Relational Symbol Grounding through Affordance Learning: An Overview of the ReGround Project

Laura Antanas[1], *Ozan Arkan Can*[2], *Jesse Davis*[1], *Luc De Raedt*[1], *Amy Loutfi*[3], *Andreas Persson*[3], *Alessandro Saffiotti*[3], *Emre Ünal*[2], *Deniz Yuret*[2], *Pedro Zuidberg dos Martires*[1]

[1]Dept. of Computer Science, KU Leuven, Belgium
[2]Koc University, Turkey
[3]Center for Applied Autonomous Sensor Systems, Örebro University, 70182 Örebro, Sweden

luc.deraedt@cs.kuleuven.be (corresponding author)

Abstract

Symbol grounding is the problem of associating symbols from language with a corresponding referent in the environment. Traditionally, research has focused on identifying single objects and their properties. The ReGround project hypothesizes that the grounding process must consider the full context of the environment, including multiple objects, their properties, and relationships among these objects. ReGround targets the development of a novel framework for "affordance grounding", by which an agent placed in a new environment can adapt to its new setting and interpret possibly multi-modal input in order to correctly carry out the requested tasks.

Index Terms: relational affordances, symbol grounding

1. Introduction

Imagine being able to simply buy a robot and customize it through either task demonstration or linguistic instructions to perform household tasks, e.g., to pick up toys and put them in the right place, or to empty the dishwasher. An essential capability for such a robot is the ability to adapt to both the language and the environment in order to perform the right tasks in the right way. During training, it will encounter new phrases referring to new objects, actions and relations, and it will have to truly ground these to perform its novel tasks. Thus it needs to identify which objects, actions and relationships in the real world the phrases refer to in order to obtain a true understanding of the world and the complex interactions that govern it.

Making this vision a reality is one of the key challenges in building intelligent robots that assist us with our daily tasks. Realizing it requires a fundamental shift in how the research community approaches the problem of symbol grounding. Currently, the focus lies on merely grounding and anchoring the *individual* symbols, which refer to *single* objects and their properties, in the environment. In contrast, the ReGround project[1] hypothesizes that *the grounding process should consider the full context of the environment*, which consists of multiple objects as well as their relationships and properties, and how these change through actions and over time. It aims to develop a novel relational grounding approach that accounts for the relationships between *multiple symbols* in the language and between *multiple referents* in the environment. Our vision is closely related to J.J. Gibsons (1979) original notion of affordances, which referred to the action opportunities (or possibilities) offered to an organism by its environment, and postulated that the organism and its environment complement each other.

[1]RegGround (http://reground.cs.kuleuven.be) is a CHIST-ERA project funded by the EU H2020 framework program.

The ReGround project also hypotheses that *affordances play a central, bi-directional role in mapping language to the world.* In one direction, linguistic clues may help identify affordances in the environment: verbs frequently used with an object provide clues about its affordances. In the other direction, what can and cannot be done with a physical object in an environment provides information relevant to learning word meanings and resolving ambiguous utterances. While affordances have been studied extensively in the robotics literature (e.g., [1, 2, 3]), and are sometimes mentioned in linguistics [4, 5, 6] the focus is on *object affordances*. In contrast, the goal of ReGround is to develop a framework for *relational affordances*, which also model relationships in the environment. Unlike most current work on affordances in robotics, the ReGround approach is to model the environment itself and to reason about it. We do so using logical and relational representations and learning techniques, which have proven useful in both language and robotics.

2. Relational Affordances

The central objective of our project is to develop a symbol grounding approach that goes beyond the current paradigm of symbol identification by learning affordances involving each symbol. The ReGround approach embraces four tenets: (1) integrating information from multiple modalities is necessary to perform symbol grounding; (2) identifying symbols is only the first step in grounding; (3) learning affordances that capture the relationships among properties of symbols, the configuration of the environment, and which actions are possible is the second, underexplored, step of grounding; and (4) performing grounding in a new environment can be aided by using previously learned affordances to make inferences about newly encountered symbols.

