To be published in 2014 IEEE International Conference on Robotics and Automation

Accelerating Imitation Learning through Crowdsourcing

Michael Jae-Yoon Chung, Maxwell Forbes, Maya Cakmak and Rajesh P. N. Rao*

***Abstract*— Although imitation learning is a powerful technique for robot learning and knowledge acquisition from naïve human users, it often suffers from the need for expensive human demonstrations. In some cases the robot has an insufficient number of useful demonstrations, while in others its learning ability is limited by the number of users it directly interacts with. We propose an approach that overcomes these shortcomings by using crowdsourcing to collect a wider variety of examples from a large pool of human demonstrators online. We present a new goal-based imitation learning framework which utilizes crowdsourcing as a major source of human demonstration data. We demonstrate the effectiveness of our approach experimentally on a scenario where the robot learns to build 2D object models on a table from basic building blocks using knowledge gained from locals and online crowd workers. In addition, we show how the robot can use this knowledge to support human-robot collaboration tasks such as goal inference through object-part classification and missing-part prediction. We report results from a user study involving fourteen local demonstrators and hundreds of crowd workers on 16 different model building tasks.**

I. Introduction

Traditional approaches to imitation learning are limited by the need for expensive human demonstrations. The research community has been addressing, or even bypassing this problem in two major ways: (a) utilizing a small quantity of good quality human demonstrations with complex learning algorithms [1]–[5], (b) simplifying the problem domain [6], [7], or (c) making use of simulators to make experiments inexpensive [8]. However, the fundamental problem of learning from small datasets still remains. The number of useful demonstrations of a task may be too small for the robot to achieve the desired goal and the robot's ability to acquire new skills and knowledge is limited by the number of users it directly interacts with. This motivates the use of crowdsourcing: the robot can acquire a variety of examples from a large pool of humans online.

Another common drawback of many imitation learning approaches is the failure to imitate the "goal." Instead, these approaches involve following action trajectories given by a human in the same space as the robot's actuator space. A trajectory-based approach to imitation fails in cases where the physical limitations of the robot's actuators make it difficult or impossible for the robot to follow the demonstrated trajectory. However, in many instances, it is not the trajectory that is important but the goal of the action. An alternative approach, referred to as goal-based imitation [6], acknowledges the fact that robots often have different actuators than humans but may still be able to achieve the same goals as a human demonstrator, albeit using different means.

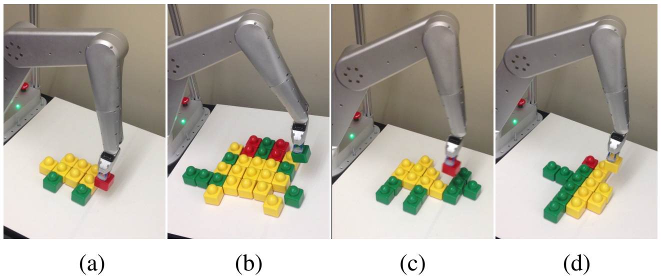


Fig. 1. **The Robot and the Object Building Task**. (a)-(d) The robot Gambit constructing four different user-generated objects obtained from crowdsourcing for the object class "Turtle" (see Fig. 6). Different parts of the object are represented using different colored Primo blocks. Gambit successfully constructed the objects in (a) and (d); the resulting constructions are shown in Fig. 6 (a) and (d). Gambit was unable to successfully build the objects in (b) and (c); such objects are estimated by our heuristic task-difficulty score (see text) to be "hard" compared to objects (a) and (d).

In this paper, we propose a framework for using crowdsourcing to accelerate goal-based imitation learning. With the use of crowdsourcing, our approach overcomes a fundamental limitation of traditional imitation learning approaches: the scarcity of data. Focusing on a simple object building task, we show how crowdsourcing can allow a robot to augment an initial human demonstration with a richer set of examples from humans online, allowing the robot to find new ways of achieving a goal that is difficult to achieve with data from a single user. We present results illustrating how the crowdsourced data can support not only imitation, but also related human-robot interaction tasks, such as goal inference based on classifying objects and their parts and predicting missing parts. In addition, we test the full system with a user study involving eleven non-expert users.

II. Related Work

Our work is closely related to two areas of research in robotics. The first, learning from demonstration, has a long history, going back to the work of Kuniyoshi, Schaal, and others [9]–[11]. Recent advances in the field have led to a number of impressive demonstrations of robotic skill learning from humans [1]–[5]. The emphasis in many of these previous approaches was on learning policies for approximating the action trajectories demonstrated by a human teacher. On the other hand, goal-based imitation ([6]–[8]) focuses on the ability to infer the intention of a human teacher. This provides a powerful framework for imitation

*This work is supported by the Office of Naval Research (ONR) grant N000141310817 and National Science Foundation (NSF) grant IIS-1318733. Michael Jae-Yoon Chung, Maxwell Forbes, Maya Cakmak, and Rajesh P.N. Rao are with the Computer Science & Engineering Department, University of Washington, Seattle, WA 98195-2350, USA {mjyc,mbforbes,mcakmak,rao}@cs.washington.edu

that allows intuitive ways of transferring task information from humans to robots, independent of their embodiment. This enables other human-robot collaboration tasks at no additional computational costs.

