Advances in Cognitive Systems 4 (2016) Submitted 4/2016; published 6/2016

Situated Mapping for Transfer Learning

Tesca Fitzgerald TESCA.FITZGERALD@CC.GATECH.EDU
Kalesha Bullard KSBULLARD@GATECH.EDU
School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA

Andrea Thomaz ATHOMAZ@ECE.UTEXAS.EDU
Department of Electrical and Computer Engineering, UT Austin, Austin, TX, USA

Ashok Goel GOEL@CC.GATECH.EDU
School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

In transferring a task model learned in a source environment to a new, target environment, the agent may encounter novel objects. Thus, analogical mapping between objects in the source and target environments is both important and complex. Two objects that share the same purpose in one task may be mapped to different objects in the context of another task. We address this context-dependent object mapping by leveraging a human teacher. The teacher provides a limited number of object correspondences from the ground-truth object mapping, from which the remainder of the object mapping can be predicted. We evaluate this Mapping by Demonstration approach both in simulation and on two physical task examples (sorting and assembly). Our results show the agent can use human guidance to quickly infer a correct object mapping, requiring assistance with only the first 1/14 and 5/12 steps of the sorting and assembly tasks, respectively.

1. Introduction

Analogical mapping is an important step in transferring knowledge, including task models, from a familiar situation to a novel situation. When an agent is presented with a demonstration of a task in a familiar "source" environment, the demonstration is observed in terms of the set of objects present in that environment. When the agent is asked to complete the task in a new, "target" environment, it will need to adapt its actions to address objects *analogical* to those in the source environment. However, identifying object mappings is difficult; if objects have been replaced, then the objects in the target environment that best fulfill the purpose of each source object must be identified.

Taylor & Stone (2009) survey a wide range of current approaches to transfer learning in robotics, several of which require a mapping between state variables in the source and target domains. Our work takes a cognitive systems approach to transfer learning, and is motivated by the importance of analogical mapping in transfer learning, focusing particularly on mapping both objects and state variables. In our work, a robot has previously learned a task in some source environment, and is asked to perform that task in a novel target environment. For example, an assembly task in a source

environment containing wood blocks may be transferred to a target environment containing Lego bricks.

While object mapping has been addressed extensively in related work (see Section 2), it is generally assumed that (i) the agent has experience in the target environment and thus has a good representation available for the target problem, and/or (ii) pre-defined features can be used to bias the mapping (e.g. blue objects are mapped to other blue objects based on their *color* feature, or cups are mapped to other cups based on their shape, etc). However, object mapping is typically context-dependent, and so we cannot assume that objects with the same shape or classification play the same role in source and target environments. For example, in a sorting task, a colored bowl can be sorted in multiple ways depending on whether object mapping is performed by the objects' affordance or color feature. Thus, the robot must determine *which features* are important to the object mapping within the context of that task. In our technique, the agent infers an object mapping using assistance from the human teacher, without additional expertise in the target environment or prior knowledge about which feature to use to map objects.

Our approach to this "situated mapping" leverages the human teacher to assist the agent by indicating the objects involved in the first few steps in the task. Through this assistance, the mapping algorithm introduced in this paper infers the mapping between source and target objects for a task, such that the remaining steps in the task can be completed autonomously. This goal presents the question: how should the agent use the teacher's assistance to infer the correct object mapping? We make two contributions: (1) identify context-dependent situated mapping as a subclass of analogical mapping problems, and (2) introduce an interactive Mapping by Demonstration approach to solving situated mapping problems.

We evaluate our approach within a 2D simulated environment and also provide case-studies on two physical tasks perceived in the real-world, under the assumption that all object features can be retrieved from a knowledge base or inferred from perception. This demonstrates how our approach can be applied to a virtual or physical agent to use a small number of hints to predict a correct object mapping.

2. Related Work

2.1 Machine learning approaches to object mapping

Lee et al. (2015) address mapping physical objects by their perceived point clouds. Their method infers a warping function that transforms the source object point cloud such that it closely matches that of the target object. However, using this warping function as a measure of similarity limits object mapping to those with similar shape. Huang et al. (2015) address a similar problem of aligning object point clouds, but prioritize alignment of certain labeled features of the objects. Overall, these approaches are effective in identifying similar objects, but assume that (i) similar objects play the same role in the source and target environments and (ii) corresponding objects are comprised of the same parts or shapes. These assumptions do not hold when the agent does not know *a priori* which features to use in object mapping.

