Advances in Cognitive Systems 6 (2018) 137–156 Submitted 6/2018; published 12/2018

# Learning Cognitive Affordances for Objects from Natural Language Instruction

**Vasanth Sarathy** VASANTH.SARATHY@TUFTS.EDU
**Bradley Oosterveld** BRADLEY.OOSTERVELD@TUFTS.EDU
**Evan Krause** EVAN.KRAUSE@TUFTS.EDU
**Matthias Scheutz** MATTHIAS.SCHEUTZ@TUFTS.EDU
Department of Computer Science, Tufts University, Medford, MA 02155 USA

## Abstract

Affordance perception refers to the ability of an agent to extract meaning and usefulness of objects in its environment. Cognitive affordance is a richer notion that extends traditional aspects of object functionality and action possibilities by incorporating the influence of changing context, social norms, historical precedence, and uncertainty. This allows for an increased flexibility with which to reason about affordances in a situated manner. Existing work in cognitive affordances, while providing the theoretical basis for representation and inference, does not describe how they can be learned, integrated, and used with a robotic system. In this work, we describe, demonstrate, and evaluate an integrated robotic architecture that can learn cognitive affordances for objects from natural language and immediately use this knowledge in dialogue-based learning and instruction.

## 1. Introduction

Using and manipulating objects in the environment requires a cognitive ability to perceive and evaluate their meaning, applicability and usefulness in relation to our own abilities to take action. Such a relational notion, known as an *affordance*, links action and behavior possibilities with objects and features present in the environment, enabling the ability to guide our behavior (Gibson, 1979; Zech et al., 2017). In robotics and AI, affordances have served as the underlying theory for action perception and have been modeled using relational and machine learning techniques such as Bayesian networks, Markov logic networks, conditional random fields, and reinforcement learning (Steedman, 2002; Montesano et al., 2007; Ugur et al., 2015; Koppula & Saxena, 2016; Sridharan & Meadows, 2017). While much of the affordance literature in robotics has focused on object or environmental affordances, some have considered "social affordances" and offered an approach to perceiving visual cues offered by social scenarios involving other agents (e.g., raised arm signaling a high-fiving affordance) (Shu et al., 2016). However, these methods do not allow for socio-contextual dependency on *object* affordances.

Sarathy and Scheutz (2016; 2018) proposed a theory of cognitive affordances to address this problem. Their theory of cognitive affordances uses a probabilistic-logic based approach capable of inferring affordances in the face of changing contexts, social norms, and epistemic uncertainty, i.e.,

it accounts for those object affordances influenced by factors beyond perceptual cues on the objects themselves ("cognitive affordances"). However, the question of how exactly to *learn* these cognitive affordances and *utilize* them on a robot is still open. The problem is especially difficult because these sorts of socio-contextual dependencies are, for humans, learned through a few exposures or instructions, and not through numerous trials and errors.

In this paper, we address these open questions directly and propose two novel contributions: (1) a grounding and integration of cognitive affordance representation within a cognitive robotic architecture, and (2) an approach to learning these cognitive affordances from natural language instruction in the presence of epistemic uncertainty. The proposed approach allows for encoding, learning and immediately actualizing of a broad class of normatively-charged cognitive affordances, accounting for aspects of objects that the agent can directly perceive (e.g., object features) and aspects that are not self-evident or directly perceivable from the object itself (e.g., context and social convention associated with the object, goals of the agent).

We will use a kitchen-helper robot from Sarathy and Scheutz (2018) as our running example, with the robot learning, from instruction, how to properly grasp a knife when using it and when handing it over to someone (at the blade). Although the proposed approach is not limited to this particular example, or even embodied robotic systems for that matter, a concrete example of this sort will help tie it to past work and explain various aspects of the representation, inference algorithm, learning approaches and integration with a cognitive robotic architecture, to allow for normative behavior capabilities.

## 2. Theoretical Aspects of Cognitive Affordances

We are interested in the class of *affordances* that possess additional properties and dimensions beyond the simple Gibsonian notions (e.g., "sitability of a chair"). As noted earlier, this class of cognitive affordances is deeply influenced by contextual and normative factors including goals and intentions, prior knowledge and interpretations, ensemble scene information, mental state, experience and developmental state, social and moral conventions, and aesthetic considerations among others. We will build on a recent theoretical model of cognitive affordances proposed by Sarathy and Scheutz (2016; 2018) that represents affordances as condition-action rules ($R$) where the left-hand sides represent perceptual invariants ($F$) in the environment together with contextual information ($C$), and the right-hand sides represent affordances ($A$) actualizable by the agent in the situation (e.g., the rule that one should grab a knife by the handle when using it would be translated by specifying the grasping parameters as $F$, the task context of "using a knife" as $C$ and the constrained grasping location together with other action parameters as $A$). Affordance rules ($R$) take the form

$$r \stackrel{\text{def}}{=} f \wedge c \underset{[\alpha,\beta]}{\Longrightarrow} a \ ,$$

with $f \in F$, $c \in C$, $a \in A$, $r \in R$, and $[\alpha, \beta] \subseteq [0, 1]$, where $[\alpha, \beta]$ is a confidence interval intended to capture the uncertainty associated with the truth of the affordance rule $r$ such that if $\alpha = \beta = 1$ the rule is logically true, while $\alpha = 0$ and $\beta = 1$ assign maximum uncertainty to the rule. Similarly, each of the variables $f$ and $c$ also have confidence intervals associated with them, and are used for

inferring affordances as described in more detail below. Thus, rules can then be applied for a given feature percept $f$ in given context $c$ to obtain the implied affordance $a$ under uncertainty about $f$, $c$, and the extent to which they imply the presence of $a$.

Given a set of affordance rules, we can determine the subset of applicable rules by matching their left-hand sides given the current context and perceivable objects in the environment together with their confidence intervals, and then determine the confidences on the fused right-hand sides (in case there are multiple rules with the same right-hand side) based on the inference and fusion algorithm in Sarathy and Scheutz (2018). We will use the confidence measure $\lambda$ defined by Nunez et al. (2013) to determine whether an inferred affordance should be realized and acted upon. For example, we could check the confidence of each affordance on its uncertainty interval $[\alpha_i, \beta_i]$: if $\lambda(\alpha_i, \beta_i) \leq \Lambda(c)$, where $\Lambda(c)$ is an confidence threshold, possibly depending on context $c$, we do not have enough information to confidently accept the set of inferred affordances and can thus not confidently use the affordances to guide action. However, even in this case, it might be possible to pass on the most likely candidates to other parts of the integrated system. Conversely, if $\lambda(\alpha_i, \beta_i) > \Lambda(c)$, then we take the inferred affordance to be certain enough to use it for further processing.