The second related area of research, crowdsourcing, has only recently received attention from the robotics community. Sorokin and colleagues [12] used Amazon Mechanical Turk to tackle the problem of robotic grasping of novel objects: human workers provided object outlines in images, grouped objects by type, and rated the extracted 3D mesh models prior to computing grasps for each object. Crick and colleagues [13] demonstrated the utility of crowdsourcing for teaching a robot to navigate a maze when users are presented with information that is limited to the robot's perception of the world. Breazeal and colleagues [14] used crowdsourcing with an online game to collect large-scale data on unscripted interactions between human players: they used this data to generate behaviors for a real robot in an interactive task at a museum. Emeli [15] investigated the use of a Twitter account for a robot so that the robot's followers can label videos of human actions for action classification. Our approach builds on these past efforts; similar to [13], [14], we use crowdsourcing as a means of gathering human demonstrations. However, unlike these approaches, our work focuses on goal-based imitation.

III. System Components

A. Robot and Object Building Task

We use the Gambit robot arm-and-gripper (Fig. 1) designed at Intel Labs Seattle. Gambit is well-suited to small-object tabletop manipulation tasks and has previously been shown to perform well in tasks with humans in the loop [16].

The task we use to illustrate our crowdsourcing-based approach is a simple imitation learning task where the robot must learn to build 2D object models on a tabletop from human demonstrations, e.g., Fig. 1. We use Primo [17]—the larger cousin of Lego—as building blocks, and we only use 1×1 unit blocks for all of the tasks[1]. The robot has a basic set of primitive actions; *i.e.* Gambit can pick and place blocks at pre-specified locations[2].

Even though Gambit is precise enough to play a game of chess [16], building 2D object models turns out to be a nontrivial task because humans typically build models in which blocks abut each other. This close block-to-block proximity means that for the robot, the difficulty of the task increases with the number of blocks in the model, e.g. Fig. 1 (b). Additionally, using inverse kinematics to place blocks on the table results in less precision near the boundaries of the robot's workspace than near the center.

For sensing the current state of objects on the table, *i.e.* perceiving human demonstrations, we use a Microsoft Kinect. The Kinect is mounted on the base frame of Gambit and looks down on the table surface. The robot takes as input the stream of RGB and depth images from the Kinect and first segments out the background and the hand of the human providing the Primo blocks. The remaining pixels are then used to determine the state of objects on the table using a simple heuristic based on a centroid and predetermined grid-positions.

B. Crowdsourcing

We use Amazon's Mechanical Turk service (AMT) for crowdsourcing. AMT is a market place for macro-tasks. Tasks which require human intelligence are posted with payments in the range of \$0.01 to \$0.2. Common tasks include categorization, audio transcription, and content filtering.

AMT has the following pros and cons: due to its publicity, AMT is very scalable; thousands of workers are online to work on hundreds of thousands of tasks daily. AMT also covers a wide range of populations. This can be beneficial for gathering data from or studying behaviors of a variety of users. Furthermore, AMT provides simple tools to control data collection and interact with workers, such as giving bonuses for good work, rejecting jobs, and promoting or blocking workers.

AMT however is not designed for complex tasks. Even though some studies [18], [19] have demonstrated the use of AMT for tackling challenging jobs, it is unclear whether these results generalize to the robotics domain. The AMT tools for managing workers and tasks are often not sufficient to block poor quality workers and spammers. Principled methods for improving the quality of results from AMT are currently being explored [20], but their applicability to the robotics domain remains to be explored.

C. Web Interface

Fig. 2 illustrates the two types of web interfaces used for data collection on AMT.[3] The first web interface (Fig. 2(a)) focuses on collecting 2D object model data. The first page of the interface contains a concise explanation of what the robot is trying to achieve with related explanatory figures.

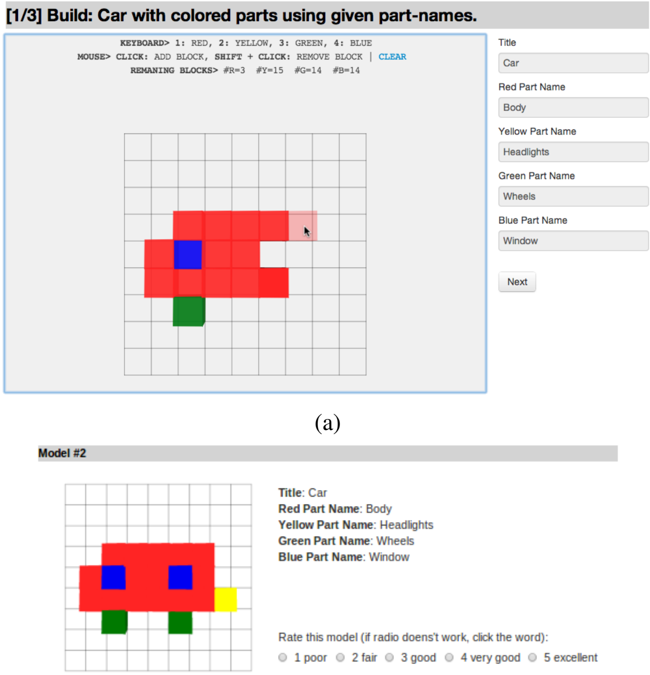
Upon agreeing to participate, the online worker is directed to the data collection page. The data collection page has a header line with gray background that displays how to build a model and what type of model to build. The online worker can use blocks to create 2D models on a 9×9 grid, corresponding to the robot's tabletop working area, by simply clicking the grid boxes. Four different colored blocks—red, yellow, green, blue—are available for use in limited quantities (15 for each). In the instructions, we encourage the worker to use a different color for each part of the model they are building. For some tasks, the worker can provide their own object title and part names by filling in the blank fields on the right side.

