


Analyzing Differences between Teachers when Learning Object Affordances via Guided-Exploration

Vivian Chu¹ and Andrea L. Thomaz²

Journal Title
XX(X):1–22
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/


Abstract

Our work focuses on robots 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 prior work has shown that people naturally focus on showing salient aspects of objects when providing demonstrations. In our work, we use a *Guided Exploration* approach that combines self and supervised learning. We present experimental results with a robot learning 3 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) a combined approach of self-exploration biased by user input. Previous analysis of this data focused on aggregate performance of these different strategies across all teachers, and showed that a combined approach is the most efficient and successful. In this article, we provide additional details on these specific strategies as well as an analysis of the variance seen across teachers in this experiment. We provide a characterization of failure cases and insights for future work in learning from naïve end-users.

Keywords

1 Introduction

As robots make their way into unstructured human environments such as homes and hospitals, they will increasingly need to learn about and model their specific environment quickly and with as little help as possible from human end-users. Our work focuses on expediting the robots exploration of new environments through a novel combination of self-exploration and human-guidance.

We take an affordance approach to this modeling problem, whereby the robot builds representations of its actions and the effects that they have on objects in the environment. The term “affordance” was first introduced by [Gibson \(1977\)](#). We use the ecological definition of “action possibilities” that appear between an agent and the environment that is commonly used in robotics [Şahin et al. \(2007\)](#); [Montesano et al. \(2008\)](#). Affordances provide a nice building block for performing tasks. Given such affordance models of an environment, a robot should be able to make inferences about the objects that have the appropriate affordances to accomplish a given task. In this work we focus on how a robot can efficiently leverage a



Figure 1. A naïve user during the user study teaching “Curi” the robot that the drawer has the open-able affordance.

¹ School of Interactive Computing, Georgia Institute of Technology

² Electrical and Computer Engineering Department, University of Texas at Austin

Corresponding author:

Vivian Chu, Georgia Institute of Technology, Atlanta, Georgia 30332–0250

Email: vchu@gatech.edu

human end-user to build affordance models of actions and objects.

In prior work, we presented and compared 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 allows the robot to perform self-exploration biased by information provided from human teachers. We evaluated these three strategies by learning 3 affordances across 4 different objects and actions. We showed that, from an aggregate viewpoint, using a human-guided approach, a robot can learn an affordance model that is as effective as exhaustive self-exploration with an order of magnitude fewer interactions with the object [Chu et al. \(2016b\)](#).

This work expands on prior work with additional details on the experiment and expanded explanations of the exploration algorithms. Furthermore, we focus on the observation from this prior work where we saw that the 10 individual teachers in our experiment showed high variance in their performance. Our aim in this paper is to analyze this variance to determine what caused some users' demonstrations to be more informative for exploration and others not.

2 Related Work

The field of robot affordance learning and exploration is by no means a new field. In this section, we go through some of the influential work in the areas of robot affordance learning, robot exploration, and guided exploration for learning affordances.

2.1 Robot Affordance Learning

Early work in affordance learning for robotics focused on using primitive actions to interact and learn about object effects. These established a framework for affordance learning using exploration. [Fitzpatrick et al. \(2003\)](#) used parametrized primitive actions to push or roll objects. [Stoytchev \(2005\)](#) used a robotic arm to use tools to bring objects within reach. [Dogar et al. \(2008\)](#) tackled the traversability affordance using visual cues and wheel encoders. [Montesano et al. \(2008\)](#) learned affordances to be used for imitation. In an effort to use and plan with affordances, [Krüger et al. \(2011\)](#) developed a rich framework that allowed for affordances to be defined as low-level primitives as well as chained to perform high-level tasks. [Hermans et al. \(2013a,b\)](#) investigated primitives for pushing objects on a flat surface. [Moldovan et al.](#)

[\(2012\)](#) looked at the relationship between affordances for multi-object manipulation tasks. In the area of using scaffolding for affordance learning, [Thomaz and Cakmak \(2009\)](#) demonstrated the importance of scaffolding for learning affordances.

More recently, [Koppula and Saxena \(2013\)](#) used affordances to predict and anticipate human activities. [Katz et al. \(2013\)](#) used grasping affordances to learn the best way to clear rubble in a pile. [Varadarajan and Vincze \(2012\)](#) built on AfNet, an open affordance initiative, by providing semantic context and household manipulation objects. All of these works, however, required specific primitive actions to be learned or programmed, and did not use human input for guidance.

2.2 Exploration

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 [Oudeyer et al. \(2007\)](#); [Vigorito and Barto \(2010\)](#); [Schmidhuber \(1991\)](#) looked at using rewards and expectations to guide the exploration without any human supervision.

The latest work on intrinsic exploration from [Ivaldi et al. \(2014, 2012\)](#) and [Nguyen and Oudeyer \(2014\)](#) 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.

2.3 Socially Guided Exploration for Affordance Learning

[Ivaldi et al. \(2012\)](#) introduced a method for human-guided exploration by having a teacher verbally command a robot to explore the environment with a set of pre-defined primitives. The work, however, does not address how to create more primitives.

Our work extends the previous work [Thomaz and Cakmak \(2009\)](#) 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. Our use of haptic affordances has been presented in [Chu et al. \(2016a\)](#), where we demonstrate that the use of a force signal is useful and necessary for the kinds of object manipulation affordances used here. The experiment in this paper was first presented in [Chu et al. \(2016b\)](#) and in this article, we provide a more detailed description

of the exploration strategies used in the experiment as well as further analysis of the data by following up on the open question of why some individuals' demonstrations worked well as seeds for exploration and others did not.

3 Affordance Learning

For a robot to learn affordances, it needs to interact with the environment and observe the effects of that interaction. This interaction needs to be done by the robot (as opposed to only observations of a human performing the skill) because affordances are action possibilities that occur between the environment and the *agent*. More concretely, there exist many objects that have affordances for a human that do not exist for all robots (e.g. jar lid is too wide for the robot to grasp). Once a robot interacts with an object and observes the effects of that interaction, 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 Figure 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. This choice of representation has gained traction in the reinforcement learning community and has shown great promise with learning actions and skills Kober et al. (2012); da Silva et al. (2014). Consider again the drawer example, now the opening action can be a primitive defined with three parameters (start, close-hand, and end poses). Note that this still results in a sample space that is infinitely large, because these actions parameters are continuous-valued poses of the end-effector.

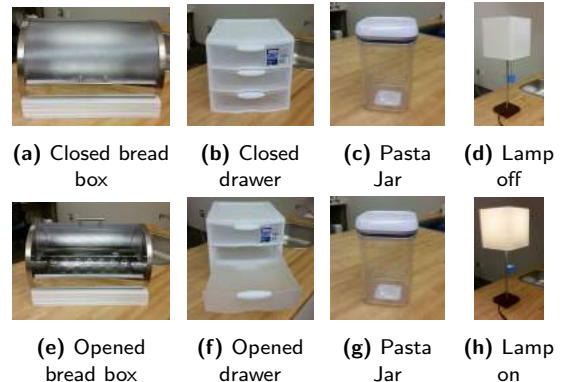


Figure 2. Shown are the various objects the robot explored. The top row are the objects before interaction and the bottom row include the same objects with the effect the robot is looking for. Note: pushing the drawer and pasta jar shift the object on the table.

Thus, we present and compare five different strategies in Sections 4 and 5 for efficiently sampling this space to collect a sufficient set of examples to build object-action affordance models.

3.1 Hardware Platform

For our experiments, we used the robot “Curi”, seen in Figure 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 and 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.

3.2 Objects and Actions

We selected four household objects (Figure 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 Thomaz and Cakmak (2009); Montesano et al. (2008) (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-action pairs for the affordance learning problem.

