

Learning Object Affordances by Leveraging the Combination of Human-Guidance and Self-Exploration

Vivian Chu, Tesca Fitzgerald, and Andrea L. Thomaz

School of Interactive Computing

Georgia Institute of Technology

Atlanta, Georgia 30332-0250

Email: {vchu,tesca.fitzgerald}@gatech.edu, athomaz@cc.gatech.edu

Abstract—Our work focuses on robots to be deployed in human environments. These robots, which will need specialized object manipulation skills, should leverage end-users to efficiently learn the affordances of objects in their environment. This approach is promising because people naturally focus on showing salient aspects of the objects [1]. We replicate prior results and build on them to create a combination of self and supervised learning. We present experimental results with a robot learning 5 affordances on 4 objects using 1219 interactions. We compare three conditions: (1) learning through self-exploration, (2) learning from supervised examples provided by 10 naïve users, and (3) self-exploration biased by the user input. Our results characterize the benefits of self and supervised affordance learning and show that a combined approach is the most efficient and successful.

I. INTRODUCTION

Robots deployed in unstructured human environments (e.g. homes and offices) will have to learn and model their specific environment quickly and with minimal effort by human end-users. Our work expedites affordance learning with robots in new environments through a novel hybrid self-guided/human-supervised exploration approach. The term “affordance” was first introduced by J.J. Gibson [2]. We use the ecological definition of “action possibilities” that appear between an agent and the environment. We represent affordances as the relationship between effects and a set of actions performed by an agent on an object (commonly used in robotics [3, 4]).

We compare three approaches to affordance learning: (1) the traditional **self-exploration** strategy where the robot exhaustively interacts with the workspace; (2) a **human-supervised exploration** strategy where a human provides example object interactions from which the robot learns; and (3) a combined **human-guided** approach that performs self-exploration biased by information provided from human teachers. We compare these three strategies by learning five affordances across four different objects and show that a human-guided approach can learn an affordance model that is as effective as exhaustive self-exploration with an order of magnitude fewer interactions.

II. RELATED WORK

Prior work in affordance learning for robotics focused on using primitive actions to interact with objects and learn



Fig. 1: Naïve user during user study teaching “Curi” the robot that the drawer has the open-able affordance

the mapping of effects to affordances. This established a framework for affordance learning using exploration [5]–[10]. These works required specific primitive actions to be learned or programmed, and did not use human input for guidance.

Many researchers are investigating how robots can explore the world. One relevant area of research is work on intrinsic motivation and curiosity-driven exploration. Early work [11]–[13] looked at using rewards and expectations to guide the exploration without any human supervision. The latest work on intrinsic exploration from [14, 15] and [16] combined intrinsic exploration with human input. Our work, by contrast, combines both human-supervision and self-exploration. Methods using intrinsic exploration assume the existence of an easily-characterized reward signal, even though such reward signals can be difficult to define for hard-to-find affordances.

Our work extends the previous work [1] in two key ways: we apply human-supervised affordance learning to more difficult-to-find affordances, and we investigate the combination of human-supervision and self-exploration.

III. AFFORDANCE LEARNING

To learn affordances, an agent interacts with the environment and observes the effects of that interaction. From this, the agent can learn what the environment *affords* for it. In our case, a robot (agent) performs a set of actions $A = \{a_1, \dots, a_N\}$ on a set of objects $O = \{o_1, \dots, o_M\}$ to model the effects that a_i can have on o_j , where $i = \{1, \dots, N\}$, $j = \{1, \dots, M\}$, and N

and M are the number of actions and objects respectively. We assume the effect of an object-action (o_j, a_i) pair is labeled as a positive or negative example of the affordance. Thus, it is a supervised learning problem and the resulting model can recognize the successful interactions of an object-action pair.

In the simplest case, if the agent’s actions are discrete, it could try all actions on all objects and model the outcomes. However, a better method is required to efficiently sample the infinitely large space of real-world actions the robot could perform to manipulate an object. For example, to open a drawer such as the one seen in Fig. 1, there are an infinite number of directions a robot could move the drawer in before discovering that it needs to pull the drawer towards itself in a horizontal line. To make the object exploration tractable, we provide the robot with a set of parameterized primitive actions. The exploration space is then defined by the continuous-valued parameters for each primitive action. In the instance of the previously mentioned drawer, we reduce the open action into three parameters (start, close, and end poses). Note that this still results in a sample space that is infinitely large. Thus, we present and compare five different strategies in Sections IV and V for efficiently sampling this space to collect the examples needed to build object-action affordance models.

A. Hardware Platform

For our experiments, we used the robot “Curi”, seen in Fig. 1. Curi has two 7 degree-of-freedom arms, each with an under-actuated 4 DOF hand. The arm can be controlled by physically moving it in a gravity compensated mode, used to kinesthetically teach the robot actions. We used the robot’s left arm for all experiments. An ATI Mini40 Force/Torque (F/T) sensor is mounted at each wrist, and an ASUS Xtion Pro RGB-D sensor is mounted above the workspace.

B. Objects and Actions

We selected four household objects (Fig. 2) for the robot to interact with. Each of these are tracked using the RGB-D sensor throughout the interaction, from which we record visual object information commonly used in affordance learning [1, 4] (in 3D space rather than 2D images). We record the color, orientation, volume of the bounding box, the dimensions of the bounding box (x, y, z), and the squareness of the object (the ratio of the number of points in the object to the area of the bounding box). We also store information from the 6-axis F/T sensor in the wrist ($F_x, F_y, F_z, T_x, T_y, T_z$) and the robot end-effector (EEF) position relative to the centroid of the object point cloud. This feature vector contains 18 values: 9 (visual), 6 (F/T), and 3 (EEF) and is how we represent the effect of object-actions pairs for the affordance learning problem.

