Autonomous Robots
https://doi.org/10.1007/s10514-018-9784-8

Semantic and geometric reasoning for robotic grasping: a probabilistic logic approach

Laura Antanas[1] · Plinio Moreno[2] · Marion Neumann[3] · Rui Pimentel de Figueiredo[2] · Kristian Kersting[4] · José Santos-Victor[2] · Luc De Raedt[1]

Received: 20 April 2017 / Accepted: 10 July 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

While *any* grasp must satisfy the grasping stability criteria, *good* grasps depend on the specific manipulation scenario: the object, its properties and functionalities, as well as the task and grasp constraints. We propose a *probabilistic logic* approach for robot grasping, which improves grasping capabilities by leveraging semantic object parts. It provides the robot with *semantic reasoning skills* about the most likely object part to be grasped, given the task constraints and object properties, while also dealing with the uncertainty of visual perception and grasp planning. The probabilistic logic framework is *task-dependent*. It semantically reasons about pre-grasp configurations with respect to the intended task and employs object-task affordances and object/task ontologies to encode rules that generalize over similar object parts and object/task categories. The use of probabilistic logic for task-dependent grasping contrasts with current approaches that usually learn direct mappings from visual perceptions to task-dependent grasping points. The logic-based module receives data from a low-level module that extracts semantic objects parts, and sends information to the low-level grasp planner. These three modules define our probabilistic logic framework, which is able to perform robotic grasping in realistic kitchen-related scenarios.

Keywords Cognitive robotics · Probabilistic logic · Task-dependent grasping · Semantic grasping · Local shape grasping

1 Introduction

To perform manipulation tasks in arbitrary and dynamic environments, robots need vision capabilities to perceive the world, reasoning skills to interpret it, and good grasping strategies. Performing good grasps, besides satisfying the grasping stability criteria, depends on the specific manipulation scenario: the object, its properties and functionalities, as well as task and grasp constraints (e.g., the gripper configuration). The problem we address in this work is how to take such information into account for robot grasping.

More specifically, instead of learning a function that directly maps visual perceptions to task-dependent grasps, we propose, as key contribution, a *probabilistic logic module to semantically reason* about the most likely *object part* to be grasped, given the object properties and task constraints. To do so, we exploit the semantic meaning of object parts that generalize across several object categories and thus, allow us to reason at a higher-level. In addition, the semantic part-based representation allows grasp transfers to novel objects that have similar parts. Our approach has several benefits: it allows for generalization over similar (multiple) object parts, it reduces the grasping inference space by reasoning about possible task-constrained grasps, and it enhances object category estimation by employing both semantic part labels as well as object shape and curvature information. The probabilistic logic module (PLM) links semantic object parts to several tasks defined as associations between parts and gripper poses. Inspired by the Gibsonian definition (Gibson 1979) of object affordances which refers to the properties

✉ Laura Antanas
laura.antanas@gmail.com

Plinio Moreno
plinio@isr.tecnico.ulisboa.pt

Luc De Raedt
luc.deraedt@cs.kuleuven.be

1 Department of Computer Science, KULeuven, Heverlee, Belgium

2 Institute for Systems and Robotics, Lisbon, Portugal

3 Department of Computer Science and Engineering, Washington University in St Louis, St Louis, USA

4 Computer Science Department, Technical University of Dortmund, Dortmund, Germany

Published online: 23 August 2018

of an object to allow action possibilities offered to it by its surrounding environment, we consider grasping affordances that each task activates according to associations between object categories, object parts and gripper poses in a household environment. To further generalize and reason across object and task categories, the PLM exploits ontologies of object and task categories.

The emerging research question that raises in this paper is: does semantic high-level reasoning help selecting a better grasp if the next task is known? Let us assume the fetch and carry scenario (Fig. 1), where a mobile robot with grasping capabilities has to collect all the bottles on the shelves and place them in a case. This assignment is a sequence of individual tasks, i.e., placing each bottle in the case in an upright pose. The environment constraints (e.g. narrow spaces to reach the object, collisions with the shelf and low-dexterity of the gripper) and task constraints (e.g. the most stable pre-grasp gripper pose for placing the bottle in the case) pose a difficult grasping problem which can be solved using semantic reasoning. Specifically, if we consider the top, middle and bottom as semantic parts of a bottle, the best part to grasp each bottle is from the middle, given that the bottle needs to be placed in the case in an upright position and the top is partially obstructed by the shelf above. Using semantic object parts, we can define common sense grasping rules for our kitchen scenario. The resulting PLM gives the robot the capability to *semantically reason* about the most likely *object part* to be grasped, given the task-object category affordance, and thus, help the grasp planner by reducing the space of possible final gripper poses. This knowledge could be transferred, via an object category ontology, to other similar objects, such as cans. Further, the robot could reuse these rules in similar environments (e.g., fetch and carry bottles in a supermarket) by updating the PLM.

Manipulation tasks such as the one mentioned above require large amounts of data to build a model for probabilistic inference. For instance, considering object variables such as shape, functionality, pose and category; and task constraints such object pose and gripper pose, the joint learning of the dependencies of these variables will require a vast amount of samples. Previous approaches only consider a subset of the variables mentioned above and learn the dependencies between those variables in order to avoid complexity issues. Works like (Madry et al. 2012a, b) separate object category variables from gripper pose w.r.t. the object during dependency learning. This allows them to build a global model by merging two smaller models. In Madry et al. (2012b), the gripper pose constraints w.r.t. the object are learned and used in simulation. Differently, our probabilistic logical approach tackles task and environmental constraints in real-world setups as educated guesses. Logic offers a natural way to integrate high-level world knowledge such as subsumption relations among object categories (encoded by an object ontology), task categories (encoded by the task ontology), and relations between object categories and tasks (encoded by object-task affordances), in a compact rule-based grasping model. Because descriptions of the perceived world are uncertain (e.g., not all cups look like the 'prototypical' cup), we consider probabilistic logic which allows reasoning about the uncertainty in the world. Based on available task and visual scene observations, we can use probabilistic logic to ask and answer queries about different aspects of the grasping action.

The main contribution of this paper is a *probabilistic logic module for task-dependent robot grasping*, which provides the robot with semantic reasoning skills (i.e. inference in an abstract representation of the world), while dealing with the uncertainty of the visual perception on two tasks: object recognition and grasp planning. The main features of the PLM include: (1) a first general rule-based model integrating task information and object category for robotic grasping; (2) a semantic part-based representation of the objects that generalizes across several categories and allows reasoning by means of logic; and (3) a new probabilistic logic approach to object grasping which uses high-level knowledge such as object-task affordances and object/task ontologies to generalize over object/task categories and improve robot grasping. This allows us to experiment with a wide range of objects of different categories, a critical aspect of autonomous agents acting independently in new environments. Our approach can be extended beyond the set of categories used, by augmenting the probabilistic logic model to cover new categories.

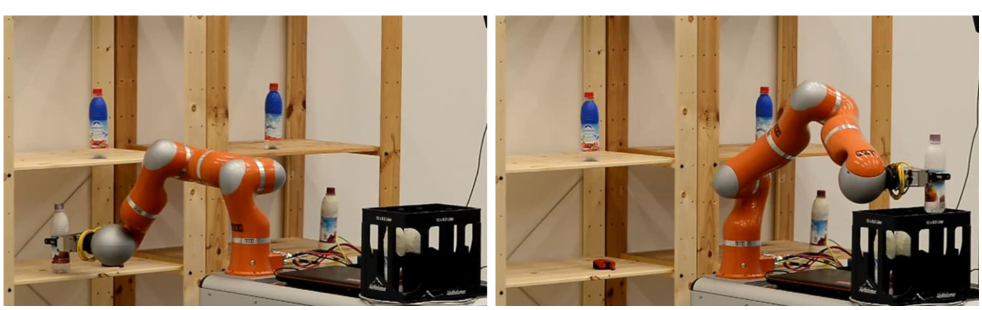


Fig. 1 Fetch and carry scenario: collect all bottles from the shelves and place them in the case

The benefits of the features listed above for task-based dependent grasping are shown experimentally. Different tasks can be successfully executed on new objects that belong to the object categories considered.

2 The proposed framework

An overview of our three module grasping framework is shown in Fig. 2. The first module is a visual perception module (detailed in Sect. 4) which maps the detected object point cloud to a set of symbolic and probabilistic visual observations of the scene. After segmenting the object point cloud and performing a full object shape reconstruction, the visual module estimates object properties such as pose, symbolic parts and geometric-based object category. The second module is the PLM for task-dependent grasping. Given the input observations, it is able to reason about the grasping scenario. It can (re)predict and improve the category of the object via the semantic object categorization model (Sect. 7.1). Further, the module can answer queries about the most likely task and most likely object part to grasp, given the task (denoted as pre-grasp). It uses probabilistic visual observations about the object provided by the first module, evidence about the task and world knowledge such as object/task ontologies and object-task affordances. Once we have identified the most likely pre-grasp, the framework calls the third module, which solves the problem of planning the grasp execution by using local shape features of the object part and completing it on the robotic platform (Sect. 8).

Figure 2 illustrates the framework on the example of an empty cup. Given the point cloud of the cup and using vision-based techniques, the framework obtains a description of the object: it has the symbolic parts *top*, *middle*, *bottom* and *handle*, stands in an upright pose and is empty. We employ global object similarity based on part information to complete the scene description with an initial prior on the object category: *cup*, *can* and *pot* with probabilities 0.56, 0.36 and 0.05, respectively. Next, using the visual description, we query the PLM for the most likely object category, most likely task and best pre-grasp. The categorical logic module reasons about the symbolic parts and recalculates the category distribution as follows: 0.98 for category *cup* and 0.02 for category *pan*. The presence of exactly one *handle* increases the probability of the object being more a *cup* rather than a *can* and identifies the object more as a *pan* rather than a *pot*. Similarly, using object-task affordances and world knowledge (e.g., the task *pour in* cannot be executed on a full object), the PLM predicts the tasks *pass*, *pick-place on* and *pick-place inside upright* with equal probability. If the task given is *pass*, the task-dependent grasping model next predicts the *middle* part of the object as most likely pre-grasp. The last step in the framework, the low-level grasping planner, predicts the best point for grasping in the pre-grasp point cloud.

We proceed as follows. First, we review related work. Then, we explain in detail each component of the proposed framework: the low-level perception module, the probabilistic logic object categorization module, the probabilistic logic grasping module and the low-level grasp planner. Before concluding, we present our experiments in simulation as well as on a real robot.

3 Related work

We review the state-of-the-art following three main axes that are addressed in our work: (i) visual grasping; (ii) task-dependent grasping; (iii) statistical relational learning for robotics.

3.1 Visual-dependent grasping

The majority of grasping methods consider mainly visual features to learn a mapping from 2D/3D features to grasping parameters (Bohg and Kragic 2010; Montesano and Lopes 2012; Lenz et al. 2015; Saxena et al. 2008). Nevertheless, these methods have a major shortcoming: it is difficult to link a 3D gripper orientation to solely local image features. Only recently, methods that take global information into account have been proposed (Aleotti and Caselli 2011; Neumann et al. 2016). The benefit is an increased geometric robustness, which is advantageous with respect to the pre-shape of

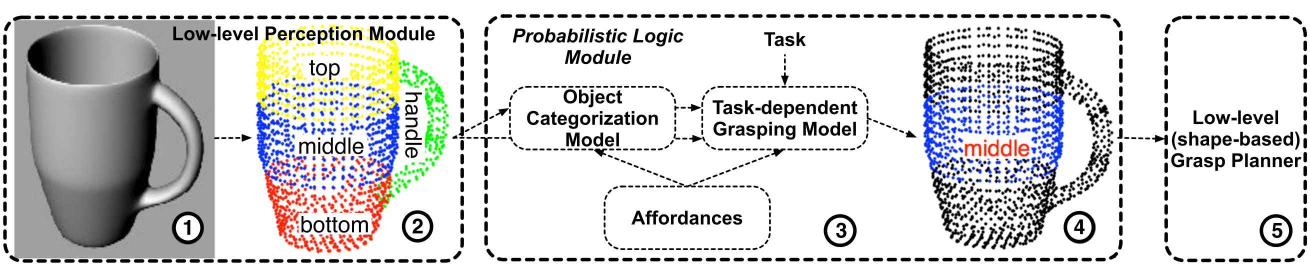


Fig. 2 The task-dependent grasping framework on a *cup* point cloud example. The boxes represent the main components. From left to right: object ①, symbolic object parts ② with labels top (yellow), middle (blue), bottom (red), and handle (green), PLM ③, predicted pre-grasp middle ④, and shape-based grasp model ⑤ (Color figure online)

the robotic hand and the general shape of the object, generating more accurate grasps. However, global information relies on a complete shape of the object.

Shape completion using a single view has been extensively studied, typically in robotics grasping applications. Usually multiple object partial views are acquired from different viewpoints, using 3D range cameras, and the gathered point clouds are then registered and aligned together in a common reference frame (e.g. ICP Besl and McKay 1992 its variants Rusinkiewicz and Levoy 2001). However, when only a single view is available and/or it is not possible to acquire several views due to time constraints or scenario/robot restrictions the shape completion problem becomes harder and some assumptions or pattern analysis must be made. Thrun and Wegbreit (2005) address the single view scenario, proposing a method based on the symmetry assumption. This method considers 5 basic and 3 composite types of symmetries that are organized in an efficient entailment hierarchy. It uses a probabilistic model to evaluate and decide which are the completed shapes, generated by a set of hypothesized symmetries, that best fit the object partial view. More recently Kroemer et al. (2012) proposed an extrusion-based completion approach that is able to deal with shapes that symmetry-based methods cannot handle. The method starts by detecting potential planes of symmetry by combining the Thrun and Wegbreit method with Mitra et al.'s (2006) fast voting scheme. Given a symmetry plane, ICP is used to decide the extrusion transformation to be applied to the object partial point cloud. Despite the fact that these methods were shown to be robust to noise and were able to deal with a wide range of object classes, they are inherently complex in terms of computational effort and thus, not suitable in real-time. Other approaches have taken advantages of common scenario structures and objects properties that are usually found in daily environments (e.g. man-made objects are typically symmetric and standing on top of planar surfaces). For example, Bohg et al. (2011) took advantage of the table-top assumption and the fact that many objects have a plane of reflection symmetry.