This approach is being empirically investigated using a concrete set-up, that will serve as our evaluation platform. This set-up consists of a kitchen table equipped with a Kinova robotic arm, a Kinect RGB-D sensor and a console for simple natural language interaction. The robot arm can be manually steered with a joy-stick, or it can perform autonomous operations. The system is equipped with some basic knowledge about kitchens in general, but has to learn about specific, possibly new objects in your kitchen in order to serve you well. For instance, it could learn that grannys glasses should be used with extreme care, should always be hand washed and not put in the dishwasher, and should be placed on a particular shelf in "grannys cupboard". It can learn all this from instructions in natural language and from demonstration: e.g., if it does not know how to wash a glass by hand, you might show it how to do this by op-

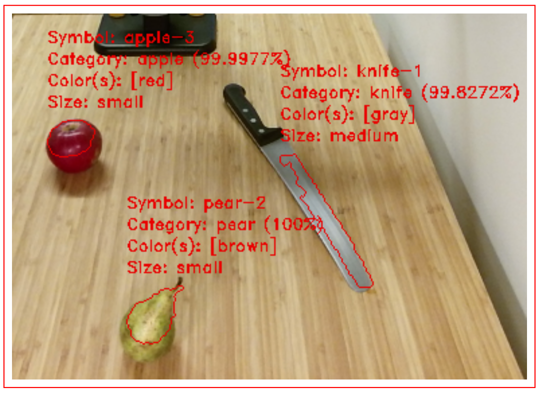

Figure 1: *Example scene from our prepare snack scenario displaying several objects perceived together with some grounded symbolic information about them.*

erating the robot arm with a joystick. It could also ask (limited) clarifying questions, in the same way a real human would. This leads to an interactive scenario that uses multi-modal information. The evaluation should be multi-modal as well – the robot should be able to understand NLP instructions and to explain in NLP what it is doing, or planning to do.

Our current development is guided by a *prepare snack scenario* – see Figure 1. The system is given instructions in natural language, using the appropriate symbols to refer to objects, their properties, and relations that hold among them. For instance:

> Pear and knife are on the table next to each other. Grasp the sharp knife and hold the soft pear. Cut the pear with the knife and put pieces on plate.

In addition to the linguistic description, the system visually perceives the scene: the objects, their (additional) properties (e.g., the knife is elongated besides being sharp), their relationships and the way the states changed over time. Observing also actions in conjunction with language narration (multi-modal input) enables the robot to learn and infer the meaning of words (e.g., knife, fruit, nextto, grasp, cut).

To best capture the information in the presented scenario, we employ relational representations. Because both language and video input interpretation is prone to error, properties and relations are also labeled with probabilities reflecting their degree of belief. Hence, we represent our scene using a probabilistic logic theory containing a set of probabilistic facts:

$0.95 :: \mathtt{object}(\mathtt{pear}).$
$0.99 :: \mathtt{object}(\mathtt{knife}).$
$0.95 :: \mathtt{is}(\mathtt{pear}, \mathtt{soft}).$
$0.9 :: \mathtt{is}(\mathtt{knife}, \mathtt{sharp}).$
$0.8 :: \mathtt{is}(\mathtt{knife}, \mathtt{elongated}). \quad \cdots$

The logical atom (or fact) $0.95 :: \mathtt{object}(\mathtt{pear})$ states that there is an object *pear* with probability 0.95. Using such a description, our goal is now to learn relational affordances of objects. For instance, in our scenario the agent might learn that the knife affords cutting because it is elongated (image data) and sharp (language), or that the pear allows cutting upon because it is a piece of fruit (image) and it is soft (language). Expressed in a relational format the agent would learn facts such as $\mathtt{affords}(\mathtt{knife}, \mathtt{cut}, \mathtt{pear})$, where the $\mathtt{affords}/3$ predicate becomes the learning target.

As the system receives more input, it continuously updates its world model, represented through a probabilistic logic pro-

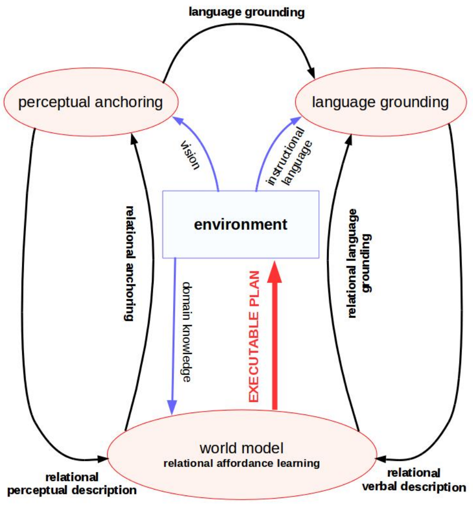

Figure 2: *The structure of the system.*

gram. The model, the language and the visual input interact in complex ways: language allows learning about affordances, perceived affordances from image data influence the language, and information from the language can improve visual perception. Once the system has built a model through a series of input in one environment, it will be placed in a different environment with the goal to exploit the learned knowledge in new situations (e.g., replacing the pear by an apple). Based on unimodal information (e.g., video), the system ought to identify and reason about the various symbols in the environment using the learned probabilistic logic model, and carry out the “cut the apple” task.