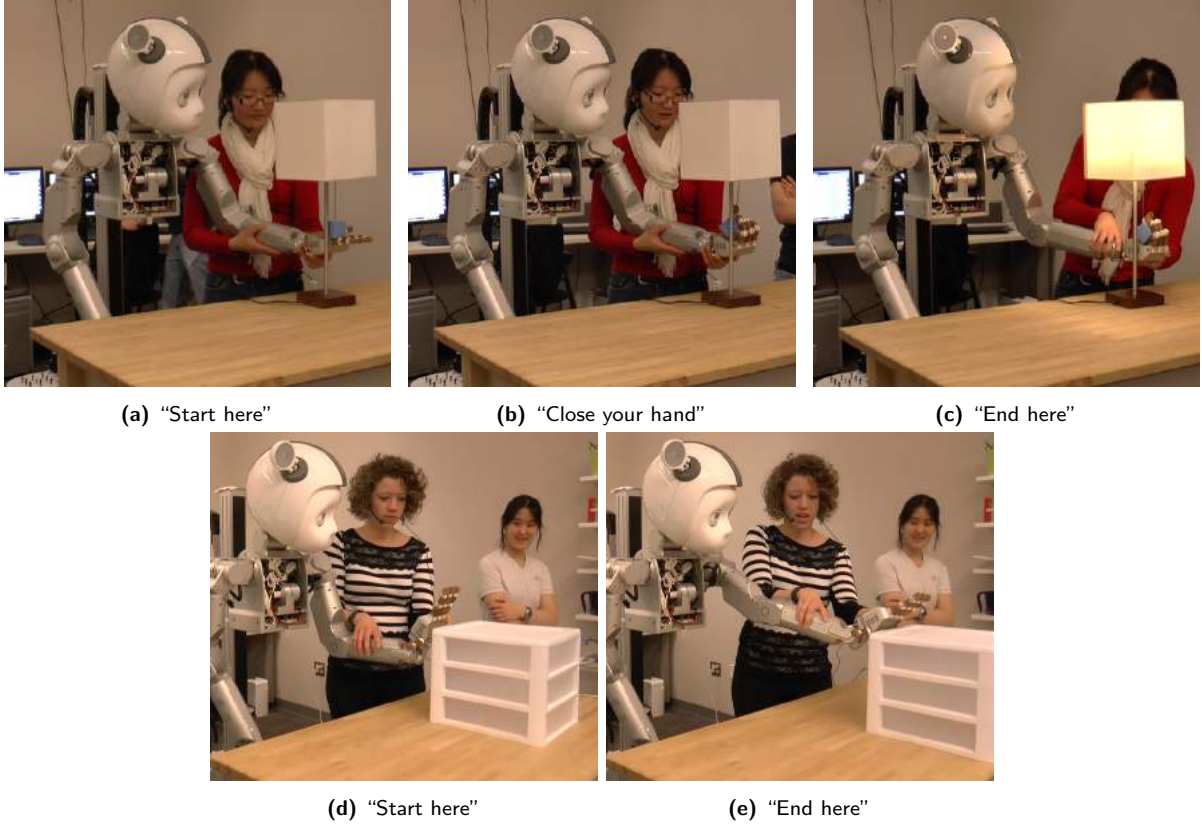


Figure 3. Shown are the two primitive actions (**pick** and **move**) taught by users during the user study by using keyframe-based kinesthetic learning from demonstration while the arm is in gravity compensated mode. The users indicate poses within a keyframe using voice commands seen below each image. The **pick** action (3a,3b,3c) consists of a start, close, and end pose and is being demonstrated on the lamp. The **move** action (3d,3e) consists of a start and end pose and is being demonstrated on the drawer

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 for the EEF by performing a quintic spline between the EEF poses with an average velocity of 1 cm/second. The two actions can be seen in Figure 3 where naïve users from our user study are demonstrating **pick** and **move** actions on the lamp and the drawer respectively.

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 Fitzpatrick et al. (2003); Hermans et al. (2013a). 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).

3.3 Affordances

The five specific object-action pairs and their corresponding affordance used in this paper are described below and summarized in Table 1. The effects of each object-action pair can be seen in Figure 2. These selected 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 variety of 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

Table 1. 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

way, representing a small subset of the object-action exploration space).

- **Bread box:** The lid of the breadbox can be opened with a move action. Affordance: **open-able**.
- **Pasta jar:** The pasta jar can be pushed across the table. Affordance: **push-able**.
- **Drawer:** The drawer unit is light enough that the robot can push it across the table. It also contains shelves that can be pulled open. Affordance: **push-able, open-able**
- **Lamp:** When the string attached to the lamp is pulled far enough, the lamp turns on. Affordance: **Turn on-able**.

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.

4 Baseline Exploration Strategies

Typical self-exploration strategies in robot affordance learning [Fitzpatrick et al. \(2003\)](#); [Stoytchev \(2005\)](#); [Hermans et al. \(2013a\)](#) 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 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 and visually in Figure 4). To understand the importance of these two variables on exploration, consider the following section.

4.1 Self-Exploration (SE)

To represent all 6 DOFs of the EEF, requires three variables (x, y, z) to describe the position and three

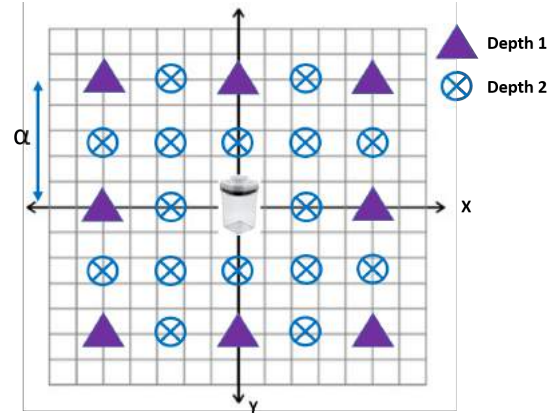


Figure 4. Shown above is a visual example of the self-exploration algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the starting position of the object and the two depths of exploration are shown in two different shapes.

Algorithm 1 Self-Exploration (SE)

```

1: procedure COMPUTEPERMUTATION( $v = [v_0, v_1, \dots, v_n]$ )
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 GENERATEEXPLORATION(expert dist.  $d$ )
6:    $\alpha \leftarrow d + 10\text{cm}$ 
7:    $D_1 \leftarrow \text{ComputePermutation}([- \alpha, 0, \alpha])$ 
8:    $D_2 \leftarrow \text{ComputePermutation}([- \alpha, -\frac{\alpha}{2}, 0, \frac{\alpha}{2}, \alpha])$ 
9:    $Unique \leftarrow (D_1 \cup D_2) - (D_1 \cap D_2)$ 
10:   $ExploreSet \leftarrow \text{Random}(Unique, 100)$ 
11:  return  $ExploreSet$ 

```

variables (rx, ry, rz) to describe the orientation in Euler space. However, in practice, 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 object-action pair in question. Even this coarse exploration of the action space results in 676 interactions per EEF pose per object-action pair and, realistically, a higher resolution search will most likely be needed to find the affordance.

It is infeasible for the robot to perform all exploratory actions for all five object-action pairs and all possible primitive action parameter poses. To reduce the number of exploratory actions, 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**. These are provided to be *ideal* for achieving the affordance. We believe it is a reasonable assumption to provide the start/close pose, because there exist state-of-the-art algorithms that find the best grasp/interaction points for a wide range of objects (e.g. the handle of the breadbox or the ball on the chain for the lamp). Furthermore, providing this information only helps self-exploration by providing expert information as to where the robot should be interacting. With this assumption, SE exhaustively explores the position (x,y,z) of the final end pose for each primitive action. While varying the orientation could provide additional ways to achieve an affordance, we fix the final pose orientation to be the same as the start pose to keep exploration tractable. Even with these constraints, the number of explorations generated can still result in an impractical number of exploratory actions. Thus, 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 until 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 . This adaptive split can be seen visually in 2-dimensions in Figure 5. 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, as mentioned earlier, to make SE tractable, we provided expert information to the algorithm in the form of 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.

4.2 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 successful or unsuccessful examples of the affordance. Our approach, HSE, builds on Thomaz and Cakmak (2009), but uses more realistic objects found in everyday homes 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, which can be seen in Figure 3. 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 two actions on two objects: lifting the lid off a jar with the **pick** action and tipping an object over with the **move** action. These affordances are not included in our analysis. Once they were comfortable with how to control the robot and had performed several example affordances, we began their real data collection.

The participants taught the robot about the 3 affordances over the 4 objects described in Table 1 for a total of 5 object-action pairs. 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. Participants were instructed during this time to also think about negative examples as a good way to teach a child about an affordance. However, we wanted to see how people teach robots about affordances naturally. Thus, we did not force users to provide negative examples or provide guidance as to how they should teach the robot. A single example for affordance learning was collected each time the robot executes

Table 2. 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.

Note: N/A means there were not enough examples for that strategy

the taught action autonomously. To generate multiple object-action examples, participants could either move the object and repeat the previous action or they could teach a new action.

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 object-action pair 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 object-action pairs 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 examples collected across all 10 users can be seen in Table 2.

5 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 object-action pair. 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.