Diuk, Cohen, & Littman (2008) introduce object-oriented MDPs to represent Reinforcement Learning (RL) problems in terms of objects and their relations, where relations between objects are

included in the agent's state representation. Objects in the same classification are assumed to serve the same purpose, regardless of their role in the learning problem. Taylor, Whiteson, & Stone (2007) introduce an approach to inter-task mapping in RL problems in which the agent trains a classifier to predict correspondences between source and target state variables, using experience in the target domain. As in our approach, this does not assume that mapping should be completed according to a particular, pre-defined object feature. However, rather than require that the agent has experience in the target domain, we use minimal human guidance to gain information about the target task.

2.2 Cognitive systems approaches to object mapping

Mapping is a core problem in making analogies and thus has received significant attention in research on cognitive systems. In general, cognitive systems approaches to analogical mapping emphasize the importance of mapping relationships among objects and not just features of the objects (Gentner, 1983; Gentner & Markman, 2006; Gick & Holyoak, 1983). Falkenhainer, Forbus, & Gentner (1989) describe the well-known and influential Structure Mapping Engine that performs analogical mapping based on the graph structure of the representations of the source and target problems. Holyoak & Thagard (1989) describe structural, semantic, and pragmatic constraints on analogical mapping, and a constraint satisfaction technique for analogical mapping. Similarly, our approach to situated mapping does not assume any predefined set of features for object mapping; instead, building on the past work on analogical mapping, it infers the underlying structure of the source and target object configurations and uses structural similarity to map the entities.

Our work also relates to two other themes in cognitive systems research on analogy: visual analogy and interactive analogy. Davies, Goel, & Yaner (2008) describe the technique of constructive analogy that interleaves analogical mapping and transfer such that while mapping constrains transfer, transfer constrains mapping. Yaner & Goel (2008) describe compositional analogy that conducts analogical mapping at multiple levels of abstraction such as shape, structure, behavior and function, with mappings at a lower level in the abstraction hierarchy constraining mapping at higher levels. Our work too is an instance of visual analogy in that low-level perception provides visuospatial information about the objects in the source and target environments.

More recently, Vattam & Goel (2012) have described the phenomenon of interactive analogical retrieval, in which designers, who are novices in biology, search for biological analogies in external sources on the web instead of retrieving them from their long-term memory. Vattam & Goel (2013) have presented a scheme for annotating biological sources to guide interactive analogical retrieval. In our work, we situate analogical mapping in the robot's external environment, with a teacher providing hints to guide the mapping.

3. Context-Dependent Mapping

We focus on the problem of enabling an agent to infer the correct analogical mapping of objects in order to perform a previously learned task in a new environment. Object mapping for task transfer is subject to an additional challenge: a different mapping may be required for the same set of objects in the source and target environments, depending on the task. Suppose that a robotic agent learns the task of sorting office supplies in a source environment S. Later, the robot may be asked to complete

the sorting task in target environment T, where a new set of supplies are present that differs from the source objects in their locations, sizes, colors, and shapes. Many mappings are possible between objects in S and T. The task of sorting by color should use an object mapping performed over the *color* feature. The task of sorting by object usage should utilize the objects' *affordance* feature (e.g. pour-able, contain-able) in mapping. While the robot does not know *a priori* the feature(s) on which mapping should be performed, the human teacher can provide information about this task context.

Current approaches to this type of context-dependent object mapping cannot address these types of problems because they operate over the assumption that one or two pre-defined object features (such as shape or location) can be used to determine an object mapping for every task. In our work, however, the features used to map objects are context-dependent, and need to be selected based on the task being transferred.

3.1 Problem Specification

When asked to transfer a task to a new environment, the agent has a representation of the source environment S and can perceive the target environment T, where $S = \{s_0, s_1, ..., s_n\}$ and $T = \{t_0, t_1, ..., t_m\}$, and s_n and t_m represent objects in the source and target environments, respectively. The hypothesis space is the set M containing all possible mappings $m : S \rightarrow T$. Each mapping hypothesis $m \in M$ is a binary $|S|$x$|T|$ matrix where: $m_{ij} = \begin{cases} 1 & \text{if} \quad S_i \mapsto T_j \\ 0 & \text{otherwise} \end{cases}$

We approach object mapping as a problem of finding the mapping $m \in M$ that maximizes the total similarity (Eqn. 1) between objects in the source environment and their counterparts in the target environment.

$$map(S, T) = \underset{m}{\operatorname{argmax}} \sum_{i=0}^{|S|} \delta(s_i, m(s_i, T)) \tag{1}$$

where $m(s_i, T)$ returns the object $t_j \in T$ for which $m_{i,j} = 1$. Equation 1 assumes some similarity function $\delta(a, b)$ that returns a measure of the suitability of object b to replace object a within the context of the task. Defining this similarity function is the primary challenge of object mapping. We assume a 1-to-1 object mapping, with future work addressing m-to-n mappings.

