Language-Guided Adaptive Perception for Efficient Grounded Communication with Robotic Manipulators in Cluttered Environments

Siddharth Patki
University of Rochester
Rochester, NY, 14627, USA
spatki@ur.rochester.edu

Thomas M. Howard
University of Rochester
Rochester, NY, 14627, USA
thoward@ece.rochester.edu

Abstract

The utility of collaborative manipulators for shared tasks is highly dependent on the speed and accuracy of communication between the human and the robot. The run-time of recently developed probabilistic inference models for situated symbol grounding of natural language instructions depends on the complexity of the representation of the environment in which they reason. As we move towards more complex bi-directional interactions, tasks, and environments, we need intelligent perception models that can selectively infer precise pose, semantics, and affordances of the objects when inferring exhaustively detailed world models is inefficient and prohibits real-time interaction with these robots. In this paper we propose a model of language and perception for the problem of adapting the configuration of the robot perception pipeline for tasks where constructing exhaustively detailed models of the environment is inefficient and inconsequential for symbol grounding. We present experimental results from a synthetic corpus of natural language instructions for robot manipulation in example environments. The results demonstrate that by adapting perception we get significant gains in terms of run-time for perception and situated symbol grounding of the language instructions without a loss in the accuracy of the latter.

1 INTRODUCTION

Perception is a critical component of an intelligence architecture that converts raw sensor observations to a suitable representation for the task that the robot is to perform. Models of environments vary significantly depending on the application. For example, a robotic manipulator may need to model the objects in its environment with their six degree-of-freedom pose for grasping and dexterous manipulation tasks, whereas a self-driving car may need to model the dynamics of the environment in addition to domain-specific semantics such as stop signs, sidewalks and pedestrians etc. to safely navigate through the environment.

The ability of robots to perform complex tasks is linked to the richness of the robot's world model. As inferring exhaustively detailed world representations is impractical, it is common to infer representations which are highly specific to the task that the robot is to perform. However, in collaborative domains as we move towards more complex bi-directional interactions, manipulation tasks, and the environments, it becomes unclear how to best represent the environment in order to facilitate planning and reasoning for a wide distribution of tasks. As shown in the Figure 1, modeling the affordance between the chips can and its lid would be unnecessary for the task of picking up the mustard sauce bottle and vice versa. Inferring exhaustively detailed models of all of the objects in the environment is computationally expensive and inconsequential for the individual tasks, and inhibits real-time interaction with these collaborative robots.

The utility of collaborative manipulators is also highly dependent on the speed and accuracy of communication between the human operator and the robot. Natural language interfaces provide intuitive and muti-resolution means to interact with the robots in shared realms. In this work, we propose learning a model of language and perception that can adapt the configurations of the perception pipeline according to the task in order to infer representations that are necessary and suffi-

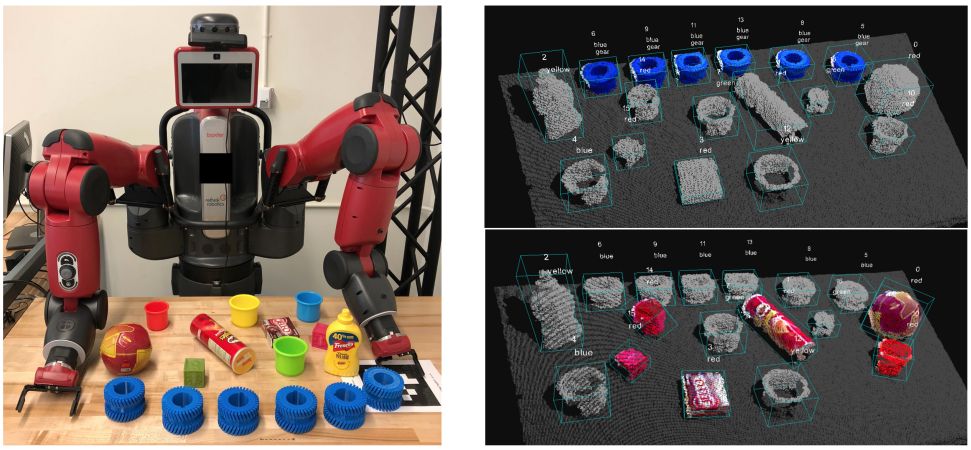

Figure 1: On the left is an image showing the Baxter Research Robot in a cluttered tabletop environment in the context of collaborative human-robot tasks. A perception system that does not use the context of the instruction when interpreting the observations would inefficiently construct detailed world model that is only partially utilized by the symbol grounding algorithm. On the right are the adaptively inferred representations using our proposed language perception model for the instructions, "pick up the leftmost blue gear" and "pick up the largest red object" respectively.

cient to facilitate planning and grounding for the intended task. e.g. the top-right image in the Figure 1 shows the adaptively inferred world model pertaining to the instruction "pick up the leftmost blue gear" which is different than the one inferred for the instruction "pick up the largest red object".