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:  $\vec{r}_{\text{change}} \leftarrow \mu_{ji} - \text{EEF}_{\text{startposition}}$ 
6:  $\vec{e}_{\text{change}} \leftarrow \frac{\vec{r}_{\text{change}}}{\|\vec{r}_{\text{change}}\|_2}$ 
7:
8: procedure GENERATEEXPLORATION
9:    $\text{exploreRegions} = [\mu_{ji}, \mu_{ji} + \sigma_{ji}^2, \mu_{ji} - \sigma_{ji}^2]$ 
10:   $\text{ExploreSet} \leftarrow \text{ComputePermutation}(\text{exploreRegions})$ 
11:   $c \leftarrow 1$ 
12:   $\vec{p}_{\text{change}} \leftarrow (0, 0, 0)$ 
13:  while  $\|\vec{p}_{\text{change}}\| < \alpha$  do
14:     $\vec{p}_{\text{change}} = \vec{e}_{\text{change}} * c * \alpha$ 
15:     $\text{ExploreSet} \leftarrow \text{ExploreSet} \cup \{\vec{p}_{\text{change}}\}$ 
16:     $c++$ 
  return  $\text{ExploreSet}$ 

```

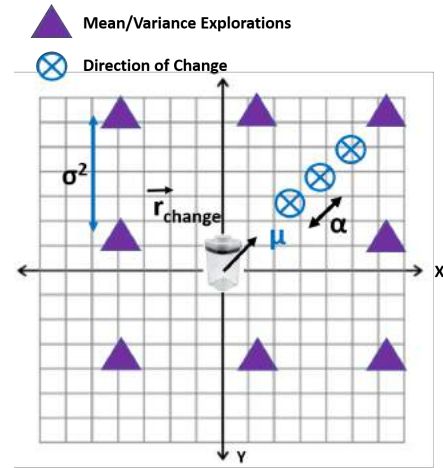


Figure 5. Shown is a visual example of the GAE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration is centered around the mean ending position of the first demonstrated by all of the users and exploration is bounded by the variance of the demonstrations.

5.1 Guided Aggregate Exploration (GAE)

Our first approach, GAE (Algorithm 2), takes an aggregate view of the guidance that people provided from HSE. The algorithm is described in detail below and shown visually in 2-dimensions in Figure 5. For each object-action pair, 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. We use only the first action from each user to create a strategy that could be generated using a person's first intuition for teaching the affordance. However, this is difficult to achieve using just one action primitive and so we built a set that contains the final position of the first action from

all users. 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 object-action 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 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 object-action pair, 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 object-action 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 object-action pair 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 $\vec{r}_{change} = \mu_{ji} - EEF_{startposition}$ 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 object-action pairs.

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  $ExploreSet$ 

```

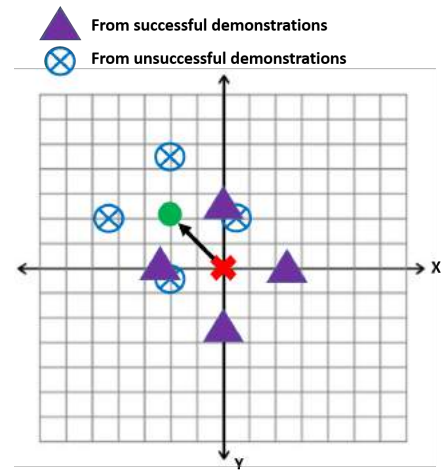


Figure 6. Shown is a visual example of the GIE algorithm. The algorithm is viewed in two-dimensions to be visually clear. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around.

5.2 Guided Iconic Exploration (GIE)

Our next approach, GIE (Algorithm 3 and shown visually in Figure 6), uses each human teacher's input individually to bias the exploration of the action space rather than relying on the aggregate of several teachers. 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