The second web interface (Fig. 2(b)) is used for obtaining worker "satisfaction" ratings of 2D object models. When the worker visits the website, 10 models as in Fig. 2 are

[1] Each block has dimensions 4.6cm × 4.6cm × 5.8cm and the tabletop workspace is 41.4cm × 41.4cm.

[2] Primo blocks were handed to Gambit's gripper one at a time to avoid manipulation complexities.

[3] Web interfaces are available at `http://homes.cs.washington.edu/~mjyc/collabot/project-site/index.html`



[1/3] Build: Car with colored parts using given part-names.

KEYBOARD> 1: RED, 2: YELLOW, 3: GREEN, 4: BLUE
MOUSE> CLICK: ADD BLOCK, SHIFT + CLICK: REMOVE BLOCK | CLEAR
REMANING BLOCKS> #R=3 #Y=15 #G=14 #B=14

Title: Car
Red Part Name: Body
Yellow Part Name: Headlights
Green Part Name: Wheels
Blue Part Name: Window

(a)

Model #2

Title: Car
Red Part Name: Body
Yellow Part Name: Headlights
Green Part Name: Wheels
Blue Part Name: Window

Rate this model (if radio doens't work, click the word):
○ 1 poor ○ 2 fair ○ 3 good ○ 4 very good ○ 5 excellent

(b)

Fig. 2. **Web Interfaces for Data collection**. (a) Web interface for 2D object model building. (b) Web interface for rating 2D object models.

displayed. The worker then rates the quality of each model from 1 (poor) to 5 (excellent).

IV. Methods

Our approach has three main steps: (1) human demonstration collection, (2) goal-model learning and goal-inference and (3) goal-based imitation.

A. Data Collection

Demonstrations for goal-based imitation are often high-level; in such cases, it is possible to abstract the task and still collect reasonable demonstrations. In our system, humans provide demonstrations either directly with the Primo blocks, or through a simulated task environment as in Fig. 2 (a). The simulated interface is parameterized so that the robot can post requests based on the input it receives from users. For example, our interface takes a parameter for deciding the source of data, *i.e.* local demonstrators vs. crowd, and takes another parameter for deciding type of instructions to be displayed to the demonstrator.

B. Graphical Models

For probabilistic goal-modeling and goal-inference we use a generative graphical model. Let X be the set of human demonstrations and C be the set of class names. Then let K be the number of sets of part names, and P_i be the i^{th} set of part names. In the context of the object building task, $x \in X$ is an object model in 9×9 colored pixel grid submitted by a demonstrator. We represent x as a 81-dimensional vector where each element can take one of five values (one for each of the four block colors, and one denoting an empty square). $c \in C$ is the name of an object model, such as "House." $P_1 = P_r, P_2 = P_y, P_3 = P_g, P_4 = P_b$ corresponds to the set of red, yellow, green and blue part names and $p_r \in P_r$ is a red part name, such as "Body."

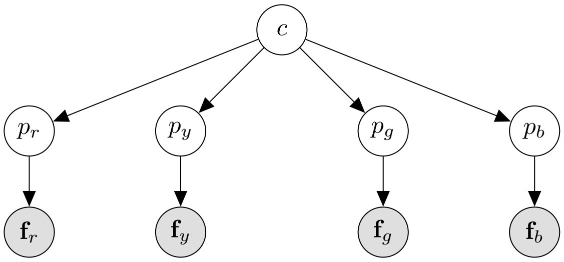


Fig. 3. **Graphical Models**. This model exploits the structure of the object-parts problem (see text for details).

Feature Extraction: To provide more informative data to the graphical model, one can extract features from x. We define M as the number of features and the i^{th} part feature vector as

$$\mathbf{f}_i = \left[\phi_i^1(x) \cdots \phi_i^M(x)\right]$$

where $\phi_i^1(x)$ is first feature transformation function for the i^{th} part. For the object building task, we build colored sliding window shape detectors for grid sizes of 1×1, 2×2, and 3×3. A single shape detector of grid size $n \times n$ can be thought of as a window with a particular pattern of a single color. That window is then placed on all possible n by n sections of the 9×9 raw pixel grid. Each time there is an exact match, that feature's count is increased by one.

Recall that workers are instructed to construct object parts from one of the four possible colors (red, yellow, green, or blue). Our 1×1 grid size shape detector is equivalent to a pixel color counter. For 2×2 and 3×3 sized grids, we exhaustively generated all possible patterns for each color (16 for a 2×2 grid and 512 for a 3×3 grid). Therefore, a total of $M = 1 + 16 + 512 = 529$ features per each color are available. We represent them as $\mathbf{f}_r, \mathbf{f}_y, \mathbf{f}_g, \mathbf{f}_b$. Since our features are independent of the parts, $\phi_r^j = \phi_y^j = \phi_g^j = \phi_b^j$ is true. In practice, we prune the unused features; from $M \times K = 529 = 2116$ possible features, only 1109 actually occurred in our dataset, and are used for the training and testing of our models.