Tenorth et al. (2013) decompose CAD models into geometric primitives and define rules for the primitives that are able to find handles, containers and supporting planes in CAD models. A method for matching the CAD models to perceived sensor data provides the functional parts of the objects. This method complements very well the main subject of this work, but it is based on CAD models and does not perform object completion for grasping actions. Starting from the work of Thrun and Wegbreit (2005) and similar in spirit to Bohg et al. (2011), we implemented an efficient shape completion approach which translates a set of environmental assumptions into a set of approximations, allowing us to reconstruct the object point cloud in real-time, given a partial view of the object.

3.2 Task-dependent grasping

Since grasping is highly correlated with the task to be performed on the object, a lot of recent work has focused on incorporating task constraints in robot grasping. This is mostly done by learning a direct mapping function between good grasps and geometrical and action constraints, action features and object attributes. A part of this work focuses on Bayesian network learning to integrate symbolic task goals and low-level continuous features such as object attributes, action properties and constraint features (Madry et al. 2012a; Song et al. 2010). The goal is to learn features of importance for grasping knowledge transfer. This work is extended to consider object categorical information as an additional feature to predict suitable task-dependent grasping constraints (Madry et al. 2012b). Further, Detry et al. (2012a, b, 2013) identify grasp-predicting prototypical parts by which objects are usually grasped in similar ways. The discrete part-based representation allows robust grasping. Differently, in addition to the semantic parts, we also consider a task-dependent setting that uses probabilistic logic and world-knowledge to reason about best pre-grasps. Several approaches make use of object affordances for grasping. While in Sweeney and Grupen (2007) the authors employ estimated visual-based latent affordances, the work in Barck-Holst et al. (2009) reasons about grasp selection by modeling affordance relations between objects, actions and effects using either a fully probabilistic setting or a rule-based ontology. In contrast, we employ a probabilistic logic-based approach which can generalize over similar object parts and several object categories and tasks.

Related to our probabilistic logic framework is the fully probabilistic one introduced in Bohg et al. (2012). It combines low-level features and Bayesian networks to obtain possible task-dependent grasps. Also, closely related is the semantical pipeline presented in Dang and Allen (2012). It employs a semantic affordance map which relates gripper approach directions to particular tasks. In contrast, we exploit additional object/task ontologies using probabilistic reasoning and leverage low-level learning and semantic reasoning. This allows us to experiment with a wide range of object categories.

3.3 SRL for robot grasping and other robotic tasks

From a different point of view, probabilistic relational robotics is an emerging area within robotics. Building on statistical relational learning (SRL) and probabilistic robotics, it aims at endowing robots with a new level of robustness in real-world situations. We review some recent successful contributions of SRL to various robotic tasks. Probabilistic relational models have been used to integrate common-sense knowledge about the structure of the world to successfully accomplish search tasks in an efficient and reliable goal-directed manner (Hanheide et al. 2011). Further, relational dependency networks have been exploited to learn statistical models of procedural task knowledge, using declarative structure capturing abstract knowledge about the task (Hart et al. 2005). The benefits of task abstraction were shown in Winkler et al. (2012), where the robot uses vague descriptions of objects, locations, and actions in combination with the belief state of a knowledge base for reasoning. The goal of this work is to robustly solve the planning task in a generalized pick and place scenario. Abstract knowledge representation and symbolic knowledge processing for formulating control decisions as inference tasks have proven powerful in autonomous robot control (Tenorth and Beetz 2009). These decisions are sent as queries to a knowledge base. SRL techniques using Markov Logic Networks and Bayesian Logic Networks for object categorization have been proposed in Marton et al. (2009) and Nyga et al. (2014).

In probabilistic planning, relational rules have been exploited for efficient and flexible decision-theoretic planning (Lang and Toussaint 2010) and probabilistic inference has proven successful for integrating motor control, planning, grasping and high-level reasoning (Toussaint et al. 2010). In mobile robotics, relational navigation policies have been learned from example paths with relational Markov decision Processes (Cocora et al. 2006). In order to compute plans comprising sequences of actions and in turn be able to solve complex manipulation tasks, reasoning about actions on a symbolic level is incorporated into robot learning from demonstrations (Abdo et al. 2012). Symbolic reasoning enables the robot to solve tasks that are more complex than the individual, demonstrated actions. In Kulick et al. (2013) meaningful symbolic relational representations are used to solve sequential manipulation tasks in a goal-directed manner via active relational reinforcement learning. Relational Markov networks have been extended to build relational object maps for mobile robots in order to enable reasoning about hierarchies of objects and spatial relationships amongst them (Limketkai et al. 2005). Related work for generalizing over doors and handles using SRL has been proposed in Moldovan et al. (2013a).

These approaches successfully intertwine relational reasoning and learning in robotics. However, none of these frameworks solves the generalization capability needed for task-dependent grasping following a semantic and affordance-based behavior. Relational affordance models for robots have been learned in a multi-object manipulation task context (Moldovan et al. 2012, 2018). We propose a probabilistic logic framework to infer pre-grasp configurations using task-category affordances. Our approach features semantic generalization and can tackle unknown objects. This research topic has great importance in robotics as robots aimed at working in daily environments should be able to manipulate many never-seen-before objects and to deal with increasingly complex scenarios.

4 Semantic vision-based scene description

The role of the visual module (cf. first module box in Fig. 2) is to obtain a semantic description of the perceived objects in terms of their pose, symbolic parts and probability distributions over possible object categories. In the context of object manipulation, a comprehensive representation must consider object semantics, shape, affordances, motion planning and the task to be performed on the object. To define such a representation, the ideal model should consider both continuous and discrete variables, while dealing with uncertainty. In this work we are driven by the motion planning constraints to define our main assumptions. The motivation comes from the space limitations for efficient planning in small household spaces (e.g., the kitchen). As a result:

- Grasping regular kitchen and workshop objects requires a significant payload. This implies larger power requirements which are coped by larger arms, reducing significantly the task workspace.
- Holistic view of top/down object parts. Consider objects placed on top of a table or in a shelf. The top or bottom regions of the object may not be available for grasping due to collision constraints with the table or the shelf, but the middle region of the object is usually available. In the same line of though, the top region of the objects are usually available for grasping when the object is in a drawer. These rules of thumb are truth without considering the object shape and affordances, so the semantic top and bottom parts allow to define rules that do not depend on the object shape. Although very general, these rules have exceptions that we consider. The use-cases of this view include kitchen cupboards and shelves, and tool storage shelves and drawers, and can be applied to chest of drawers, tables and shelves in general.
- Tool holistic parts. Our main assumption is that we can estimate the bounding box of every semantic part from a partial view of the object. To accomplish this, we assume that all the objects are symmetric [either global or partial symmetry (Thrun and Wegbreit 2005)]. Also, we assume that usable area(s) and handle can cover most of the tools and are appropriate for the definition of tasks such as: pass the tool to a person (the object should be grasped from the usable area) and pick and place (the handle is usually the better candidate for grasping).
- Discrete object poses. We consider poses upright, upside-down, and sideways. Semantically, upright is distinguished from upside-down according to what the object affords, meaning that if the object is a container, the posi-

tion of the lid on top corresponds to the upright pose and the lid on bottom corresponds to upside-down one.
- Low-level grasp planning is customized to the particular end-effector and it is learned independently of the task, using only point cloud statistics in the selected part. In this work the robot has a two finger gripper in practice, so the grasp planning is tailored for that type of end-effector. In order to apply the semantic and geometric reasoning to other end-effectors, the end-user just needs to learn the grasp planning model for other type of hands.
- Grasping control and stability. We do not address this in our model but rely on the force and grip control provided by the end-effector selected for the experiments (Schunk WSG). The current technological developments in tactile sensing are limited in terms of frequency and sensor density (Yousef et al. 2011), so the application of human dexterity control procedures that maximize comfort (Flanagan et al. 2006) are out of the scope of this work.

In practice, in a kitchen scenario we consider the categories *pan, cup, glass, bottle, can, knife*, and in a workshop scenario we consider the categories *hammer and screwdriver*. We are able to distinguish between upright and sideways poses (i.e. semantic poses). The considered scenarios and object categories cover a large set of regular household objects and their common poses in such environments. Regarding the parts, we are able to segment objects and label each one of their parts with one of the following labels: top, middle, bottom, handle and usable area. The object symmetry assumption allows us to apply shape completion algorithms, only on the parts or on the whole object according to a decision scheme. Then, using the shape-completed point cloud, we segment the object into its parts and assign them a semantic label. This reduces the search space for robot grasp generation, prediction and planning. The next subsections explain our symmetry-based method for shape-completion and the division of the completed point cloud into a set of semantic parts.

4.1 Object shape completion

As any other type of reconstruction based on a partial view of an object, computing its bounding-box is an ill posed problem due to lack of observability of the self-occluded part. However, the actual bounding shape provides a large set of reaching poses (i.e. pre-grasp hypotheses), which lead to successful grasping. Thus, we complete the object shape by assuming either global or partial rotational symmetry. Inspired by the work of Thrun and Wegbreit (2005) and Bohg et al. (2011) and keeping in mind the computational efficiency, we propose an approach that translates a set of assumptions and rules of thumb observed in many daily environments into a set of heuristics and approximations: (a) the objects stand on top of a planar surface (Muja & Ciocarlie); (b) the camera is at a higher viewpoint; (c) the objects have rotational symmetry; (d) their main geometrical axes are either orthogonal or parallel to the supporting plane; (e) the axis of symmetry corresponds to one of the main geometrical axes; and (f) the direction of the axis of symmetry indicates the object's pose (i.e., upright or sideways). These constraints model perfectly simple box-like and cylinder-like object shapes, such as kitchen-ware tools, and are reasonable assumptions for many other approximately symmetric objects, such as tools (Fig. 3). These heuristics allow us to reconstruct the unobserved part of an object point cloud in real-time, given a partial view (Fig. 4).

To find the main geometrical axis, we employ a global PCA representation. After we identify the PCA direction with more energy, the decision scheme for finding the object parts has two main steps:

- If the direction is perpendicular to the table, the object may have or not a *handle*. The object has a handle if the projection of the points onto the table plane fits very likely a circle. Otherwise, the object does not have a handle. The handle is defined as the set of projected points out of the circle. Then, the symmetry completion is applied separately to the handle and to the rest of the object, which is decomposed into *top*, *middle* and *bottom* parts.
- If the direction is parallel to the table, we consider the height w.r.t. the table. If the height is greater than a threshold, the object is not a tool and the procedure described in the previous point is applied. Otherwise, the object is a tool that has a *handle* and an *usable area*, and the object is completed with linear symmetry.

Analogous to Bohg et al. (2011), we consider one type of symmetry, that is the *line reflection symmetry* (Thrun and Wegbreit 2005) as it copes better with the object categories that we want to detect. More details on the assumptions and the implementation can be found in Figueiredo et al. (2017). The bounding boxes of the object parts define the pre-grasp hypotheses, providing two hypotheses for each face of a box, as illustrated in Fig. 5. The set of pre-grasp poses is used by the low-level grasp planner (Sect. 8) for local-shape grasping prediction and motion trajectory planning. The software and its ROS node that implements the object pose and part segmentation of Sect. 4.1 is available online.[1]

4.2 Object properties

Given the completed object point cloud and its semantic parts, we can estimate additional properties of the object such as pose, category and containment.

[1] Available at: https://github.com/vislab-tecnico-lisboa/object_recognition/tree/master/ist_object_details.

Fig. 3 Objects having global rotational symmetry

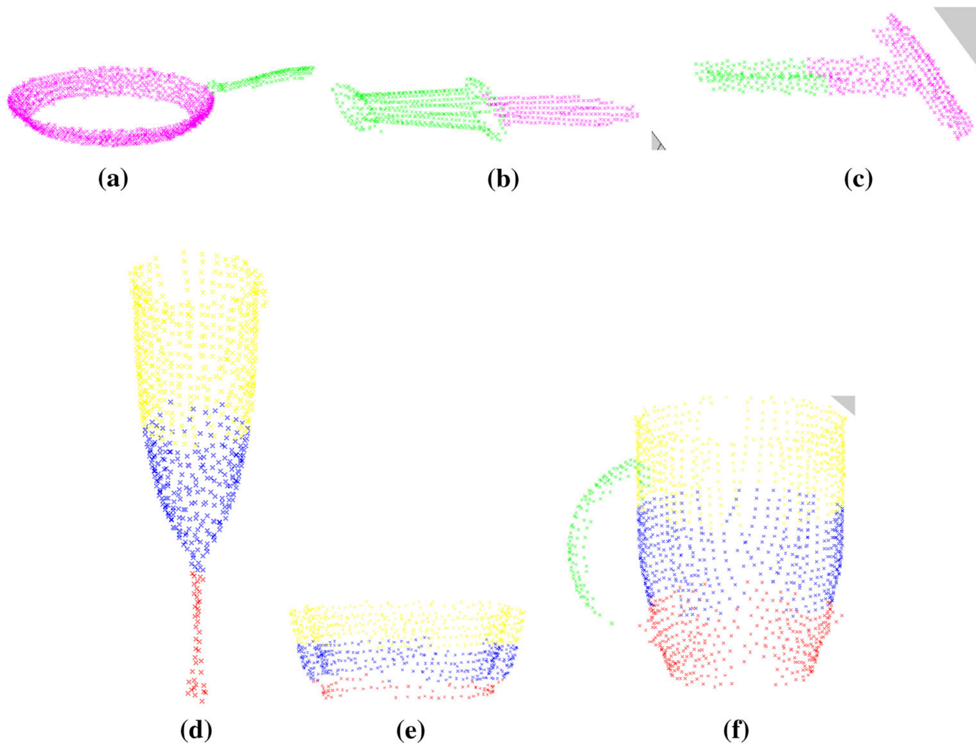


Fig. 4 Semantic parts for several objects after applying the completion algorithm. The colors correspond to parts as follows: yellow—top, blue—middle, red—bottom, green—handle, and magenta—usable area. **a** Pan, **b** Knife, **c** Hammer, **d** Glass, **e** Bowl, **f** Mug (Color figure online)

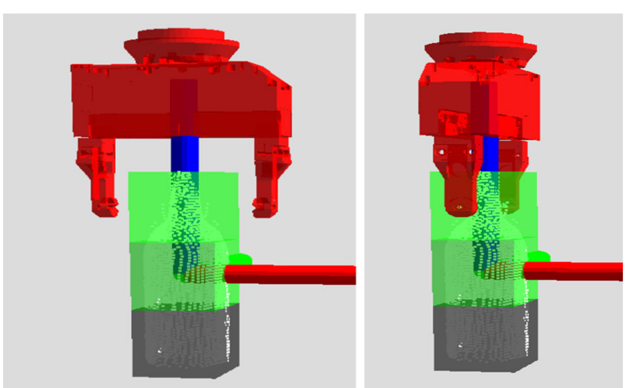


Fig. 5 Examples of the pre-grasp gripper poses for the superior face of the top part of a bottle

Object pose The orientation of the axis of symmetry with respect to the object's supporting plane indicates the pose of the object (upright, upside-down or sideways). In our real-world experiments we do not estimate the upside-down pose.