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  $ExploreSet$ 

```

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 (e.g. the small lip of a drawer).

5.3 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 and visually in Figure 7), 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 (x, y, z) for

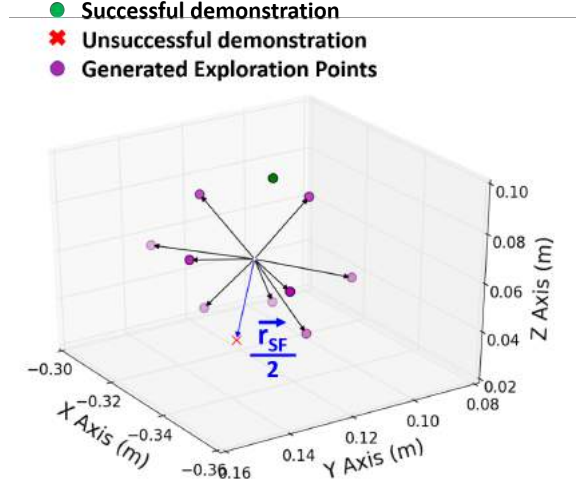


Figure 7. Shown is a visual example of the GBE algorithm. The exploration uses the first successful and first unsuccessful demonstrations to determine the resolution of exploration as well as where in the space to explore around.

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.

6 Affordance Modeling

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

Each interaction was hand labeled as “Success” or “Failure” depending on whether or not the object interaction achieved the affordance. An example interaction with the breadbox can be seen in Figure 8. We used the following cutoffs for “Success”:

* GIE 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.

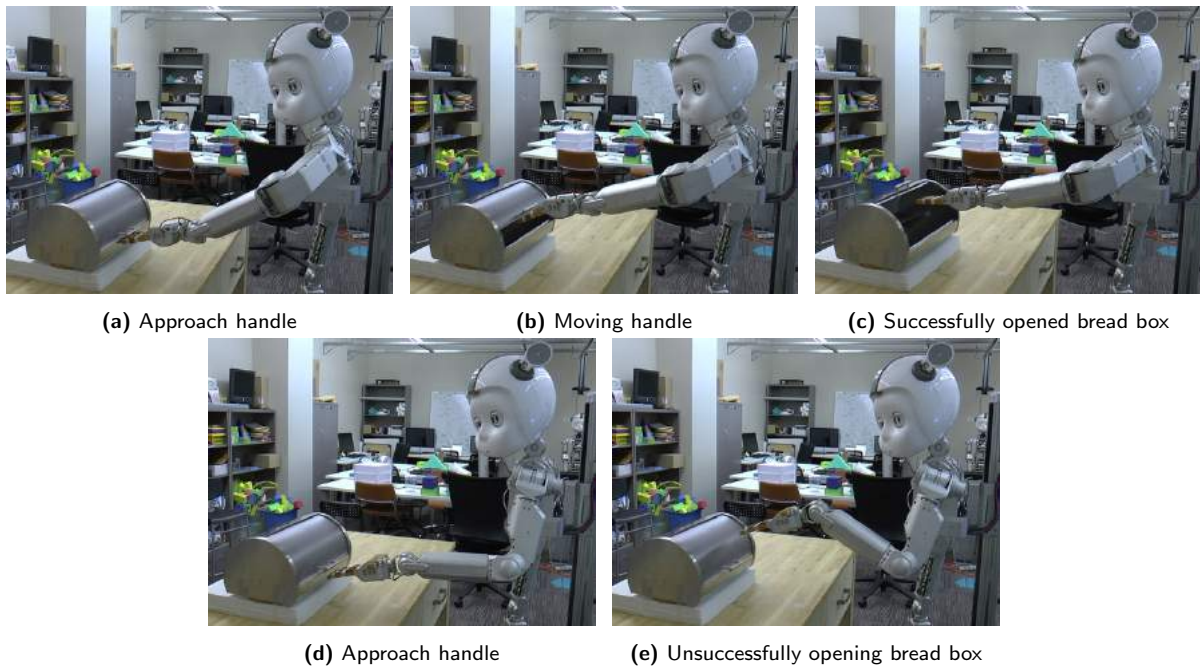


Figure 8. Shown are example interactions of Curi executing the **move** action on the bread box to find the open-able affordance. The top row (8a,8b,8c) show Curi successfully finding the open-able affordance. The bottom row (8d,8e) shows an example of Curi failing to find the affordance.

- **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 object-action pair 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 ($HSE = 10$, $GIE = 10$, $GBE = 10$).

6.1 Model Representation

We represent each object-action pair using two Hidden Markov Models (HMMs) [Rabiner and Juang \(1986\)](#), where one model is built from successful 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 object-action pair.

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 (nxn), B the emission probability distributions ($nx1$), and π the initial state probability vector ($nx1$). 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. 3.2. To select the number of states n for each HMM, we performed 5-fold cross-validation within the training set described in Section 6.2. For our implementation, we used the Python machine learning library scikit-learn [Pedregosa et al. \(2011\)](#).

Table 3. 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

6.2 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 object-action 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?).

7 Aggregate Results

7.1 Exploration Coverage

Exploration coverage can be broken down into two categories (1) the total area of the action space of the object and (2) the ratio between successful and unsuccessful interactions. We are interested in the coverage of the action space because the end goal for our evaluation is to find the separating line between interactions that find an affordance and interactions that do not. Successfully finding this line allows us to build a model that can correctly determine what it means to find and not find the affordance. For SE, this boundary is completely unknown and the best it can do is exhaustively search, which results in a large number (100 per object-action pair) of executions. In contrast, guidance from human-users provides information on where this boundary might lie and as a result, each of the guided exploration strategies requires fewer interactions (seen in Table 2).

Table 4. Classification Scores on All Exploration Strategies

(o_j, a_i)	Strat.	n	Precision	Recall	F ₁ Score
Breadbox Move	SE	1	0.73	0.78	0.75
	HSE ^a	9	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

We also showed that by having human-input, it is possible to design strategies (GIE and GBE) that will search around this boundary to provide a balanced ratio of successful and unsuccessful interactions (see in Table 3).

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 4 indicates the number of HSE or Guided strategies with sufficient data to build the object-action HMM model.

7.2 Model Performance

Our prior results [Chu et al. \(2016b\)](#) showed that interactions from HSE (Table 4) were overly focused on positive examples and the number of trials (approx. 5) were insufficient to build models that could perform on par to self-exploration. Furthermore, Guided exploration bridged this performance gap while still requiring far less (an order of magnitude) interactions than self-exploration with GIE performing better than GBE and GAE. The results noted the high variance between the performance of user-specific models, but did not explore why this occurred. In

Table 5. User Specific Classification Scores for Strategies HSE, GIE, and GBE

Object-Action	User	Human			Iconic			Boundary		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Breadbox Move	User 1	1.00	0.01	0.03	N/A	N/A	N/A	N/A	N/A	N/A
	User 2	0.74	0.29	0.42	0.69	0.62	0.65	0.71	0.90	0.79
	User 3	0.71	0.84	0.77	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	0.60	0.13	0.22	N/A	N/A	N/A	N/A	N/A	N/A
	User 6	0.72	0.97	0.82	0.72	1.00	0.84	0.83	0.57	0.68
	User 7	0.73	0.99	0.84	0.72	0.99	0.83	N/A	N/A	N/A
	User 8	1.00	0.09	0.16	0.80	0.06	0.11	0.90	0.13	0.23
	User 9	0.73	0.97	0.83	0.82	0.46	0.58	N/A	N/A	N/A
Pastajar Move	User 1	1.00	0.09	0.17	0.77	0.61	0.68	0.65	0.67	0.66
	User 2	0.70	0.58	0.63	N/A	N/A	N/A	0.00	0.00	0.00
	User 3	1.00	0.03	0.06	0.53	1.00	0.69	1.00	0.15	0.26
	User 6	N/A	N/A	N/A	N/A	N/A	N/A	0.48	1.00	0.65
Drawer Move	User 2	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 3	1.00	0.03	0.06	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	N/A	N/A	N/A	0.51	0.88	0.65	0.50	0.56	0.53
	User 8	0.21	0.15	0.17	N/A	N/A	N/A	N/A	N/A	N/A
Drawer Pick	User 3	0.65	0.98	0.78	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	N/A	N/A	N/A	0.69	0.84	0.76	1.00	0.02	0.04
	User 5	1.00	0.02	0.04	0.67	0.87	0.76	N/A	N/A	N/A
	User 6	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 7	N/A	N/A	N/A	0.67	1.00	0.80	1.00	0.09	0.16
	User 8	1.00	0.02	0.04	N/A	N/A	N/A	N/A	N/A	N/A

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

the next section, we investigate the difference between users to determine why this might have been the case.

8 User Specific Results

To understand why some user specific models performed better than others, we take a deeper look at each user. We first look at the precision, recall, and F_1 scores for each user specific strategy (HSE, GIE, and GBE). This can be seen in Table 5. As described previously, several users outperformed self-exploration (5 users in HSE, 5 users in GIE, and 1 user in GBE across all four object-action pairs). Table 5 also shows why GIE outperforms GBE and HSE on aggregate. While many of the models from HSE and GBE do equally as well as models from GIE, there are several models in HSE and GBE that perform poorly. In contrast only 1 model in GIE performs poorly (Breadbox-Move: User 8).

Interestingly, for some users, even though the user provided enough positive or negative examples to build models for HSE, GIE and GBE were not able to find enough examples. This tells us that the first successful and unsuccessful demonstrations were not diverse enough to provide a sufficient amount of exploration range for GIE or GBE. Specifically, for