A mapping refers to a function which maps objects in the source environment to the target, and is determined by the object features relevant to that task. As a result, we can associate any mapping function m with a feature set $f \in F$ such that objects mapped by m share similar values for features in f. Thus, part of the challenge of selecting a mapping $m \in M$ is to select which feature set $f \in F$ is best supported by the task. We define the feature set space $F = \mathcal{P}(R) - \varnothing$, where $\mathcal{P}(R)$ produces the power set of the feature vector R (which we define next).

We define the feature vector R over which mapping may occur. Holyoak & Thagard (1989) discuss three strategies for analogical mapping, which we adapt to refer to different object features: spatial relations among objects (referred to as a structural mapping strategy), object affordances (the actions enabled by an object, referred to as a semantic strategy), and the relation between the object and the task goal (object properties such as whether a cup used in a pouring task is *full*, referred to as a pragmatic strategy). We build on these mapping features to include features relevant to a physical

Perceived Features	Derived Features	Knowledge-base Features
Location	Spatial relations	Affordances
Hue	Hue-shift	Properties
Size	Size-shift	

Table 1: Source of Each Object Feature's Value

object mapping task, representing the pragmatic mapping strategy based on an object's properties and adding additional mapping features based on each object's hue and size. Thus, we represent objects by the feature vector $R = <c, ch, d, dh, s, a, p>$ containing (1) the color c represented by the object's hue, (2) the area of the object's bounding box, d, (3) the affordance set $a = \{a_0, a_1, \ldots, a_n\}$ where a_i is an affordance of the object, and (4) the property set $p = \{p_0, p_1, \ldots, p_n\}$ where p_j is a property associated with each affordance of the object (e.g. full, open, closed, empty). Features ch and dh are derived features and are introduced shortly. This results in F being the set of all 127 combinations of the feature vector R.

We consider three additional, *derived* features which are calculated over a set of objects. The hue-shift ch represents the average difference in hue between source objects and their corresponding objects in the target environment as observed in mapping hints provided by the human teacher. This is used to represent mappings in which objects specifically correspond to different-colored objects (e.g. all blue objects now map to green objects). The size-shift dh represents the average change in size between source objects and their known corresponding objects in the target environment as observed in mapping hints provided by the human teacher. This is used to represent mappings where source and target objects exist at a different scale (e.g. objects in the target environment are twice the size of source objects). Finally, the spatial relation set $s = \{s_0, s_1, \ldots, s_n\}$ contains spatial relations $s_n = <r, o>$ for relation $r \in \{$left-of, right-of, above, below$\}$ to the object o. In our real-world evaluation, each feature's value is obtained as noted in Table 1. In our simulated evaluation, object feature values are generated at random to realize simulated source and target environments.

4. Situated Mapping

We consider situated mapping in interactive environments, in which the agent is taught by a human teacher available to assist in the mapping between the source and target environments. We assume that in the source environment, the agent learned a task consisting of a sequence of steps each involving a single object interaction. Given this, by demonstrating one step of the task in a new target environment, the human teacher provides a supervised example of mapping one object from the source to another object in the target. In this work, we call these examples *hints* about the true object mapping, where a single hint consists of an object in the source environment and its corresponding object in the target environment. If the teacher provides a hint for enough steps in the target environment, the mapping hypothesis space will eventually be reduced to a single mapping. However, our goal is to infer the object mapping with as few mapping hints as possible.