Similarity-based object classification We can estimate the object class by retrieving the objects with the most similar global properties. Due to good results for grasping point prediction (Neumann et al. 2016), we employ the manifold-based graph kernel approach proposed in Neumann et al. (2013) to asses global object similarity. It ensures a strong appearance-based predictor for object category. The prediction, in the form of a probability distribution on object labels, is used further as a prior by the probabilistic logic module. Other global object similarity methods can be also used instead.

We obtain the distribution on object categories for a particular object by retrieving the objects in an object database being most similar in terms of global shape and semantic part information. We represent the objects by labeled graphs, where the labels are the semantic part labels derived by the visual module and the graph structure given by a k-nearest neighbor (k-nn) graph of the completed object's point cloud. For each completed object we derive a weighted k-nn graph by connecting the k nearest points w.r.t. Euclidean distance in 3D.

The nodes have five semantic labels encoding object part information *top*, *middle*, *bottom*, *handle* and *usable area*. To capture manifold information as graph features in presence of full label information we use a diffusion scheme of the labels corresponding to the diffusion graph kernel, simply referred to as propagation kernel in Neumann et al. (2012). The similarity measure among objects is a kernel function over counts of similar node label distributions per diffusion iteration. We consider a maximum number of iterations. Given a new object G^* that the robot aims to grasp, we first select the top n most similar graphs $\{G^{(1)}, \ldots, G^{(n)}\}$, where $n = 10$ in all our experiments. Second, we build a weighted average over the categories of the objects corresponding to $\{G^{(1)}, \ldots, G^{(n)}\}$. This average is used as a prior distribution on object categories for the object graph G^*. The prior for the scenario in Fig. 2 is: $P = 0.56$ cup, $P = 0.36$ can, $P = 0.05$ pot, and $P = 0.02$ pan. It is used by the PLM to reason about different prediction tasks.

Object containment In our scene description we additionally consider observations about the object containment. It can be estimated by comparing top visual images of empty and full objects (opened containers) or by observing human (or humanoid) lifting and transportation actions (closed containers) (Palinko et al. 2014). We assume either that the containment property is not observed (e.g., for tools), the object is empty, or the object is full with a certain probability.

5 CP-logic for task-dependent grasping

We encode the extracted semantic description of the scene using Causal Probabilistic (CP) Logic (Vennekens et al. 2009) and introduce its syntax and semantics w.r.t. our setup. We observe one object at a time.[2] The conjunction of observations is: given task, object parts and visual properties of the object, that is pose, containment and category. Visual observations are represented using deterministic facts and ground probabilistic facts. For the scenario in Fig. 2 a deterministic fact is `obsObject(o)`, stating that an object `o` is observed, and a probabilistic fact is `0.8::part(top,o)`, stating that `o` has a top part with probability 0.8. The observation `obsObject(o)` is a logical atom and `obsObject/1` is a predicate symbol of arity 1.

[2] The work can be extended to consider several objects simultaneously.

The object identifier `o` is a constant denoted in lower case and represents a ground term. Terms can also be variables when denoted in uppercase. Ground atoms, such as `obsObject(o)` and `part(top,o)`, do not contain variables and represent particular relations. They possess truth-values, that is they are either true or false. When ground atoms are true, they are also called facts.

Additional visual observations are encoded via CP-rules capturing a prior distribution over the object category. For object `o` such a distribution is:

```
0.56::cup(o); 0.36::can(o); 0.05::pot(o);
0.02::pan(o) ← obsObject(o).
```

The CP-rule states that an object `o` belongs to a category with a certain probability, that is, it is either a cup with probability 0.56 or a can with probability 0.36 or a pot with probability 0.05 or a pan with probability 0.02. The arrow means implication (if). The left side of the arrow represents the head of the CP-rule. It is a disjunction of logic atoms indicating the possible outcomes of the body, represented by the logical atom `obsObject(o)`. In any CP-rule, the sum of the possible outcomes can be at most 1.[3] The probability should be interpreted as: if the body is true, then it causes the consequence to become true with a causal probability. The body of a rule can be empty, i. e., always true. In Example 1, this is formalized as: `0.8::part(top,o)`. If the head contains only one atom with probability 1.0, we may write it deterministically, e.g., `obsObject(o)`. More deterministic CP-rules are illustrated in Sect. 6.

We can also have a prior on the task type. In our scenario, if the task is not given, we assume a uniform distribution. Similarly, if the prior on the object category is not observed, we consider a uniform prior instead. Example 1 illustrates the set of visual observations made about the world in Fig. 2 as CP-rules.

Example 1 Visual observations for the scenario in Fig. 2:

```
obsObject(o).
0.8::part(top,o).
1.0::part(handle,o).
1.0::part(middle,o).
1.0::part(bottom,o).
0.5::pose(o,upright).
empty(o).
0.56::cup(o); 0.36::can(o); 0.05::pot(o);
0.02::pan(o) ← obsObject(o).
```

There are several advantages of using CP-Logic to model complex uncertain dependencies over the more popular graphical models such as Markov networks or Bayesian Networks (BNs). First, CP-Logic is designed to explicitly model causal events (or relationships between random variables). Robotic grasping is characterized by a number of causal

[3] If the sum is less than 1, there is a non-zero probability that nothing happens.

uncertain events which sometimes involve different consequences. For example, if the object has a usable area and a handle, it is likely to be any specific tool category, including a ‘hammer’. This rule is a general, but local piece of information which does not consider other possible causes for the object being a ‘hammer’. This is rather difficult to encode with a BN, as querying for the object being a hammer involves knowing all the possible causes and how they interact with the set of observations. Similarly, if the object is a ‘tool’ and the task is ‘pass’, then it should be rather grasped by the usable area instead of the handle. This involves again local causation.

Second, a CP-Logic theory is more efficient as it is more compact, requires fewer parameters (Meert et al. 2008) and allows parameter sharing by generalizing over similar situations. This is due to its first order nature. Differently, BNs are a probabilistic extension of propositional logic, and thus, they inherit limitations of propositional logic: they have a rigid structure and cannot handle interactions between a variable number of objects in a generic way. Consider building a probabilistic model of a set of kitchen table settings, where the goal is stacking objects in order to clear the table. This is rather difficult to be done with BNs as a kitchen table scenario involves configurations of objects, such as plates, knives, forks, cups. The structure of different settings is, at an abstract level, fairly similar. However, each particular setting would need to be modeled by its own specific BN. In contrast, CP-Logic can generalize over similar object configurations.

Third, due to its first order nature, CP-Logic can intuitively integrate (hierarchical) world knowledge as logic rules. For example, we can easily integrate and exploit object ontologies to reason about object (super-)categories or task-object affordance models. This integration is not obvious with BNs.

6 World knowledge: ontologies and affordances

After we described the visual scene using CP Logic, we introduce the knowledge base used by our PLM. This part is essential to generalize over similar semantic object parts and across object/task categories. The knowledge base makes use of an object category ontology, a task ontology, object-task affordances and environmental constraints.

We consider the object category ontology illustrated in Fig. 6. It structures 11 object categories: $C = \{$*pan, pot, cup, glass, bowl, bottle, can, hammer, knife, screwdriver, cooking_tool*$\}$. The super-categories are defined based on the object functionality, and are represented by: *kitchenContainer, dish, openContainer, canister, container, tool, object*. The super-category *dish* subsumes the categories *bowl*, *glass* and *cup*. By making use of the ontology structure, the grasping model makes abstraction of the fine-grained object categories. The ontology can be extended with new categories.

The ontological knowledge is translated by our probabilistic logic module into deterministic logical rules. They are obtained by encoding the super-category in the head and the finer-grained category in the body. Such mappings are shown in Example 2. The rule `canister(X) ← can(X)` is deterministic and states that “any can is a canister”. The argument X, denoted in uppercase, is a term in the form of a variable. All variables are universally quantified. In other words, the rule `canister(X) ← can(X)` specifies that for all objects X, when X is a can, X is a canister as well.

Example 2 Object ontology mappings to logical rules:

```
kitchen_container(X) ← pan(X).
kitchen_container(X) ← pot(X).
...
canister(X) ← can(X).
canister(X) ← bottle(X).
...
open_container(X) ← kitchen_container(X).
open_container(X) ← dish(X).
container(X) ← open_container(X).
container(X) ← canister(X).
object(X) ← container(X).
object(X) ← tool(X).
```

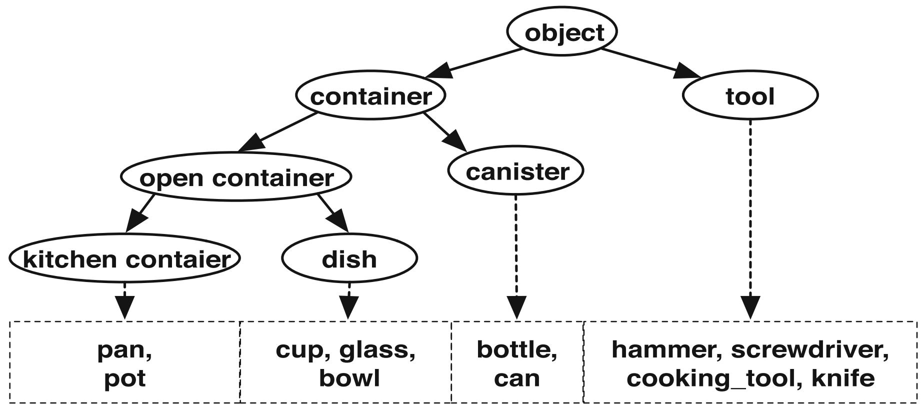
Next, we consider as world knowledge the task ontology in Fig. 7. It structures 7 tasks: $T = \{$*pass, pourOut, pourIn, p&pInUpright, p&pInUpsidedown, p&pInSideways, p&pOn*$\}$.[4] The task *pass* refers to grasping and passing the object to a human in the exact same pose, the tasks *pourOut* and *pourIn* to the actions of pouring liquid out of and inside the object, respectively, after grasping it. Tasks *p&pInUpright*, *p&pInUpsidedown* and *p&pInSideways* refer to picking the object from the current pose and placing it (in a shelf/in a cupboard/on a table) in the upright, upside-down and sideways poses, respectively. Finally, the task *p&pOn* is defined as picking and placing the object on a surface in the same initial pose. Depending on the object pose, its parts and the task to be performed, the object should be grasped in different ways. The task super-categories are: *p&pIn, pickPlace, pour, task*. Task ontology knowledge is translated by our probabilistic logic module into deterministic logical rules as well. The super-task is encoded in the head, while finer-grained task in the body as Example 3 shows. The rule `pour(T) ← pourIn(T)` specifies that “any task of pouring liquid in to fill some object is a pouring task”.

Example 3 Task ontology mappings as logical rules:

```
p&pIn(T) ← p&pInUpright(T).
p&pIn(T) ← p&pInUpsidedown(T).
p&pIn(T) ← p&pInSideways(T).
...
task(T) ← pickPlace(T).
task(T) ← pour(T).
task(T) ← pass(T).
```

[4] The notation *p&p* is abbreviation for *pick and place*.

Fig. 6 Object category ontology



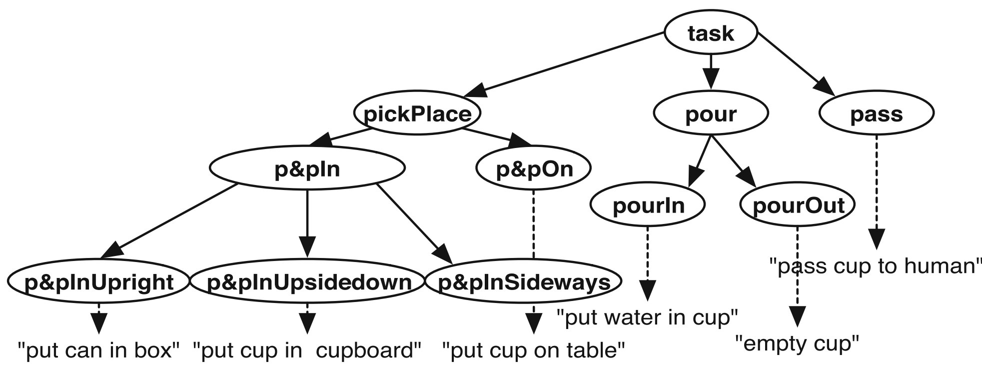


Fig. 7 Task category ontology

While the broader Gibsonian definition of affordance refers to the properties of an object to allow action possibilities offered to it by its surrounding environment, we narrow down the notion of object-task affordances as the tasks afforded by objects in our object grasping scenario, considering the manipulation capabilities of the two-finger gripper mounted on a robotic arm (Fig. 9d). Figure 8 illustrates common sense affordances in the form of a table. They allow us to relate object-task concepts and help us to define the grasping model in a relational way. Both the affordances table and object ontologies were defined by human experience and inspired by *AfNet: The Affordance Network*.[5] They can be extended to include new object or task categories.

By looking at the table, we can define possible object-task affordances. Our probabilistic logic grasping model encodes them as probabilistic logical rules (Example 4). The rule `0.9::possible(X,T) ← object(X), pass(T)` states that any object in our scenario afford the task of passing with a high probability. The body of the rule is a disjunction of the atoms `object(X)` and `pass(T)`. We can also generally state that `0.9::possible(X,T) ← container(X), pour(T)`, that is any container affords the task of pouring. However, this is not always true, as pouring liquid in a canister is an almost impossible task, even for a human. We encode such constraints via the `impossible/2` predicate. The rule `1.0::impossible(X,T) ←canister(X), pourIn(T)` states that a canister does not afford the task of pouring in. The `possible/2` and `impossible/2` predicates are combined by the `affords/2` predicate, giving us the final object-task affordance knowledge.