Factorization: For a multiple part assembly task, such as the object model building task, we use a graphical model given by the factorization as shown in Fig. 3. This can be expressed as

$$P(c, p_{1:K}, \mathbf{f}_{1:K}) \quad \propto \quad P(c) \prod_{i=1}^{K} P(\mathbf{f}_i \mid p_i) P(p_i \mid c).$$

$P(\mathbf{f}_i \mid p_i)$ is the probability of seeing features $\mathbf{f}_i$ given part name p_i. Since the features are vectors of counts (similar to a Naïve Bayes model), we use a multinomial distribution to model $P(\mathbf{f}_r \mid p_r), P(\mathbf{f}_y \mid p_y), P(\mathbf{f}_g \mid p_g)$ and $P(\mathbf{f}_b \mid p_b)$. $P(p_i \mid c)$ is the probability of seeing part label p_i given the object class name c and $P(c)$ is a prior over object class names. We model both $P(p_i \mid c)$ and $P(c)$ with categorical distributions.

Learning: The problem of learning from human demonstration becomes a parameter learning problem of the graphical model. We assume fully labeled data is always available. For example, during data collection, the system either specifies the object model name and four part names, or asks online workers to provide labels, e.g. the four part names; we use only the system-specified data. In all the experiments with fully labeled data, we use maximum likelihood estimation to learn the parameters of the graphical model (Fig. 3).

Inference: We equate the "goal" of the demonstrator with the unobserved variables, which are c and $p_{1:K}$. In our case, we equate the "goal" of the demonstrator with the object class name and its four part names s/he constructed with the blocks. The goal inference problem then becomes: given a particular constructed model, which object does the construction correspond to and what are its part names? With the graphical model (Fig. 3), we can compute marginals and most probable explanations (MPEs) of $P(c \mid \mathbf{f}_{1:K})$ (object class name inference), and $P(p_i \mid \mathbf{f}_{1:K})$ (part name inference). Taking advantage of our model's tree structure, we use a belief propagation algorithm to compute marginals and MPEs efficiently.

C. Imitation as Search

We model the problem of goal-based imitation as a problem of searching for a "good" demonstration to imitate from collected human demonstrations. Let X_c be the set of instances that belong to object class c. We assume that the robot has an single, identified user-goal, e.g. the mode of $P(c \mid \mathbf{f}_{1:k})$. Then, the search can be expressed as

$$x^* = \operatorname*{argmin}_{x \in X_c} \; score(x). \tag{1}$$

In all of our experiments, the search space is relatively small $|X_c| \leq 100$, so we always exhaustively search through the entire search space.

The "goodness" of an instance is measured by score functions. We use three score functions:

$$\begin{aligned} d(x) &= w_n d_n(x) + w_b d_b(x) && (2)\\ sd(x) &= w_d d(x) + w_s s(x) && (3)\\ vd(x) &= w_d d(x) + w_v v(x,z) && (4) \end{aligned}$$

The task-difficulty score function, Eq. 2, approximates the difficulty of building given instance object model x based on two sources $d_n(x)$ and $d_b(x)$; $d_n(x)$ measures the difficulty related to the number of blocks and $d_b(x)$ measures the difficulty related to proximity to boundary. We set weights w_n and w_b empirically.

However, considering only task-difficulty may not yield good imitation results; the easiest-to-build model may be too simple and therefore can be unpleasing or unconvincing. For this purpose, we propose a satisfaction score $s(x)$. This function is created from averaging and normalizing[4] the crowd satisfaction data collected using the second web interface introduced in Section III-C. The new combined score, Eq. 3, incorporates both task-difficulty and crowd-satisfaction. We set weights to $w_d = 0.5$ and $w_s = 0.5$, but they can be tweaked based on the system designer's preference.

Another interesting score for imitation can be an object model's "visual-distance" with a given object model from the user. We measure this distance, $v(x, z)$, based on the counting number of different blocks between an object model x and the user given demonstration z. To make our distance metric robust against translations, we use the minimum distance from all possible distances measures with different offsets between x and z. We propose another score function, Eq. 4, by combining the visual-distance and task-difficulty scores. Similar to the previous score function, we set the weights empirically ($w_d = 0.25$ and $w_v = 0.75$).

V. Experiments and Results

A. Example Scenario

To demonstrate how crowdsourcing can accelerate goal-based imitation learning, we conducted a single user experiment with the following scenario.[5]

(1) **User** builds a 2D object model and labels its class name and four part names.
(2) **Robot** collects data by posting jobs on AMT.
(3) **Robot** learns the graphical model using collected data.
(4) **Robot** infers the user's goal from the provided object model in step (1).
(5) **Robot** finds three candidate object models using task-difficulty, satisfaction combined and visual-distance combined scores then asks the user to select one.
(6) **User** makes a selection.
(7) **Robot** builds the selected object model from step (6).

Fig. 4 shows the progression of the scenario. First, the user provided a physical object model of "Tree" as shown in Fig. 4(a). The physical model was then converted to a virtual model, Fig. 4(b), via sensing. The user labeled the class name as "Tree" and labeled the four part names as "fruit" (red), "trunk" (yellow) , "leaves" (green) and "" (empty—blue).

Once the user finished the demonstration, the robot posted jobs on AMT asking for demonstrations of "Tree" with the four specified part names given by the user. After one day of data collection, the robot learned the graphical model parameters from the collected data, which includes the data collected for the other eight classes described in Section V-B. The eight classes were added because otherwise the goal-inference task becomes trivial. Then the robot inferred the goal, a class name, and four part names from the user-provided "Tree" object model. Fig. 4(c) shows the marginal distributions of the four part names and the class name. There

[4] We also reverse the direction of $s(x)$ (so that a lower score is better) to make it work with Eq. 1

[5] Visualization of the data collected for the experiments is available at `http://homes.cs.washington.edu/~mjyc/collabot/project-site/index.html`.