From a systems standpoint, in order to process cognitive affordances, several functional units were proposed by Sarathy and Scheutz (2018). During inference, the functional units are meant to search through all available affordance rules of the form specified above in the agent's long term memory and populate a working memory with the relevant rules. Once the rules are in the working memory, the system can use these rules as the basis for perception and inference. An example cognitive affordance rule instantiation in this past work had the form

$$r \stackrel{\text{def}}{=} hasSharpEdge(O) \wedge domain(X, kitchen) \underset{[0.8,1]}{\Longrightarrow} cutWith(X, O) \; .$$

While this work presented some crucial early theoretical foundations for using and performing inference with cognitive affordances, it was missing two key components. First, the past work did not suggest how these rules could could be grounded in a robotic system. For example, Sarathy & Scheutz (2018) state that the results from affordance inference are "passed to the robot's action management system," but they do not discuss how exactly this interaction might work and how an action management system might be able to use this information in connection with its own action repetoire and action knowledge. Thus, an open question is how can an agent use $cutWith(X, O)$, and what exactly do the predicates and variables in the logical representation mean in a robotic architecture. In this paper, we describe such a grounding for an exemplary architecture and provide a grounded rule representation consistent with the cognitive affordance theory, but also tightly integrated with the robot's actuation and perceptual systems. In doing so, we will also need to revisit and modify the above-mentioned cognitive affordance rule example to tie the predicates in the rule representation to perceptual and action knowledge actually available in the system as well as contextual knowledge associated with the task the agent is performing.

Moreover, while Sarathy and Scheutz have outlined an approach for performing inference with cognitive affordance rules, it is still an open problem as to how these rules might be learned. Here, we propose a solution based on learning from instruction, which at times, might be the only option available to an agent, for example, in situations where the agent does not have enough time to observe or if the agent is not able to collect enough observational data. Recent work by Scheutz

et al. (2017) discusses an approach for learning percepts and actions from instruction. Here, we propose extending this approach for learning not only perceptual and action predicates, but the rules themselves. By combining these ideas from past work, we provide a novel approach for learning normatively-guided affordances from natural language.

## 3. Grounding and Learning

To choose and manipulate everyday objects in socially-appropriate and context-dependent ways, we claim that *any cognitive system will require mechanisms for learning, representing and immediately applying arbitrary socio-contextual rules associated with these objects*. While the cognitive affordance theory provides a suitable rule representation, the rules must be grounded within the cognitive system (Section 3.1). Action management components must be able to guide perceptual and action components to check if a rule applies, and then apply the rule by constraining action choices and parameters, all under conditions of epistemic uncertainty (Section 3.3). Moreover, much of these social norms are conveyed via natural language. Thus, the natural language components must be equipped to parse speech into the grounded rule representations (Section 3.2).

### 3.1 Enabling Affordance Processing in a Cognitive Robotic Architecture

To integrate affordance processing into a cognitive robotic architecture, we developed a separate component for maintaining the affordance rules and the inference algorithm for DIARC, an example architecture (Scheutz et al., 2007). In addition we updated several components of the architecture to be able to handle the new types of information enabled by the new affordance component. This grounding within the architecture gives the affordance reasoning mechanisms described in Section 2 a concrete medium through which new rules can be added dynamically based on an agent's interactions with its environment. This extends the functionality and utility of the theoretical model which previously was limited to a fixed set of abstract rules.

We selected DIARC over other cognitive architectures, such as SOAR (Laird et al., 1987) and ACT-R (Anderson et al., 2004), because of its integration of social behaviors, and specifically its natural language understanding and production capabilities, which allow for more natural human-robot interaction, as well as the ability to lean new concepts through natural language (Scheutz et al., 2017). None of the current architectures (including DIARC) are currently able to represent and reason about cognitive affordances.

So regardless of the choice of architecture, an affordance component of the type described here could be desirable to enable affordance processing and enhanced social interaction capabilities. Whichever architecture is chosen, the affordance component will still need to be connected to other high and low level components in order to influence perception and action.

Figure 1 depicts the integration of the AFFORDANCE component (AFF in the figure) in DIARC. The subcomponents of AFFORDANCE work closely with sensory and perceptual systems (e.g., vision) and other components in the architecture to coordinate perceptual and action processing. AFFORDANCE is connected to the GOAL MANAGER (GM/AM) component and, during execution of actions, GOAL MANAGER sends affordance requests to AFFORDANCE. These requests provide information about the current action to be performed and the context. AFFORDANCE returns the

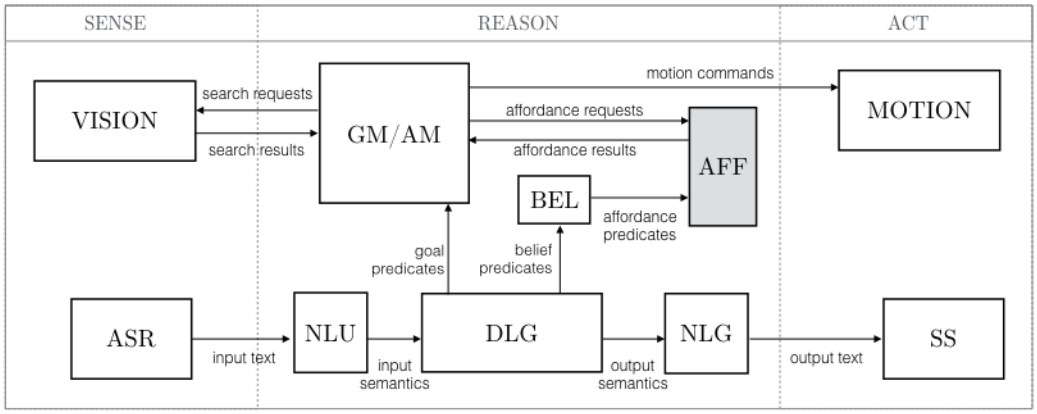


*Figure 1.* Diagram of the extended DIARC architecture, with the AFFORDANCE (AFF) component highlighted. Other relevant components are SPEECH RECOGNITION (ASR), NATURAL LANGUAGE UNDERSTANDING (NLU), DIALOGUE MANAGER (DLG), NATURAL LANGUAGE GENERATION (NLG), SPEECH SYNTHESIS (SS), BELIEF MODEL (BEL), MOTION CONTROL (MOTION), VISION, and GOAL MANAGER/ACTION MANAGER (GM/AM). During operation, AFF receives semantic information, uses GM/AM to direct VISION to look for environmental features relevant to social norms, and then guides MOTION via GM to perform a socially-appropriate action.

specific perceptual features that must be found in the environment. This allows GOAL MANAGER to direct the attention of low-level perceptual modules like VISION to search in a focused manner, only looking for perceptual features relevant to the applicable rules in AFFORDANCE. The presence or absence of these perceptual features (along with information about perceptual uncertainty) is passed back to AFFORDANCE, which performs uncertain logical inference (logical AND and modus ponens) on the rules.