Example 4 Object-task affordances as probabilistic logical rules:

```
0.9::possible(X,T) ← object(X), pass(T).
0.9::possible(X,T) ← container(X),
pour(T).
0.9::possible(X,T) ← object(X), p&pOn(T).
0.9::possible(X,T) ← container(X),
p&pIn(T).
0.9::possible(X,T) ← tool(X),
p&pInSideways(T).
```

Impossible object-task affordances:

```
1.0::impossible(X,T) ← canister(X),
pourIn(T).
1.0::impossible(X,T) ←
kitchenContainer(X), pourOut(T).
...
```

The final object-task affordance model is given by the rule:

```
affords(X,T) ← possible(X,T),
not(impossible(X,T)).
```

[5] Available at: www.theaffordances.net.

affordances task/object			container							tool			
			open container					canister					
			dish			kitchen							
			cup	glass	bowl	pan	pot	bottle	can	hammer	knife	screwdr	cooking
pass			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
pour	in		✓	✓	✓	✓	✓	-	-	-	-	-	-
	out		✓	✓	✓	-	-	✓	✓	-	-	-	-
p&p	in	upright	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
		upsidedown	✓	✓	✓	-	-	-	-	-	-	-	-
		sideways	-	-	-	-	-	-	-	✓	✓	✓	✓
	on		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fig. 8 Object-task affordances

Finally, next to the affordance rules, we include some other common sense object-task constraints, which hold for our kitchen setup. In Example 5 some of them are shown. They are encoded via the $impossible/2$ predicate and illustrate situations when the object is observed to be filled with content, i.e., $full/1$ is true. If extra object/task categories are added to the scenario, the list of constraints can be increased.

Example 5 Other common sense exceptions:

```
impossible(X,T) ← pan(X), full(X),
pass(T).
impossible(X,T) ← container(X), full(X),
p&pInUpsidedown(T).
...
```

We employ a manually-defined probabilistic affordance model. However, the parameters of the model can be learned to obtain better estimations of the object-task affordances (Moldovan et al. 2012, 2018). They can be also re-estimated by our probabilistic logic module, given the prior distribution over object categories. When the task is not given, it can be inferred from the model. *Querying for the most likely task* in our grasping scenario, implies calculating $argmax_T\, P(\textit{affords(o,T)}|\mathbb{V},\mathbb{M})$, where T is the task variable which unifies with possible tasks. Without grounding T, one obtains a probability distribution over possible tasks. For the cup scenario in Fig. 2, the resulted distribution over the considered tasks is: $P(\textit{affords(o,pass)}) = 0.32$, $P(\textit{affords(o,p\&pOn)}) = 0.32$, $P(\textit{affords(o,p\&pInUpright)}) = 0.32$, $P(\textit{affords(o,pourIn)}) = 0.03$, $P(\textit{affords(o,p\&pInUpsidedown)}) = 0.01$, $P(\textit{affords(o,p\&pSideways)}) = 0.0$, and $P(\textit{affords(o,pourOut)}) = 0.0$ (the cup is empty). This shows the flexibility of our approach.

Now that we introduced our knowledge base, we explain next in more detail the probabilistic logic module.

7 Probabilistic logic module

Our PLM for grasping is defined using CP Logic. Its role is to answer three types of queries: most probable object category, most affordable task and best semantic pre-grasp. Thus, the resulting CP-Logic module for object grasping includes three parts in the form of probabilistic logic rules: knowledge about the world (object/task ontologies and object-task affordances explained in Sect. 6), the object category model and the task-dependent grasping model. After introducing the world knowledge in the previous section, we present the remaining two parts of the PLM in the following two subsections. Note that for all prediction tasks we make the mutually exclusiveness assumption. For object category prediction this implies that an object cannot have several categories at the same time. Similarly, for task selection, this translates into the fact that only one task can be executed at any point in time. We use a ProbLog implementation (Fierens et al. 2011) for the CP-Logic module and we show experimentally that by putting together probabilistic and logical reasoning we improve the grasping performance.

7.1 Probabilistic logic for object categorization

Our probabilistic logic object categorization model answers the query: what is the most probable object category? We can query an object instance o for being, for example, a hammer by using the predicate `category/2` and by calculating the probability of the ground atom $P(category(o, hammer))$. It is possible to ask the query without grounding the specific object category. *Querying for the most likely object category* is equivalent to calculating $argmax_C\, P(category(o, C)|\mathbb{V},\mathbb{M})$, where the variable C indicates the category of o and unifies with defined object categories, $\mathbb{M}$ is the PLM and $\mathbb{V}$ is the conjunction of observations made about the world: the given task, the observed object parts, pose and containment (presented in Sect. 4.2).

The CP-model for object categorization is encoded using CP-events to indicate object category consequences based on the object parts and properties and deterministic rules. Example 6 illustrates some rules extracted from our object category model. The first deterministic rule reads as: if the object has a usable area and a handle and it poses sideways, then it is a tool. Similarly, we can define CP-events showing several possible outcomes in the head of the rule when its body is true. When observed parts are, for example, *bottom*, *middle* and *top*, no handles are detected and the pose is sideways or upright, then the object can be a glass, a bowl or a canister (rules 2 and 3). If the observed pose is upsidedown then the object can be a glass, a bowl or a can (rule 4). If exactly one handle is observed, then the object may be a cup or a pan (rule 5).

Example 6 Rules extracted from the object category model:

```
tool(X) ← part(ua,X), part(ha,X),
pose(X,sideways). (1)

0.25::glass(X); 0.25::bowl(X);
0.5::canister(X) ← part(top,X),
part(middle,X), part(bottom,X),
no_handle(X), pose(X,upright). (2)

0.25::glass(X); 0.25::bowl(X);
0.5::canister(X) ← part(top,X),
part(middle,X), part(bottom,X),
no_handle(X), pose(X,sideways). (3)

0.33::glass(X) ; 0.33::bowl(X)
; 0.33::can(X) ← part(top,X),
part(middle,X), part(bottom,X),
no_handle(X), pose(X,upsidedown). (4)

0.75::cup(X); 0.25::pan(X) ←
part(middle,X), part(top,X),
part(bottom,X), part(handle,X),
pose(X,upright). (5)
...
```

These rules encode generality also by using object super-category atoms in the head (e.g., tool and canister). In order to estimate the object category in these cases, we adapt the original object ontology defined in Sect. 6 to a similar ontology which, in addition, models category distributions with respect to the super-categories across the ontology. This is part of the object categorization model and is done also using CP-events. The causal probabilities are estimated based on the number of categories in the leafs. Example 7 pictures the distribution over *hammer*, *knife*, *screwdriver* and *cooking_tool* caused by the super-category *tool*, and the distribution over *can* and *bottle* caused by the object being a *canister*.

Example 7 Examples of category distributions with respect to the super-categories:

```
0.25::hammer(X); 0.25::knife(X);
0.25::screwdriver(X); 0.25::cooking_tool(X)
← tool(X).

0.5::can(X); 0.5::bottle(X) ←
canister(X).
```

In our experiments, during inference, the object categorization rules are used together with world knowledge and visual observations to form the CP-theory for object categorization. Its parameters should not be interpreted as the conditional probability of the head atom given the body, e.g., $P(can(o)|canister(o)) = 0.5$ is incorrect. It is part of the semantics of CP-Logic that each rule independently makes a head atom true when triggered. Thus, the conditional probability that o is a can, given that o is a canister, may be different than 0.5, in case there is a second possible cause, e.g., a prior knowledge that o is a can with probability 0.36, which contributes to $P(can(o))$. To illustrate the benefits of the categorization model let us reconsider our cup example in Fig. 2. Initially, the manifold shape model predicts for the detected object categories cup, can and pot with probabilities 0.56, 0.36 and 0.05, respectively. However, after querying our model, the new distribution is $P(cat(o, cup)) = 0.98$, $P(cat(o, pan)) = 0.02$, while $P(cat(o, can))$ and $P(cat(o, pot))$ become 0.0. The result is a better probability distribution over object categories than the observed prior.

There are different levels of generalization with respect to the rules of the theory. We experimented also with more general rules to investigate the suitability of our model. They were able to improve the object category prior (see Sect. 9), showing similar behavior and results to the more specific one. Example 8 shows more general rules, which replace, for example, the more specific head `0.25::glass(X); 0.25::bowl(X); 0.5::canister(X)` with the super-category `1.0::container(X)`, while keeping the same body. For the cup example the new distribution with the more general theory becomes $P(cat(o, cup)) = 0.93$, $P(cat(o, pot)) = 0.05$, $P(cat(o, pan)) = 0.02$.

Example 8 More general rules from the object category model:

```
1.0::container(X) ← part(middle,X),
part(top,X), part(bottom,X), no_handle(X),
pose(X,upright).

1.0::container(X) ← part(middle,X),
part(top,X), part(bottom,X), no_handle(X),
pose(X,upsidedown).

0.6::dish(X); 0.4::canister(X)
← part(top,X), part(middle,X),
part(bottom,X), no_handle(X),
pose(X,sideways).

0.5::cup(X); 0.5::kitchen_container(X)
← part(top,X), part(middle,X),
part(bottom,X), one_handle(X).
...
```

7.2 Probabilistic logic for task-dependent grasping

We introduce next our probabilistic logic task-dependent grasping model. Its role is to answer the query: what is the best semantic pre-grasp? We examine possible pre-grasps by querying *P(grasp(o,t,Part))*, where *t* is the given task and *Part* is a variable which unifies with possible object parts. *Querying for the most likely semantic pre-grasp* of an object is equivalent to calculating $argmax_{Part} P(grasp(o,t,Part)|\mathbb{V}, \mathbb{M})$, where $\mathbb{M}$ is the PLM and $\mathbb{V}$ is the conjunction of world observations, to obtain a probability distribution over possible object parts or pre-grasps.

The CP-model for estimating the best pre-grasp is a set of causal probabilistic (CP) events. Each such event generates as consequence the graspability of a certain object part conditioned on the part existence, task, object (super-category and properties). The feasibility of the semantic grasp is encoded via the causal probability. Some examples from the grasping model for the *dish* super-category are illustrated in the Example 9. The first three rules state that if the object is a full dish in the upright pose, the task is pass, and the object affords the task, then, the object can be grasped by the observed *middle* part with probability 0.7, *top* part with probability 0.2 or *handle* with probability 0.1. The next three rules refer to the situation when the dish is in the same pose, but it is empty. Then, it can be passed by grasping it from the detected *bottom* part with probability 0.1, *middle* part with probability 0.7 or *top* part with probability 0.2. Finally, the last rule in the example considers the scenario when the task to be performed on the dish is *pourOut*. In this case, the object can be grasped by the observed *middle* part with probability 1.0.

Example 9 Rules extracted from the semantic pre-grasp model:

```
0.7::grasp(X,T,middle) ← dish(X),
pose(X,upright), full(X), pass(T),
affords(X,T), part(middle,X).
0.2::grasp(X,T,top) ← dish(X),
pose(X,upright), full(X), pass(T),
affords(X,T), part(top,X).
0.1::grasp(X,T,handle) ← dish(X),
pose(X,upright), full(X), pass(T),
affords(X,T), part(handle,X).

0.1::grasp(X,T,bottom) ← dish(X),
pose(X,upright), empty(X), pass(T),
affords(X,T), part(bottom,X).
0.7::grasp(X,T,middle) ← dish(X),
pose(X,upright), empty(X), pass(T),
affords(X,T), part(middle,X).
0.2::grasp(X,T,top) ← dish(X),
pose(X,upright), empty(X), pass(T),
affords(X,T), part(top,X).

1.0::grasp(X,T,middle) ← dish(X),
pourOut(T), affords(X,T), part(middle,X).
...
```

Our CP-theory can also enforce constraints to model impossible pre-grasps. Examples of such constraints are showed below (Example 10). For example, when the object is a *tool* and the task is *pour*, we have an impossible affordance and thus, also an impossible pre-grasp position. This is specified through the first constraint which states that it is impossible for the pre-grasp atom `grasp(X,T,R)` to be true when the body is true. It guarantees that the probability of such grasps is equal to 0.0. Further, when attempting to grasp and a collision happened during grasping execution, our CP model can incorporate feedback to the high-level reasoning module announcing an impossible pre-grasp position. The second constraint rule shows that we can connect the reasoning module to the execution planner by enforcing the probability of a pre-grasp to 0.0, if there are environmental constraints for the gripper. Finally, the third constraint indicates that if the object is a pan in the upside down pose then no task should be executed, as grasping the object in this situation is very difficult.

Example 10 Constraint for impossible affordances:

```
false:- grasp(X,T,R), task(T), object(X),
impossible(X,T),part(R,X).
```

Constraint for collision:

```
false:- grasp(X,T,R), task(T), object(X),
part(R,X), collision(R).
```

Grasping constraint:

```
false:- grasp(X,T,R), pose(X,upsidedown),
pan(X), task(T), part(R,X).
```

In our implementation, the grasping CP-theory used during task-dependent reasoning includes, besides the semantic pre-grasp rules, the object category rules, world knowledge and visual observations about the world. For our cup example in Fig. 2 the distribution over possible grasping parts when the task $t = pass$ is: *P(grasp(o,pass,middle))=0.87*, *P(grasp(o,pass,top))=0.08*, *P(grasp(o,pass,bottom))=0.03*, *P(grasp(o,pass,handle))=0.01*.

Similar to the object categorization theory, there are different levels of generalization with respect to the rules. To test the fittingness of the theory we experimented also with more general rules, by generalizing over the object pose and containment with respect to several tasks and thus, reducing the number of rules. Example 11 generalizes the excerpt of the theory presented above for task *pass* and super-category *dish*.

Example 11 Part of a more general theory of the semantic pre-grasp model:

```
0.1::grasp(X,T,bottom) ← affords(X,T),
pass(T), dish(X), part(bottom,X).
0.6::grasp(X,T,middle) ← affords(X,T),
pass(T), dish(X), part(middle,X).
0.2::grasp(X,T,top) ← affords(X,T),
pass(T), dish(X), part(top,X).
0.1::grasp(X,T,handle) ← affords(X,T),
pass(T), dish(X), part(handle,X).

1.0::grasp(X,T,middle) ← dish(X),
pourOut(T), affords(X,T), part(middle,X).
```

Our (non-optimal) models were defined using human experience and “educated guesses”. They can be augmented by adding extra rules to include new object/task categories. The world knowledge was encoded as general as possible while still reflecting the ontologies and task-object affordances. The parameters of the rules composing the models can, in principle, be learned from data (Meert et al. 2008) to best represent the application domain. Our current results with the quite rigid affordance model can be improved by learning better probability estimates for object-task affordances from data. We made the code of the PLM available online.[6]

8 Low-level grasping planner

The PLM presented in the previous section selects the object pre-grasp given the task to be performed by taking into account semantic information. This high-level result is further complemented by the specialized grasping module. It considers only local shape information (rightmost box in Fig. 2). The low-level grasping module is a two stage planner which estimates: (i) the grasping probability of given a pre-grasp pose and (ii) the arm motion planner for reaching the pre-grasp pose.