The robot can perform two parameterized action primitives: **move** and **pick**. Each is a sequence of EEF poses relative to the centroid of the object point cloud. The EEF pose is the position and orientation of the robot hand for all 6 degrees-of-freedom (DOF). A **move** action has two EEF poses (start and end). The **pick** action has three EEF poses (start, where Curi closes its hand, and end). For both primitives, we generate a trajectory

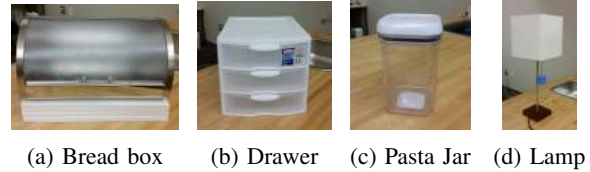


Fig. 2: Objects the robot explored

TABLE I: Affordances

Object	Action	Effect	Affordance
Breadbox	Move	Moves up	open-able
Pasta jar	Move	Moves	push-able
Drawer	Move	Moves	push-able
Drawer	Pick	Pulls out	open-able
Lamp	Pick	Pulls down	turn-on-able

for the EEF by performing a quintic spline between the EEF poses with an average velocity of 1 cm/second. While all poses are needed to define the primitive action, this paper will only modify the parameters of the final pose for each primitive action due to the sheer number of object interactions needed to explore the continuous-valued parameters of *all* poses in a primitive action. This is a reasonable simplification since the start pose can be initialized by putting the EEF near the object, as is common in existing affordance work [5, 8]. For both primitive actions in this work, the final pose has the largest impact on successful execution (e.g. the final pose is key in making the move action succeed in pushing an object).

C. Affordances

The five specific object-action affordances used in this paper are summarized in Table I and seen in Figure 2. These affordances represent a range of difficulty: *simple* affordances that can be found in a large part of the action space during exploration (e.g. push-able can be found in a various ways) while *complex* affordances require interacting with the object along a specific dimension of the action primitive space (e.g. open-able on the drawer requires the robot to pull the object towards itself in a particular way, representing a small subset of the object-action exploration space).

The question we ask in this work is how to best sample the space of our primitive actions’ continuous-valued parameters to interact with objects and collect effective examples for affordance modeling in a way that is *efficient*. We present two baseline approaches, Self-Exploration (SE) and Human-Supervised Exploration (HSE), and compare these to three strategies that represent a *combined* approach: Guided Aggregate Exploration (GAE), Guided Iconic Exploration (GIE), and Guided Boundary Exploration (GBE). All five of these are detailed in the following two sections.

IV. BASELINE EXPLORATION STRATEGIES

A. Self-Exploration (SE)

Typical self-exploration strategies in robot affordance learning [5, 6, 8] exhaustively sample the space of action parameters. These strategies know only that it should perform actions around the object and the main decisions needed to discretize

Algorithm 1 Self-Exploration (SE)

```
1: procedure COMPUTE PERMUTATION( $v = [v_0, v_1, \dots, v_n]$ )
2:    $P_{set} \leftarrow$  set of permutations of  $(x, y, z) \forall x, y, z \in v$ 
3:    $P_{set} \leftarrow P_{set} - \{(0, 0, 0)\}$ 
4:   return  $P_{set}$ 
5: procedure GENERATE EXPLORATION(expert dist.  $d$ )
6:    $\alpha \leftarrow d + 10\text{cm}$ 
7:    $D_1 \leftarrow$  ComputePermutation( $[-\alpha, 0, \alpha]$ )
8:    $D_2 \leftarrow$  ComputePermutation( $[-\alpha, -\frac{\alpha}{2}, 0, \frac{\alpha}{2}, \alpha]$ )
9:    $Unique \leftarrow (D_1 \cup D_2) - (D_1 \cap D_2)$ 
10:   $ExploreSet \leftarrow$  Random(Unique, 100)
11:  return  $ExploreSet$ 
```

the space of action parameters relate to (1) what range the robot should explore around the object and (2) the resolution (step-size) to use in sampling. We present our version of self-exploration, SE, below (Algorithm 1).

Representing the EEf requires three variables (x, y, z) to describe the position and three variables (rx, ry, rz) to describe the orientation in Euler space. However, in reality, it is infeasible for the robot to perform exploration in all six dimensions. For example, suppose we only vary the orientation of the EEf between -90° and 90° , with a step-size of 90° and the position between $-\alpha$ and α with a step-size of α . Assume α is a constant selected to guarantee the search covers some maximum distance needed for the EEf to have a chance at achieving the affordance in question. Even this coarse exploration of the action space results in 676 interactions per EEf pose per affordance and, realistically, a higher resolution search will most likely be needed to find the affordance.

To reduce this, we only sample the space of parameters of the final pose of each primitive action. An expert (one of the authors) provides a starting pose (position and orientation) for **move**, and a start and close pose for **pick**, and these are provided to be *ideal* for achieving the affordance.¹ With this assumption, SE exhaustively explores the position (x, y, z) of the final end pose for each primitive action. We fix the final pose orientation to be the same as the start pose. Even with these constraints, the number of explorations generated can still result in an impractical number of exploratory actions. We limit the number of samples to 100 actions per object.