In dialogue-driven tasks, GOAL MANAGER receives language-based goals via the natural language pipeline (ASR $\rightarrow$ NLU $\rightarrow$ DLG), while the BELIEF (BEL) component is the recipient of language-based knowledge. BELIEF maintains a history of all declarative knowledge passing through the architecture and is capable of performing various logically-driven knowledge-representation and inference tasks. Thus, it serves as a convenient holding area for cognitive affordance information that has been partially processed through the natural-language pipeline, which can then be retrieved and processed by AFFORDANCE.

Integrating AFFORDANCE into DIARC, or any cognitive architecture for that matter, requires more than simply depositing it into the system. Various other components (GOAL MANAGER, VISION, BELIEF MODEL, etc.) must also be modified so they can provide the additional capabilities required for cognitive affordance processing. For example, the natural-language pipeline must be updated to allow for the recognition and understanding of affordance-related words (“cutting”, “sitting”, “enclosing” etc.), and GOAL MANAGER must be updated to recognize these semantic representations and consult AFFORDANCE at the appropriate points in action selection and execution. In the next sections, we will discuss in more detail the specific architectural modifications and the resulting functionality that enables these new operations.

As an additional benefit, these modifications provide the architecture with the ability to understand references to objects by their affordance (e.g., a knife as not just an object with some visual property, but as an "object used for cutting"). An affordance-enabled cognitive robotic architecture also allows an agent to account for context, and helps constrain and guide behavior. Actions need no longer be performed the same way each time, but can vary depending on context. For example, a kitchen helper robot may grab a knife differently if the context of the grab is that the robot will use it to cut something, as opposed to the robot grabbing it so it can be handed to a human. Or it might carry plates of food differently in the context of serving them versus the context of busing a table.

### 3.2 Learning Affordance Rules from Instruction

As mentioned earlier, we will use as our guiding example the two instructions "a knife is often used for cutting" and "to pickup a knife grab it by the handle". These are presented to a robot that does not know about a functional affordance of a knife ("cutting") that is epistemically uncertain ("often") or that its grasp affordance ("pick up") is context sensitive ("by the handle").

We previously outlined how the affordance model described in Section 2 can be integrated into a cognitive robotic architecture to expand the capabilities of the robot. This integration also gives that model a mechanism through which new rules can be added on the fly, letting it better adapt to and represent real-world scenarios. In order to do this, various DIARC components must be extended. Natural language utterances that contain cognitive affordance rules must be converted to general-purpose facts stored in BELIEF and then used by AFFORDANCE to generate the rules described in Section 2, which can then ultimately be used to perform uncertain logical inference. With these extensions to existing DIARC components, we are able to leverage DIARC's mechanisms for learning through instruction (e.g., Scheutz et al., 2017) to enable learning new affordance rules about concepts the agent already *understands*, as well as completely novel concepts that have been learned on the fly.

The role of the Natural Language Understanding component is converting the text form of spoken utterances into a semantic form which can be *understood* (used) by the other components within DIARC. We extended this component through the addition of new parsing and pragmatic inference rules which enable the generation of new semantic forms.

In order to learn from natural language instructions, a cognitive robotic architecture must be able to ground the content of the utterances containing the instructions in terms that it can understand. AFFORDANCE understands affordance rule descriptions that are represented in the predicate form,

$$implies(antecedents, consequents, confidence)\ ,$$

where the predicate's arguments represent the antecedents, consequents, and confidence interval of an affordance rule. The natural-language processing components of DIARC (ASR, NLU, and DM in Figure 1) convert spoken language into a predicate of this form and assert it into BELIEF.

When an utterance is spoken to the agent, the SPEECH RECOGNITION (ASR) component converts the acoustic speech signals to text. The NATURAL LANGUAGE UNDERSTANDING (NLU) component receives the utterance in text form from the speech recognition component and performs two steps of processing. The first step parses the text into a form that can be used by the rest of the system. The

*Table 1.* A subset of the relevant rules used by the NATURAL LANGUAGE UNDERSTANDING component.

| Label | Syntax | Semantics |
|---|---|---|
| to | (S/C)/C | $\lambda x \lambda y.implies(x, y, high)$ |
| pickup | C/NP | $\lambda x.pickUp(?ACTOR, x)$ |
| a | NP/N | $\lambda x.x$ |
| knife | N | $knife$ |
| grasp | C/NP | $\lambda x.grasp(?ACTOR, x)$ |
| the | NP/N | $\lambda x.x$ |
| by | (NP/NP)\NP | $\lambda x \lambda y.partOf(x, y)$ |
| handle | N | $handle$ |

second performs pragmatic inference to add a notion of the speaker's intent to the representation of the utterance (Scheutz et al., 2013).

In the parsing step, the natural-language understanding component uses a parser to determine the syntactic structure and the semantic interpretation of the utterance. The parser used in this configuration of DIARC is an extended incremental version of the Combinatory Categorial Grammar parser from Dzifcak et al. (2009), described in more depth by Scheutz et al. (2017). It contains a dictionary of parsing rules each composed of three parts: a lexical entry, a syntactic definition, and a semantic definition in lambda calculus. An example set of rules can be found in Table 1. These rules are a subset of the complete set of rules used by the system. They are selected because of their relevance to the empirical demonstration in Section 4.

An example of a cognitive affordance rule spoken in natural language and its accompanying semantics are

"To pickup a knife grab it by the handle" .

$$STATEMENT(Sam, self, implies(pickUp(self, knife), graspObject(self, partOf(handle, knife)), high)) .$$

Here, "Sam" is the name of the human (and trusted source) speaking to the robot. This representation denotes a statement from Sam to the agent, whose semantics are the *implies* predicate above.

The parsing step produces a notion of the meaning of the spoken utterance. The pragmatic inference step uses that meaning and a set of inference rules to determine the speaker's intention. The pragmatic inference system used in our configuration of DIARC is described in work by Scheutz et al. (2013). In the case of our working example the semantic representation generated in the parsing step matches the left-hand side of

$$STATEMENT(A, B, X) \implies wantBelieve(A, B, X),$$

which is a general rule for utterances of the type *STATEMENT*, and can be interpreted as "when a person tells an agent something it wants the agent to believe it". The resulting DIARC representation produced by the NATURAL LANGUAGE UNDERSTANDING component is the predicate

$$wantBelieve(Sam, self, implies(pickUp(self, knife), graspObject(self, partOf(handle, knife)), high)) .$$

This semantic representation from NATURAL LANGUAGE UNDERSTANDING is received by the DIALOGUE MANAGER, whose role is to respond appropriately to utterances from other agents. In the case of our example, DIALOGUE MANAGER recognizes that Sam wants the agent to believe a predicate. It checks if Sam is a trusted source of information and, if so, asserts the predicate into BELIEF. Upon confirmation that the information has been successfully stored, the module submits a goal to GOAL MANAGER to verbally acknowledge understanding.