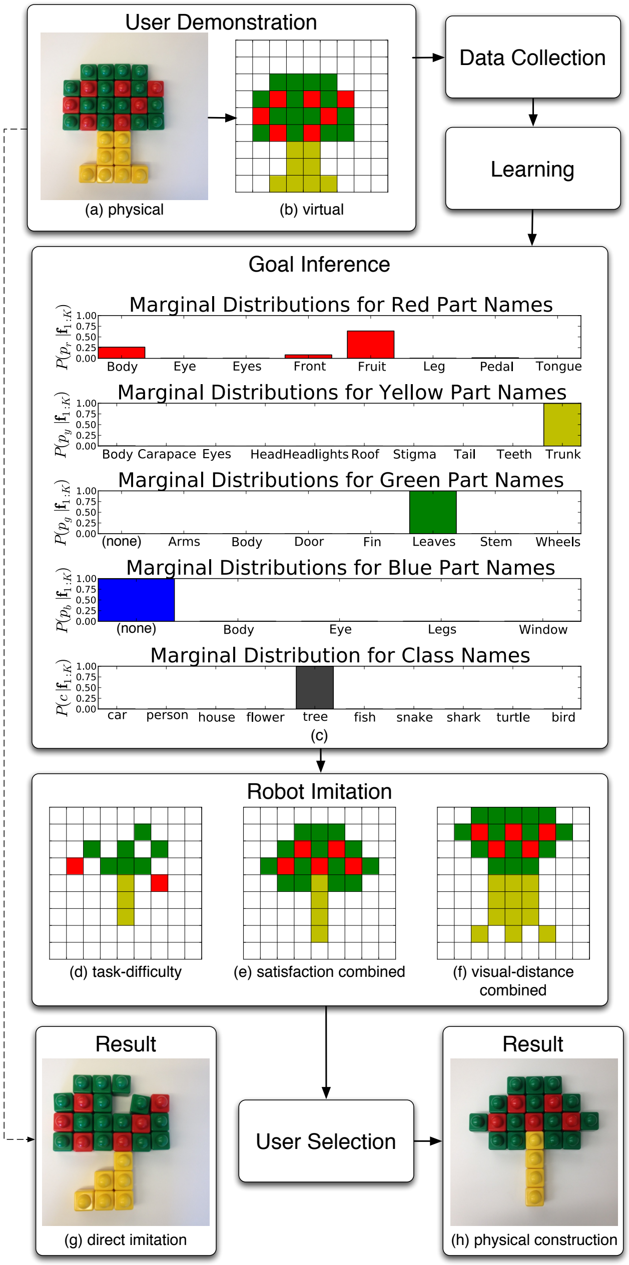


Fig. 4. **Example Imitation Learning Scenario**. Intermediate results from following the example imitation learning scenario (see the main text) are visualized. The results can be seen in order by following solid arrows. The dotted arrow show the scenario of directly imitating the user's original demonsration (a)&(b), which we compared our method against.

is some uncertainty about the red part name, but the other four part names have marginal distributions with almost no uncertainties. Because of this, the marginal distribution for the class name is also quite certain.

With the most probable class name, the robot searched for candidate models using three different score functions: Eq. 2, Eq. 3 and Eq. 4. The three candidates found are shown in Fig. 4(d),(e),(f). The robot asked the user to make a selection and the user selected (e). Following the request of the user, the robot successfully constructed the selected model, Fig. 4(e), as shown in Fig. 4(h). We also commanded the robot to build the original object model demonstrated by the user, (g), but the robot was not be able to successfully build it due to the "hardness" (task-difficulty) of the object.

B. Experiment 1: Goal-Based Imitation

In our first experiment, three local participants[6] gave demonstrations of eight objects: a car, person, house, flower, fish, snake, turtle and chick (small bird). We evaluated the object models provided by the robot as the results of our approach with the following variations.

1) *Source of Demonstrations:* The robot could collect demonstrations either by requesting demonstrations from local demonstrators, or by posting jobs to Mechanical Turk (online demonstrators). When the robot was collecting data from local demonstrators, it asked for one example per class from each demonstrator. When the robot was collecting data from online demonstrators, it posted 100 jobs, which corresponded to 100 examples per class. Once the data had been collected, the robot again used Mechanical Turk to rate the obtained models. As mentioned in Section III-C, we have the crowd rate ten models at the same time, where all models are drawn from the same class. The rating data were then used for building the crowd-satisfaction score function for the crowdsourced data.
2) *Score Functions:* The robot used three different methods of imitation which were based on using three different score functions described in Section IV-C.

We considered four experimental conditions: (a) collecting data from local demonstrators and using the task-difficulty score, which we denoted as `L`, (b) collecting data from the crowd and using the task-difficulty score, which we denoted as `C`, (c) collecting data from the crowd and using the satisfaction combined score, which we denoted as `CS`, and (d) data collected from the crowd and using the visual-distance combined score, which we denote as `CV`. We conducted experiments similar to the scenario in Section V-A, except that steps (2) and (5) were modified appropriately and the experiments were stopped at step (5).

We observe that `C` outperforms the other methods in terms of task-difficulty. However, a low task-difficulty does not necessarily lead to a model that is desirable, (see Fig. 6 for examples[7]), or is similar to the user's demonstration. Fig. 5(a)-(c) illustrate this issue more clearly. `C` performs worst in terms of satisfaction (Fig. 5(b)) and results in models that do not resemble the original demonstration (Fig. 5(c)). By incorporating crowd satisfaction ratings `CS` results in more preferable models (Fig. 6(b)), but at the expense of greater task difficulty (Fig. 6(a)).