Algorithm 1 Mapping by Demonstration

1: $E \leftarrow$ evaluateHypotheses(M, F, S, T)
2: $correctMapping \leftarrow$false
3: **while** $correctMapping$ = false **do**
4: $srcID, tgtID \leftarrow$ receiveMappingAssist()
5: $M \leftarrow$ pruneMappingHypotheses(M, $srcID$, $tgtID$)
6: $F \leftarrow$ pruneFeatureSpace(F, $srcID$, $tgtID$)
7: $E \leftarrow$ evaluateHypotheses(M, F, S, T)
8: $p \leftarrow$ predictMapping(M, F, E)
9: $correctMapping \leftarrow$ requestValidation(p)
10: **return** p

4.1 Mapping by Demonstration Algorithm

We introduce the *Mapping by Demonstration* (MbD) algorithm (Algorithm 1). Initially the algorithm considers all possible mapping hypotheses between source S and target T, with every feature being potentially relevant, and thus every feature set $f \in F$ being equally plausible. The mapping hypothesis space M and feature set space F are intialized as described in Section 3.1. As the teacher provides hints, M and F are pruned, removing hypotheses and feature sets that are inconsistent with the hints seen. This process continues until the correct mapping is determined. We do this so as to fully evaluate the number of hints needed for the algorithm to succeed, leaving the task of deciding when to stop requesting mapping hints to future work. We describe steps 5-8 in more detail in the following sections.

4.1.1 Prune Mapping Hypothesis Space and Feature Space

When a hint is provided, any correct mapping must contain that object correspondence. As a result, the mapping hypothesis space is pruned to include only those that support the correspondence indicated by the object mapping hint. Reducing the mapping hypothesis space in this way (which our Evaluation section refers to as *hypothesis pruning*) increases the likelihood that a mapping selected from the hypothesis space will be correct. Our approach increases this likelihood further by using the mapping hint to infer the feature(s) over which the true mapping is based. A feature set is removed from the feature space if its variance across all remaining (unmapped) objects is 0. For example, if all remaining objects have the same affordance, the feature set $\{a\}$ is removed from the feature space F since it does not support any mapping hypothesis over another.

4.1.2 Evaluate Hypotheses

A matrix is generated which contains an evaluation metric of every possible object correspondence, where the score reflects similarity between two objects according to each feature. We generate a $|S|$x$|T|$x$|R|$ evaluation matrix E, where $E_{i,j,r} = \Delta r_{i,j}$, representing the similarity between objects i and j according to feature r as: $\Delta r_{i,j} = <\Delta \mathbf{c}_{i,j}, \Delta \mathbf{ch}_{i,j}, \Delta \mathbf{d}_{i,j}, \Delta \mathbf{dh}_{i,j}, \Delta \mathbf{s}_{i,j}, \Delta \mathbf{a}_{i,j}, \Delta \mathbf{p}_{i,j}>$, where similarity between two feature values relies on the distance function D. We estimate the

difference between two feature values along a Gaussian curve as follows:

$$\mathbf{D}(v_1, v_2, r) = \frac{\mathcal{N}(v_2|v_1, 1\sigma r)}{\mathcal{N}(v_1|v_1, 1\sigma r)} \tag{2}$$

where $\mathcal{N}(x|\mu, \sigma)$ is the probability density of x over the normal distribution, v_1 and v_2 are the value of the two objects' features, and r is the range of their possible values. Using this distance function, we define the similarity function for each feature as follows:

- *Hue sim.:* $\Delta\mathbf{c}_{i,j} = D(0, \tan^{-1}\left(\frac{\sin(h_i - h_j)}{\cos(h_i - h_j)}\right), 360)$
- *Shifted-hue sim.:* $\Delta\mathbf{ch}_{i,j} = D(c_j, c_i + \mu_c, 360)$
- *Size sim.:* $\Delta\mathbf{d}_{i,j} = D\left(1, \frac{|d_j - d_i|}{d_i} + 1, 1\right)$
- *Size-change sim.:* $\Delta\mathbf{dh}_{i,j} = D\left(0, \frac{|d_j - d_i|}{d_i} - \mu_d, 1\right)$
- *Spatial sim.:* $\Delta\mathbf{s}_{i,j} = D(|s_i|, |s_i \cap s_j|, |s_i|)$
- *Affordance sim.:* $\Delta\mathbf{a}_{i,j} = D(|a_i|, |a_i \cap a_j|, \min(|a_i|, |a_j|)$
- *Property sim.:* $\Delta\mathbf{p}_{i,j} = D(|p_i|, |p_i \cap p_j|, \min(|p_i|, |p_j|)$

4.1.3 Evaluate Mapping Hypotheses

Using the evaluation matrix, a mapping can be evaluated according to a feature set as follows:

$$\mathrm{V}(m, f) = \frac{1}{|f|} \sum_{f^{(i)} \in f} \mathrm{sum}(m \circ E_{f^{(i)}}) \tag{3}$$

Where the function $A \circ B$ returns the entrywise product of matrices A and B. We use the notation $E_{f^{(i)}}$ to represent the previously defined |S| x |T| evaluation matrix derived from using feature $f^{(i)}$. Equation 3 returns a similarity score for any mapping $m \in M$, using the feature set f as the similarity metric between objects in that mapping.