To generate these 100 samples, we demonstrate one successful interaction with each object to calculate the maximum distance (α) the EEf must travel to achieve each affordance. To ensure this is a conservative estimate we extend the expert demonstrated distance, d , ($\alpha = d + 10\text{cm}$), resulting in the maximum distance the SE samples to create exploratory actions. Rather than provide more information to SE about the resolution to sample within these maximum bounds, we adaptively split the action parameter space in half until we reach the designated 100 samples. Thus, we start with a coarse

exploration of the space, and continue to sample at a higher resolution we reach 100 samples of the action space. First, we explore all possible permutations of the three dimensions (x, y, z) for the discrete values: $-\alpha$, 0, and α . This has 27 different permutations, but we remove the interaction where nothing changes $(0, 0, 0)$ for a total of 26 EEf poses to execute as exploratory actions on the object, which we call D_1 . To sample at a higher resolution, we split the step-size in half, resulting in five discrete values: $-\alpha$, $-\frac{\alpha}{2}$, 0, $\frac{\alpha}{2}$, and α and a total of 125 permutations. Again, we remove $(0, 0, 0)$ as well as any actions already included in D_1 , resulting in 98 new EEf poses, which we call D_2 . To limit each object-action pair to 100 samples, we randomly select 74 interactions from D_2 to add to the 26 interactions of D_1 . Together, D_1 and D_2 compose the exhaustive set of interaction samples for SE.

Note, to make SE tractable, we provided the start position and orientation of the EEf as well as the maximum distance (α) that the EEf has to explore to find the affordance.

B. Human-Supervised Exploration (HSE)

The next baseline approach uses a human teacher to fully supervise the collection of examples of object-action interactions. Through action demonstrations, the human teacher provides various ways to explore the object to result in successful or unsuccessful examples of the affordance. Our approach, HSE, builds on [1], but uses more complex affordances and generates actions in the full 6 DOF range of the robot EEf.

For HSE, we collect data from people in the campus community who had not interacted with our robot before. They used the same two action primitives (**move** and **pick**) that the robot uses during SE. Users teach a **move** action by moving the arm to a start pose and then an end pose, and a **pick** action by moving the arm to a start, grasp, and end pose. The robot creates an action trajectory in the same manner as SE, by splining between the action poses. The data used for affordance learning is collected when the robot autonomously executes this human-taught action on the given object. This allows the robot to record the visual and haptic sensory data without erroneously capturing noise from user contact.

We conducted a user study with 10 participants (5 male, 5 female) from a college campus. At the start of their session, participants were instructed briefly on the definition of affordances as well as how to verbally command and move the robot for kinesthetic teaching. For practice, they taught **pick** and **move** on two objects, and the data was not used. Once comfortable, we began their real data collection.

The participants taught the robot about the 5 affordances over the 4 objects described in Table I. For each object, they were told the specific action (**move** or **pick**) to use and the effect to show the robot. We instructed them to think about what strategy they might use if they were to teach a child about that specific affordance. A single example for affordance learning was collected each time the robot executes the taught action autonomously. To generate multiple affordance examples, participants could either move the object and repeat the previous action or they could teach a new action.

¹We believe it is a reasonable assumption to provide the start pose, because there exist state-of-the-art algorithms that find the best grasp/interaction points for a wide range of objects (e.g. handle of the breadbox or chain on the lamp).

TABLE II: Number Examples from Each Exploration Strategy

Object	Action	SE	HSE	GAE	GIE ^a	GBE ^a
Breadbox	Move	100	64	31	12	9
Pasta jar	Move	100	48	30	12	9
Drawer	Move	100	51	31	12	9
Drawer	Pick	96	41	31	12	9
Lamp	Pick	100	51	N/A	N/A	N/A

^a These are the number of examples generated for each user model, since these approaches operate on an individual user basis. N/A means there were not enough examples for that strategy

For the *complex* object-action pairs (i.e. breadbox-move, drawer-pick, and lamp-pick), participants were given 10 minutes to provide examples to the robot. For the *simple* pairs (pasta jar-move and drawer-move), they were given 5 minutes. The motivation for this difference was based on pilot studies. For simple affordances, users quickly developed strategies for teaching, whereas complex affordances required more time and trials for the user to develop a strategy to get the robot to perform the desired user action. The selected time constraints facilitate the collection of several interactions of each affordance and limit each study to within an hour, thus preventing user fatigue. To control for ordering effects in the data, we counter-balanced the order in which the five affordances were taught across users. At the end of the experiment, participants answered a single open-ended survey question that asked them about their teaching strategy. The total number of affordance examples collected across all 10 users can be seen in Table II.

V. GUIDED EXPLORATION STRATEGIES

While users provide key information and useful examples of affordances, it is cumbersome to have people provide an exhaustive set of examples for each affordance. During self-exploration, the robot can easily generate an exhaustive search, but has no real concept of where to focus that search. Combining the strengths of both approaches should yield the best of both worlds. Our primary research question is how to effectively bias SE with information from human teachers. In this section, we present three novel strategies that differ in how they integrate teacher input for exploration.

A. Guided Aggregate Exploration (GAE)

Our first approach, GAE (Algorithm 2), takes an aggregate view of the guidance that people provided from HSE. For each affordance, we build a new set of samples in the action space based on the mean and variance of the *final* EEF position of each first action shown by the ten people in our study. More concretely, let p^n be the final EEF pose from the first demonstration by user n . Now we define $P_{(j,i)}$ as the set of final EEF positions from all users' first demonstrations for an affordance pair (o_j, a_i) : $P_{(j,i)} = \{p^1 \dots p^n\}$ for $n = 1 \dots 10$. We compute the mean (μ_{ji}) and variance (σ_{ji}^2) of $P_{(j,i)}$, which represents an aggregate of the human provided input, and use them to generate new sample points in the action space. Note that each value contains three numbers (for each axis).