In summary, the results show that the robot can exploit data from the crowd to learn object models that are easier to build than purely local user-provided object models.

[6] Graduate students in University of Washington Computer Science & Engineering department who are not focusing on the field of robotics. Two of them are male and one of them is female.

[7] Videos of Gambit constructing these four object models are available at `http://homes.cs.washington.edu/~mjyc/collabot/project-site/index.html`

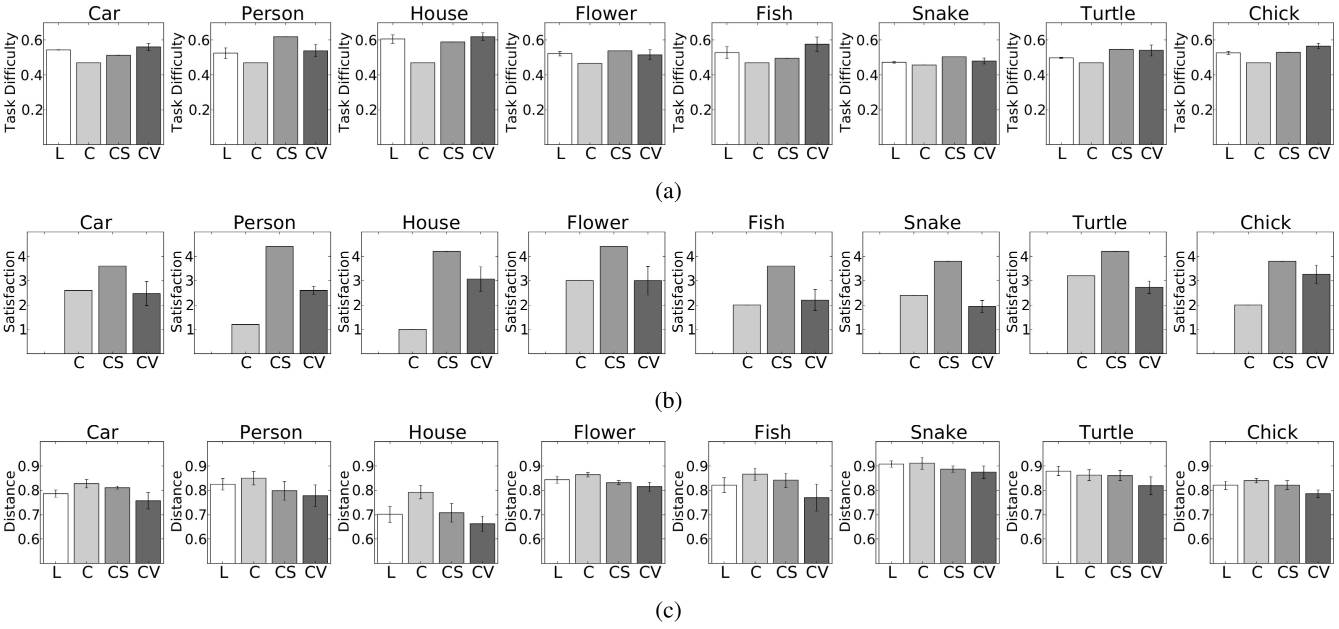


Fig. 5. **Task Difficulty, Crowd Satisfaction and Visual Distance for Four Experimental Conditions**. For every object class, the results from four experimental conditions– L, C, CS, CV (see the main text)–are plotted with three different score metrics. When using local demonstrator data or using visual distance scores, there are multiple ways to get the results. Therefore we report means and standard deviations for them. The L results in (b) are empty because the crowd did not rate the local user demonstrations.

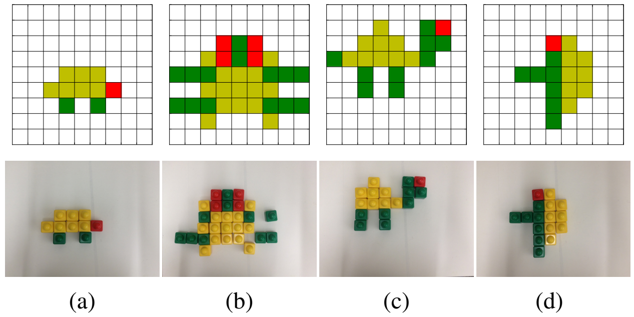


Fig. 6. **Crowd-provided and Gambit-constructed Models for Object Class "Turtle"**. (Top Row) Online worker constructions on Mechanical Turk. (Bottom Row) Results of Gambit attempting to construct the same models. (a) and (b) are the models with the lowest and highest task-difficulty, respectively, estimated using our heuristic score (Gambit fails to build (b)). When we use a single score that combines task-difficulty with crowd rating of "satisfaction with a model," (c) and (d) emerge as the models with the highest and lowest ratings, respectively.

However, just considering task-difficulty can yield unsatisfying results, so introducing additional scores such as crowd satisfaction and visual distance can be useful in selecting good models.