4.1.4 Predict Mapping

Given an evaluation matrix containing an evaluation for every mapping and feature set pair, we solve for the feature set f which yields the maximum-scoring mapping as follows:

$$\arg\max_{f \in F} \left(\max_{m} \left(\frac{1}{|S|} \mathrm{V}\,(m, f) \right) \right)$$

5. Evaluation

5.1 Simulated Evaluation

We evaluated the system with three categories of simulated tasks: containing $n = 5$, $n = 6$, or $n = 7$ objects in the source and target environments. These categories represent incrementally more difficult problems; as the number of objects increases, the mapping hypothesis space from which the agent must choose a single mapping increases factorially. For each category of problem, 10 mapping problems (each representing a task and consisting of a source and target environment) were generated randomly, such that each object's perceived and knowledge-base features had random value assignments, with derived features automatically generated from perceived features. To simulate how some objects may be present in both the source and target environments (as in a realistic mapping problem), each source object had a 50% likelihood of being reused in the target environment, with the remaining objects being randomly generated. Reused objects retained intrinsic feature values (size, color, and affordances), but were given a randomly assigned location and properties, since the values of these features can be changed without replacing the object (e.g. moving a cup to a new location or emptying a cup so that its "isFilled" property changes).

Finally, ground truth mappings for each task were generated corresponding to feature sets consisting of one or two features (except for redundant feature sets {size, size-change} and {hue, hue-change}), resulting in a set of 26 possible ground truth mappings. The source and target environments had the same number of objects, so a bijective mapping was generated. This resulted in a total of 780 evaluations (3 categories x 10 tasks x 26 ground truth mappings). Note that this does not result in 780 *unique* ground truth mappings, since two feature sets may result in the same ground truth mapping.

The mapping hints were also generated for each mapping problem instance. In a realistic mapping problem, one hint will be provided for each step of the task as described in Section 4. As a result, the hint ordering will be dictated by the order in which objects are used in the task plan. Since the task plan is undefined in our simulated tasks, we evaluated the MbD algorithm using every possible hint ordering to observe the impact of hint ordering on mapping performance. This results in evaluating over a permutation of ${}^{n}P_{n-2}$ potential hint orderings. The ordering of the last two hints does not matter, since hint $n - 1$ leaves only one object remaining, resulting in the complete ground truth after $n - 1$ hints.

We ran our evaluation as follows:

1. Problem instances, ground truths, and hint orderings are generated a priori.
2. For each problem instance, the agent iteratively retrieves the next hint to be provided by the teacher to the object mapping algorithm and checks its predicted solution.
3. If the predicted mapping is correct, the agent halts and records the number of hints needed to get the correct solution using this hint ordering. If the prediction is incorrect or if multiple predictions are returned, the agent repeats the process by retrieving the next hint.