The grasping probability of a pre-grasp pose $P(grasp|pre\text{-}grasp\ pose,\ shape)$ relies on the completed point cloud shape of the selected object part (Sect. 4.1). It is estimated by training a binary classifier which discriminates between graspable and non-graspable pre-grasp poses and by mapping the classification output of a Support Vector Machine onto a probability. The classifier is trained using local shape depth features of pre-grasps as explained below.

Depth difference features The local shape features are computed in the volume enclosed by the gripper, which is a bounding box located and oriented according to the pre-grasp hypothesis pose. Depth changes in the objects were shown helpful to recognize graspable regions, even in cluttered environments where objects cannot be segmented accurately (Fischinger et al. 2013). The symmetry height accumulated feature (Fischinger et al. 2013) is robust, but constrained to top grasps only. We introduce a feature with computations based also on heights, however, it can be computed for any grasping orientation. Our feature, called *depth gradient image* (DGI), computes the gradient of the depth image in the volume enclosed by the gripper. This volume defines a depth value (i.e., the height in mm) as the z-component of the distance from the gripper base to the object point. Figure 9c shows an example of the selected region of an object and Fig. 9d illustrates the volume of interest enclosed by the gripper.

The depth image requires a discrete sampling of the volume, which was defined as boxes of $7 \times 7 \times 15$ (mm) and is defined as:

$$\mathrm{DI}(u, v) = \begin{cases} \min\{z\} & \text{if } z \in \mathrm{box}(u, v) \\ -1 & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathrm{box}(u, v)$ represents the set of points inside the box defined at the pixel (u, v). Equation (1) performs an orthogonal projection of the closest point to the base of the gripper for every pixel of the depth image. Figure 10 shows the depth image for the selected volume in Fig. 9d. Finally, the DGI is computed on the depth image by applying pixel differences in u and v as follows:

$$\mathrm{DI}_u(u, v) = \mathrm{DI}(u+1, v) - \mathrm{DI}(u-1, v), \tag{2}$$
$$\mathrm{DI}_v(u, v) = \mathrm{DI}(u, v+1) - \mathrm{DI}(u, v-1), \tag{3}$$
$$\mathrm{DGI}(u, v) = \sqrt{\mathrm{DI}_u(u, v)^2 + \mathrm{DI}_v(u, v)^2}. \tag{4}$$

The DGI acts as a local shape descriptor for the grasping predictor.

Grasping probability Given DGI shape features $\mathbf{x}_i$ and their labels y_i, we use SVMs (Cortes and Vapnik 1995) with Radial Basis Function kernel to discriminate between successful and failed grasps. Before applying the sign function, we map the SVM output onto a probability by applying a sigmoid function to the decision value $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$. We employ the parametric sigmoid to estimate

$$P(grasp|pre\text{-}grasp\ pose,\ shape) = \frac{1}{1 + \exp(Af(\mathbf{x}) + b)}, \tag{5}$$

where the parameters A and b are obtained by generating a hold-out set and cross-validation. Its advantages were shown empirically in Platt (1999).

The arm motion planning searches for collision free paths between the current pose of the arm and a pre-grasp pose. This problem is solved by an existing algorithm, the tree-based motion planner (Sánchez and Latombe 2003) available in the Open Motion Planning Library (Şucan et al. 2012). The software and its ROS node that implements the low-level grasp planning of this section is available online.[7]

9 Experiments

We address the benefits of the PLM for robot grasping experimentally. After we introduce the different experimen-

[6] Available at: https://people.cs.kuleuven.be/~laura.antanas/.

[7] Available at: https://github.com/vislab-tecnico-lisboa/grasping/tree/master/ist_grasp_generation/ist_grasping_point_prediction.

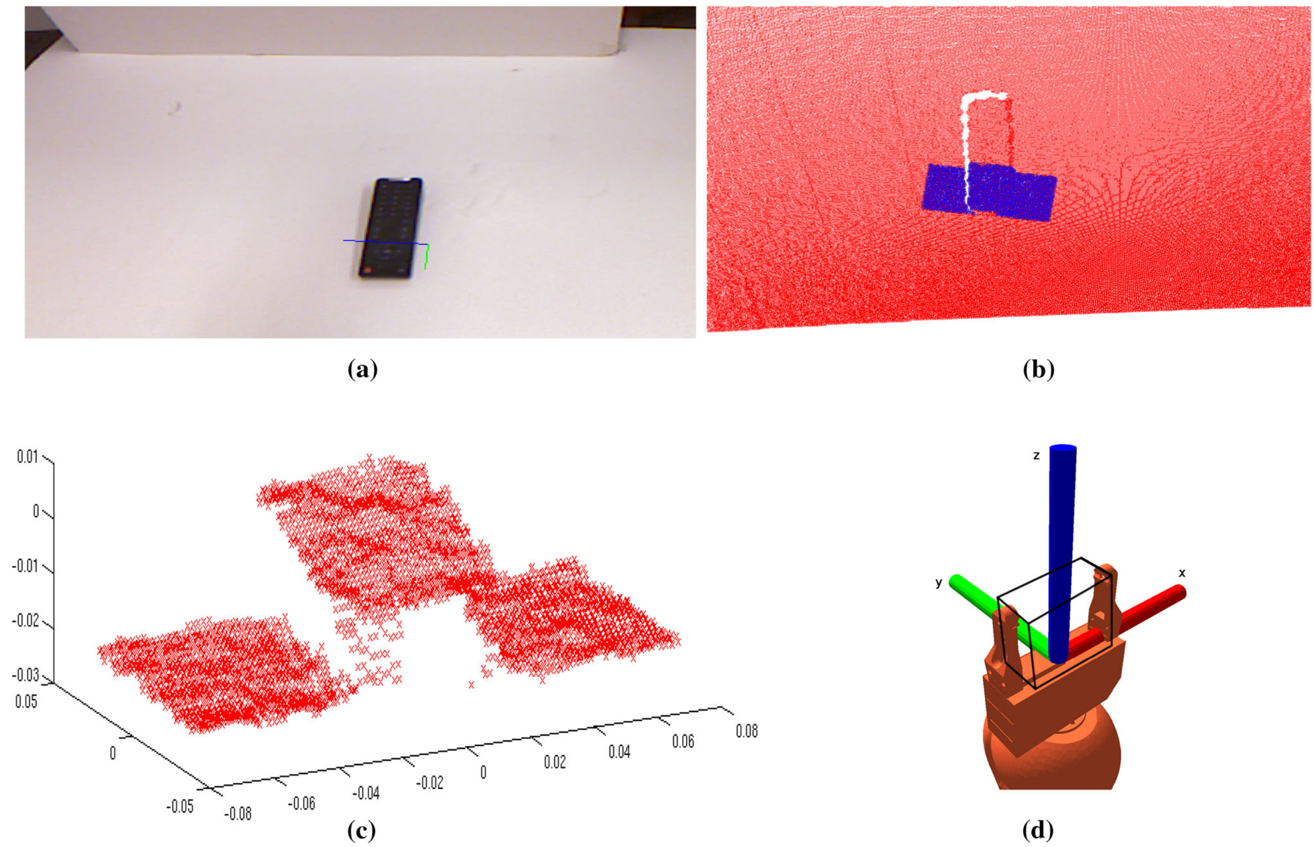


Fig. 9 Illustration of the point cloud region selected for grasping probability computation. **a** Object (remote control). **b** Correspondent point cloud. The blue points show the selected points of a graspable region of the remote control. **c** Cropped grasping cloud of the object; points enclosed by the gripper volume. **d** Gripper and volume of interest, showing the reference frame origin for the orthogonal projection of the DI image (Color figure online)

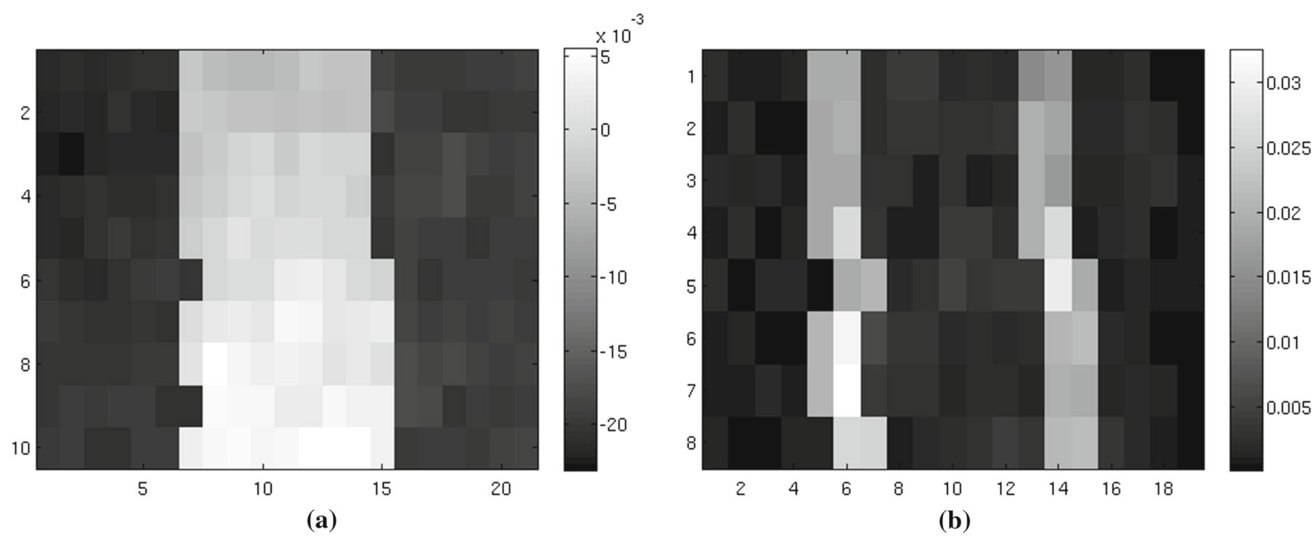


Fig. 10 Example of a depth image (10 × 21 pixels) and its corresponding gradient magnitude (8 ×19 pixels). **a** Depth image. **b** Depth gradient image

tal setups, datasets and evaluation criteria in Sect. 9.1, we present our results in Sect. 9.2 and analyze them in detail in Sect. 9.3. We investigate two main groups of questions: (q1) how robust is the probabilistic logic module, how well does it generalize and cope with missing information? and (q2) does the integration of high-level reasoning and low-level learning improve grasping performance upon local shape features? To answer (q1) we investigate if the PLM: (i) can improve upon similarity-based object classification using the logic-based object categorization model, (ii) can correctly predict suitable tasks for a given scenario, and (iii) can properly predict the pre-grasp region using the task-dependent grasping model. Further, we investigate its robustness with respect to object category, task, pre-grasp and grasping point prediction. We perturb either the visual observations about the world when dropping the prior on the object category, or the CP-theory by employing more general rules. Additionally, we determine how well the PLM can generalize by comparing results obtained with a more general CP-theory to those gathered with a more domain-dedicated one. We evaluate the performance of the framework considering different options. To answer (q2) we additionally learn a classifier that maps points sampled from full objects to successful grasps using solely local shape features. Given a new object, we then directly predict the most likely grasp using local shape information. This is our local shape-based baseline. We compare the baseline with the framework classifier, which further maps points from predicted semantic pre-grasps to successful grasps, using similar local shape features. Given a new object, we first predict, using high-level reasoning, the most likely object pre-grasp region. We then use the classifier to predict, for a given task, good grasping points among the set of points in the inferred pre-grasp part. As in (q1), we evaluate the performance of the full framework, considering different options. To support the research questions, we also investigate how good are the visual perception module and the low-level grasping planner. Specifically, we investigate the performance of the object pose and part detector, the global similarity-based object classifier and the grasping pose classifier without considering the rest of the framework.

9.1 Setup and datasets

We consider three setups to obtain the datasets on which we quantitatively investigate the robustness and power of generalization of our SRL approach. In a first setup, the object point clouds are obtained from 3D meshes and the object parts are manually labeled. In this case the dataset is synthetic and actual grasps are not executed. For the second setup, data samples are obtained from the ORCA simulator (Baltzakis). In the third setup, data samples are obtained from a real robot platform.

Fig. 11 The table is in front of the mobile platform. The range sensor is marked by the green rectangle

Synthetic setup It considers flawless visual detection of objects from 3D meshes. The object points are distributed uniformly on the object surface according to their size by applying the midpoint surface subdivision technique. Point normals are correctly oriented, the object pose and its parts are manually labeled as well as the object containment. This "perfect scenario" serves as an upper-bound comparison scenario to the more realistic scenarios, allowing an extensive evaluation of the generalization capabilities of the PLM. The dataset contains 41 objects belonging to all categories in our ontology and 102 grasping scenarios. We denote this dataset S_{SYN}.

ORCA setup The second setup is used to evaluate all the modules and the full framework in simulation. We use ORCA, which provides the sensors [laser range camera Asus Xtion PRO (ASUS) and the Universal Gripper WSG 50 (W. Robotics) force sensor], robotic arm (KUKA LightWeight Robot (LWR), K. Robotics), objects and interface to physics engine (Newton Game Dynamics library, Jerez & Suero) for robot grasping simulation. The other modules are external to ORCA and interfaced both with the simulated and real robot. These modules include: object completion, part and pose detection, global shape similarity, probabilistic logical reasoning modules, local shape grasping prediction and the tree-based motion planner (Sánchez and Latombe 2003) available in the Open Motion Planning Library (OMPL) (Şucan et al. 2012). Each object is placed on top of a table. Further, we consider four possible settings:

- S_{REAL_semi} Object pose is not estimated but given by the ground truth, while the parts are estimated from the completed point cloud, as explained in Sect. 4. The scene description may have missing parts when they are occluded or not detected. We assign to all detected parts probability 1.0;
- S_{REAL} Both object pose and its parts are estimated from the completed point cloud. While the pose has associated a likelihood, we keep highly confident parts;
- S_{REAL_noisy} We provide, in addition, a part likelihood according to the limitations of the detection algorithm;
- S_{GRASP_noisy} Includes actual grasping tests with the simulated robot. It comprises a subset of the scenarios from the third setting, where all containers are empty and all objects are graspable by the robot. The rationale behind is that it is very difficult to check whether a container is full or if some objects do not fit the gripper capabilities. Due to the ambiguity between upright and upside-down poses when using global shape representations, object poses considered are *upright* or *sideways*.