C. Experiment 2: Goal Inference

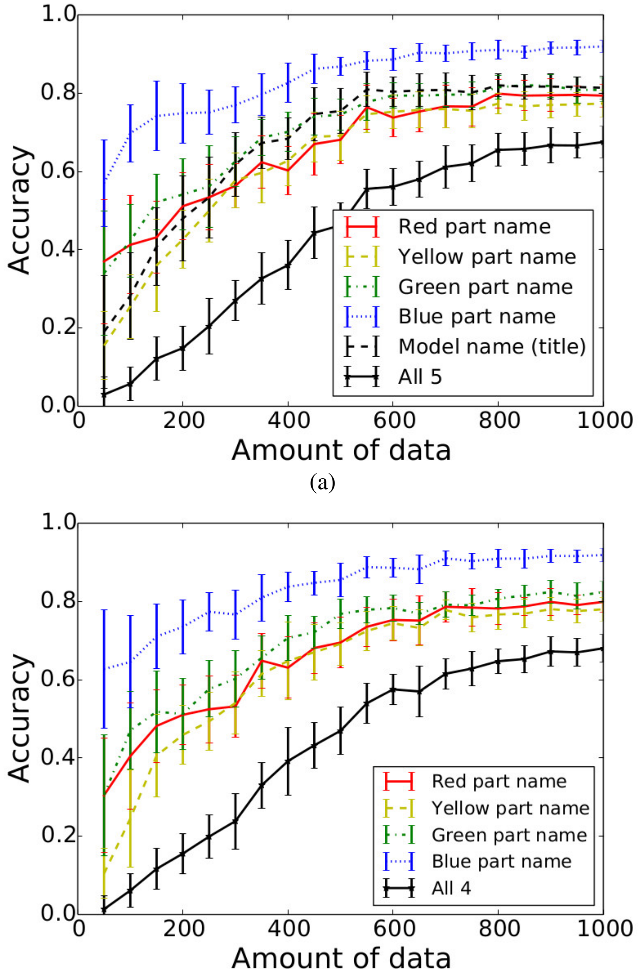
The second experiment tests whether the data obtained from online workers allows the robot to infer an object and its parts given a novel construction built from blocks. Fig. 7(a) shows the performance of classifiers predicting object and part names using the proposed graphical model. In Fig. 7(b) the graphical model is trained on complete models, but must then predict the name of an object part that has been removed from the input. All plots use data described in Section V-B-1). Plots are averaged over ten runs with random 80%/20% train/test splits over the data.

There were two interesting aspects of these results. First, in both cases, the performance of "Predicting all 4" classifiers just started to mature when the amount of input data reached 800. Second, our unmodified graphical model was robust against missing pieces. These results lend further support to the viability of crowdsourcing as a vehicle for collecting large-scale data. In addition, the results also demonstrate one possible benefit of using the goal-based imitation method: missing part prediction is possible without any modification to the framework.

D. Experiment 3: User Preferences on Imitation Methods

Lastly, we conducted a user study to investigate people's preferences among the three different imitation methods described in Section V-B: using the task-difficulty, satisfaction combined, and visual-distance combined scores.

The study consisted of three steps: collecting data from the participants, a "Best Representation" survey, and a "Best Imitation" survey. In the participant data collection step, the participants were asked to build the eight 2D object models described in the second experiment. The models built by participants were used as inputs to the imitation methods in the following steps. In the "Best Representation" survey step, the participants were asked to rate each model based on "how well it represent its title" on a Likert scale (1: poor, 5: excellent). They were shown with three object models retrieved by three imitation methods in random order per each class. In the best imitation survey step, the participants were asked to rate the models based on "how well it imitated your model" on a Likert scale (1: poor, 5: excellent). The same retrieved models in the second step were shown to the participants in the same ordering for each class. In addition, the models built by the participants were provided to them



(a)

(b)

Fig. 7. **Object and Parts Classification Performance**. (a) shows the performance of the graphical model predicting object and part names as a function of the amount of data. (b) shows the graphical model predicting the name of a single missing piece.

along with "Your Model" annotations to remind them of their own models. Note that in this study, the robot did not physically construct the final candidate object models.

Our study was completed by 11 participants (6 males, 5 females) who were undergraduate or graduate students. The results are shown in Fig. 8. We observe that the imitation method using the *satisfaction combined* score is rated the highest in the "Best Representation" survey, the imitation method using the *visual-distance combined* score is rated the highest in the "Best Imitation" survey, and the imitation method using *task-difficulty* score alone is rated the lowest in both surveys. However, these differences were not statistically significant in a Three-Way ANOVA test.

VI. Discussion

Crowdsourcing Quality Control: Although we demonstrated the benefits of utilizing crowdsourcing the context of robotic imitation learning, crowdsourcing needs to be used with caution. The quality control of crowdsourcing was non-trivial. Since we collected our data with live crowdsourcing without any quality control mechanism, we were vulnerable to spammers. There were some spammers who produced repeated data from multiple accounts, and even others who produced data without following the instructions [8]. We are planning to adopt more complex quality control techniques developed from the HCI community to improve the quality of crowdsourced data in the future [18], [19].

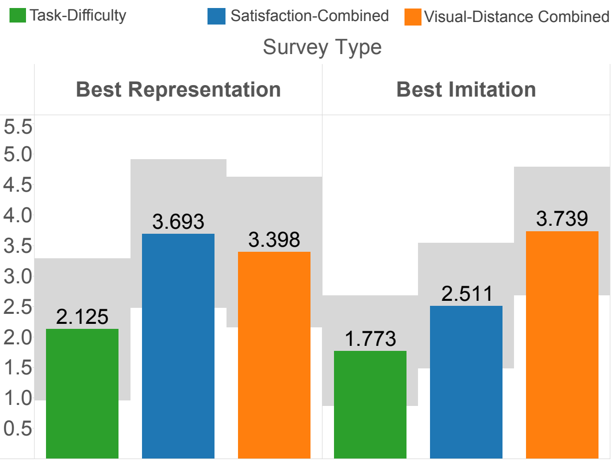


Fig. 8. **User Study Results:**. Average ratings over three imitation methods across 11 participants and 8 object models under two surveys types; Best Representation and Best Imitation. The grey colored error bars represent standard deviations.