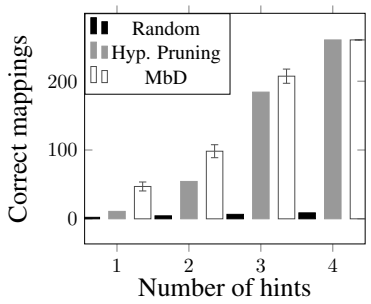


Figure 1: Performance in 5-Object Tasks

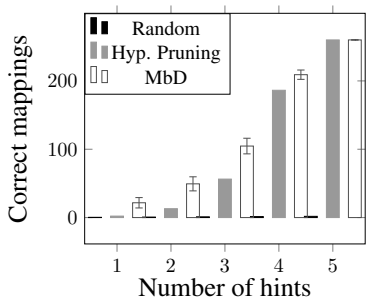


Figure 2: Performance in 6-Object Tasks

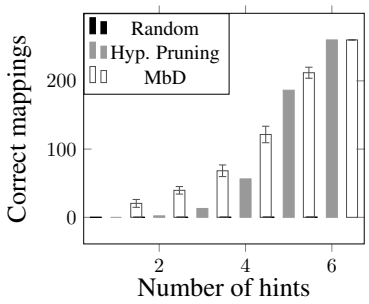


Figure 3: Performance in 7-Object Tasks

5.1.1 Results

For each class of n-object problems, we collected data on the algorithm's performance over all possible hint orderings, where we measure performance as the number of correct mappings after each hint is provided. For all of these problems, the full ground truth mapping is provided at $n-1$ hints, since only one source and target object remains to be mapped after hint $n-1$. Figures 1 - 3 compare the expected performance of the MbD algorithm over all hint orderings, with error bars denoting one standard deviation, to two baselines: (i) expected performance when selecting a random mapping without utilizing mapping hints, and (ii) expected performance when using mapping hints to *only* prune the hypothesis space (described in the Pruning step in Section 4.1), and then choosing a random mapping from the remaining hypothesis space (rather than using the hint to infer features on which mapping is based).

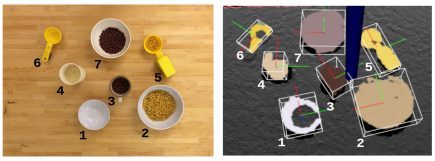
5.2 Real-World Case Studies

We have provided simulation results as a systematic analysis of the approach. The following real-world examples provide case studies demonstrating example tasks for which situated mapping problems exist and how they can be addressed using the MbD approach. To evaluate the suitability of the MbD algorithm for a robot learner, we tested it on two physical tasks: a dish sorting task (shown in Figure 4(a)) and a stacking assembly task (shown in Figure 4(c)). The robot passively observed its environment, and did not record or execute any actions. Similar to the simulated evaluation, we ran the real-world evaluation as follows:

1. The mapping ground truth(s), task plan, and object affordances/properties are defined a priori. In the sorting task, there are several correct object mappings, whereas the assembly task has a single correct object mapping.

2. We set up the source environment and recorded the robot's observation of the scene, repeating this step for the target environment.

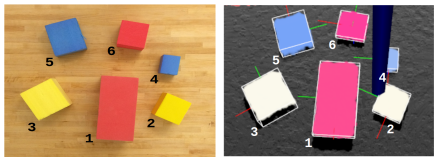
3. The algorithm accepts the hint corresponding to the next object used in the task plan and predicts a mapping solution.



(a) Sorting Task Source Environment

```
clean dish 1
put dish 1 away
clean dish 4
put dish 4 away
clean dish 6
put dish 6 away
clean dish 3
put dish 3 away
clean dish 7
put dish 7 away
clean dish 2
put dish 2 away
clean dish 5
put dish 5 away
```

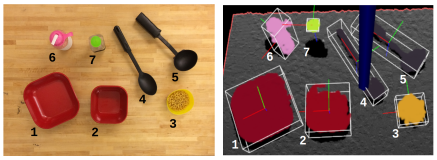
(b) Sort Plan



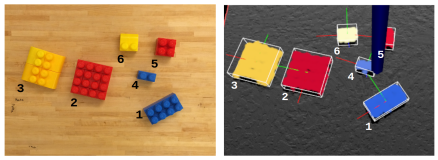
(c) Assembly Task Source Environment

```
pick up part 1
install part 1
pick up part 3
install part 3
pick up part 5
install part 5
pick up part 6
install part 6
pick up part 2
install part 2
pick up part 4
install part 4
```

(d) Stack Plan



(e) Sorting Task Target Environment



(f) Assembly Task Target Environment

Figure 4: Sorting and Assembly Task Environments

4. If correct, the algorithm halts and records the number of hints needed to get a correct solution using this hint ordering. If the prediction is incorrect, the algorithm repeats the previous step and accepts the next mapping hint.

Rather than generate feature values as in the simulated evaluation, in this evaluation all objects' perceived features listed in Table 1 are observed from an RGBD sensor above the environment. We incorporate perception to provide an example of how our mapping strategy may be applied to real-world problems, but do not consider the perception aspect itself to be a contribution of our work. Our main focus is on the underlying mapping strategy, and thus we indicate the perceptual features which our mapping algorithm uses and identify sources from sensors and a knowledge base.