3. Techniques

To realize the overall ReGround vision, we integrate techniques from (statistical) relational learning and reasoning, language processing and symbol grounding, perception, child language learning, and simulation. We present these in turn. The overall system structure is illustrated in Figure 2.

3.1. Object Anchoring

In order to refer to objects in the environment, and subsequently learn the spatial relation between objects and the actions that apply to objects, the system must first create and maintain a consistent world model of perceived objects. For this purpose, we rely on the computational theory of perceptual anchoring [7]. Hence, the relation between the perceptual sensory data and symbolic knowledge that refer to the same physical objects is maintained in internal representations called anchors. Figure 1 depicts some examples of anchored objects with the corresponding symbolic information. Note that our technique for object anchoring technique is *bottom-up*, following an approach for anchoring with multiple sensor modalities [8], both visual and non-vision-based. In contrast to traditional symbolically driven *top-down* anchoring, the bottom-up approach permits sensor driven acquisition of anchors.

The first step in our object anchoring procedure is object segmentation to detect objects of interest in the environment. This is based on organized 3D data [9] from the Kinect sensor. Secondly, each segmented object, here denoted as a *percept*, is processed in order to extract both visual and geometrical attributes, as well as grounding each extracted attribute to corresponding predicate symbols. To exemplify, a *color* attribute is extracted as a color histogram (in HSV color space), from the 2D visual part of the percept. This color histogram is then associated, through a *predicate grounding relation*, to a corresponding predicate symbol, e.g. the symbol `'yellow'`. In a similar manner, the *shape* and *position* attributes are extracted from the 3D geometrical part of the percept. In order to categorize objects, we use a Convolutional Neural Network (CNN) architecture, which is based on the 1 K GoogLeNet model [10], suitably modified and fine-tuned for 100 objects categories that are relevant for a household domain, e.g., `'mug'`, `'spoon'`, `'apple'`.

The extracted perceptual and symbolic information for each perceived object is then encapsulated in an internal data structure α_t^x, called *anchor*, indexed by time t and identified by a unique identifier x. The goal of the anchoring system is to manage these anchors. Based on the result of a *matching procedure*, that compares the attribute of an unknown candidate object against the attributes of all previously maintained anchors, anchors are either created or maintained through two general functionalities:

- *Acquire* – initiates a new anchor whenever a candidate object is received that does not match any existing anchor α^x.
- *Re-acquire* – extends the definition of a matching anchor α^x from time $t-k$ to time t. This functionality assures that the percepts pointed to by the anchor are the most recent perceptual representation of the object.

As noted in recent work on probabilistic anchoring [11], proper data association is essential for object anchoring. In ReGround, we have integrated the concept of perceptual anchoring with a joint probability data association technique. The anchoring system uses the beliefs about objects positions, and how objects relate to each other, to compensate for falsely acquired anchors, e.g., as a result of erroneous sensor readings. In order to fully exploit this integration, we have extended a third anchoring functionality, previously defined in [12], as follows:

- *Track* – extends the definition of an anchor α^x from time $t-1$ to time t. This functionality is directly responding to the state of the data association system, which assures that the percepts pointed to by the anchor are an adequate perceptual representation of the object.

3.2. Natural language processing and grounding

One of the aims of ReGround is to develop a novel relational grounding approach that accounts for relationships between *multiple symbols* in the language and *multiple referents* in the environment. We developed a neural model to ground the meaning of multiple words compositionally onto physical objects, their properties and mutual relations. The model transforms an instruction given in natural language into a neural network architecture, which then uses the world representation as input to predict a task-related target. We first applied this model in an artificial task, in order to gain understanding. The task is finding the location of an object that is described by a set of adjectives and/or prepositional phrases using landmark objects in a 3D grid world.

We adapted the neural modules proposed in [13] to this grid world domain. The model consist of two parts, the network layout generator and the neural language grounder. We start with the language grounder part to introduce the neural components, later we describe the algorithm to generate the network layout.