During SE, we sampled the final position of the EEF by adaptively splitting the action space about the starting position using an expert defined α . In GAE, we instead replace α with

Algorithm 2 Guided Aggregate Exploration (GAE)

```

1:  $\alpha \leftarrow$  expert demo dist. +10cm
2:  $p_{(j,i)} \leftarrow \{p^1 \dots p^n\}$  for  $n = 1 \dots 10$ 
3:  $\mu_{ji} = \text{mean}(p_{(j,i)})$ 
4:  $\sigma_{ji}^2 = \text{variance}(p_{(j,i)})$ 
5:  $\vec{r}_{change} \leftarrow \mu_{ji} - EEF_{start\ position}$ 
6:  $\vec{e}_{change} \leftarrow \frac{\vec{r}_{change}}{\|\vec{r}_{change}\|_2}$ 
7:
8: procedure GENERATEEXPLORATION
9:    $exploreRegions = [\mu_{ji}, \mu_{ji} + \sigma_{ji}^2, \mu_{ji} - \sigma_{ji}^2]$ 
10:   $ExploreSet \leftarrow \text{ComputePermutation}(exploreRegions)$ 
11:   $c \leftarrow 1$ 
12:   $\vec{p}_{change} \leftarrow (0, 0, 0)$ 
13:  while  $\|\vec{p}_{change}\| < \alpha$  do
14:     $\vec{p}_{change} = \vec{e}_{change} * c * \alpha$ 
15:     $ExploreSet \leftarrow ExploreSet \cup \{\vec{p}_{change}\}$ 
16:     $c++$ 
  return  $ExploreSet$ 

```

the computed σ_{ji}^2 and center the sampling of the final position of the EEF using μ_{ji} . This generates an action primitive that starts at the same position defined by the expert and ends using all permutations of the three dimensions (x, y, z) for the discrete values: $\mu_{ji} + \sigma_{ji}^2$, μ_{ji} , and $\mu_{ji} - \sigma_{ji}^2$. For each affordance, we have 27 sample locations and use the same EEF orientation used during SE. This strategy explores along the dimensions (x, y, z) of high variance, which are locations in the action space where the affordance can be discovered in a variety of positions. It also constrains the exploration in dimensions of low variance as these are important to finding the affordance.

Additionally, while collecting the SE interactions, we noticed that each affordance had a direction of change. For example, the open-able drawer affordance, requires moving the EEF perpendicular to the drawer towards itself and the open-able breadbox at an angle away from itself. To focus the exploration along this direction of change (\vec{e}_{change}), we do an additional sampling of the EEF action space along this dimension. The \vec{e}_{change} is actually the unit vector between the start (or close) and end positions of the EEF in the action primitive. We scale \vec{e}_{change} by different magnitudes and use the resulting vector as the position in the final EEF pose.

To compute \vec{e}_{change} , we subtract and normalize the expert selected starting position from μ_{ji} . For consistency, we use the same resolution from SE (α) as the base increments to the magnitude. Precisely, $\vec{e}_{change} = \frac{\vec{r}_{change}}{\|\vec{r}_{change}\|_2}$ where $\vec{r}_{change} = \mu_{ji} - EEF_{start\ position}$ and the final EEF position is $\vec{e}_{change} * c * \alpha$ where $c = \{1 \dots C\}$. C is the max number of times we can increase the magnitude by before we reach the max exploration distance allowed (set in SE: α). This results in 3 new interactions for pasta jar-move and 4 for all other affordances.

B. Guided Iconic Exploration (GIE)

Our next approach, GIE (Algorithm 3), uses each human teacher's input individually to bias the exploration of the action space rather than relying on the aggregate of several teachers.

Algorithm 3 Guided Iconic Exploration (GIE)

```

1:  $S \leftarrow$  EEF position of final pose in  $a^n(S)$ 
2:  $F \leftarrow$  EEF position of final pose in  $a^n(F)$ 
3:  $\vec{r}_{SF} \leftarrow (F - S)$ 
4: procedure GENERATEEXPLORATION
5:    $ExploreSet \leftarrow \emptyset$ 
6:   for  $p$  in  $[S, F]$  do
7:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x \pm \|\vec{r}_{SF}\|_2, p_y, p_z]\}$ 
8:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x, p_y \pm \|\vec{r}_{SF}\|_2, p_z]\}$ 
9:      $ExploreSet \leftarrow ExploreSet \cup \{[p_x, p_y, p_z \pm \|\vec{r}_{SF}\|_2]\}$ 
10:  return  $ExploreSet$ 

```

Specifically, we use only two samples (the first successful $a^n(S)$ and the first failed $a^n(F)$ interaction) from user n to generate a new set of samples. We select $a^n(S)$ and $a^n(F)$ because this provides crucial information on the location of the boundary between affordance success and failure in the action space. Furthermore, selecting $a^n(S)$ and $a^n(F)$ allows us to determine the viability of having a user provide two samples of the space and having the robot take over afterwards.