Objects are perceived by abstracting a set of segmented objects from the point cloud using the algorithm described by Trevor, Gedikli, Rusu, & Christensen (2013). A bounding box is fitted to each segmented object to approximate its centroid $< x, y, z >$ and dimensions $< width, depth, height >$ as shown in Figure 4(a). The object's overall hue is derived from average hue of each pixel located at the surface of the object. Once perceived features are obtained, object locations and dimensions are used to derive a set of spatial relations between objects. Finally, size and color are used as a heuristic to assign an ID to each object, which is then used to retrieve its knowledge-base features: the affordances and properties associated with that object. Currently, we manually provide the affordances and properties associated with each object ID. In future work, we plan for the robot to autonomously retrieve this knowledge using an object classifier.

5.2.1 Cleaning and Sorting Task

In the first task, each environment consists of two utensils, two cups, and three bowls, which are to be cleaned and sorted separately based on their object type (indicated by their affordances). The source and target environments are shown in Figures 4(a) and 4(e), respectively. Table 2 lists the affordance and property values provided for these objects. The remaining features listed in Table 1 are derived from perception.

Object	Affordances	Properties
Cups	Fillable, Pourable	Empty, Full, Upright
Bowls	Fillable, Stackable	Empty, Full, Upright
Utensils	Scoopable, Pourable	Empty, Full, Upright
Blocks	Stackable	N/A

Table 2: Provided Object Knowledge Base

Task	Hint	Predicted Mapping
Sorting	$\mathbf{1 \mapsto 3}$	$1 \mapsto \mathbf{3}, 2 \mapsto 1, 3 \mapsto 6, 4 \mapsto 7, 5 \mapsto 5, 6 \mapsto 4, 7 \mapsto 2$
Assembly	$1 \mapsto 2$	$1 \mapsto 2, 2 \mapsto 6, 3 \mapsto 3, 4 \mapsto 1, 5 \mapsto 4, 6 \mapsto 5$
	$3 \mapsto 3$	$1 \mapsto 2, 2 \mapsto 6, 3 \mapsto 3, 4 \mapsto 1, 5 \mapsto 4, 6 \mapsto 5$
	$\mathbf{5 \mapsto 1}$	$1 \mapsto 2, 2 \mapsto 6, 3 \mapsto 3, 4 \mapsto 4, 5 \mapsto 1, 6 \mapsto 5$

Table 3: Predicted Mappings After Each Hint

In contrast to simulated evaluations, there are several correct mappings for this task, since any bowl in the source environment can be mapped to any target bowl, either source utensil can be mapped to either target utensil, and either source cup can be mapped to either target cup. Whereas the task plan was undefined in the simulated evaluations, and thus the algorithm was evaluated over every hint ordering, we use the task plan shown in Figure 4(b) to define the hint ordering. Hints were provided in the order in which objects would be used to complete the sorting task in the source environment: starting with the object closest to the robot's left hand, and continuing in order of increasing distance from the robot's hand.

The algorithm returned eight mappings (all of which were correct for the task) after the first hint: the correspondence between the small white bowl and small yellow bowl. This hint, and one of the returned mappings, is listed in Table 3.

5.2.2 Assembly Task

In the second task, each environment consists of six colored blocks of various sizes. The goal of this task is to assemble a model by stacking the blocks in a repeating color sequence (red-yellow-blue) and in order of decreasing size (such that large blocks are placed first). Thus, objects should be mapped according to their hue and relative size. The source and target environments each contain a different kind of block, and are shown in Figures 4(c) and 4(f). Affordance and property values were provided as listed in Table 2, and remaining features were derived from perceptual information.

As in the cleaning and sorting task, object hints were provided in the order in which objects would be used to complete the task in the source environment. Objects in this task are stacked in order of the required color sequence and in decreasing size as listed in Figure 4(d) (the task plan referencing objects in the source environment); thus, the object corresponding to the large red block was provided first, then the object corresponding to the large yellow block, and so forth. There is one correct mapping for this task, which was returned by the algorithm after three hints. Each of the

provided hints and the predicted mapping after each hint is listed in Table 3, with the bolded hint being the final one provided before the algorithm returned the correct mapping.

6. Discussion

In any mapping problem, the number of possible object mappings increases factorially with the number of objects present. Graph isomorphism is an intractable problem and thus this attribute is inherent to any technique for object mapping. This motivates situating mapping in the task environment. While all mapping hypotheses are considered, leveraging human assistance helps (i) prune the hypothesis space after each hint, (ii) prune the feature set space, and (iii) re-evaluate the remaining mapping hypotheses to produce a mapping prediction.