Once the information from the utterance has been asserted into BELIEF it is accessible to AFFORDANCE. Epistemically, an utterance is a piece of evidence received by the agent in support of the truth of the affordance rule it represents. Thus, we use the confidence directly to represent the degree of support for the rule.[1] The confidence value may be used to capture the inherent uncertainty in the utterance (e.g., when qualifiers such as "sometimes" or "maybe" are used), the trust placed in the interlocutor (e.g., a rule taught by a superior or boss may hold more water), the uncertainty in speech detection mechanisms, or some combination of these factors. The linguistic placeholder "high" represents a preset confidence (0.95). That value is used because the speaker is a priori known to be trustworthy by the agent, but nothing in our system requires this particular, method of assigning confidence values.

Functional affordances can be learned in the same way as the action affordance that we described above. For example, the utterance "A knife is sometimes used for cutting" would be translated to the DIARC predicate representation $implies(knife, cutting, mediumLow)$ and then deposited into the BELIEF model.

Scheutz et al. (2017) describes how DIARC agents can learn new concepts on the fly through natural language instruction. When the agent encounters an unknown word it is able to infer its syntax and semantics based on the parser's knowledge about the syntax and semantics. In the case of utterances related to cognitive affordance inference rules, the syntax and semantics of previously unknown antecedents or consequents can be inferred by recognizing the pattern of the rest of the utterance. Novel consequents or antecedents introduced this way can be recognized in subsequent utterances and their representation in the set of rules in AFFORDANCE will be consistent. This enables the agent to understand cognitive affordance rules with previously unknown consequents and antecedents, which provides the agent the ability to continuously adapt its knowledge base.

To clarify, it is worth noting that the architecture proposed by Scheutz et al. was limited to learning concepts that have direct perceptual correlates (speech signals or visual attributes) and was not able to learn and utilize nonperceptual or cognitive concepts (like cognitive affordances). These involve nonperceivable attributes (contexts), and relationships between agent capabilities (actions) and perceptual entities (visual features) tied together in compact natural language utterances. In the next section, we describe how an agent having learned cognitive affordance rules can apply this knowledge immediately in a command-based task.

1. The "confidence" here is different from the confidence measure $\lambda$ discussed in Section 2. $\lambda$ is a singular measure of the degree of uncertainty for an uncertainty interval (somewhat akin to the width of the interval) typically used in conjunction with Dempster-Shafer theory of uncertainty. We can use $\lambda$ when executing affordance-based commands (Section 3.3) and deciding which action to perform when there are multiple choices. However, the confidence value mentioned here is used to directly represent the degree of support for rule, i.e., it represents the single-valued precision assigned to the rule when received as evidence via an utterance.

*Figure 2.* Left: knife; Right: Grasp candidates across the knife. Cognitive affordances can serve as a normative constraint when selecting one of the many grasp possibilities.

### 3.3 Executing Affordance-Based Commands

There are numerous examples of DIARC and other cognitive robotic architectures enabling robots to engage in task-based dialogues where a human instructs a robot to perform tasks using commands given in natural language. The integration of AFFORDANCE into such architectures allows for incorporation of affordance information when discussing a task. This supports more natural dialogue and gives the human and robot more flexibility in the objects they discuss.

Returning to our running example of using a knife, consider a human uttering a command to the robot: "Pass me something used for cutting." Currently, DIARC would fail because the GOAL MANAGER would not be able to handle pairing a known action of "passing" with a nonspecific object reference "something" and a functional affordance concept of "cutting." Moreover, even a more specific request of "pass me a knife" would often fail, because there is no guarantee that the robot will choose to pass the knife by grasping the blade (the normatively appropriate option), as opposed to the handle, which has similar – if not better – grasp possibilities. As shown in Figure 2, taken from Scheutz et al. (2017), there are many available grasp candidates distributed all across the knife on the handle and on the blade.

In order to *understand* the command, the information contained in it needs to be grounded within the system. We start with the system knowing nothing about knives or how to pass them. We use the features of DIARC described in Scheutz et al. (2017) to teach the system what a knife is and how to pass something. At this point, the robot knows that an observed 3D point cloud is a "knife" and that certain subsets of this point cloud constitute "handle" and "blade". Using the object grasping mechanism described in Ten Pas and Platt (2014), it is capable of generating candidate grasp points (from the geometry of the point cloud) and then scoring these grasp points to determine which ones are likely to succeed. We use a four-layer deep convolutional neural network to make grasp predictions based on projections of antipodal grasp points contained between fingers.

Using the approach described in Section 3.2, we assume that a human has taught the robot cognitive affordance rules about a knife in three utterances as follows:

"A knife is used for cutting",

"To pickup a knife grab the knife by the handle",

"To pass a knife grab the knife by the blade."

As described earlier, the BELIEF component produces five predicates:

$implies(knife, cutting, high)$,
$implies(pickUp(self, knife)), graspObject(self, partOf(handle, knife)), high$,
$implies(pass(self, knife), graspObject(self, partOf(blade, knife)), high)$.

Now that the system understands how to pass knives in the context of cutting, we can instruct it to do so using the natural language mechanisms described earlier:

1. Utterance: "pass me something used for cutting"
2. Parse: $INSTRUCT(Sam, self, pass(self, usedFor(something, cutting)))$
3. Relevant Pragmatic Rule:
   $INSTRUCT(A, B, X) \implies want(A, X)$
4. DIARC Semantic Representation:
   $want(Sam, pass(self, usedFor(something, cutting)))$
5. Submitted Goal Predicate:
   $pass(self, usedFor(something, cutting))$

Upon goal submission GOAL MANAGER executes the action script associated with the goal. An action script is hierarchically organized with actions and subactions, with bottom-level actions representing commands issued to the action component (MOTION CONTROL). The hierarchy for the "pass" action is shown in Figure 3. Executing an action script of this form involves performing a preorder traversal of its tree. At each node, we carry out three operations for applying learned affordances, in addition to the operation related to the action itself.