all object-action pairs except for Pastajar-Move, there were users where HSE could build a model, but GIE or GBE could not. To better understand this, we looked at the pose of the first successful and unsuccessful demonstration provided from users that generated good GIE/GBE models and compared them to users who did not. In Figure 9, we can see the comparison of the final pose of each user’s first successful and unsuccessful demonstration. Visually, the general location of the demonstrations seem relatively similar. However, computing the average Euclidean distance between the successful and unsuccessful positions (shown in Table 6), shows that participants who provided demonstrations that were further apart in distance, allowed GIE and GBE to generate better models. This makes intuitive sense as both GIE and GBE rely on the user’s demonstration to determine the resolution to search within. If the resolution is too small, then the algorithm does not explore a large enough range to capture a balanced set of positive and negative interactions.

8.1 Exploration Coverage

In Chu et al. (2016b), we visually showed the difference between strategies in the action space (EEF position relative to the object). Here we visualize the action

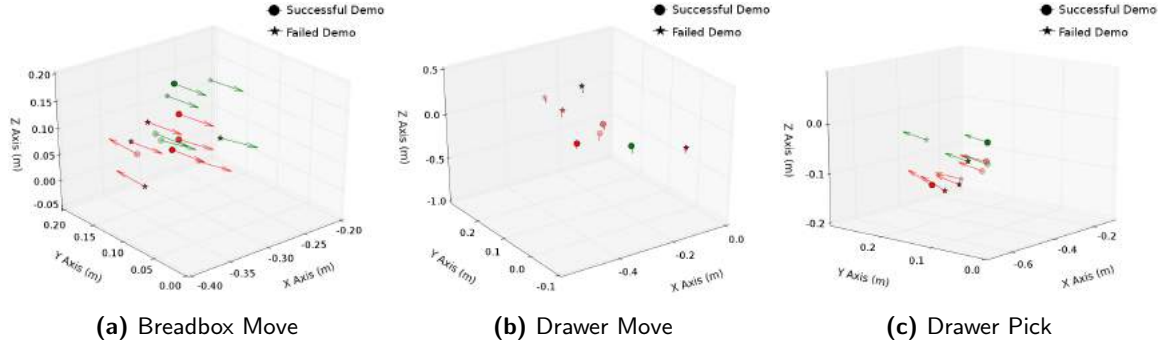


Figure 9. This shows the first successful and unsuccessful demonstration for the affordances Breadbox-Move, Drawer-Move, and Drawer-Pick. The symbols indicate if the demonstration given was a success or fail demonstration. The colors separated the users that could generate GIE/GBE models and those who could not (green - models were generated, red - models were not). Note: this figure requires color to fully understand.

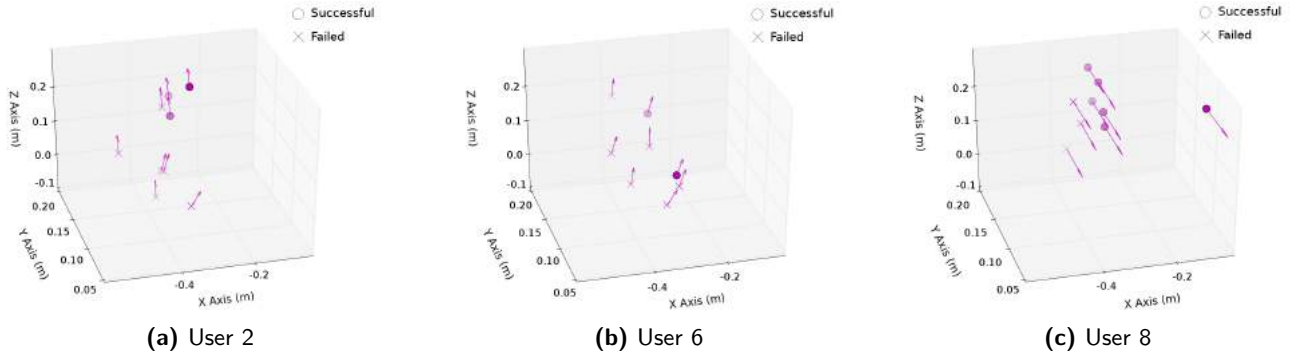


Figure 10. Shown is the action space (EEF relative to the object) for strategy Guided Iconic Exploration for users 2, 6, and 8 for the object-action pair Breadbox-Move. The arrow indicates the direction the EEF palm is facing. Successful interactions are circles and failed interactions are crosses.

Table 6. Average Distance Between Demonstrations

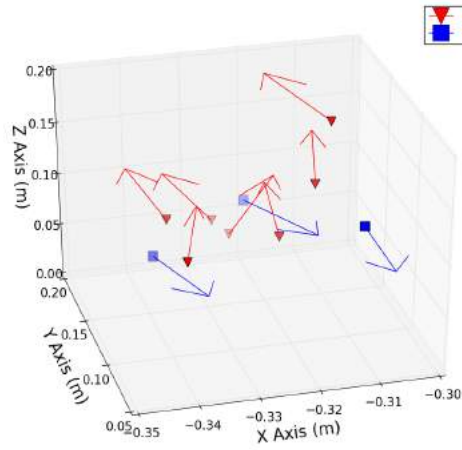
Object-Action	Good (cm)	Poor (cm)
Breadbox Move	8.46 ± 0.05	7.3 ± 0.19
Drawer Move	32.94 ± 0.0	24.60 ± 0.14
Drawer Pick	8.98 ± 0.01	8.08 ± 0.019

space to provide insight into why a model based on a certain user's demonstrations might not have generated a good model. Table 5 shows that only one user's GIE model performed poorly. Furthermore, this particular user (user 8) does poorly across all of the user specific strategies (HSE, GIE, GBE) for the object-action pair Breadbox-Move. The next set of graphs will be presented as a case study to determine what differences exist between user 8 and the rest of the users.

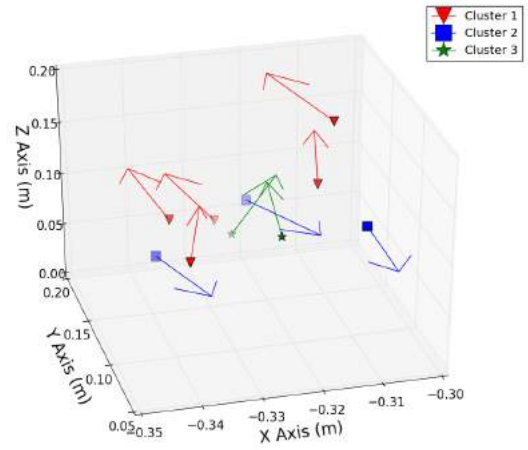
In Figure 10, we see three different users' exploration points generated for the object-pair Breadbox-Move for the strategy GIE. Circles represent successful interactions while crosses represent failed interactions. The orientation of the palm of the EEF is also shown

as a vector. The figure shows that the orientation of the EEF played a clear role in differentiating user 8 from the rest of the users. User 8 chose a different orientation when opening the bread box and video verifies that user 8 had Curi's palm facing down as opposed to up to lift the handle.

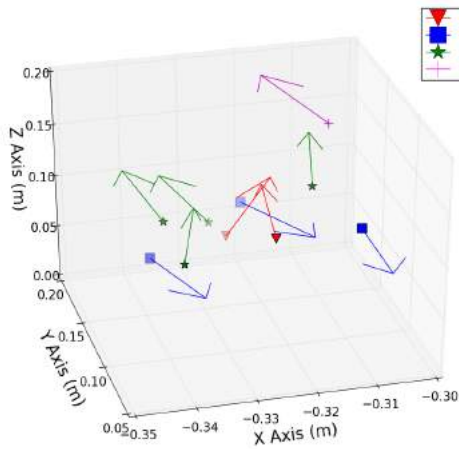
This suggests that the demonstration from user 8 was not bad, but rather *different* from the other demonstrations provided by other users. Furthermore, we hypothesize that if there existed a subset of the evaluation set that is similar to the demonstrations from user 8, then the performance for that user would increase. To understand and determine which users were most similar to each other, we take a simple approach of clustering all of the user's first demonstrations (those used to seed GIE and GBE) using a standard unsupervised clustering algorithm k-means.



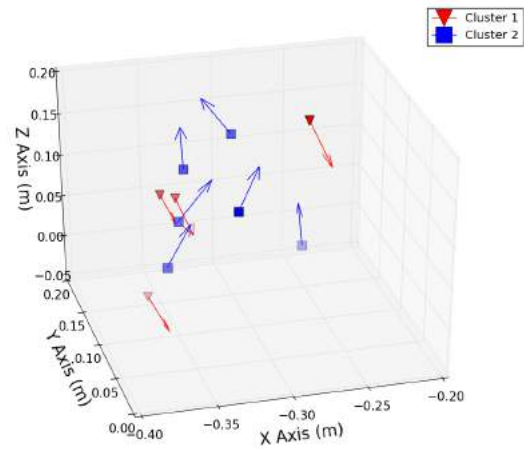
(a) Success; Size: 2



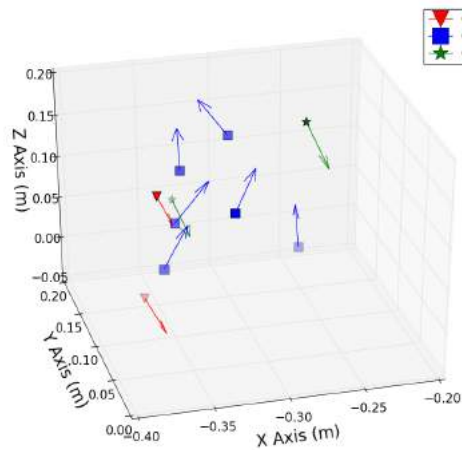
(b) Success; Size: 3



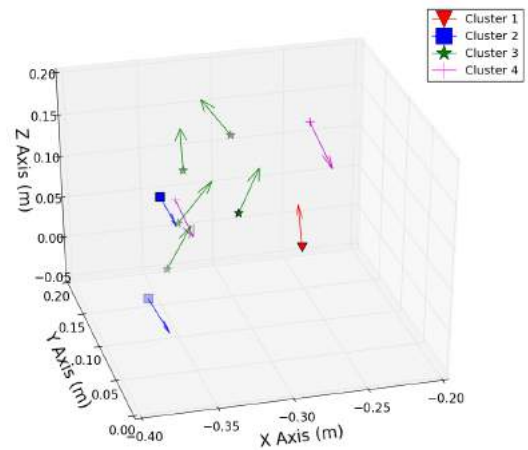
(c) Success; Size: 4



(d) Fail; Size: 2



(e) Fail; Size: 3



(f) Fail; Size: 4

Figure 11. Displayed are the first success and fail demonstration from each user in action space (EEf relative to object) for different clusters sizes (2,3,4). The arrow indicates the direction the EEf palm is facing. Note: this figure will be easier to decipher with color.

8.2 Clustering

To cluster the demonstrations using k-means, there are two decisions that need to be made (1) what metric to use for distance between demonstrations and (2) the number of clusters we expect to see. We chose to use the Euclidean distance of the EEF position and orientation of the palm relative to the object. This decision allows us to focus on what the user demonstrated relative to the object without looking at the effects generated by the demonstration. Furthermore, Euclidean distance is a natural metric between points in three-dimensions. While orientation of the EEF is stored as a quaternion, when computing the distance between demonstrations, orientation is represented by the normalized unit vector of the direction the palm of the EEF is facing (shown in Figure 10).

We chose several cluster sizes and compared user performance within clusters. As a reminder, we use both the first successful and first unsuccessful demonstration from each user when generating exploration points. We cluster successes and failures separately. While successful demonstrations are (typically) intentional, failures are not guaranteed to be intentional. Often during the HSE, the human-user’s first failure was a result of failing to demonstrate a successful interaction.

We visually show which cluster the user’s first demonstrations (successes and failures) fall into for the object-action pair Breadbox-Move, which can be seen in Figure 11. For clusters of size two, it seems that orientation of the EEF plays a larger role in cluster membership. As the cluster sizes increase, position plays a larger role.

8.3 Clustering Performance

To verify that the difference in initial demonstrations impacts the final performance of a user-specific model, we hypothesize that there exists a subset of robot interactions that are similar to the user and the model would perform well on this subset. We generate the user-specific test set by taking a portion of the original test set (20% of each strategy). This subset is determined based on the cluster membership of the user-specific model. For example, to generate the test set for user 1, we first determine what cluster generated from k-means the user falls into (for both success and fail). Then the test interactions associated with all users in that cluster are pulled and these interactions make up the test set for user 1. Note, this means that the original test set does not contain any interactions that are not associated with a specific user (i.e. only test interactions from GIE, GBE, and HSE are used).

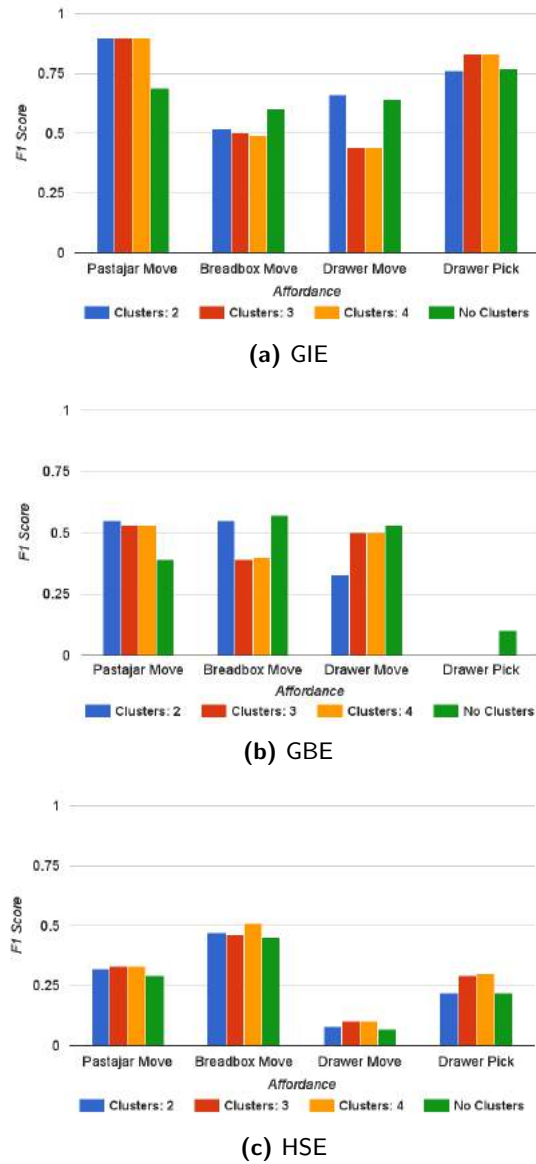


Figure 12. Aggregate F_1 values across users and affordances and cluster sizes.

The aggregate performance of each user-specific model for each object-action pair for all cluster sizes are shown in Figure 12. The average F_1 scores are shown in comparison to the original aggregate user-specific scores for GIE and GBE. Overall, with the exception of Pastajar-Move, selecting a subset of the test set based off of the user’s first demonstration was unable to improve the performance of the user-specific models. As expected, there does not exist a single cluster size that is favored across object-action pairs. We believe this is due to the inherent differences in each pair (i.e. each pair has its own subset of unique interactions).