We define \vec{r}_{SF} to be the vector extending from S to F , where S is the position (3D) of the EEF in the final pose of $a^n(S)$, and F is the final position of the EEF in $a^n(F)$. The L_2 norm of \vec{r}_{SF} provides a crucial piece of information that, during SE, we had to get from an expert: the exploration resolution the robot should use to achieve the affordance. We can look for the iconic or prototypical examples of successful and failed interactions by adding and subtracting $\|\vec{r}_{SF}\|_2$ from the final pose of the EEF in the action primitive provided by the user in all dimensions (x, y, z). This results in 6 final EEF poses for $a^n(S)$ and 6 final EEF poses for $a^n(F)$ for a total of 12 final EEF poses. Each of the computed final EEF poses are used to generate primitive actions by replacing the final EEF pose of the primitive action provided by the user.

Note that all poses in the primitive action are generated from the user provided sample. Therefore, not only are we inferring the resolution of the search space with $\|\vec{r}_{SF}\|_2$, but we also no longer need an expert to define the start or close pose of the EEF primitive action. This is particularly important for instances where the a robot manipulator is not standard or easily modeled, or the object handle is not visually distinct.

C. Guided Boundary Exploration (GBE)

In GIE, we inferred the boundary between success and fail in the action space by concentrating the new action samples around $a^n(S)$ and $a^n(F)$. Now we introduce GBE (Algorithm 4), which explicitly samples along the boundary. This strategy also uses two action samples ($a^n(S)$ and $a^n(F)$) from each user, and S, F , and \vec{r}_{SF} are the same as before.

To generate the boundary between success and failure in the action space, we use the midpoint between S and F , and coarsely generate multiple vectors circling the midpoint. Specifically, we take $\frac{\vec{r}_{SF}}{2}$ and translate it to the position halfway between S and F . We rotate this new vector about each axis

Algorithm 4 Guided Boundary Exploration (GBE)

```

1:  $S \leftarrow$  EEF position of final pose in  $a^n(S)$ 
2:  $F \leftarrow$  EEF position of final pose in  $a^n(F)$ 
3:  $\vec{r}_{SF} \leftarrow (F - S)$ 
4: procedure GENERATEEXPLORATION
5:    $ExploreSet \leftarrow \emptyset$ 
6:   for  $\theta$  in  $[-\frac{\pi}{2}, \frac{\pi}{2}, \pi]$  do
7:      $ExploreSet \leftarrow ExploreSet \cup \{rotateX(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
8:      $ExploreSet \leftarrow ExploreSet \cup \{rotateY(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
9:      $ExploreSet \leftarrow ExploreSet \cup \{rotateZ(S + \frac{\vec{r}_{SF}}{2}, \theta)\}$ 
10:  return  $ExploreSet$ 

```

(x, y, z) for the angles $\frac{\pi}{2}$, $-\frac{\pi}{2}$, and π . We hypothesize that one of these vectors is the real boundary for the action space.

GBE generates 9 different final EEF poses in the action space (3 for each axis) that try to find the boundary between the successful and failed affordance interactions. Similar to GIE, we generate each sample by replacing the EEF position in the final EEF pose in $a^n(S)$. Note that since we are using the vector from S to F , we only use $a^n(S)$ and not $a^n(F)$. Just like GIE, we no longer need an expert for the start pose, close pose, or orientation of the actions primitives.

VI. AFFORDANCE MODELING

We used all five exploration strategies to select actions for the robot to execute to collect example object interactions for all 5 affordances. In total, the robot executed 1219 interactions with the environment (SE (496), HSE (255), GAE (123), GIE and GBE (345)²).

Each interaction was hand labeled as “Success” or “Failure” depending on whether or not the object interaction achieved the affordance. We used the following cutoffs for “Success”:

- **Breadbox (open-able)** - the breadbox had to be completely open. Any interactions where the robot only opened the box partially is a failure.
- **Pasta jar (push-able)** - the jar is pushed any distance without tipping.
- **Drawer (push-able)** - the drawer is pushed any distance.
- **Drawer (open-able)** - the robot has to pull the drawer out greater than or equal to 5.5 inches (the halfway point)
- **Lamp (turn-on-able)** - the robot has to turn on the lamp without causing the lamp to tip/wobble

To compare the five search strategies, we attempt to train 32 separate models for each affordance using the collected data; 2 for strategies that used the holistic approach to search (SE = 1, GAE = 1) and 30 models from the strategies that build a model per user (HSE = 10, GIE = 10, GBE = 10).

A. Model Representation

We represent each affordance using two Hidden Markov Models (HMMs) [17], where one model is built from success-

²GIE and GBE often explored similar locations around the object. As a result, we collected GIE and GBE as a single set and removed similar interactions using a 2cm threshold for position and 45° threshold for orientation.

ful interactions and one model is built from failed interactions. We build two models so that during classification, we can compute the log likelihood (a representation of the probability) of an interaction occurring for *both* models and select the model label that has the higher likelihood. Using relative likelihood avoids tuning a likelihood threshold for each affordance.

We selected HMMs because of the time-varying nature of the interaction. Furthermore, using a generative model that contains information about the EEF trajectory may allow us to generate actions for exploring new objects in future work. The trained HMMs are ergodic (all states are reachable from all other states) and the parameters of the n -state HMM (A, B, π), are learned using Expectation Maximization (EM), where A is the transition probability distribution ($n \times n$), B the emission probability distributions ($n \times 1$), and π the initial state probability vector ($n \times 1$). B is modeled using a continuous multivariate Gaussian distribution. The observation state-space O is composed of visual information, F/T information, and EEF relative to the object as described in Sec. III-B. To select the number of states n for each HMM, we performed 5-fold cross-validation within the training set described in Section VI-B. For our implementation, we used the python machine learning library scikit-learn [18].