First, an affordance request is sent to AFFORDANCE to $getFeatures()$, which involves assimilating newly learned affordance rules, identifying relevant affordance rules, and returning perceptual invariants (F) from the antecedents. AFFORDANCE queries BELIEF for any new $implies(X, Y, Z)$ predicates learned since its last call, then maps the arguments into the perceptual invariants (F), contextual items (C), and affordances (A) in the affordance model. Here, we assume that context itself is a higher-order action and therefore is captured as a functor name in the DIARC semantic representation of the utterance. Thus, "pass" is the action context in $implies(pass(self, knife), graspObject(self, partOf(handle, knife)), high)$. We recognize that context is not always knowable or definable in advance but, in this situation, contextual information is explicitly provided in the utterance and is therefore available for the system to use. In other instances, the context may be implicit and the agent may need to infer it; the approach does not preclude such inference because the affordance model is general enough to capture contextual predictions regardless of how they are obtained, but we leave it for future work. The perceptual invariant (F) is available in the argument to the action context and thus, for example, "knife" is a perceptual invariant to be added to the rule. The affordance information (A) is available from consequents where the "graspObject" predicate is flattened. This process leads to three affordance rules in our motivating example,

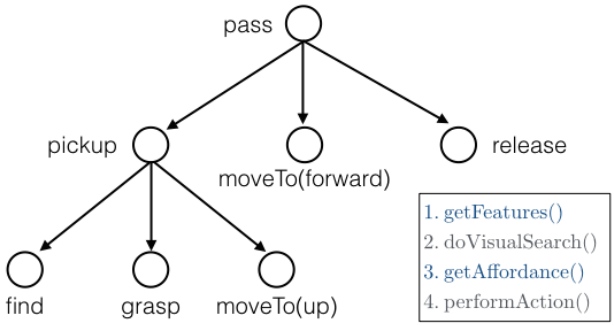


*Figure 3.* Key affordance-related operations during action execution using an exemplary *pass* action script. For every node in the action tree four operations are performed: extracting perceptual features and contextual items from the relevant affordance rules, running a visual search to determine whether these features exist in the agent's environment, performing inference with the rules and observations to obtain constraints on actions, and performing the action with inferred constraints.

$$r^1 \stackrel{\text{def}}{=} knife(K) \underset{[0.95,1]}{\Longrightarrow} findObject(cutting, knife(K)) \ ,$$

$$r^2 \stackrel{\text{def}}{=} knife(K) \wedge context(C = pickup) \underset{[0.95,1]}{\Longrightarrow} graspObject(knife(K), handle(P), partOf(P, K)) \ ,$$

$$r^3 \stackrel{\text{def}}{=} knife(K) \wedge context(C = pass) \underset{[0.95,1]}{\Longrightarrow} graspObject(knife(K), blade(P), partOf(P, K)) \ .$$

Once the rules have been updated to include new additions from BELIEF, AFFORDANCE selects the rules relevant to the current situation. We do not provide an in-depth comparison of various rule-selection approaches, but we take a straightforward approach that selects those rules relevant to the current action (i.e., the current active node in the action tree) and affordance. We select rules with consequents containing functor names that match the current action. This is possible since the syntax and semantics of the affordance predicates match the grounded representations of actions in GOAL MANAGER and MOTION CONTROL. In addition to the current action, we use goal predicate information (including affordance) obtained from the current command to further prune the rules if necessary. AFFORDANCE obtains this information by querying BELIEF for $usedFor(X, Y)$.

During the "find" action, the only match is rule $r^1$ and thus the only rule that is selected is associated with functional affordance of the knife. The output from $getFeatures()$ is then sent as a search request by GOAL MANAGER to VISION to locate them in the agent's visual field of view. For example, the perceptual invariant ($knife(K)$) obtained in the "find" action is then sent back to GOAL MANAGER, which then initiates a visual search to look for a knife. Upon finding a match, VISION provides a detection confidence for the object it has identified as being a knife and AFFORDANCE uses this confidence to perform inference to determine if the deduced action "find" is above a certain confidence threshold. If so, the object identified as a knife is selected for further processing.

If a representative visual object is identified, then a second request is made to the affordance component to $getAffordance()$, during which affordance inference is performed and the best action or object constraint is returned. The constraints are then used in connection with the motion

commands and sent to MOTION CONTROL. Thus, instead of the general command to grasp a knife, which could result in the agent selecting one amongst a countless number of high-scoring grasp candidates on the knife, the agent may be constrained to only selecting those on the handle.

At the next node (*grasp*), this process is repeated, but now there are two rules associated with *grasp*. However, only one is associated with the context of "pass." Thus, inference is performed on this one rule and the constraints $knife(K), blade(P), partOf(P, K)$ is returned and used for identifying grasp points on the blade of the knife. The agent can then correctly (from a normative standpoint) pass the knife by the blade.

## 4. Evaluation

We take a two-step approach to evaluating the affordance-enabled cognitive robotic architecture. First, we evaluate whether the system exhibits correct behavior (i.e., is it taking the correct actions when instructed with a cognitive affordance-based natural language command?). We do so through an extended simulation, in which the entire architecture was tested without external noise or sensor fluctuations that typically occur in real-world settings. Clearly, real-world runs of the system are important and show how the architecture can perform, not just in simulation, but on an embodied robot in which real-time constraints apply. So, in our second step of the evaluation we tested the architecture on a PR2 robot and provide an uncut video. We describe each step of the evaluation in this section.

### 4.1 Simulation Experiment and Empirical Demonstration

We first tested the correctness of the approach in an extended simulation involving several household objects and over two dozen rules. As noted earlier, the goal was to be able to test the approach in a simulation without perceptual and sensory noise experienced in the real world in order to focus on evaluating the correctness of the underlying algorithms.

For the experiment, we considered eight household objects, each composed of two parts: knife (handle, blade), spatula (handle, blade), spoon (handle, bowl), shoe (upper, sole), hammer (handle, head), glass (bowl, stem), mug (handle, barrel), and screwdriver (handle, shaft). We considered five different affordances in the spirit of those used in computer-vision data sets associated with affordance detection (Myers et al., 2015; Varadarajan, 2015): containing, cutting, pounding, rolling, and poking. We restricted the agent's action repertoire to the actions *pass*, *pickUp*, and *pointTo*. Unlike *pass* and *pickUp*, *pointTo* involves finding but no grasping. With these objects and actions, we generated 15 different commands (five affordances for each of three actions) of the form "[action] something used for [affordance]" (e.g., point to something used for pounding).

We are interested in learning functional affordances and action affordances that contain a notation of confidence. To capture this, we used four terms to represent different degrees of confidence – occasionally, sometimes, often, and generally – which we then mapped to specific numerical values (Kerdjoudj & Curé, 2015). Thus, given eight objects, five affordances, four uncertainties, and three actions, we can generate 288 possible affordance rules, comprising 160 functional affordance rules and 128 action affordance rules.