The neural language grounder is a collection of neural modules with connections among them as specified by the layout

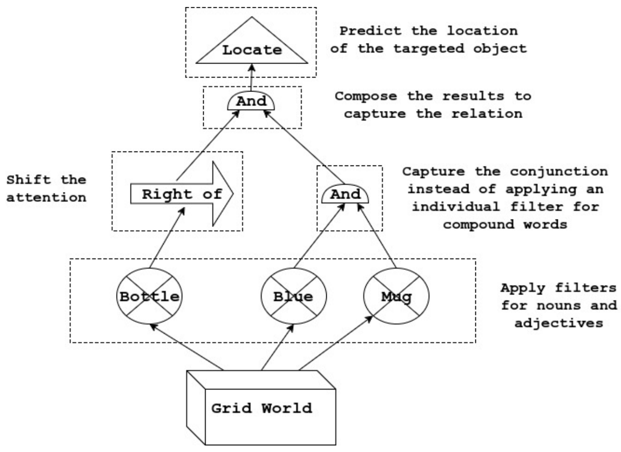
Locate
Predict the location of the targeted object
And
Compose the results to capture the relation
Shift the attention
Right of
And
Capture the conjunction instead of applying an individual filter for compound words
Bottle
Blue
Mug
Apply filters for nouns and adjectives
Grid World

Figure 3: *The layout generated for the instruction: "Pick the blue mug on the right of the bottle".*

generator in response to an instruction — see figure 3. There are four different types of modules, named ***Detect***, ***And***, ***Shift*** and ***Locate***, and each module has a specific grounding role. The ***Detect*** module is a simple convolutional neural network with a single filter that captures a noun or adjective on the grid world. Although we can use ***Detect*** modules to represent the word compounds, the model cannot generalize unseen word compounds even if we have a module for each component of the compound. To overcome this problem, we use an ***And*** module to combine two or more modules. This module element-wise multiplies the output of the incoming modules, and it allows us to capture the composition of nouns and adjectives (e.g., the blue mug) without using additional filters. The ***Detect*** modules produce an attention map over the grid world to capture locations that match a noun or an adjective. The prepositions are represented by ***Shift*** modules which move this attention location in a specific direction, e.g., right of the blue mug. A ***Shift*** module shifts the incoming attention mapping in the direction of the preposition that the module represents. The last module, ***Locate***, takes the output of a subtree and predicts the location of the targeted object by producing a probability distribution over the grid cells.

The layout generator takes the instruction and produces a network layout based on the parse tree of the description. Our algorithm iterates over words and then selects nouns in the input. Then it assigns modifiers to each noun found in the previous step. These nouns usually have a direct correspondence to the objects in our 3D scene. Each noun and its modifier(s) create a filter component and a structure that combines their output. After that we extract prepositions, together with the words that these prepositions relate, from the parser output. Each preposition corresponds to a shift module in our layout. Finally, verbs like Put or Pick tell us how to finalize the layout generation by adding a locate component at the end.

We tested this model in a real world scenario using the ReGround physical table-top setup described above. Our system is triggered by the user with an instruction given in natural language. The model gets the perceptual information from the anchoring system, including the classes, locations and attributes of all the objects, and it maps this information to the grid world internal to the model. The layout generator parses the instruction and generates the neural network layout. The language grounder assembles the corresponding network and predicts the location of the anchor targeted with the instruction. This prediction is sent to the robotic arm and the related action is performed to complete the instruction.

3.3. Learning relational affordances

Learning relational affordances in real-world set-ups relies on two major abilities: i) capturing relations among objects, their properties, actions and other aspects of the environment; and ii) endowing them with probabilities as the real world is inherently continuous and uncertain. The first ability boils down to learning the structure of relational models, while the second one means that we have to describe these models in terms of continuous random variables and enable parameter estimation of probabilistic relation models.

We assume previously processed language or visual primitives is given in the knowledge base (i.e., objects, actions, properties, spatial relations, etc). Our goal is to learn (and then recognize and reason about) relational affordances, that is combinations of primitives, such as object-action, property-action or actionable upon, action-effects (e.g., knife affords cutting). The learning problem can be formalized as: given a set of relational affordance examples consisting of probabilistic facts or relations describing the target predicate and a probabilistic logic theory containing information about the examples, find a rule that defines the probabilistic target relation. Certain affordances or relations can be given as generalized rules expressing pieces of background knowledge (e.g., any reasonably sized knife can be grasped) and also used during learning.