B. Training and Testing

We split the data collected from each strategy into two sets: train and test. The train set for each strategy contains a randomly selected 80% of the samples from that strategy. The test set is comprised by merging the remaining 20% of the samples from each of the strategies. This results in a test set that contains examples from all strategies. Thus, each strategy trains using 80% of its own sample set, but is tested on a common test set that contains samples from all strategies. We evaluated the affordance models using standard metrics for binary classification of precision, recall, and F_1 score such that $\text{precision} = \frac{tp}{tp+fp}$; $\text{recall} = \frac{tp}{tp+fn}$; and $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ where tp is the number of true positives, fp false positives, tn true negatives, and fn false negatives. Precision is a measure of quality (e.g. how accurate is the model when it does label an interaction with the drawer as open-able?) and recall is a measure of completeness (e.g. of all interactions with the drawer, did the model miss any instances of open-able?).

VII. RESULTS

In this section we present (1) a characterization of the action space coverage achieved by each exploration strategy (2) the classification performance of the models for each strategy, and (3) qualitative results from the user study survey question.

A. Exploration Coverage

We first compare the exploration strategies by the total number and percentage of interactions that successfully achieve the affordances. Seen in Table II, the different strategies result in dramatically different number of samples. HSE resulted in around 5 samples per affordance, whereas SE had 100 samples. By design, all of the Guided strategies fall somewhere

TABLE III: Percentage of Positive Interactions Per Strategy

Object-Action	SE	HSE ^a	GAE	GIE ^a	GBE ^a
Breadbox-Move	8%	50%	39%	35%	42%
Pasta Jar-Move	44%	77%	93%	34%	33%
Drawer-Move	43%	78%	100%	28%	36%
Drawer-Pick	18%	38%	74%	33%	44%
Lamp-Pick	2%	4%	N/A	N/A	N/A

^a Values are averaged across each user

Note: Darker shading denotes higher scores and N/A means there were not enough examples for that strategy

TABLE IV: Classification Scores on All Exploration Strategies

Aff.	Strategy	n	Precision	Recall	F ₁ Score
Breadbox Move	SE	1	0.73	0.78	0.75
	HSE ^a	10	0.69±0.28	0.48±0.42	0.45±0.34
	GAE	1	0.75	0.53	0.62
	GIE ^a	5	0.75±0.05	0.62±0.35	0.60±0.27
	GBE ^a	3	0.81±0.08	0.53±0.31	0.57±0.24
Pastajar Move	SE	1	0.54	0.97	0.70
	HSE ^a	3	0.90±0.14	0.23±0.24	0.29±0.25
	GAE	0	N/A	N/A	N/A
	GIE ^a	2	0.65±0.12	0.80±0.20	0.69±0.01
	GBE ^a	4	0.53±0.36	0.45±0.40	0.39±0.28
Drawer Move	SE	1	1.00	0.35	0.52
	HSE ^a	3	0.40±0.43	0.06±0.06	0.08±0.07
	GAE	0	N/A	N/A	N/A
	GIE ^a	1	0.51±0.00	0.88±0.00	0.65±0.00
	GBE ^a	1	0.50±0.00	0.56±0.00	0.53±0.00
Drawer Pick	SE	1	0.66	0.93	0.77
	HSE ^a	4	0.66±0.41	0.26±0.42	0.22±0.33
	GAE	1	0.69	0.93	0.79
	GIE ^a	3	0.68±0.01	0.90±0.07	0.77±0.02
	GBE ^a	2	1.00±0.00	0.06±0.03	0.10±0.06

^a Reported values are averaged across the n user or user-biased models.

Note: Darker shading equates to higher scores and N/A means no model could be built using the example

between these two extremes. In prior work [1] it was shown that self exploration resulted in mostly negative examples, and their conclusion was that human teachers are good at showing the robot salient positive instances of object affordances. Our data also supports this conclusion. Table III shows the percentage of successful interactions per affordance. The human teachers (HSE) in our study showed a heavy bias for positive examples, with three of the five affordances having at least half successful examples. This positive bias carries over to the GAE strategy. In terms of coverage of the affordance space, GIE and GBE achieved what we wanted. Biasing SE with supervised examples results in a small number of samples (12 or 9 compared to 100) that have more positive examples than the SE strategy, and more negative examples than HSE.

For the Lamp-Pick affordance, only one of ten users and two SE interactions were able to complete the action successfully. To train and test a success HMM, we need a minimum of three successful interactions, otherwise the Guided exploration strategies cannot be generated. Thus we exclude Lamp-Pick in the rest of the results. Furthermore, given the limited data from human teachers, some users did not provide sufficient data to build both HMM models (i.e. min of 3 positive and 3 negative), and in some cases this carried over to the user-biased data sets as well. Column n in Table IV indicates the number of HSE or Guided strategies with sufficient data to build the affordance HMM model.