*Table 2.* Ground truth rules in Scenario A, with Scenario B obtained by reversing the list of each affordance. Point confidences in parentheses.

| **Affordance** | **Generally (0.95)** | **Often (0.75)** | **Sometimes (0.5)** | **Occasionally (0.25)** |
|---|---|---|---|---|
| containing | [mug, glass] | | spoon | shoe |
| cutting | knife | | screwdriver | [spatula,spoon] |
| pounding | hammer | shoe | spatula | mug |
| rolling | | glass | screwdriver | |
| poking | | screwdriver | knife | |

In any given learning scenario, the agent is taught a set of rules chosen from these 288 possible rules, thus generating $2^{288}$ different possible trajectories. Also, since we have eight objects, we can generate 256 possible scenes involving these objects that, when combined with the 15 possible commands, gives 3840 different problem situations (scene-command combinations). Testing the architecture across all possible learning scenarios ($3840 \times 2^{288}$) and problem situations is infeasible.

Instead, we evaluated the system by (1) choosing a random subset of our evaluation space, and (2) establishing some general performance expectation for the system in this space. We limited our evaluation space by randomly choosing ten different scenes and testing the agent's performance for all 15 commands. With regards to the selection of rules, we arbitrarily chose two different rule sets representing two different normative standards and provided some expectations for how the agent should act based on these two distinct learning scenarios. In Scenario A, for each of the five affordances, we generated a list of objects (ranked highest to lowest confidence) that possess this affordance. In Scenario B, we reversed the ranking of objects. For example, in Scenario A, a mug is top-ranked object for the containing affordance while a shoe is a bottom ranked (but still feasible) object. In Scenario B, the shoe is top-ranked and the mug is bottom ranked. These two scenarios represent our ground truth rules, which have the values shown in Table 2.

We used Table 2 to derive functional affordance rules of the form "[object] is [uncertainty] used for [affordance]" (e.g., "a spatula is sometimes used for pounding"). For action affordances, we generated 16 rules corresponding to physical grasp affordance rules in the *pass* and *pickup* context for all eight objects, with a single confidence setting of "generally." For example, "To pass a shoe, generally grab the shoe by the sole." Of the 288 rules initially stated, many are somewhat nonsensical by our own normative or practical standards (e.g., a mug being used for cutting). However, the robot did not know this, and we therefore expected it to perform the necessary affordance inference without this additional commonsense knowledge.

We performed multiple trials during which we generated various tabletop scenes using combinations of the eight objects. We tested both learning sets of affordance rules, and acting on sets of commands using the proposed architecture. We evaluated the performance of the system by checking if four principles held true in all trials: (1) if all the objects are in the scene, then the robot must select the top-ranked object for the required affordance; (2) if the top-ranked object is not available, then the robot must select the next lower ranked object; (3) if there is more than one top-ranked object with equal measures of confidence, then the robot may select either; and (4) if there are

*Table 3.* High-level syntax of understandable utterances, in JSpeech Grammar Format (JSGF).

| | **Utterance Templates** |
|---|---|
| <statement> | a <object> is [<qualifier>] <implies> <use> \| to <goal> a <object> [<qualifier>] <primitive> <object> <mod> <part> |
| <command> | [now \| okay \| first \| then] (<goal> \| <primitive>) something <implies> <use> \| <primitive> <object> <mod> <part> |
| | **Grounded Concepts** |
| <qualifier> | sometimes \| often \| generally \| always |
| <object> | mug \| knife \| wine glass \| spatula \| spoon \| shoe \| screwdriver \| rock |
| <implies> | used for |
| <primitive> | grab \| grasp |
| <goal> | pass \| pickup \| point to |
| <mod> | by the |
| <part> | red \| green \| blue |
| <use> | cutting \| containing \| pounding \| rolling |

no objects available with the required affordance, the robot must tolerate the failure condition and provide a suitable response.

We ran the experiment across ten randomly generated scenes of varying sizes, including one with all the objects. During each run, we taught the 32 above-mentioned rules in each of Scenario A and B, then we presented a randomly generated scene and issued each of the 15 commands in sequence. We ensured that sets of scenes in combination with the commands covered the above-mentioned four performance expectations. Table 3 shows the general form of the two types of affordance-related utterances (Utterance Templates) our system can handle and the component parts of those templates that can be expanded as needed for the system's applications (Grounded Concepts), provided they can be grounded in the architecture. It is important to note that these utterance types are not the only language DIARC can understand: they are *added functionality* that coexists with prior functionality.

In this evaluation, we are interested in whether the integrated system correctly learned the rules from natural language expressions and then immediately applied this knowledge correctly to select the best action. The two learning scenarios described above provide a ground truth of sorts, as the objects are ranked from best to worst in each scenario. It is important to note here that we are not interested in evaluating robustness of the underlying low-level perceptual and action systems themselves. Accordingly, we avoid sensor noise and motion imperfections by running this experiment as a simulation and focus exclusively on evaluating the proposed architecture with AFFORDANCE. Moreover, the rule uncertainties were set to four distinct and separated values to ensure that the ground truth rule sets themselves were not noisy, i.e., without overlapping uncertainty intervals. Since we are evaluating a normative system, we lose the ability to clearly establish a ground truth if the underlying rules themselves were noisy. For example, if the uncertainties of a knife and screwdriver as objects used for cutting are very close to one another and overlapping, then which object is a better object becomes a more difficult question and without a clear answer supported in the ground

truth. Thus, given a clear ground truth and no sensory noise, the architecture should learn the rules and act correctly all of the time.

As expected, we obtained a 100% success rate with the robot inferring the correct functional affordance and choosing the correct object (for all actions) and choosing the correct grasp locations (for pass and pickup). We observed this success rate across all scenes measured. As one example, when all the objects were presented, the robot chose the mug when asked to select an object with the containing affordance. Likewise, the robot correctly identified top-ranked objects for each of the four affordances. This meant that, for Scenario B, the robot correctly identified the shoe as being the best candidate with a containing affordance.

Our simulation further suggests that any performance below 100% must be due to sensor noise. If the agent is unable to correctly detect that an object on the table is in fact a knife, when asked for something used for cutting, then the agent is less likely to find this object as a suitable candidate - affordance inference will yield an uncertainty that might be below a threshold confidence measure described earlier.

In addition to the simulation, we further provide an empirical demonstration of a DIARC agent with a fully integrated module for cognitive affordance reasoning in a task-driven dialogue involving multiple human interlocutors. In this demonstration, we show the system's ability to learn new cognitive affordance rules on the fly and to reason about these newly learned rules. A video of this demonstration is located at http://tiny.cc/affordanceNL2018. We use the motivating example utterance "Pass me something used for cutting" spoken from a human to robot running cognitive affordance-enabled DIARC.