Given that clustering did not work uniformly across all object-action pairs, we only present detailed user-specific scores for Pastajar-Move and Breadbox-Move to understand why some aggregates went up while others went down. This can be seen in Table 7. Overall the models generated from HSE did not drastically change. This is likely because HSE models have access to all demonstrations provided within a single user whereas GIE and GBE are limited to the first successful and first failed demonstration. This is amplified by users going out of their way to provide different and interesting demonstrations for each pair during the user study. Looking at Pastajar-Move, on the aggregate level, clustering the test set improves the overall performance. On an individual level, the models that were already performing well improved and the models that were not forming well either dropped or didn't change. This is seen for several users in Breadbox-Move as well. While it is not surprising that existing high-performing models improve when looking a subset of interactions that are most similar to it, it is surprising that the models that were performing poorly, performed even worse. This suggests that there is something else occurring within the effect space of the affordance that we are not capturing by clustering the EEF pose relative to the object. We explore this in Section 8.5.

8.4 Ratio of Success and Failure

Before we look into the observation space of the learned HMM, we look at one final metric presented in previous results (Table 3), the ratio between successful and unsuccessful interactions. Previously, we concluded that overall GBE and GIE had a more balanced set of positive vs. negative examples compared to HSE or self-exploration. We now take a look at this ratio on a per user basis. Results are summarized in Table 8.

Similar to Table 5, we only show the users that had enough positive and negative interactions to build a HMM. As mentioned earlier, we need a minimum of 3 examples of both positive and negative to build a model. For the models that could be built, we can look back at the detailed results in Table 5, and pull out specific users that performed well and performed poorly. For strategy GIE for the object-action pair Breadbox-Move, users 2, 6, 7, and 9 outperformed user 8. We can see that user 8 has a much higher percentage of positive executions than these other users. This is also true for the strategy GBE, where user 8 performs poorly compared to 2 and 6. This trend is consistent across the rest of the pairs as well: users with a particularly high number of success demonstrations have models that perform poorly.

Table 8. Percentage of Positive Interactions Per User For Strategies HSE, GIE, and GBE

Object-Action	User	HSE	GIE	GBE
Breadbox Move	User 1	0.50	0.11	0.17
	User 2	0.40	0.33	0.40
	User 3	0.33	0.11	0.17
	User 4	0.60	0.50	0.50
	User 5	0.50	0.10	0.14
	User 6	0.38	0.25	0.43
	User 7	0.50	0.29	0.67
	User 8	0.50	0.67	0.67
	User 9	0.50	0.44	0.20
	User 10	0.75	0.12	0.12
Pastajar Move	User 1	0.50	0.50	0.43
	User 2	0.60	0.20	0.67
	User 3	0.60	0.50	0.67
	User 4	0.75	N/A	N/A
	User 6	0.67	0.12	0.33
	User 7	0.67	0.12	0.20
	User 8	0.67	N/A	N/A
	User 10	0.75	N/A	N/A
Drawer Move	User 1	0.75	0.14	0.25
	User 2	0.60	0.11	0.14
	User 3	0.67	0.89	0.88
	User 4	0.67	N/A	N/A
	User 5	0.75	0.25	0.33
	User 8	0.50	0.11	0.14
	User 10	0.67	N/A	N/A
Drawer Pick	User 3	0.50	0.10	0.14
	User 4	0.33	0.33	0.67
	User 5	0.50	0.33	0.67
	User 6	0.50	0.12	0.20
	User 7	0.67	0.22	0.67
	User 8	0.50	0.14	0.20
	User 9	0.67	N/A	N/A
	User 10	0.25	0.75	0.67

Note: Darker shading equates to higher scores and N/A means no model could be built using the examples (recall we need a minimum of 3 examples each of success and failure to build models).

Digging deeper, we discovered that for some of the models that performed poorly, they essentially classify everything as not having the affordance, resulting in low recall and non-existent precision values. This indicates that when users have too many examples of successful interactions we cannot build a good HMM representing the expected effects of these successful interactions. We believe this happens due to the nature of the F/T data we are using to represent the effects of affordances. Recall that our exploration strategies are taking a couple of human demonstrations as seed examples and then varying these slightly in the end-effector space to get several new examples *around* the original ones. However, even though a slight change in position of the end-effector to the object may still

Table 7. User Specific Classification Scores for HSE, GIE, and GBE After Clustering for Breadbox-Move and Pastajar-Move

Object-Action	User	Human			Iconic			Boundary		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Breadbox Move	User 1	1.00	0.02	0.04	N/A	N/A	N/A	N/A	N/A	N/A
	User 2	0.62	0.33	0.43	0.42	0.54	0.47	0.48	0.83	0.61
	User 3	0.56	1.00	0.72	N/A	N/A	N/A	N/A	N/A	N/A
	User 4	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A	N/A
	User 5	1.00	0.18	0.31	N/A	N/A	N/A	N/A	N/A	N/A
	User 6	0.66	0.95	0.78	0.67	1.00	0.80	0.80	0.64	0.71
	User 7	0.86	1.00	0.92	0.88	0.96	0.92	N/A	N/A	N/A
	User 8	1.00	0.14	0.24	0.50	0.02	0.04	0.67	0.05	0.09
	User 9	0.68	1.00	0.81	0.76	0.43	0.55	N/A	N/A	N/A
Pastajar Move	User 1	1.00	0.15	0.27	0.83	0.77	0.80	0.85	0.85	0.85
	User 2	0.80	0.62	0.70	N/A	N/A	N/A	0.00	0.00	0.00
	User 3	0.00	0.00	0.00	0.37	1.00	0.54	1.00	0.15	0.27
	User 6	N/A	N/A	N/A	N/A	N/A	N/A	0.74	1.00	0.85

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

result in a successful interaction (e.g. the breadbox still moves), it can drastically change the signal seen on the F/T plate at the robot’s wrist. Thus, a dataset that includes a large number of successful examples is more likely to be highly varied. It is difficult to build a model that represents all of these different effects at once. On the other hand datasets with a limited number of success interactions, are more likely to only include a single *way* of achieving the affordance, that is more consistent in the sensory space and easier to model.

8.5 Affordance Effect Space

In Section 8.3 and Section 8.4 we have seen evidence that the effect space of the object-action pairs play large role in the quality of models built. To understand why, we take a deeper look at the multivariate Gaussian distribution that represent the observation space of the learned HMMs. We choose to look at the last state because achieving an affordance is highly dependent on the final pose of the interaction. This also allows us to focus on a specific snapshot in time where the effect of the affordance is most likely to have occurred as opposed to the entire trajectory of the interaction.

Recall we have an 18-value feature vector that represents the observation state-space of the HMM. To focus on the specific dimensions that have the greatest change in the effect space, we perform principal component analysis (PCA) on the observation space. Concretely, we compute the principal components of the set of all mean values of the multivariate Gaussian distribution for all of the generated HMMs. This is done specific to the set of successful HMMs and unsuccessful HMMs. This results in two transforms - one for each set of means. We selected the