Each of the first three settings contains 26 different objects, instances of categories *pan, bowl, cup, glass, bottle, can, hammer, screwdriver, knife* and 126 grasping scenarios. The fourth setting contains 18 objects, instances of categories *pan, cup, glass, bottle, can, hammer, screwdriver, knife* and 113 grasping scenarios.

Real robot setup The robot scenario (Fig. 11) considers the same type of tests as those included in S_{GRASP_noisy}. In addition, we evaluate the performance of the framework when two or more objects are in the field of view of the camera and the field of action of the arm, considering three settings of increasing complexity in terms of path planning. The less complex setting (scenario1), considers only two objects which are instances of *glass* and *bottle*, in a way that planning constraints are similar to a single object on the table. The setting with intermediate complexity (scenario2) includes three objects which are instances of *can, hammer* and *screwdriver*. The more complex setting (scenario3) considers four objects which are instances of *bottle, glass* and *cup*. Figure 12 shows the objects of every scenario. In addition to the larger number of objects, we also consider object placement as another criterion for evaluation. Object placement is performed in two steps: (i) plan from the grasp pose to a post-grasp pose and (ii) plan from the post-grasp pose to the grasp pose. We denote this dataset S_{ROBOT}.[8]

[8] The synthetic dataset and part of ORCA datasets are available for download at http://www.first-mm.eu/data.html.

9.1.1 Evaluation measures

We evaluate our experiments in terms of accuracy given by $\frac{\#successes}{\#tries} \cdot 100\%$. We assess a *success* in several ways. Depending on the prediction task, the ground truth is either one value (object categorization, pose detection) or a set of values (task and pre-grasp prediction, part detection). A correct pose detection is considered when the discrete pose predicted matches the ground truth. For object categorization we take as prediction the category with the highest probability and consider it a success if it matches the ground truth category. For the uniform prior, it can be that two or more categories are predicted with the same probability. This case is reported as a false positive. For object part detection we consider a success if all detected object parts match the ground truth.

For task and pre-grasp prediction evaluation, the ground truth Gt of each instance is a set (e.g., the tasks *p&pOn* and *p&pInUpright* may be equally possible in a particular scenario). In this case, we compare the set of best predictions Pr to Gt, where $|Pr| \leq |Gt|$. If $Pr \subseteq Gt$ a success is reported. We present results for different sizes of Pr, such that $|Pr|$ belongs to the set $\{|Gt| - i\}$, with i ranging from 0 (the most restrictive evaluation setting) to $|Gt| - 1$ (the most pertinent setting). We denote the possible evaluation settings as E_i. For the scenarios with robot grasping execution (S_{GRASP_noisy} and S_{ROBOT}) the evaluation must consider the success of grasping execution with respect to the valid grasping hypotheses provided by the framework. In this case the accuracy is given by $\frac{\#correctly\ estimated\ task/part}{\#valid\ grasping\ hypotheses} \cdot 100\%$. As a baseline for the global performance evaluation measure consider the minimum overall chance level of the framework, which depends on the information provided to the robot. For instance, when the object label is provided the overall chance level will take into account the chance level of: (i) the task, (ii) pre-grasp pose and (iii) low-level grasping. this overall chance applies to the S_{ROBOT} experiments.

9.2 Results

In the following we present quantitative experimental results. For all results the best performance is indicated in bold.

9.2.1 Visual perception module

Object pose and part detection results are shown in Table 1. We do not consider the upside-down pose for the tests. Object part detection suffers from part occlusion for particular object poses, reducing the framework performance for object category prediction. In addition to the accuracy, we stress the execution time for the object completion and pose detection. The average execution times are 27.5 and 15.71 ms on a PC using one core of the Intel Xeon (2.67 GHz). These numbers

Fig. 12 Experimental settings with the real robot. Each picture shows the objects utilized for each scenario. Additional object constraints are: the gray bottle of scenario3 is full with water, the white bottle is empty and the coffee container is full of coffee (Color figure online)

Table 1 Accuracy (%) for object part and pose detection

Dataset	Part detection	Pose detection
S_{GRASP_noisy}	84.56	100
S_{ROBOT}	82.14	100

Table 2 Grasping accuracy (%) on the rectangle dataset (Jiang et al. 2011)

Approach	Grasping accuracy
Jiang et al. (2011)	84.7
Lenz et al. (2015)	93.7
Our work	92.63

confirm the computational efficiency of our approach, which allows to make fast decisions. We note that the PCA global representation is able to cope well with object pose detection, considering the table-top assumption and the object categories used.

We train our grasping pose classifier on the grasping rectangle dataset introduced in Jiang et al. (2011). It contains camera images, point clouds and the pre-grasp poses for both successful and failed grasps. The dataset is balanced and contains more than 5K positive and negative samples. We remove objects with very noisy point clouds, so in total we have 4708 samples (2424 positive and 2284 negative). We apply fourfold cross-validation to find the best RBF and sigmoid parameters. The grasping accuracy defined in Jiang et al. (2011) selects the top grasping region per object and then compares it to the ground truth. Table 2 shows that our approach using only depth features has good performance, improving the result in Jiang et al. (2011). A better result is obtained in [6], however they rely on both image and depth features.

Object category prediction We evaluate the object category prediction, using two approaches: (i) manifold-based object model (i.e. part labels in a graph + object shape), and (ii) the PLM categorization model. As a baseline for the accuracy, the chance classification level is 9%, in the problem of 11 categories along with the part labels. Table 3 reports the classification accuracy of the various methods in order to answer question (q1-i). We compare results of a linear kernel among label counts (Label fractions), propagation kernels (Global similarity), and the PLM with and without the similarity-based prior. Label fractions is a baseline which computes a linear base kernel using only label counts as features without any manifold information. In order to evaluate the performance of the PLM's object category predictor when dealing with missing information, we experiment also without the manifold prior on the object category. PLM$_{uniform}$ indicates the setting with a uniform prior on the object category. We vary the generality of our categorization theory to test the robustness of the PLM. $PLM^{general}$ indicates the more general theory with the manifold prior. $PLM^{general}_{uniform}$ indicates experiments with the more general theory setting and the uniform prior.

9.2.2 Task prediction and pre-grasp selection

We investigate whether our PLM can correctly predict suitable tasks for a given scenario (q1-ii). As a baseline, the chance level of task assignment in the problem of 7 tasks is 14.3%, which is challenging due to the subtle differences between some of the tasks (e.g. p&pInUpright, p&pInSideways, p&pInUpsidedown). Results on all S_{REAL} and S_{SYN} are reported in Table 4 with the most restrictive evaluation setting E_0 for both general and more specific object categorization theories. Figure 13 presents results for the more specific object categorization theory using all evaluation settings, with and without the manifold prior for all the datasets.

Next, we investigate (q1-iii), that is if our PLM can properly predict the pre-grasp region when the task is given. As a baseline for the accuracy, the chance level of random selection of an object part is in between 25 and 50%, corresponding to the number of detected parts (2–4). We consider the spe-

Table 3 Accuracy (%): PLM versus global similarity versus baseline for object categorization

Dataset	Label fractions	Global similarity	$\text{PLM}_{uniform}$	PLM	$\text{PLM}^{general}$
S_{SYN}	63.4	*87.8*	31.37	**93.14**	92.16
S_{REAL_semi}	*39.7*	*39.7*	14.29	**49.21**	*46.83*
S_{REAL}	*39.7*	*39.7*	14.29	**48.41**	*46.83*
S_{REAL_noisy}	**39.7**	**39.7**	14.29	**39.7**	**39.7**

The random category assignment is 9%. Italics indicate similar values for different settings

Table 4 Accuracy (%): PLM for task and pre-grasp prediction

Dataset	PLM		$\text{PLM}^{general}$	
	Pre-grasp	Task	Pre-grasp	Task
S_{SYN}	**85.29**	**72.55**	84.73	**72.55**
S_{REAL_semi}	85.26	**95.24**	**86.73**	93.65
S_{REAL}	**85.26**	**95.24**	84.69	93.65
S_{REAL_noisy}	85.49	**93.65**	**86.73**	**93.65**
S_{GRASP_noisy} E_0	75.51	35.71	–	–
S_{GRASP_noisy} E_1	75.51	50.00	–	–
S_{ROBOT} E_0	66.7	25.00	–	–
S_{ROBOT} E_1	66.7	75.00	–	–

The random task assignment is 14.3%, and the random pre-grasp assignment is in between 25–50%

cific object categorization theory and experiment with both a more specific and a more general task-dependent grasping theory. The results for both settings are shown in Table 4 for evaluation setting E_0 (top rows). Results of the PLM pre-grasp predictor for all evaluation settings are reported in Fig. 14.

9.2.3 Low-level grasping planning versus semantic and geometric reasoning

The selection of grasping points using only local shape descriptors bias the ranking of the object points towards the most visually graspable, disregarding other constraints such as the pre-grasp pose for task execution, path planning and post-grasp object pose. By adding those constraints, regions with lower visually graspable probability will become more important when considering the task execution probability and vice versa. Thus, in order to answer question (q2) we compute the percentage of grasping points that have a low "visually graspable" probability (Sect. 8), but still lead to successful grasps when taking into account task constraints. Results are shown in Table 5, having as baseline the local shape-based approach. We also investigate its robustness with respect to grasping point prediction by dropping the prior on the object category ($\text{Full}_{uniform}$). The full framework selects in average more points having low visually graspable probabilities than the other options. This behavior confirms the importance of task constraints on the computation of the grasping probability. We note that the complete framework clearly improves upon using the local shape-based approach. This answers affirmatively (q2). Additionally, the results obtained by $\text{Full}_{uniform}$ show again the robustness of the framework.

Finally, we present results for the grasp and place action discriminated by the level of complexity for the S_{ROBOT}

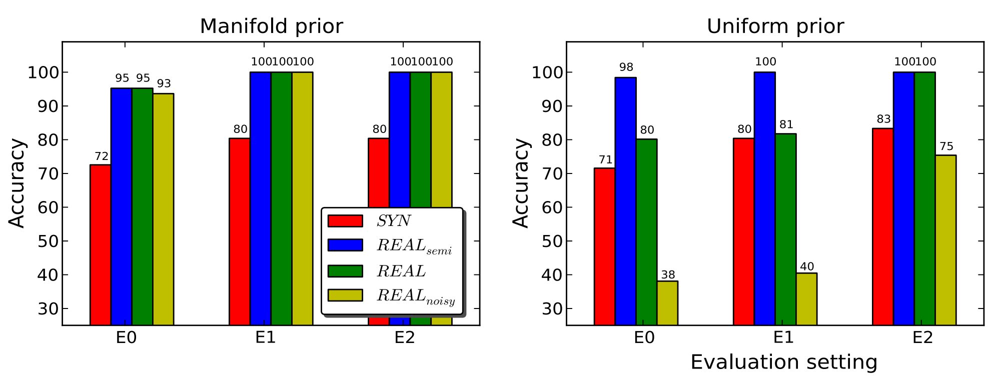


Fig. 13 Accuracy (%): PLM for task prediction using all evaluation settings

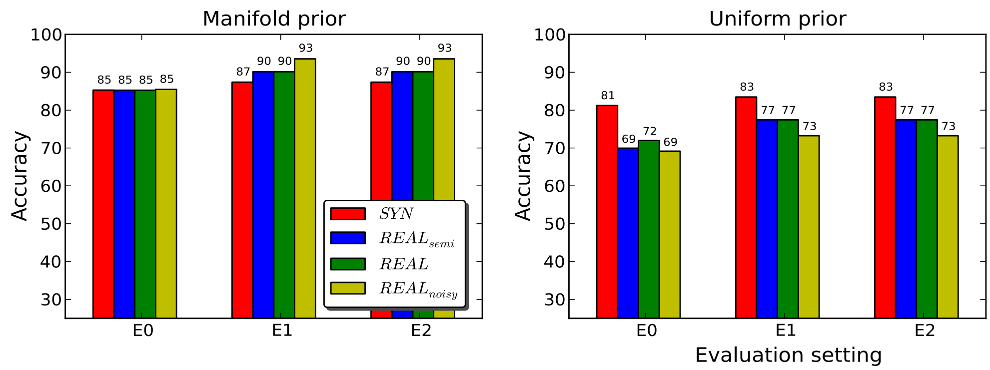


Fig. 14 Accuracy (%): PLM for pre-grasp prediction for all evaluation settings

Table 5 Percentage (%) of successfully graspable points that have visually graspable probability less than (lt) 0.3, 0.4 or 0.5: full versus local shape grasp prediction

Feature (DGI)	Measure	Local (%)	Full (%)	$\text{Full}_{uniform}$(%)
S_{GRASP_noisy}	lt 0.5	44.55	**52.38**	50
	lt 0.4	27.27	**28.57**	21.15
	lt 0.3	4.55	**7.94**	7.69
S_{ROBOT}	lt 0.5	37.5	**50**	46.15
	lt 0.4	25	**42.86**	23.08
	lt 0.3	12.5	**28.57**	23.08

Table 6 Percentage of successful grasps in the real robot scenarios

Scenario	Total tests	Reachable pre-grasps	Grasped objects	Placed objects
Scenario1	10	9 (90%)	7/9 (77.8%)	7/9 (77.8%)
Scenario2	15	10 (66.7%)	7/10 (70%)	7/10 (70%)
Scenario3	20	16 (80%)	8/16 (50%)	8/16 (50%)
Total	45	35 (77.8%)	22/35 (62.9%)	22/35 (62.9%)

Different levels of S_{ROBOT} complexity

dataset in Table 6. As a baseline for the global performance we consider the overall chance level of the framework ($3.21\% = 0.09 \times 0.14.3 \times 0.25 \times 100$). When there are few objects on the table (scenario1 and scenario2), the framework is able to grasp successfully on 73.7% of the experiments.

9.3 Discussion

In this section we analyze the results and discuss the impact of the different parts of the semantic reasoning.