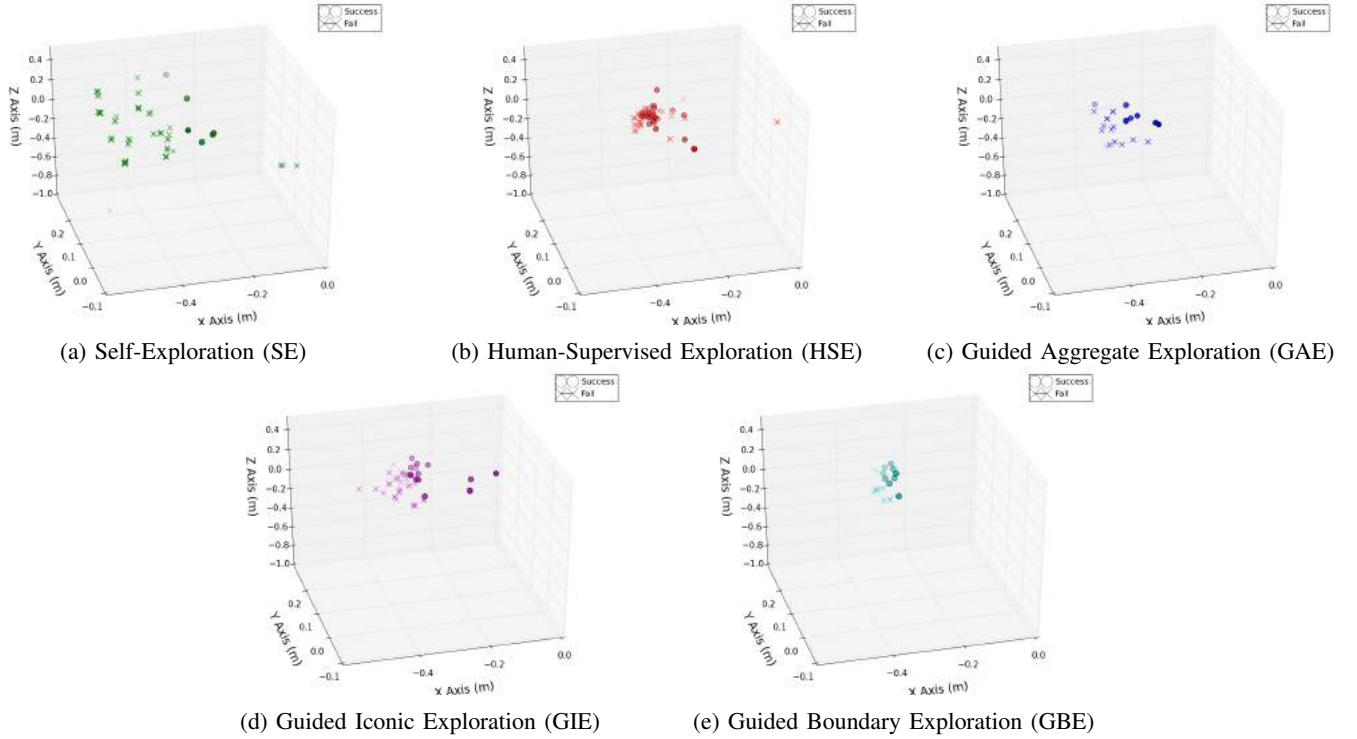


Fig. 3: The action space (EEF relative to the object) for all five strategies for the affordance Breadbox-Move. Successful interactions are circles and failed interactions crosses. Note: 3b, 3d, and 3e are aggregates over all of the user models.

Coverage can also be evaluated by the physical space the EEF explored and can be visualized by plotting the final position of the EEF relative to the object for successful and failed executions. Fig. 3 shows an example visualizing all five strategies for the object-action pair Breadbox-Move. The successful interactions have a high dependency on the x - and z -axis. This makes sense as the EEF must lift the handle away from itself to open the breadbox. In SE, we can see the grid structure of the exhaustive strategy and the large coverage of the action space. The explorations for HSE are highly concentrated and fixated on nearly the same locations in the space. GAE finds examples of successful and failed executions for Breadbox-Move (reflected in Table III), but without a clear action space boundary as in GIE and GBE. GBE examples span a wider range in the action space than GIE.

B. Model Performance

The performance of the models built from each strategy is shown in Table IV. HSE has the worst performance. This could be due to the fact that overall, users tended to overly focus on positive interactions and that these models were built from the least amount of data (5 examples compared to 9, 12, 30 or 100). These results support our hypothesis that the limited data we can collect from a person in 5-10 minutes of interaction is not sufficient to build models on par with exhaustive self-exploration consisting of 100 interactions with an object.

Next we turn to the question of whether or not the Guided strategies can help bridge this performance gap between SE

and HSE. Our results show that both GIE and GAE achieve higher performance than GBE and approach the performance of SE with significantly less data. GIE outperforms GAE in two ways: (1) across the four affordances, GAE only generates 2 working affordance models (due to the focus on successful examples) whereas GIE generated 11 (seen in column n in Table IV) and (2) GIE is likely to be the more practical approach compared to GAE since it can be used with a single individual as opposed to requiring data from multiple teachers.

Surprisingly, the simple affordances (Pastajar-Move & Drawer-Move) performed worse on average across the strategies than the complex affordances. The only exception is GIE for Pastajar-Move, which slightly outperforms Breadbox-Move. One possibility for the discrepancy between these affordances could be related to how affordances are not really on a binary spectrum, but rather there are varying levels (e.g. slightly push-able vs. very push-able). This suggests that the task should be a regression task where we label the affordances with values (e.g. 1cm vs. 5cm). In our case, we set hard cutoffs for judging the success of each affordance. Successful interactions were more obvious for complex tasks (e.g. drawer or breadbox fully opening) compared to simple tasks (e.g. shifting the pasta jar or drawer 10 cm vs. 1 cm across the table; both of which were considered successful interactions).