### 4.2 Commands with Implicit Affordances

Thus far, we have presented examples where the requested affordance was explicitly stated, such as "pass me something used for pounding." However, the approach presented in this paper is not limited to such cases and is capable of handling cases where the requested affordance is not explicit. For example, a command "pound the nail" contains an implicit request for a tool that can do the pounding. In some sense, the command might actually be suggesting "pound the nail *with something used for pounding*," without saying so explicitly. As before, the robot is taught the normative affordance rule that a hammer can be used for pounding. But, in order for the robot to be able to make use of this affordance rule, it must already have an action script that describes how it should perform the pounding action. Much like the action script depicted as a tree in Figure 3, we consider an example action script for $pound$ that is composed of a $pickup$ action. It also contains a $moveAbove(nail)$ action and a series of repeated $raise$ and $lower$ actions to generate the pounding motion. The $pickup$ subaction, in turn, contains *find*, $grasp$ and $moveTo(up)$ subsubactions. The $pound$ action can be made more complex, containing visual actions of sensing the depth of the nail and identifying when to stop pounding.

In addition to this action description, the action script needs additional knowledge to handle cases when the action is called with a tool explicitly mentioned (e.g., pound the nail with the hammer) and when the action is called with the tool affordance implicitly suggested (e.g., pound the nail). In the implicit case, the highest level $pound$ action must be able to supply a $usedFor(something, pounding)$ argument to the child $pickup$ action. Thus, the $pound$ action

must first review the arguments of the goal predicate $pound(self, nail)$ received and parsed from the command utterance and then provide the arguments $self$ and $usedFor(something, pounding)$ to the the $pickup$ action. It is possible that the implicit command "pound the nail" was intended to be interpreted as "pound the nail *with the hammer*". In this case, the action script would need to consider other factors (e.g., the intent of the speaker) in order to determine what exactly was left unsaid – i.e., was it that the speaker intended for the agent to use a specific tool, namely the hammer or any tool with a pounding affordance.

With this translation, the rest of command execution proceeds as described in the previous sections. This example shows that, with suitable modifications to the action script, we can handle commands that contain affordance information explicitly as well as implicitly. It is important to reiterate a key assumption: that the robot is already equipped with the above-mentioned $pound$ action script. Learning these action scripts (from natural language or however else) as well as determining interpretations of unsaid action arguments is beyond the scope of this paper and the subject of ongoing work.

## 5. Discussion

The above walkthrough and simulation show how a set of new social norms, previously unknown to the agent, can be acquired, in one-shot, from natural language instruction. The process of learning an implication rule of the form described is generalizable to other rules as long as the agent is familiar with the entities being described; for instance, the agent already knows what a knife, handle, and blade mean. Critically, the new knowledge of the social norm encoded as an affordance rule is now available for inference by any and all subsystems in the cognitive robotic architecture. As shown in the evaluation, these rules can be put to immediate use for follow-up requests from a human. These are, to our knowledge, the first demonstrations of an agent learning an unknown affordance norm from natural language instruction and then performing an action sequence conforming with the rule that it just learned. Moreover, an affordance norm of this sort may be beneficial not just to an action subsystem of an architecture, but to planning and other subsystems. A general rule-based structure, coupled with an inference mechanism presented here, allows these other subsystem to query and access these affordance norms, as well.

Note that the above demonstration also shows that the instructions and actions do not have to pertain to a particular set of sensors or actuators and do not depend on a particular robotic platform. Rather, the same inference and learning mechanisms can be carried out in other agents with different action capabilities. It is also important to note that the approach is not limited to the particular examples demonstrated; being implementations of a general framework for reasoning about affordances to guide normative behavior, they are only limited by the agent's knowledge of natural language and by its sensory and actuation capabilities. For example, a Nao robot may not possess adequate gripper capabilities to grab a knife, but will still be able to reason about the normative aspects of other action capabilities like pointing and can still learn from instructions about these normative aspects of these actions.

Finally, to demonstrate the extent of learning, we note that current state of the art vision systems can identify and label objects and object features with a high level of accuracy. Thus, an agent can potentially become familiar with names and descriptions for thousands of objects. Along the same

lines, agents can be trained through existing methods to build a substantial vocabulary and grammar allowing for an infinite possible set of descriptors for perceptual invariants, contexts, and actions. Hence, it is not possible, nor does it make sense, to evaluate the system exhaustively by generating every possible combination of rules and checking whether the agent can learn them. The strength of our system is that, no matter what set of rules we give it, it can learn and reason about affordances provided it has the sensory information to ground them.

## 6. Related Work

While affordances have been studied for decades in psychology and philosophy, few computational approaches have been presented for modeling them in normative contexts, and none for learning them *from* natural language, which is an important open problem in affordance-related research in robotics (Zech et al., 2017). We believe that our approach represents a significant advance over existing ones. Research in cognitive robotics and AI more generally originated from philosophical and psychological theories and diverged in two directions: statistical and ontological. The statistical approaches have modified and implemented these general theories in specific domains using mathematical formalisms to represent and compute affordances (Steedman, 2002; Montesano et al., 2007; Aleotti et al., 2014; Chan et al., 2015; Ugur et al., 2015; Koppula & Saxena, 2016). The affordances were modeled as a statistical relationship between an object, actions performed on the object, and the effects of those actions (i.e., success or failure). Some preliminary work has extended this approach by incorporating "environment" as a fourth entity, thereby providing some degree of situatedness and context (Kammer et al., 2011). The ontological approaches have focused on developing a detailed knowledge base of conceptual, functional, and part properties of objects, and used a combination of detection and query-matching algorithms to pinpoint the affordances for objects (Varadarajan, 2015). However, neither approach has considered the influence of social or normative (and nonperceptual) factors in affordance determination.

More recently, Shu et al. (2016) presented a framework for reasoning about "social affordances" and provided a system that can act in social scenarios. However, the underlying affordance model is largely devoid of contextual or normative reasoning, (non-perceivable aspects of affordances) and is focused just on physical geometries of objects (perceivable aspects) in these scenarios, in this case skeletal geometries. Other work in robotics has explored mechanisms for detecting context and social contextual perception at both an individual level (O'Connor & Riek, 2015; Nigam & Riek, 2015; Parashar et al., 2015), as well as in group-level activities (Okal & Arras, 2014). However, these approaches do not provide a generalized model or integration of normative affordance perception of objects in a robotic architecture.

Thus, more generally, despite these past efforts, the task of computationally modeling affordances faces many challenges that have not been overcome in the previous work. These past efforts do not allow for reasoning about normative affordances and, from an architectural standpoint, most affordance processing is subsumed by sensory processing (e.g., vision) or higher-level cognition (e.g., planning), which does not allow for an effective interaction between top-down and bottom-up processing of information. Moreover, none of the current approaches show how affordances can be learned from natural language.

