462–3468, 2016.

[34] Mohan Sridharan, Jeremy Wyatt, and Richard Dearden. Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs. *Artificial Intelligence*, 174(11):704–725, 2010.

[35] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[36] Gyan Tatiya and Jivko Sinapov. Deep multi-sensory object category recognition using interactive behavioral exploration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7872–7878. IEEE, 2019.

[37] Gyan Tatiya, Ramtin Hosseini, Michael C. Hughes, and Jivko Sinapov. A framework for sensorimotor cross-perception and cross-behavior knowledge transfer for object categorization. *Frontiers in Robotics and AI*, 7: 137, 2020.

[38] Gyan Tatiya, Yash Shukla, Michael Edegware, and Jivko Sinapov. Haptic knowledge transfer between heterogeneous robots using kernel manifold alignment. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2020.

[39] Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J Mooney. Learning multi-modal grounded linguistic semantics by playing "I Spy". In *IJCAI*, pages 3477–3483, 2016.

[40] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J Mooney. Opportunistic active learning for grounding natural language descriptions. In *Conference on Robot Learning*, pages 67–76. PMLR, 2017.

[41] Jesse Thomason, Jivko Sinapov, Raymond Mooney, and Peter Stone. Guiding exploratory behaviors for multi-modal grounding of linguistic descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[42] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. SwingBot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 5633–5640, 2020.

[43] Shiqi Zhang, Mohan Sridharan, and Christian Washington. Active visual planning for mobile robot teams using hierarchical POMDPs. *IEEE Transactions on Robotics*, 29(4):975–985, 2013.

[44] Shiqi Zhang, Piyush Khandelwal, and Peter Stone. Dynamically constructed (PO) MDPs for adaptive robot planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[45] Kaiyu Zheng, Yoonchang Sung, George Konidaris, and Stefanie Tellex. Multi-resolution POMDP planning for multi-object search in 3d. *arXiv preprint arXiv:2005.02878*, 2020.