9.3.1 Object category prediction

Table 3 shows that, for the synthetic dataset S_{SYN}, the addition of local object structure improves the accuracy more than 20% upon using label counts only. We also note a performance drop of propagation kernels on the S_{REAL} scenarios. This is due to the fact that part assignment is achieved by applying the part detector introduced in Sect. 4.1. This detector was designed to work for general objects (i.e. we assume the scenarios to be as realistic as possible) and its performance is rather poor for some object categories. Nevertheless, manifold information gives a good prior distribution on object categories (details in Neumann et al. 2013).

Further, the PLM improves object categorization accuracy upon global similarity on three of the datasets and equals it on S_{REAL_noisy}. This result confirms that the uncertainty introduced by the global-similarity approach is corrected in certain situations by the high-level rules of the PLM. Thus, it improves the semantic perception of the object and obtains better results. Its lower performance on S_{REAL} compared to S_{SYN} is linked to the drop in performance of the similarity-based object label predictor. By removing the manifold prior ($\text{PLM}_{uniform}$ column in Table 3), the PLM gives a result on S_{SYN} that is 3 times better than random category assignment (which gives an accuracy of 9%). It improves accuracy with 22.27% on S_{SYN} and with 5.19% on S_{REAL} upon random category assignment. Thus, $\text{PLM}_{uniform}$ behaves reasonably well, given that it relies solely on high-level rules and it is not provided with any object shape information.

We note that increasing the generality of the theory (PLM^{general}) keeps good results and it is still able to improve upon the prior. Furthermore, it does not loose much in terms of performance compared to the more specific theory when manifold information is used, keeping similar results as those of $PLM_{uniform}$. The results obtained using this evaluation setting explain the good performance for the other grasping tasks. More specifically, the PLM assigns an equal probability to all categories that belong to a super-category. Estimating the category of an object as any of the sub-categories of a super-category in the ontology is, thus, satisfactory to predict good semantic pre-grasps.

9.3.2 Task prediction

Table 4 indicates that the PLM's accuracy is above 70% on task prediction, and that PLM^{general} does not loose much in terms of accuracy when compared to PLM. In addition, Fig. 14 shows that PLM$_{uniform}$ performances are close the ones of PLM, except for S_{REAL_noisy} where the PLM is clearly better. There were few cases for S_{GRASP_noisy} and S_{ROBOT} where the uniform prior provided better results. These are explained by the fact that the complete framework failed on the PLM experiment more than the uniform prior due to uncertainty. In addition, the usage of the manifold prior gives significantly better results in the most difficult setting, S_{REAL_noisy}. For the other evaluation settings, in most situations, the PLM returns, although not always as first option, a correct possible task with or without a prior. In the scenarios with grasp execution (bottom rows in Table 4), the evaluation settings E_0 and E_1 consider the outcome of the grasping action. The additional source of failures on grasping include uncertainty on the pose of the objects and the gripper, which are caused by the uncertainty of the point cloud and the object completion. These issues have effects on the performance of the planner, for instance placing the gripper a bit misaligned or hitting the object before closing the gripper. We also stress the advantage of using a prior probability distribution over the object categories, rather than the top category. The same experiments only with the top predicted category give accuracies of 95.24, 89.68 and 89.68% for S_{REAL_semi}, S_{REAL} and S_{REAL_noisy}, respectively, which are lower than using the full prior.

9.3.3 Pre-grasp selection

With respect to pre-grasp selection we note that increasing the generality of the model (PLM^{general}), we do not loose in terms of accuracy (Table 4). This result confirms generalization over similar object parts and object/task categories, which implies that if the input object is an unseen category, such as a paint roller or a vase, the grasping framework is robust enough to return a good grasping part. This allows us to experiment with a wide range of object/task categories and lets us to believe that our approach can be extended beyond the categories used (by augmenting the PLM with extra rules). The uniform prior model has a lower accuracy than the PLM, in average 8% less over all datasets (Fig. 14). In all cases the object category provided by the manifold gives a better performance than the uniform category prior, showing the benefits of the framework. Again, it is important to consider the full prior distribution as input to the PLM, instead of only the top category. Our experiments using solely the top category resulted in accuracies of 81.63, 81.63 and 82.88, respectively, for the three S_{REAL} datasets, which are lower than the ones using the full prior.

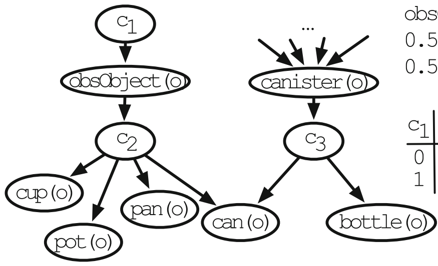
We emphasize that in the scenarios with grasp execution, the evaluation considers the outcome of the grasping action. We also note that it does not consider mutual exclusiveness among possible pre-grasp outcomes. For the other datasets we performed experiments both with and without this assumption. The means of the differences between the mutually exclusive and the non-exclusive results for object categorization, task prediction and pre-grasp prediction (1.6, 8.5 and 2.6%, respectively) indicate a better result for the mutually exclusive setup. This lets us to believe that current robot results can be improved by using the exclusiveness assumption.

9.3.4 Real scenario evaluation of the semantic reasoning

We emphasize the good performance of the framework in Table 6 for the very complex scenario3, where the presence of multiple objects reduce largely the number of available pre-grasp poses. Considering all three scenarios, the framework leads to successful grasps in around 63% of the experiments where there are motion planning solutions in the scenario, which is around 20 times better than the overall chance level. This result show in practice that the semantic task selection and the semantic object part selection of the PLM guide the robot to execute successfully grasp and place actions.

9.3.5 Comparison to BNs

We compare CP-Logic to BNs for our setup using concrete examples. Figure 15 shows three object categorization CP-rules and their equivalent BN, which has the same parameters. In the converting procedure (Meert et al. 2008), the CP-theory parameters appear in the network's CPTs, and all other CPT entires are either 0.0 or 1.0. For every atom in the CP-theory, a Boolean variable is created in the BN. For every rule in the CP-theory a choice variable is created, such that this variable can take $n+1$ values, with n being the number of atoms in the head. If an atom is in the head of a rule, an edge




```
obsObject(o).
0.56::cup(o); 0.36::can(o); 0.05::pot(o); 0.02::pan(o) <- obsObject(o).
0.5::can(o); 0.5::bottle(o)  <- canister(o).
```

c_1
0
1

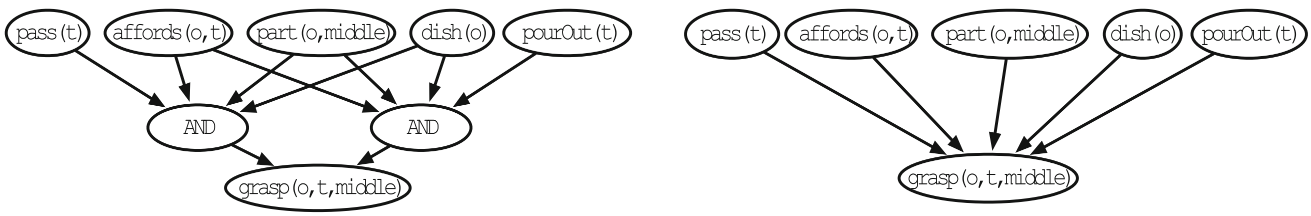
obsObject(o)	c_1	
	0	1
T	0	1
F	1	0

c_2		obsObject(o)	
		T	F
	0	0	1
cup(o)	1	0.56	0
can(o)	2	0.36	0
pot(o)	3	0.05	0
pan(o)	4	0.02	0

can(o)	c_2, c_3							
	0,0	0,1	0,2	...	2,0	2,1	2,2	...
T	0	1	0	...	1	1	1	...
F	1	0	1	...	0	0	0	...

Fig. 15 The equivalent BN (left) for the exemplified CP-theory (top right) extracted from our probabilistic logic object categorization model. The CP-theory parameters appear in the CPTs of the choice nodes (right). The CPT for the *can(o)* node represents the deterministic relationship $c_2 = 1 \vee c_3 = 1$

```
0.6::grasp(o,t,middle) <- affords(o,t), pass(t), dish(o), part(middle,o).
1.0::grasp(o,t,middle) <- affords(o,t), pourOut(t), dish(o), part(middle,o).
```



grasp(o,t,middle)	affords(o,t), part(o,middle), dish(o), pass(t), pourOut(t)						
	T,T,T,T,T	T,T,T,T,F	T,T,T,F,T	T,T,F,T,T	...	T,F,F,F,F	F,F,F,F,F
T	1	0.6	1	0	...	0	0
F	0	0.4	0	1	...	1	1

Fig. 16 Regular BN for CP-rules with multiple atoms in the rule bodies. CP-rules from our probabilistic logic grasping model (top). Regular BN combining noisy-OR and AND (middle right) and its CPTs (bottom). BN with additional nodes for the deterministic AND (middle left)

is created from its corresponding choice variable towards a Boolean variable. The CPTs are constructed in the following way. The choice variable takes the value i if the ith atom from the head is chosen the the probabilistic process. If the body of the rule is true, then the probability that the variable takes the value i is the causal probability given in the head of the rule. The CPT of an atom variable represents a deterministic OR function of the different rules having the atom in the head. If one of the choice variables representing a rule with the given atom in position i of its head takes the value i, then the atom will be true with probability 1.0. In all other cases it will be false, with probability 1.0.

One should, however, compare equivalent BNs to regular BNs defined over the original atoms, that is networks with precisely one Boolean node for each atom in the domain and no additional (unobserved) nodes. CP-theories with at most one atom in the body and at most one atom in the head can be represented as a BN consisting of only noisy-OR nodes. Nevertheless, our PLM consists of CP-rules with more than one atom in the body or/and in the head. When this happens, it is no longer possible to represent the theory as a BN with only noisy-OR nodes. In the following we show that, using regular BNs instead of CP-Logic implies more parameters, less interpretability and intractable learning (more details in Meert et al. 2008).

Let us consider the cup grasping scenario in Fig. 2. Example 11 shows pre-grasp rules for any dish when the tasks considered are pass and pour out. We look at the second and last rules which give the highest probability to the middle part. They are CP-rules with multiple atoms in the rule bodies. To represent them as regular BNs, one either has to add additional nodes for the deterministic AND between the literals in the body, or one has to combine the noisy-OR and AND function in one CPT. Figure 16 illustrates both possibilities. Combining noisy-OR and AND results in many entries having the same value in the CPT. The addition of separate AND nodes avoids this redundancy and is easier to interpret.

When there are multiple atoms in the head, we reconsider the object categorization example in Fig. 15. The regular BN conversion has to consider the mutually exclusiveness of the head atoms. Modeling this relationship without any additional (choice) nodes requires a highly connected network (including cycles). Figure 17 shows the resulting BN. Edges between the head atoms are mandatory as one atom is true only if the others are false. Also, even though the fact that a canister does not cause directly the presence of a cup, there is a direct edge from $canister(o)$ to $cup(o)$, as

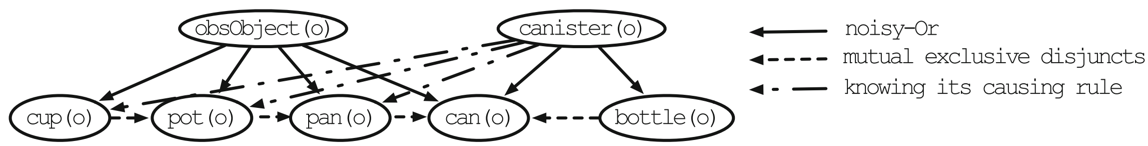


Fig. 17 Regular BN for CP-rules with multiple atoms in the rule heads

the two atoms are not independent. When $canister(o)$ is not true, the cup will be triggered, for example, by the super-category $dish$ and when $canister(o)$ is true, the object must be either a can or a bottle, but there still is a possibility that cup is triggered by the super-category $dish$. Learning such complex networks is intractable, unless dedicated techniques are developed. The equivalent BN avoids the fully connected network by introducing the choice nodes: the choice variable encodes exactly which of the head atoms is selected by the probabilistic process.

Thus, CP-Logic offers an elegant way of encoding mutually exclusive consequences in the head and joint conditions in the body. Not only CP-Logic is more intuitive to understand, and thus, better to incorporate world knowledge and constraints, but it also requires fewer parameters than BNs. This makes such relationships easier to learn in CP-Logic compared to BN, where, unless dedicated techniques, learning such complex networks is intractable (Meert et al. 2008). An experimental comparison to a BN could be further defined, however, unless both the BN and the CP-model are learned, the comparison is highly subjective and beyond the purpose of this paper.

10 Conclusions

We proposed a new probabilistic logic framework combining high-level reasoning and low-level learning for robot grasping. The high-level reasoning leverages symbolic world knowledge, in the form of object/task ontologies and object-task affordances, object category and task-based information. The low-level learning is based on visual shape features. The non-trivial realization of high-level knowledge relies on logic, which exploits world knowledge and relations to encode compact grasping models that generalize over similar object parts and object/task categories in a natural way. When combined with probabilistic reasoning, our proposed framework shows robustness to the uncertainty in the world and missing information.

Our experiments confirm the importance of high-level reasoning and world-knowledge as opposed to using solely local shape information for robot grasping. The integration of high-level reasoning and low-level learning improves grasping performance. First, the PLM generalizes well over object/task categories and copes well with missing information in object categorization and grasp planning. Second, the framework is able to transform low-level visual perception into semantic object parts which represent the task-based dependent grasping adequately. Third, the final grasp execution relies on both the low-level grasping classifier and the probabilistic logic output. Thus, the combination of low-level percepetion and high-level reasoning provides good semantic task-based grasping skills. The robot is able to perform 7 tasks on 11 object categories while dealing with uncertainty in realistic scenarios.

We point out three important directions for future work. The first suggestion is to learn the parameters and structure of our logic theory from data (Antanas et al. 2018). Further, we would like to exploit recent advances in the research on persistent homology (Zhu 2013). Persistent homology is a tool from topological data analysis performing multi-scale analysis on point clouds identifying clusters, holes, and voids therein. Incorporating such a technique into our framework can improve the current part detector. Another direction to be considered is planning a sequence of actions in order to fulfill the task-dependent pre-grasp poses. Since planning in presence of multiple objects raises complexity and generalization issues, considering relational planners similar to those in Moldovan et al. (2013b) is another promising avenue of future work.