As an illustration, consider the snack scenario in Section 2 where, in addition, there may be also a spoon and a full cup. In this setup, we can learn that the knife allows cutting the paer, given that the knife is elongated, hard and sharp, and the pear is a soft fruit. This goal translates into learning the relational target predicate $\mathtt{affords}/3$ from an example such as $\mathtt{affords}(\mathtt{knife}, \mathtt{cut}, \mathtt{pear})$. Similarly, we could learn via the example $\mathtt{affords}(\mathtt{spoon}, \mathtt{stir}, \mathtt{coffee})$ that a spoon allows stirring the coffee because the spoon is an elongated hard object and coffee is a liquid. Several possible target atoms or learning examples in our scenario are illustrated below. Target predicates could be also labeled with target probabilities:

$\mathtt{1.0} :: \mathtt{affords}(\mathtt{knife}, \mathtt{cut}, \mathtt{pear}).$

$\mathtt{1.0} :: \mathtt{affords}(\mathtt{spoon}, \mathtt{stir}, \mathtt{coffee}).$

$\mathtt{0.5} :: \mathtt{affords}(\mathtt{knife}, \mathtt{stir}, \mathtt{tea}).$

$\mathtt{0.0} :: \mathtt{affords}(\mathtt{knife}, \mathtt{cutUpon}, \mathtt{cup}).$ $(negative)$

$\mathtt{0.02} :: \mathtt{affords}(\mathtt{spoon}, \mathtt{cut}, \mathtt{pear}).$ $(negative)$

There are several relational affordances that can be learned from such examples. Two of them, shown here:

$\mathtt{0.75} :: \mathtt{affords}(\mathtt{X}, \mathtt{cut}, \mathtt{Y}) \leftarrow \mathtt{object}(\mathtt{X}), \mathtt{is}(\mathtt{X}, \mathtt{sharp}), \mathtt{is}(\mathtt{X}, \mathtt{elongated}), \mathtt{allows}(\mathtt{X}, \mathtt{grasp}), \mathtt{is}(\mathtt{Y}, \mathtt{fruit}), \mathtt{is}(\mathtt{Y}, \mathtt{soft}), \mathtt{allows}(\mathtt{Y}, \mathtt{hold}).$

$\mathtt{0.75} :: \mathtt{affords}(\mathtt{X}, \mathtt{stir}, \mathtt{Y}) \leftarrow \mathtt{object}(\mathtt{X}), \mathtt{is}(\mathtt{X}, \mathtt{hard}), \mathtt{is}(\mathtt{X}, \mathtt{elongated}), \mathtt{allows}(\mathtt{X}, \mathtt{grasp}), \mathtt{is}(\mathtt{Y}, \mathtt{liquid}), \mathtt{in}(\mathtt{Y}, \mathtt{cup}).$

are general affordance rules that correspond to the knife-cutting and spoon-stir examples and represent (parts of) the structure of the relational model. The learned rules can be used to predict the target affordance predicate in new situations. For example, the second rule can be exploited by replacing the variable X, which in our scenario is a spoon, with a knife or any stick, if a spoon is not available. The probability of the rule is the degree of belief with which the rule holds, and represents a parameter of the relational model.

To learn such probabilistic relational affordances, we use probabilistic rule learning, given that the example descriptions, their classification and the rules to be learned have a probabilistic (or uncertain) aspect. This setting arises naturally when example descriptions are obtained from perception. By leveraging probability theory and logic, it promises to be the perfect approach to what we need. Our examples (or queries) are facts that are entailed by the given probabilistic logic theory. Furthermore, the probability of a new query represents, in fact, the probability that it is entailed by the theory together with the (potential) background knowledge. As a result we proceed with a setting that learns from entailment and propose two approaches.

In a first approach, we employ ProbFOIL+ [14] which builds on ProbLog [15], a simple probabilistic Prolog, and FOIL [16], a first-order rule learner, and employs principles and heuristics of rule learning. It upgrades the traditional inductive logic programming system FOIL towards a full probabilistic setting and generalizes other probabilistic extensions such as nFOIL and ProbFoil. ProbFOIL+ learns probabilistic rules of the form $\mathtt{p} :: \mathtt{target} \leftarrow \mathtt{body}$. All input (probabilistic) facts for such rules are independent of one another. The probability $\mathtt{p}$ will be determined by the rule learning algorithm which follows the typical sequential covering approach (used by the vanilla rule-learner [17]). In an outer loop, the algorithm starts from an empty set of clauses and repeatedly adds clauses to the hypothesis until no more improvement is observed with respect to a global scoring function. The clause to be added is obtained by a greedy search for a clause that maximizes a local scoring function, using a refinement operator. Besides providing a general probabilistic setting for first-order rule learning, ProbFOIL+: i) integrates rule and parameter learning in a one-step process, rather than learning them separately, and ii) efficiently computes the probability of candidate clauses by combining the sequential covering approach with local optimization.