top three principal components, which account for 99.9% of variance for both sets of HMMs (successful and unsuccessful). We compare this reduced set of components from user models that produced good object-action pairs vs. those who did not. As a reminder, successful HMMs were HMMs generated from robot trials that successfully found the affordance whereas unsuccessful HMMs were generated from trials that did not. Furthermore, good user models (or good HMMs) are models that performed well at classifying unseen interactions whereas poor user models (or poor HMMs) did not score well in classifying new unseen interactions.

Figure 13 shows the different principal component values for each individual model for the object-action pair, Drawer-Move. The green bars show the users that had good HMMs and the red bars show users with HMMs that performed poorly. While the means do differ, it is unclear if this difference is enough to account for the poor performance. As a result we do not show the rest of the object-action pairs. When we compare the variance of the means of the set of good user models and poor user models (in Figure 14) there is a clear difference in the variance of the HMMs that do well vs. poorly. Aside from Pastajar-Move, the poorly performing HMM models have a much larger variance in the observation state of the HMMs. The high performing models clearly had greater consistency in the observational values as opposed to those from the poorly trained models. This supports our hypothesis in Section 8.4 that the HMMs were having difficulty capturing a larger variety of demonstrations, whereas those trained with a smaller set converged to a specific and consistent observational state-space for the HMM. For Pastajar-Move, we believe the difference

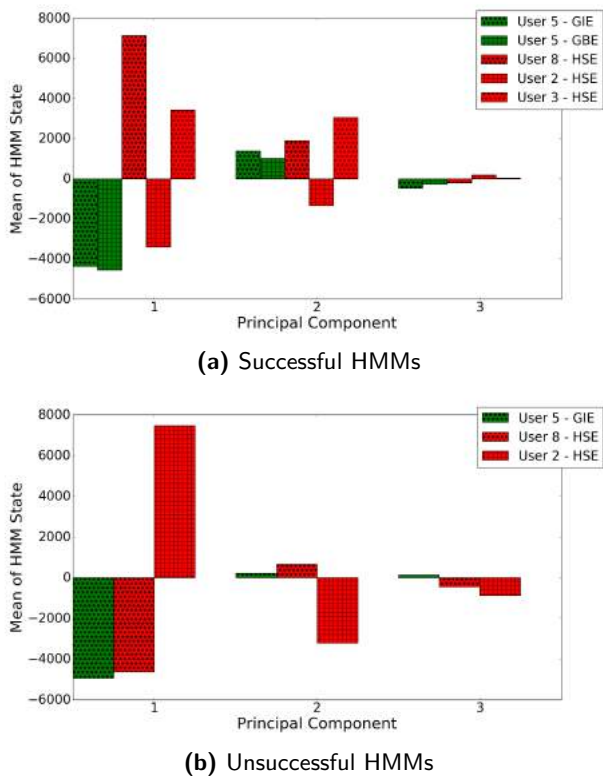


Figure 13. Displayed are the top three principal components of the final state of each user specific model for the object-action pair Drawer-Move. The values are separated in color by user models that performed well (green) and user models that performed poorly (red).

in variance is not high because none of the Pastajar-Move models themselves performed as well as the other object-action pairs.

8.6 Qualitative Observations

We presented qualitative observations from the user study based on anecdotes and common threads from a single open-ended question survey administered at the end of the user study in the previous results. Some of the findings are repeated here because they are relevant for providing insight on why differences arose between users when teaching affordances.

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 pairs like Pastajar-Move, users tended to get bored quickly and many wanted to move onto the next pair before the allotted time. The bread box was particularly favored because it was simple enough

to provide many examples of success and failure, but “difficult” in comparison to the pasta jar.

Users’ dislike of failure resulted in an expressed preference to not provide examples of failure when teaching affordances. Not surprisingly, people dislike failure. However, it was surprising that users preferred not to provide example of failure even when instructed that providing negative examples of an affordance could be beneficial. 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.

Another common thread in the reported teaching strategy was the focus on providing “different ways to achieve the same outcome” and “show[ing] the affordance in multiple ways”. Half of the users reported using this method in their teaching strategy. This thread is interesting because it could account of the difference in variance across users when showing examples of interactions.

While users focused on providing varied and different interactions for the same effect, The vast majority of users did so by changing the robot’s action as opposed to the environment. Going into the study, the authors had believed that users would take advantage of the fact that they could modify the environment as opposed to reteaching actions to provide different interactions. For example, showing a negative example of Pastajar-Move could be achieved but just putting the jar out of reach and this was in fact demonstrated to all participants before the study began as part of the tutorial on affordances. Even with this priming, users did not use this strategy, with only one user mentioning that they would “slowly modify the environment by repositioning the object”. For the users that did use reuse an action, this generated very similar interactions since the same action is executed with a slightly different object position.

9 Discussion

In this work, we provided an in-depth comparison of three different approaches to affordance learning: self-exploration, human-supervised exploration, and a *combined* human-guided approach defined as self-exploration biased by information provided from human teachers. Prior results showed that a combined

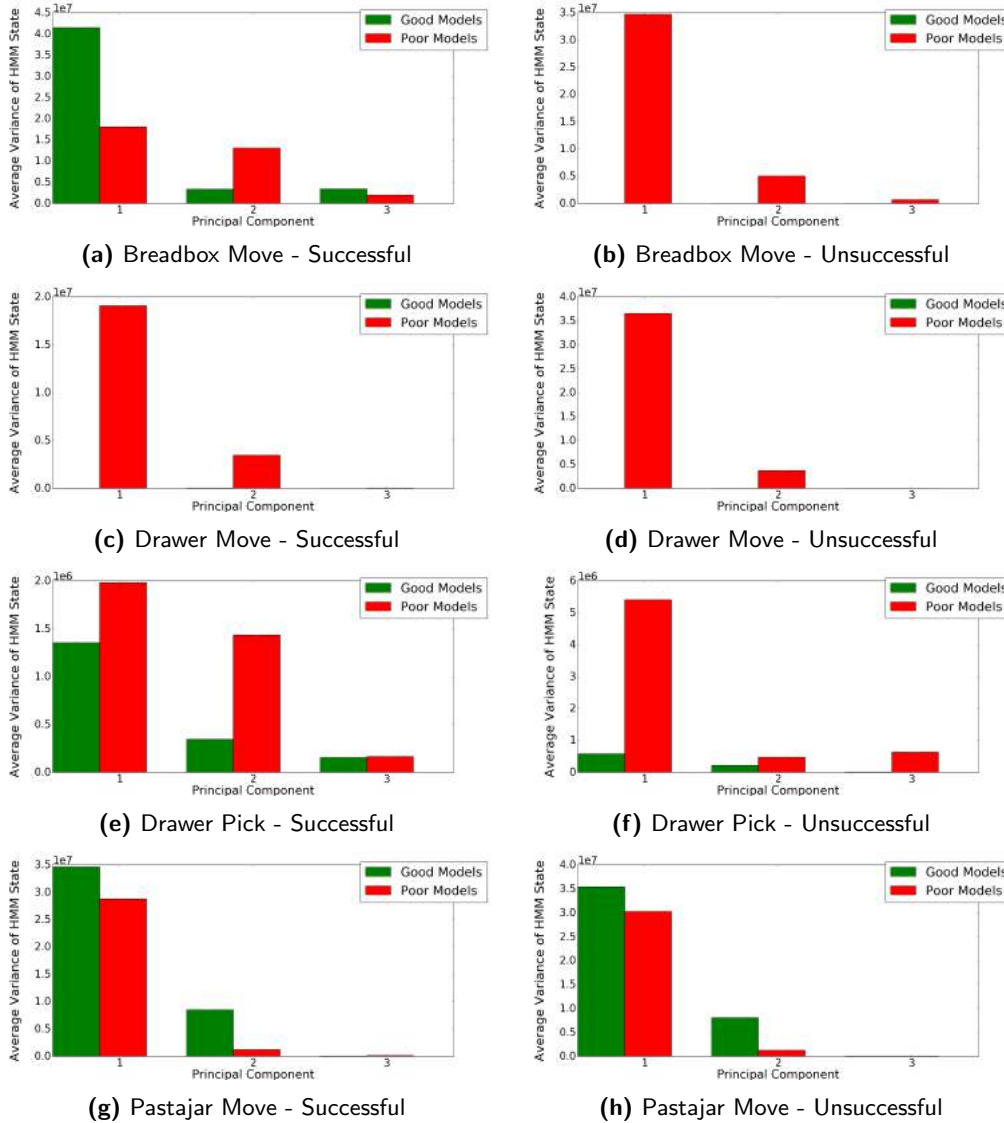


Figure 14. Shown are the comparisons of variance across the mean HMM values between good user models and poor performing user models for all four object-action pairs. The means are separated by successful HMMs and unsuccessful HMMs. Poor performing user models overall have higher variance than high performing user models.

approach, GIE, can learn affordance models on par with those generated from exhaustive SE, but using an order of magnitude fewer interactions with the object. The results of an individual analysis of each user-specific model provide several interesting pieces of insight that can guide future work for learning affordances from naive users.

At the heart of the analysis presented in this work is characterizing how users differ when teaching affordances and how that impacts the performance of their resulting models. To characterize the difference between users, we looked at clustering individuals based off of their demonstrations. However, while

clustering the users based off of the EEF pose clearly showed that users provide very different approaches to teaching an affordance, merely clustering users based off of this was too simplistic to improve performance across all affordances. More importantly, we discovered that the impact of positive vs. negative interactions plays a large role in the performance of the users. We show that having many successful interactions causes the performance of individual models to decrease. Given that end-users do not intuitively provide many examples of failure, this suggests we need to explicitly ask users to provide more examples of failure.

While our hypothesis suggests that we should gather from users very similar successful interactions to model the affordance with a low amount of data due to the impact of F/T sensing, this is misleading because what we really need to learn are all of the different ways it feels like to find the affordance. This suggests that not only do we need to gather varied interactions, but we also need to develop new modeling techniques that can capture the high variability due to F/T sensing (whether that be with a different representation where we model F/T felt with respect to the object or generate a library of models that encompasses this variance).

10 Future Work