## 7. Conclusions and Future Work

The expressive framework of cognitive affordances treats such relations as normative condition-action rules. In a sense, it extends the traditional Gibsonian notion of an affordance as a relation between an object and an action to include other nonperceptual aspects that influence action selection such as context, intentions, and social conventions. In this paper, we provide two contributions: (1) a grounding and integration of this theoretical framework within a robotic architecture, and (2) an approach to learning cognitive affordances from natural language instruction. To accomplish this task, we extended recent work in instruction-based one-shot learning to parse and learn cognitive affordance rules. The predicates and terms that constitute the rules contain perceptual and action concepts that are grounded within the DIARC cognitive robotic architecture. For each action that the robot must carry out, we proposed several operations that obtain sensory information from the perceptual system, perform inference over relevant affordance rules that impose constraints, and execute the constrained action. We evaluated the approach through an extended simulation and real-world runs of the robotic architecture as implemented on a PR2 robot. Critically, we showed that not only can an agent learn normative behavior from instruction, but it can immediately apply this newly acquired knowledge to the task at hand. To our knowledge this is the first conceptual and robotic demonstration of an agent learning an unknown affordance norm from natural language instruction and performing an action sequence conforming to the rule it just learned. We believe that these capabilities are necessary to let agents work effectively with humans and to dynamically learn tasks in ways that respect prevailing social norms. The approach presented in this paper does not currently incorporate commonsense knowledge about objects and their similarity to similar objects. Thus, the affordance rules that are learned from natural language are limited to the particular object explicitly taught. One direction for future work is to explore how to induce new cognitive affordance rules using such commonsense knowledge. For example, we would like the system to know and use the fact that both knives and screwdrivers are tools with pointed ends that must be handled carefully. When we teach the robot how to safely pass a knife, it should subsequently induce a comparable rule for the screwdriver.

## Acknowledgements

This research was supported in part by Grants N00014-14-1-0149 and N00014-14-1-0751 from the Office of Naval Research, which is not responsible for its contents.

## References

Aleotti, J., Micelli, V., & Caselli, S. (2014). An affordance sensitive system for robot to human object handover. *International Journal of Social Robotics*, *6*, 653–666.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*, 1036–1060.

Chan, W. P., Pan, M. K., Croft, E. A., & Inaba, M. (2015). Characterization of handover orientations used by humans for efficient robot to human handovers. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1–6). Hamburg, Germany: IEEE.

Dzifcak, J., Scheutz, M., Baral, C., & Schermerhorn, P. (2009). What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4163–4168). Kobe, Japan: IEEE.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton, Mifflin and Company.

Kammer, M., Schack, T., Tscherepanow, M., & Nagai, Y. (2011). From affordances to situated affordances in robotics-why context is important. *Frontiers in Computational Neuroscience Conference / Abstracts. doi:10.3384/conf.fncom.2011.52.00030*.

Kerdjoudj, F., & Curé, O. (2015). Evaluating uncertainty in textual document. *Proceedings of the Eleventh International Workshop on Uncertainty Reasoning for the Semantic Web* (pp. 1–13). Bethlehem, PA: Springer.

Koppula, H. S., & Saxena, A. (2016). Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*, 14–29.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Montesano, L., Lopes, M., Bernardino, A., & Santos-Victor, J. (2007). Modeling affordances using bayesian networks. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4102–4107). San Diego, CA: IEEE.

Myers, A., Teo, C. L., Fermüller, C., & Aloimonos, Y. (2015). Affordance detection of tool parts from geometric features. *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1374–1381). Seattle, WA: IEEE.

Nigam, A., & Riek, L. D. (2015). Social context perception for mobile robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3621–3627). Hamburg, Germany: IEEE.

Nunez, R. C., Dabarera, R., Scheutz, M., Briggs, G., Bueno, O., Premaratne, K., & Murthi, M. N. (2013). DS-based uncertain implication rules for inference and fusion applications. *Proceedings of the Sixteenth International Conference on Information Fusion* (pp. 1934–1941). Istanbul, Turkey: IEEE.

O'Connor, M. F., & Riek, L. D. (2015). Detecting social context: A method for social event classification using naturalistic multimodal data. *Proceedings of the Eleventh IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 1–7). Ljubljana, Slovenia: IEEE.

Okal, B., & Arras, K. O. (2014). Towards group-level social activity recognition for mobile robots. *Proceedings of the Workshop on Assistance and Service Robotics in a Human Environments Workshop at the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL: IEEE.

Parashar, P., Fisher, R., Simmons, R., Veloso, M., & Biswas, J. (2015). Learning context-based outcomes for mobile robots in unstructured indoor environments. *Proceedings of the Fourteenth IEEE International Conference on Machine Learning and Applications* (pp. 703–706). Miami, FL: IEEE.

Sarathy, V., & Scheutz, M. (2016). Cognitive affordance representations in uncertain logic. *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning*. Cape Town, South Africa: AAAI Press.

Sarathy, V., & Scheutz, M. (2018). A logic-based computational framework for inferring cognitive affordances. *IEEE Transactions on Cognitive and Developmental Systems*, *10*, 26–43.

Scheutz, M., Briggs, G., Cantrell, R., Krause, E., Williams, T., & Veale, R. (2013). Novel mechanisms for natural human-robot interactions in the DIARC architecture. *Proceedings of the AAAI Workshop on Intelligent Robotic Systems*. Bellevue, WA: AAAI Press.

Scheutz, M., Krause, E., Oosterveld, B., Frasca, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. *Proceedings of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems*. Sao Paolo, Brazil: IFAAMAS.

Scheutz, M., Schermerhorn, P., Kramer, J., & Anderson, D. (2007). First steps toward natural human-like HRI. *Autonomous Robots*, *22*, 411–423.

Shu, T., Ryoo, M. S., & Zhu, S.-C. (2016). Learning social affordance for human-robot interaction. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 3454-3461*. New York, NY: AAAI Press.

Sridharan, M., & Meadows, B. (2017). Learning affordances for assistive robots. *Proceedings of the Ninth International Conference on Social Robotics* (pp. 1–11). Tsukuba, Japan: Springer.

Steedman, M. (2002). Formalizing affordance. *Proceedings of the Twenty-Fourth Annual Meeting of the Cognitive Science Society* (pp. 834–839). Fairfax, VA: Cognitive Science Society.

Ten Pas, A., & Platt, R. (2014). Localizing handle-like grasp affordances in 3-D points clouds using taubin quadric fitting. *Proceedings of the International Symposium on Experimental Robotics*. Marrakech, Morocco: Springer.

Ugur, E., Nagai, Y., Sahin, E., & Oztop, E. (2015). Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese. *IEEE Transactions on Autonomous Mental Development*, *7*, 119–139.

Varadarajan, K. (2015). Topological mapping for robot navigation using affordance features. *Proceedings of the Sixth International Conference on Automation, Robotics and Applications* (pp. 42–49). Queenstown, NZ.

Zech, P., Haller, S., Lakani, S. R., Ridge, B., Ugur, E., & Piater, J. (2017). Computational models of affordance in robotics: A taxonomy and systematic classification. *Adaptive Behavior*, *5*, 235–271.