Imitation Method: In our user study (Section. V-D), even though there was no statistical significance, the participants preferred the imitation methods we expected them to prefer in two different surveys. This preliminary results showed a potential to perform multi-purpose imitation learning using our current framework. For example, if the user's objective is to make the robot be able to build an instructed object model without failure, he/she can choose the imitation method using task-difficulty score. In another case where the user wants the robot to build an object model that is visually similar to what he/she demonstrated, he/she can use the imitation method using the visual-distance combined score. In addition, for the purpose of knowledge acquisition and general object model construction tasks (not the imitation task), the combined satisfaction score can be effectively used.

Partially Specified Data: For all the results reported in the experiment section, we used what we call "Fully Specified Data". There were two ways to specify labels when the user was providing an initial demonstration. The user could (1) specify both class name and part names of the object model which we call "Fully Specified Data," or (2) only specify the class name of the object model which we call "Partially Specified Data". During the data collection, in the case of (1), the robot asked workers for object models and fully specified the task by providing the title of the model it needed and the names for each part (block color) during data collection. In the case of (2), the robot asked for a 2D object model but provided only the title of the model, not the part names. The demonstrator was then free to name

[8] See our visualization of the dataset available at the project website to see examples.

all parts as desired. Utilizing "Partially Specified Data" in our framework is an interesting direction but brings multiple problems, the primary of which is the widely varying part names provided by the crowd. We are interested in exploring techniques in natural language processing to address such variance in a structured manner, which would allow us to utilize this rich data set.

VII. Conclusions

Our results suggest that crowdsourcing can be a powerful tool for accelerating imitation learning and robotic knowledge acquisition by: (1) augmenting an initial human demonstration with a richer set of examples to learn from, (2) using these examples to find new ways of achieving a goal that was difficult to achieve with just one or a few initial demonstrations, and (3) using crowdsourced data to support human-robot interaction tasks such as goal inference based on object-parts classification and missing-part prediction. Future work will focus on exploring the application of the approach to more challenging tasks such as 3D object assembly and tool use, self-learning of heuristic measures of task-difficulty, and endowing the robot with the ability to use decision theoretic methods to decide when to crowdsource and when to ask the physical user for more examples.

VIII. Acknowledgments

We thank Daniel Butler, Kathleen Tuite, Rahul Benerjee, Jinna Lei and Lydia Chilton who helped inspire and shape this paper during discussions. We thank Cynthia Matuszek for helping with programming the Gambit robot. This work was supported by ONR grant no. N00014-13-1-0817 and NSF grant no. IIS-1318733.

References

[1] P. Kormushev, S. Calinon, and D. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 3232–3237.

[2] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, 2010.

[3] D. B. Grimes and R. P. Rao, "Learning actions through imitation and exploration: Towards humanoid robots that learn from humans," in *Creating Brain-Like Intelligence*. Springer, 2009, pp. 103–138.

[4] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Advances in Neural information processing systems*, vol. 21, pp. 849–856, 2009.

[5] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[6] D. Verma and R. Rao, "Goal-based imitation as probabilistic inference over graphical models," *Advances in Neural information processing systems*, vol. 18, p. 1393, 2006.

[7] A. L. Friesen and R. P. Rao, "Gaze following as goal inference: A Bayesian model," in *Annual Conference of the Cognitive Science Society*, 2011.

[8] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1579–1585.

[9] L. Berthouze and Y. Kuniyoshi, "Emergence and categorization of coordinated visual behavior through embodied interaction," *Autonomous Robots*, vol. 5, no. 3-4, pp. 369–379, 1998.

[10] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[11] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng, "Discovering optimal imitation strategies," *Robotics and autonomous systems*, vol. 47, no. 2, pp. 69–77, 2004.

[12] A. Sorokin, D. Berenson, S. S. Srinivasa, and M. Hebert, "People helping robots helping people: Crowdsourcing for grasping novel objects," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2117–2122.

[13] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins, "Human and robot perception in large-scale learning from demonstration," in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 339–346.

[14] C. Breazeal, N. DePalma, J. Orkin, S. Chernova, and M. Jung, "Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 82–111, 2013.

[15] V. Emeli, "Robot learning through social media crowdsourcing," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2332–2337.

[16] C. Matuszek, B. Mayton, R. Aimi, M. Deisenroth, L. Bo, R. Chu, M. Kung, L. LeGrand, J. Smith, and D. Fox, "Gambit: An autonomous chess-playing robotic system," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 4291–4297.

[17] "Primo," http://www.lugnet.com/primo/, 2013, [Online; accessed 9-15-2013].

[18] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Turkit: human computation algorithms on Mechanical Turk," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 2010, pp. 57–66.

[19] L. B. Chilton, G. Little, D. Edge, D. S. Weld, and J. A. Landay, "Cascade: crowdsourcing taxonomy creation," in *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*. ACM, 2013, pp. 1999–2008.

[20] P. Dai, C. H. Lin, D. S. Weld, *et al.*, "POMDP-based control of workflows for crowdsourcing," *Artificial Intelligence*, 2013.