Finally, it is important to note the high variance for exploration strategies that generated models per user. While the average GIE performance is similar to SE, many individual models achieved performance that surpassed the SE models,

with an order of magnitude fewer examples. This shows that certain users provided better examples than others, and future work is to understand how to accomplish this with all users.

C. Qualitative Observations

There were a few interesting anecdotes and common threads from the survey administered at the end of the user study. In general, users tended to view the hour long session as “fun” and compared getting the robot to successfully find the affordance to puzzle solving. For simple affordances like Pastajar-Move, users tended to get bored quickly and many wanted to move onto the next affordance before the allotted time. Users’ dislike of failure resulted in an expressed preference to not provide examples of failure when teaching affordances. Users were allowed to discard demonstrations and one user used this as a feature to “test the action [they] wanted to teach [the robot] without her recording to see if her interaction with the object would behave as [they] expected.” Another user reported feeling dejected that he could not get the robot to successfully find the affordance and felt that it was due to a lack of ability and intelligence. Interestingly, while only a few negative examples were provided, 6 out of 10 users reported thinking about providing negative examples in the survey.

VIII. FUTURE WORK

To evaluate the quality of the affordance model, we used a binary classification task (i.e. did the interaction find the affordance?). While a commonly used metric, this will favor models that excel at finding the boundary cases of an affordance because the best examples to train the models are the ones that can help determine the separating line between success and failure. However, for the robot to use the learned model on novel objects, it will likely need to generate trajectories that are *prototypical* interactions and not ones near the boundary. Future work will investigate if the models that performed well for monitoring can perform well for trajectory generation on novel objects.

While on average the GIE strategy performed comparably with SE, the variance of the scores showed that certain user models did better than others. Future work will look at understanding what exactly a user needs to provide to build a robust affordance model. One hypothesis is that the type of failed interaction might have a strong impact on the exploration space. For example, while many users provided failed explorations unintentionally, others consciously provided an example of not finding the affordance.

IX. CONCLUSION

This paper compared three different approaches to affordance learning: self-exploration, human-supervised exploration, and a *combined* human-guided approach that performs self-exploration biased by information provided from human teachers. We outline two specific implementations of existing self and human-supervised approaches (SE and HSE) and introduced three novel combination strategies: GAE, GIE, and GBE. We compared these affordance exploration strategies by

learning five affordances across four objects, resulting in 1219 robot executions. Our results show the combined approach, GIE, can learn affordance models that perform on par with those generated from exhaustive SE, but using an order of magnitude fewer interactions with the object.

ACKNOWLEDGMENT

This work is funded through Office of Naval Research Grant N000141410120 and NSF Graduate Research Fellowship DGE-1148903.

REFERENCES

- [1] A. L. Thomaz and M. Cakmak, “Learning about objects with human teachers,” in *International Conference on Human Robot Interaction (HRI)*, 2009, pp. 15–22.
- [2] J. Gibson, “The concept of affordances,” *Perceiving, acting, and knowing*, pp. 67–82, 1977.
- [3] E. Şahin, M. Çakmak, M. R. Doğan, E. Uğur, and G. Üçoluk, “To afford or not to afford: A new formalization of affordances toward affordance-based robot control,” *Adaptive Behavior*, vol. 15, pp. 447–472, 2007.
- [4] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, “Learning object affordances: From sensory-motor coordination to imitation,” *Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.
- [5] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, “Learning about objects through action - initial steps towards artificial cognition,” in *International Conference on Robotics and Automation (ICRA)*, Sept 2003, pp. 3140–3145.
- [6] A. Stoytchev, “Behavior-grounded representation of tool affordances,” in *International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 3060–3065.
- [7] M. R. Dogar, E. Ugur, E. Sahin, and M. Cakmak, “Using learned affordances for robotic behavior development,” in *International Conference on Robotics and Automation (ICRA)*, 2008, pp. 3802–3807.
- [8] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, “Learning stable pushing locations,” in *Development and Learning and Epigenetic Robotics (ICDL)* 2013, 2013, pp. 1–7.
- [9] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, “Learning relational affordance models for robots in multi-object manipulation tasks,” in *International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4373–4378.
- [10] D. Katz, A. Venkatraman, M. Kazemi, J. A. D. Bagnell, and A. T. Stentz, “Perceiving, learning, and exploiting object affordances for autonomous pile manipulation,” in *RSS Berlin*, June 2013.
- [11] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, “Intrinsic motivation systems for autonomous mental development,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 2, pp. 265–286, April 2007.
- [12] C. Vigorito and A. Barto, “Intrinsically motivated hierarchical skill learning in structured environments,” *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 2, pp. 132–143, June 2010.
- [13] J. Schmidhuber, “Curious model-building control systems,” in *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, Nov 1991, pp. 1458–1463 vol.2.
- [14] S. Ivaldi, S. M. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P.-Y. Oudeyer, and O. Sigaud, “Object learning through active exploration,” *Autonomous Mental Development, IEEE Transactions on*, vol. 6, no. 1, pp. 56–72, March 2014.
- [15] S. Ivaldi, N. Lyubova, D. Gerardeaux-Viret, A. Droniou, S. Anzalone, M. Chetouani, D. Filliat, and O. Sigaud, “Perception and human interaction for developmental learning of objects and affordances,” in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, Nov 2012, pp. 248–254.
- [16] S. Nguyen and P.-Y. Oudeyer, “Socially guided intrinsic motivation for robot learning of motor skills,” *Autonomous Robots*, vol. 36, no. 3, pp. 273–294, 2014.
- [17] L. Rabiner and B.-H. Juang, “An introduction to hidden markov models,” *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.