While the experimental results and analysis of this work provide concrete guidelines for generating models from naive users for object-action pair learning, there remains the open question on how we can use these models for the ultimate goal of task execution using affordances. Assuming a robot is given a task plan that requires a series of objects with specific affordances, the robot needs to locate objects in the room with the candidate affordances and test these objects to see if they can be used to perform the task. Currently, this system only addresses the first half of the equation where we are answering the question of “how can the robot learn about the object efficiently”? The second half requires the robot to apply an existing learned model to a new object with a similar affordance and test the object for that affordance. For this to occur, the robot needs to generate actions from its existing models.

While this work does not look into generation of trajectories from learned models, future work will look at this specific problem. In particular, as mentioned in Section 6, we chose to model affordances using HMMs because HMMs are generative and give us the ability to sample from its states the parameters necessary to create new trajectories. While the HMMs learned in this work show great promise at modeling object-action pairs, it is unclear whether these HMMs, which have been optimized for classification, are suitable for action generation. Classification favors models that excel at finding the boundary cases of an object-action pair. However, for the robot to generate primitive actions, it will likely need to generate actions that are prototypical interactions and not ones nears the boundary.

11 Conclusion

Our work uses a *Guided Exploration* approach to affordance learning with human teachers. A previously reported experiment compared the impact of three types of exploration on affordance learning performance: (1) learning through self-exploration, (2) learning from supervised examples provided by 10 naïve users, and (3) a combined approach of self-exploration biased by user input. That work analyzed aggregate performance of the teachers and showed that a combined approach is the most efficient and successful. In this article we extend prior work by providing more analysis of the exploration algorithms and in particular, we focus on the variance seen across teachers in this experiment.

After considering several alternatives, we conclude that individuals with seeds that lead to a relatively limited set of ways to interact with the object resulted in data that achieved a model with consistently good recognition capabilities. By contrast, individuals whose guided-exploration resulted in a large number of different ways to successfully interact with the object had poor model performance. This points out important future work needed in novel techniques for efficiently modeling the multi-modal sensory information inherent in object affordances.

Acknowledgements

We would like to thank Tesca Fitzgerald for help collecting data used for the analysis of this paper. We’d also like to thank Dr. Sonia Chernova for her insight on analyzing the individual cluster results.

Funding

This work is funded through Office of Naval Research Grant N000141410120.

References

- Vivian Chu, Baris Akgun, and Andrea L. Thomaz. Learning haptic affordances from demonstration and human-guided exploration. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 119–125, April 2016a.
- Vivian Chu, Tesca Fitzgerald, and Andrea L. Thomaz. Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 221–228, March 2016b.
- Bruno Castro da Silva, George Konidaris, and Andrew G. Barto. Active learning of parameterized skills. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1737–1745, 2014.

- Mehmet Remzi Dogar, Emre Ugur, Erol Sahin, and Maya Cakmak. Using learned affordances for robotic behavior development. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3802–3807, May 2008. doi: 10.1109/ROBOT.2008.4543794.
- Paul Fitzpatrick, Giorgio Metta, Lorenzo Natale, Sajit Rao, and Giulio Sandini. Learning about objects through action - initial steps towards artificial cognition. In *International Conference on Robotics and Automation (ICRA)*, pages 3140–3145, Sept 2003.
- James J. Gibson. The concept of affordances. *Perceiving, acting, and knowing*, pages 67–82, 1977.
- Tucker Hermans, Fuxin Li, James M Reh, and Aaron F Bobick. Learning stable pushing locations. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7, Aug 2013a.
- Tucker Hermans, James M Reh, and Aaron F Bobick. Decoupling behavior, perception, and control for autonomous learning of affordances. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4989–4996, May 2013b.
- Serena Ivaldi, Natalia Lyubova, Damien G  rardeaux-Viret, Alain Droniou, Salvatore M Anzalone, Mohamed Chetouani, David Filliat, and Olivier Sigaud. Perception and human interaction for developmental learning of objects and affordances. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 248–254, Nov 2012.
- Serena Ivaldi, Sao Mai Nguyen, Natalia Lyubova, Alain Droniou, Vincent Padois, David Filliat, Pierre-Yves Oudeyer, and Olivier Sigaud. Object learning through active exploration. *Autonomous Mental Development, IEEE Transactions on*, 6(1):56–72, March 2014. ISSN 1943-0604.
- Dov Katz, Arun Venkatraman, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Anthony (Tony) Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In *RSS Berlin*, June 2013.
- Jens Kober, Andreas Wilhelm, Erhan Oztop, and Jan Peters. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, 33(4):361–379, 2012.
- Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *RSS, Berlin*, June 2013.
- Norbert Kr  ger, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin W  rg  tter, Ale   Ude, Tamim Asfour, Dirk Kraft, Damir Omr  en, Alejandro Agostini, and R  diger Dillmann. Object-action complexes: Grounded abstractions of sensory-motor processes. *Robotics and Autonomous Systems*, 59(10):740–757, 2011.
- Bogdan Moldovan, Plinio Moreno, Martijn van Otterlo, Jos   Santos-Victor, and Luc De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4373–4378, May 2012.
- Luis Montesano, Manuel Lopes, Alexandre Bernardino, and Jos   Santos-Victor. Learning object affordances: From sensory-motor coordination to imitation. *Transactions on Robotics*, 24(1):15–26, 2008.
- Sao Mai Nguyen and Pierre-Yves Oudeyer. Socially guided intrinsic motivation for robot learning of motor skills. *Autonomous Robots*, 36(3):273–294, 2014. ISSN 0929-5593.
- Pierre-Yves Oudeyer, Fr  dric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286, April 2007. ISSN 1089-778X.
- 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*, 12:2825–2830, 2011.
- Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.
- Erol   ahin, Maya   akmak, Mehmet R Do  ar, Emre U  ur, and G  kt  rk    oluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.
- Juergen Schmidhuber. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pages 1458–1463 vol.2, Nov 1991.
- Alexander Stoytchev. Behavior-grounded representation of tool affordances. In *International Conference on Robotics and Automation (ICRA)*, pages 3060–3065, April 2005.
- Andrea L Thomaz and Maya Cakmak. Learning about objects with human teachers. In *International Conference on Human Robot Interaction (HRI)*, pages 15–22, 2009.
- Karthik Mahesh Varadarajan and Markus Vincze. Afrob: The affordance network ontology for robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1343–1350, Oct 2012.

Christopher M Vigorito and Andrew G Barto. Intrinsically motivated hierarchical skill learning in structured environments. *Autonomous Mental Development, IEEE Transactions on*, 2(2):132–143, June 2010. ISSN 1943-0604.