The results indicate that using mapping hints increases the agent's likelihood of predicting a correct mapping quickly. Even when mapping hints are only used to prune the hypothesis space, as shown in the Hypothesis Pruning results in the simulated evaluation, the agent's likelihood of selecting a correct mapping is dramatically increased over that of choosing an object mapping at random. The MbD algorithm provides further benefit by inferring additional information from the hint; rather than only use the mapping hint to prune the hypothesis space, it is also used to infer the feature(s) on which mapping may be performed. This benefit is especially evident as the hypothesis space increases. Particularly, in the 7-object task (the category of mapping problems with the largest hypothesis space), the MbD algorithm correctly solves significantly more problems within the first 1-4 hints than either the Hypothesis Pruning or Random Mapping baselines.

The two perceived tasks (sorting and assembly) demonstrate the use of MbD on physical tasks such as those a robot would need to encounter. By assisting the robot with the initial steps of a task in the target environment, the robot can complete the rest of the task autonomously once it has inferred a correct mapping. The cleaning and sorting task described in Section 5.2.1 would consist of at least two steps per object (clean dish x, put dish x away), resulting in a task containing a total of 14 steps. Only one mapping hint was needed for the robot to predict a correct mapping; as a result, the robot requires assistance from the teacher only during the first step of the cleaning and sorting task (shown in bold in Figure 4(b)), and can execute the remaining 13 steps autonomously. Similarly, the assembly task described in Section 5.2.2 would consist of two steps per object (pick up part x, install part x), resulting in a task containing 12 steps. Three mapping hints were needed for the robot to predict the correct mapping; as such, the robot requires assistance from the teacher only during the first five steps of the assembly task (shown in bold in Figure 4(d)), and is able to execute the remaining seven steps autonomously.

7. Conclusion and Future Work

We have presented an approach to the problem of mapping objects observed in a source environment such that learned tasks can be reused in a new target environment. In this work, we have framed this as a "situated mapping" problem and have proposed an interactive approach to address it by using the human-provided mapping hints to predict the most appropriate mapping for the task. We validated our approach in simulated mapping tasks and two real-world tasks, confirming our

hypothesis that the proposed approach is able to predict the correct situated object mappings with few hints.

The current algorithm continues receiving hints until it returns a mapping that is equal to the ground truth mapping. We do this so as to validate our overall mapping strategy by fully evaluating the number of hints needed for the algorithm to succeed. In future work, we intend to (1) address the problem of deciding when to stop requesting mapping hints by incorporating the learner's confidence in its mapping solution and (2) consider mapping problems which are not 1-to-1.

References

Davies, J., Goel, A. K., & Yaner, P. W. (2008). Proteus: Visuospatial analogy in problem-solving. *Knowledge-Based Systems*, *21*, 636–654.

Diuk, C., Cohen, A., & Littman, M. L. (2008). An object-oriented representation for efficient reinforcement learning. *Proceedings of the 25th International Conference on Machine Learning* (pp. 240–247). ACM.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy*. *Cognitive Science*, *7*, 155–170.

Gentner, D., & Markman, A. B. (2006). Defining structural similarity. *The Journal of Cognitive Science*, *6*, 1–20.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, *15*, 1–38.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive science*, *13*, 295–355.

Huang, S. H., Pan, J., Mulcaire, G., & Abbeel, P. (2015). Leveraging appearance priors in non-rigid registration, with application to manipulation of deformable objects. *International Conference on Intelligent Robots and Systems*.

Lee, A. X., Gupta, A., Lu, H., Levine, S., & Abbeel, P. (2015). Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation. *Intelligent Robots and Systems (IROS)*.

Taylor, M. E., & Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, *10*, 1633–1685.

Taylor, M. E., Whiteson, S., & Stone, P. (2007). Transfer via inter-task mappings in policy search reinforcement learning. *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems* (p. 37). ACM.

Trevor, A. J., Gedikli, S., Rusu, R. B., & Christensen, H. I. (2013). Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration*.

Vattam, S. S., & Goel, A. K. (2012). Interactive analogical retrieval. *First Annual Conference on Advances in Cognitive Systems*.

Vattam, S. S., & Goel, A. K. (2013). Biological solutions for engineering problems: a study in cross-domain textual case-based reasoning. In *Case-Based Reasoning Research and Development*, (pp. 343–357). Springer.

Yaner, P. W., & Goel, A. K. (2008). Analogical recognition of shape and structure in design drawings. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *22*, 117–128.