References

Abdo, N., Kretzschmar, H., & Stachniss, C. (2012). From low-level trajectory demonstrations to symbolic actions for planning. In *ICAPS workshop on combining task and motion planning for real-world applications* (pp. 1–8).

Aleotti, J., & Caselli, S. (2011). Part-based robot grasp planning from human demonstration. In *ICRA* (pp. 4554–4560).

Antanas, L., Dries, A., Moreno, P., & De Raedt, L. (2018). Relational affordance learning for task-dependent robot grasping. In N. Lachiche & C. Vrain (Eds.), *Inductive logic programming* (pp. 1–15). Cham: Springer.

ASUS, Asus xtion pro. http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/. Accessed 15 July 2014.

Baltzakis, H. Orca simulator. http://www.ics.forth.gr/cvrl/_software/orca_setup.exe. Accessed 15 July 2014.

Barck-Holst, C., Ralph, M., Holmar, F., & Kragic, D. (2009). Learning grasping affordance using probabilistic and ontological approaches. In *International conference on advanced robotics* (pp. 1–6).

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-d shapes. *TPAMI*, *14*(2), 239–256.

Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., et al. (2011). Mind the gap—robotic grasping under incomplete observation. In *ICRA* (pp. 686–693).

Bohg, J., & Kragic, D. (2010). Learning grasping points with shape context. *RAS*, *58*(4), 362–377.

Bohg, J., Welke, K., Leon, B., Do, M., Song, D., Wohlkinger, W., et al. (2012). Task-based grasp adaptation on a humanoid robot. In *IFAC symposium on robot control* (pp. 779–786).

Cocora, A., Kersting, K., Plagemann, C., Burgard, W., & Raedt, L. D. (2006). Learning relational navigation policies. In *IEEE/RSJ ICIRS* (pp. 2792–2797).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Dang, H., & Allen, P. K. (2012). Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *IEEE/RSJ ICIRS* (pp. 1311–1317).

Detry, R., Ek, C. H., Madry, M., & Kragic, D. (2012a). Compressing grasping experience into a dictionary of prototypical grasp-predicting parts. In *International workshop on human-friendly robotics* (pp. 1–1).

Detry, R., Ek, C. H., Madry, M., & Kragic, D. (2013). Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *ICRA* (pp. 601–608).

Detry, R., Ek, C. H., Madry, M., Piater, J. H., & Kragic, D. (2012b). Generalizing grasps across partly similar objects. In *ICRA* (pp. 3791–3797).

Fierens, D., Van den Broeck, G., Thon, I., Gutmann, B., & De Raedt, L. (2011). Inference in probabilistic logic programs using weighted CNF's. In F. Gagliardi Cozman, A. Pfeffer (Eds.) *UAI* (pp. 211–220).

Figueiredo, R., Moreno, P., & Bernardino, A. (2017). Automatic object shape completion from 3d point clouds for object manipulation. In *International joint conference on computer vision, imaging and computer graphics theory and applications* (Vol. 4: VISAPP, pp. 565–570).

Fischinger, D., Vincze, M., & Jiang, Y. (2013). Learning grasps for unknown objects in cluttered scenes. In *ICRA* (pp. 609–616).

Flanagan, J. R., Bowman, M. C., & Johansson, R. S. (2006). Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, *16*(6), 650–659.

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

Hanheide, M., Gretton, C., Dearden, R., Hawes, N., Wyatt, J. L., Pronobis, A., et al. (2011). Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *IJCAI* (pp. 2442–2449).

Hart, S., Grupen, R. A., & Jensen, D. (2005). A relational representation for procedural task knowledge. In *AAAI* (pp. 1280–1285).

Jerez, J., & Suero, A. Newton game dynamics. Open-source physics engine. http://www.newtondynamics.com. Accessed 15 July 2014.

Jiang, Y., Moseson, S., & Saxena, A. (2011). Efficient grasping from rgbd images: Learning using a new rectangle representation. In *ICRA* (pp. 3304–3311).

Kroemer, O., Amor, H. B., Ewerton, M., & Peters, J. (2012). Point cloud completion using extrusions. In *Humanoids* (pp. 680–685).

Kulick, J., Toussaint, M., Lang, T., & Lopes, M. (2013). Active learning for teaching a robot grounded relational symbols. In *IJCAI* (pp. 1451–1457). AAAI Press.

K. Robotics, Kuka lightweight robot (LWR). http://www.kuka-robotics.com/en/products/addons/lwr/. Accessed 15 July 2014.

Lang, T., & Toussaint, M. (2010). Planning with noisy probabilistic relational rules. *JAIR*, *39*, 1–49.

Lenz, I., Lee, H., & Saxena, A. (2015). Deep learning for detecting robotic grasps. CoRR arXiv:1301.3592.

Limketkai, B., Liao, L., & Fox, D. (2005). Relational object maps for mobile robots. In *IJCAI* (pp. 1471–1476).

Madry, M., Song, D., Ek, C. H., & Kragic, D. (2012a). Robot bring me something to drink from: Object representation for transferring task specific grasps. In *ICRA Workshop on semantic perception, mapping and exploration* (pp. 1–6).

Madry, M., Song, D., & Kragic, D. (2012b). From object categories to grasp transfer using probabilistic reasoning. In *ICRA* (pp. 1716–1723).

Marton, Z. C., Rusu, R. B., Jain, D., Klank, U., & Beetz, M. (2009). Probabilistic categorization of kitchen objects in table settings with a composite sensor. In *IEEE/RSJ ICIRS* (pp. 4777–4784).

Meert, W., Struyf, J., & Blockeel, H. (2008). Learning ground CP-logic theories by leveraging bayesian network learning techniques. *Fundamenta Informaticae*, *89*(1), 131–160.

Mitra, N. J., Guibas, L., & Pauly, M. (2006). Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics*, *25*(3), 560–568.

Moldovan, B., Antanas, L., & Hoffmann, M. (2013a). Opening doors: An initial SRL approach. *Lecture Notes in Computer Science*, *7842*, 178–192.

Moldovan, B., Moreno, P., Nitti, D., Santos-Victor, J., & De Raedt, L. (2018). Relational affordances for multiple-object manipulation. *Autonomous Robots*, *42*(1), 19–44.

Moldovan, B., Moreno, P., & van Otterlo, M. (2013b) On the use of probabilistic relational affordance models for sequential manipulation tasks in robotics. In *ICRA* (pp. 1290–1295).

Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., & De Raedt, L. (2012). Learning relational affordance models for robots in multi-object manipulation tasks. In *ICRA* (pp. 4373–4378).

Montesano, L., & Lopes, M. (2012). Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions. *RAS*, *60*(3), 452–462.

Muja, M., & Ciocarlie, M. Table top segmentation package. http://www.ros.org/wiki/tabletop_object_detector. Accessed 15 July 2014.

Neumann, M., Garnett, R., Bauckhage, C., & Kersting, K. (2016). Propagation kernels: Efficient graph kernels from propagated information. *Machine Learning*, *102*(2), 209–245.

Neumann, M., Moreno, P., Antanas, L., Garnett, R., & Kersting, K. (2013). Graph kernels for object category prediction in task—dependent robot grasping. In *MLG* (pp. 1–6).

Neumann, M., Patricia, N., Garnett, R., & Kersting, K. (2012). Efficient graph kernels by randomization. In *ECML/PKDD* (pp. 378–393).

Nyga, D., Balint-Benczedi, F., & Beetz, M. (2014). PR2 Looking at things: Ensemble learning for unstructured information processing with Markov logic networks. In *IEEE international conference on robotics and automation (ICRA)*, Hong Kong, China.

Palinko, O., Sciutti, A., Rea, F., & Sandini, G. (2014). Weight-aware robot motion planning for lift-to-pass action. *International conference on human-agent interaction* (pp. 193–196). New York, NY, USA: ACM.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers* (pp. 61–74). Cambridge: MIT Press.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *3DIM* (pp. 145–152).

Sánchez, G., & Latombe, J.-C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. *Robotics research, Volume 6 of Springer tracts in advanced robotics* (pp. 403–417). Berlin: Springer.

Saxena, A., Wong, L. L. S., & Ng, A. Y. (2008). Learning grasp strategies with partial shape information. In *AAAI* (pp. 1491–1494). AAAI Press.

Song, D., Huebner, K., Kyrki, V., & Kragic, D. (2010). Learning task constraints for robot grasping using graphical models. In *IEEE/RSJ ICIRS* (pp. 1579–1585).

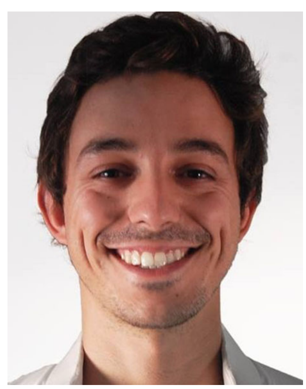
Şucan, I. A., Moll, M., & Kavraki, L. E. (2012). The open motion planning library. *Robotics & Automation Magazine*, *19*(4), 72–82.
Sweeney, J., & Grupen, R. A. (2007). A model of shared grasp affordances from demonstration. In *Humanoids* (pp. 27–35).
Tenorth, M., & Beetz, M. (2009). Knowrob knowledge processing for autonomous personal robots. In *IEEE/RSJ ICIRS* (pp. 4261–4266).
Tenorth, M., Profanter, S., Balint-Benczedi, F., & Beetz, M. (2013). Decomposing cad models of objects of daily use and reasoning about their functional parts. In *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5943–5949). IEEE.
Thrun, S., & Wegbreit, B. (2005). Shape from symmetry. In *ICCV* (pp. 1824–1831).
Toussaint, M., Plath, N., Lang, T., & Jetchev, N. (2010). Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference. In *ICRA* (pp. 385–391).
Vennekens, J., Denecker, M., & Bruynooghe, M. (2009). CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, *9*(3), 245–308.
Winkler, J., Bartels, G., Msenlechner, L., & Beetz, M. (2012). Knowledge enabled high-level task abstraction and execution. *Conference for Advances in Cognitive Systems*, *2*(1), 131–148.
W. Robotics. Universal gripper WSG 50. http://www.weiss-robotics.de/gripper-systems/gripper-modules/universal-gripper-wsg-50.html. Accessed 15 July 2014.
Yousef, H., Boukallel, M., & Althoefer, K. (2011). Tactile sensing for dexterous in-hand manipulation in roboticsa review. *Sensors and Actuators A: Physical*, *167*(2), 171–187 (Solid-State Sensors. Actuators and Microsystems Workshop).
Zhu, X. (2013). Persistent homology: An introduction and a new text representation for natural language processing. In *IJCAI* (pp. 1953–1959). AAAI Press.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Laura Antanas is a post-doc in the Computer Science Department, DTAI group at KULeuven, Belgium working with Prof. Luc De Raedt. She received her M.Sc. in Computer Engineering from the Ecole Superieure Des Telecommunications de Bretagne (France, 2007). She received her Ph.D. from KULeuven, in 2014. Her research interests include artificial inteligence, machine learning and its applications. Her work is focused on employing (statistical) relational learning techniques to solve computer vision and robotics problems.

Plinio Moreno is a visiting Professor at Electrical and Computers Engineering Department of Instituto Superior Ténico and senior Post-doc researcher at Computer and Robot Vision Laboratory, located in the Instituto de Sistemas e Robótica, one of the research centers at Instituto Superior Técnico, which is the Engineering School of Universidade de Lisboa. He completed a Ph.D. degree in Electrical and Computers Engineering by the Instituto Superior Técnico (Lisbon, Portugal) in 2008. He has more than 30 published papers in the areas of computer vision, pattern recognition and robotics.

Marion Neumann is a senior lecturer on machine learning and cloud computing at the CSE faculty, Washington University in St. Louis from July 2015. She received her Ph.D. from the Knowledge Discovery and Machine Learning research group at the University of Bonn (2015). Her main research interests can be summarized under machine learning withgraphs. This includes graph mining, Gaussian processes, random walks, information retrieval, statistical relational learning and the application of ML/SRL methods to solve problems in robotics, biology, medicine, and agriculture.



Rui Pimentel de Figueiredo is a Ph.D. Student at Technical University of Lisbon, Institute for Systems and Robotics. His research interest lies at the intersection of machine learning, action recognition and cognition, and robotics with the ultimate goal of designing human(oid) cognitive companions. The areas covered include: sensorimotor coordination, multisensory fusion, attention, human gesture recognition, uncertainty modelling, and learning from demonstration.

Kristian Kersting is an Associate Professor in the Computer Science Department at the Technical University of Dortmund, Germany. He received his Ph.D. from the University of Freiburg, Germany, in 2006 and moved to the Fraunhofer IAIS and the University of Bonn using a Fraunhofer ATTRACT Fellowship in 2008 after a PostDoc at MIT, USA. Before moving to the TU Dortmund University in 2013, he was appointed Assistant Professor for Spatio-Temporal Patterns in Agriculture at the University of Bonn in 2012. His main research interests are data mining, machine learning, and statistical relational artificial intelligence, with applications to medicine, plant phenotpying, traffic, and collective attention.

José Santos-Victor received the Ph.D. in Electrical and Computer Engineering from Instituto Superior Tcnico (IST), Lisbon, Portugal, in 1995. He is currently Full Professor at IST, Director of the Institute for Systems and Robotics (ISR-Lisboa) and head of the Computer and Robot Vision Laboratory (VisLab). He is the scientific responsible for IST in many European and national research projects in cognitive and bio-inspired computer vision and robotics. He published more than 300 articles in international journals and conferences (h-index 37).

Luc De Raedt received his Ph.D. in Computer Science in 1991 from KULeuven (Belgium). From 1986 till 1999 he held positions as teaching assistant (1986–1991), postdoctoral researcher of the Fund for Scientific Research, Flanders (1991–1999), part-time assistant professor (1993–1998) and part-time associate professor (1998–1999) at KULeuven, DTAI group. From 1999 till 2006 he was a Professor of Computer Science of the Albert-Ludwigs-University Freiburg and Chair of the Machine Learning and Natural Language Processing Lab research group. Since 2006, he has a research professorship at the Department of Computer Science of the KULeuven. His research interests are artificial intelligence, machine learning and data mining, as well as their applications. He is currently working on probabilistic logic learning and machine learning, the integration of constraint programming with data mining and machine learning principles, the development of programming languages for machine learning, and analyzing graph and network data. He is interested in applications of these methods to chemo- and bio-informatics, NLP, vision, robotics and action and activity learning. He has been a coordinator or a principle investigator in several European and national projects.