In a second and extended approach, we use an adaptation of this technique to accomodate also continuous distributions. It employs the DDCTL framework, which extends ProbLog and can be used to build relational affordances via dynamic distributional clauses (DDCs) [18]. Local distributions are modeled using relational regression trees, such that each leaf has a linear or logistic regression model.

4. Conclusions

We have outlined our current work within the ReGround project. ReGround lifts symbol grounding to the relational level, where an agent reasons about the relationships between multiple symbols in the language and multiple referents in the environment, and learns affordances that capture the relationship between objects and their properties, actions, and the environment. Our system integrates techniques from diverse areas such as natural language processing and grounding, object anchoring, and relational affordance learning.

Future work in ReGround will focus on evaluating the ability to use the learned concepts to deal with new situations. The robot will be trained from demonstrations involving inputs from multiple modalities, e.g., language and perception; and it will then be placed in an unseen environment where it will be requested to carry out tasks, possibly using unimodal input, i.e., only language or only perception.

5. References

[1] E. Rome, J. Hertzberg, and G. Dorffner, Eds., *Towards Affordance-Based Robot Control*, ser. Lecture Notes in Computer Science. Springer, 2008, no. 4760.

[2] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner, “Learning predictive features in affordance based robotic perception systems,” in *IROS*, 2006, pp. 3642–3647.

[3] R. Jain and T. Inamura., “Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools,” *Artificial Life and Robotics*, vol. 18, no. 1-2, pp. 95–103, 2013.

[4] M. Steedman, “Plans, affordances, and combinatory grammar,” *Linguistics and Philosophy*, vol. 25, no. 5-6, pp. 723–753, 2002.

[5] V. Krunic, G. Salvi, A. Bernardino, L. Montesano, and J. Santos-Victor, “Affordance based word-to-meaning association,” in *ICRA*, 2009.

[6] P. Gorniak and D. Roy, “Situated language understanding as filtering perceived affordances,” *Cognitive Science*, vol. 31, no. 2, pp. 197–231, 2007.

[7] S. Coradeschi and A. Saffiotti, “An introduction to the anchoring problem,” *Robotics and Autonomous Systems*, vol. 43, no. 2, pp. 85–96, 2003.

[8] A. Loutfi, S. Coradeschi, and A. Saffiotti, “Maintaining coherent perceptual information using anchoring,” in *IJCAI*, Edinburgh, UK, 2005, pp. 1477–1482.

[9] A. Trevor, S. Gedikli, R. Rusu, and H. Christensen, “Efficient organized point cloud segmentation with connected components,” in *3rd Workshop on Semantic Perception Mapping and Exploration (SPME), Karlsruhe, Germany*, 2013.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVPR*, 2015, pp. 1–9.

[11] J. Elfring, S. van den Dries, M. van de Molengraft, and M. Steinbuch, “Semantic world modeling using probabilistic multiple hypothesis anchoring,” *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, 2013.

[12] A. Persson, M. Länkvist, and A. Loutfi, “Learning actions to improve the perceptual anchoring of objects,” *Frontiers in Robotics and AI*, vol. 3, p. 76, 2017.

[13] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Learning to compose neural networks for question answering,” in *Proceedings of NAACL-HLT*, 2016, pp. 1545–1554.

[14] L. De Raedt, A. Dries, I. Thon, G. Van den Broeck, and M. Verbeke, “Inducing probabilistic relational rules from probabilistic examples,” in *IJCAI*, 2015, pp. 1835–1843.

[15] L. De Raedt, A. Kimmig, and H. Toivonen, “Problog: A probabilistic prolog and its application in link discovery,” in *IJCAI*, M. M. Veloso, Ed., 2007, pp. 2462–2467.

[16] J. R. Quinlan, “Learning logical definitions from relations,” *Machine Learning*, vol. 5, pp. 239–266, 1990.

[17] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[18] D. Nitti, I. Ravkic, J. Davis, and L. De Raedt, “Learning the structure of dynamic hybrid relational models,” in *ECAI*, vol. 285, no. 22, 2016, pp. 1283–1290.