2 BACKGROUND

The algorithms and models presented in this paper span the topics that include robot perception and natural language understanding for human-robot interaction. Perception is a central problem in the the field of situated robotics. Consequently, a plenty of research has focused on developing representations that can faciliate planning and reasoning for highly specific situated tasks. These representations vary significantly depending on the application, from two-dimensional costmaps (Elfes, 1987), volumetric 3D voxel representations (Hornung et al., 2013, 2010), primitive shape based object approximations (Miller et al., 2003; Huebner and Kragic, 2008) to more rich representations that model high level semantic properties (Galindo et al., 2005; Pronobis and Jensfelt, 2012), 6 DOF pose of the objects of interest (Hudson et al., 2012) or affordances between objects (Daniele et al., 2017). Since inferring exhaustively detailed world models is impractical, one solution is to design perception pipelines that infer task relevant world models (Eppner et al., 2016; Fallon et al., 2014). Inferring efficient models that can support reasoning and planning for a wide distribution of tasks remains an open research question.

Natural language interfaces provides intutive and multi-resolution means to interact with the collaborative robots. Contemporary models (Tellex et al., 2011; Howard et al., 2014; Boularias et al., 2015; Matuszek et al., 2013) frame the problem of language understanding as a symbol grounding problem (Harnad, 1990). Specifically, of inferring correspondences between the linguistic constituents of the instruction and the symbols that represent perceived entities in the robot's environment such as objects and regions or desired actions that the robot can take. (Howard et al., 2014) frames this problem as one of inference in a probabilistic graphical model called a Distributed Correspondence Graph (DCG). This model leverages the hierarchical structure of the syntactically parsed instruction and conditional independence assumptions across constituents of a discrete symbol space to improve the run-time of probabilistic inference. Other variations include the Hierarchical DCG (Propp et al., 2015) and Adaptive DCG (Paul et al., 2016) to further improve the run-time performance in cluttered environments with known environment models. Recently, these models have been used to augment perception and representations. (Daniele et al., 2017) uses DCG for supplementing perception with linguistic information for efficiently inferring kinematic models of articulated objects. (Duvallet et al., 2014;

Hemachandra et al., 2015) use DCG to augment the representations by exploiting information in language instruction to build priors over the unknown parts of the world. A limitation of current applications of probabilistic graphical models for natural language symbol grounding is that they do not consider how to efficiently convert observations or measurements into sufficiently detailed representation suitable for inference. We propose to use DCG for the problem of adapting the perception pipelines for inferring task optimal representations.

Our work is most closely related to that of (Matuszek et al., 2013). Their work presents an approach for jointly learning the language and perception models for grounded attribute learning. Their model infers the subset of objects based on color and shape which satisfy the attributes described in the natural language description. Similarly, (Hu et al., 2016) proposes deep learning based approach to directly segment objects in RGB images that are described by the instruction. We differentiate our approach by expanding the diversity and complexity of perceptual classifiers, enabling verbs to modify object representations, and presenting an end-to-end approach to representation adaptation and symbol grounding using computationally efficient probabilistic graphical models. In the following sections we introduce our approach to adapting perception pipelines, define our experiments, and present results against a suitable baseline.

3 TECHNICAL APPROACH

We describe the problem of understanding natural language instructions as one of probabilistic inference where we infer a distribution of symbols that express the intent of the utterance. The meaning of the instruction is taken in the context of a symbolic representation (Γ), observations ($\mathbf{z}_t$) and a representation of the language used to describe the instruction (Λ). A probabilistic inference using a symbolic representation that is described by the space of trajectories $\mathbf{X}\,(t)$ that the robot may take takes the form of equation:

$$\mathbf{x}(t)^* = \underset{\mathbf{x}(t)\in\mathbf{X}(t)}{\arg\max}\ p(\mathbf{x}(t)|\Lambda, \mathbf{z}_t) \tag{1}$$

Solving this inference problem is computationally intractable when the space of possible trajectories is large. Contemporary approaches (Tellex et al., 2011; Howard et al., 2014) frame this problem as a symbol grounding problem, i.e. inferring the most likely set of groundings (Γ^{s*}) given a syntactically parsed instruction $\Lambda = \{\lambda_1, ..., \lambda_m\}$ and the world model Υ.

$$\Gamma^{s*} = \underset{\gamma_1 ... \gamma_n \in \Gamma^s}{\arg\max}\ p(\gamma_1 ... \gamma_n|\Lambda, \Upsilon) \tag{2}$$

Here, the world model Υ is a function of the constructs of the robot's perception pipeline (P), and the raw observations $\mathbf{z}_t$.

$$\Upsilon \approx f(P, \mathbf{z}_t) \tag{3}$$

The groundings Γ^s are symbols that represent objects, their semantic properties, regions derived from the world model, and robot actions and goals such as grasping the object of interest or navigating to a specific region in the environment. The set of all groundings $\Gamma^s = \{\gamma_1, \gamma_2, ..., \gamma_n\}$ is called as the *symbol space*. Thus the symbol space forms a finite space of interpretations in which the instruction will be grounded. The DCG is a probabilistic graphical model of the form described in equation 2. The model relates the linguistic components $\lambda_i \in \Lambda$ to the groundings $\gamma_j \in \Gamma^s$ through the binary correspondence variables $\phi_{ij} \in \Phi$. DCG facilitates inferring the groundings at a parent phrase in the context of the groundings at its child phrases Φ_{ci}. Formally, DCG searches for the most likely correspondence variables Φ^* in the context of the groundings γ_{ij} , phrases λ_i, child correspondences Φ_{ci} and the world model Υ by maximizing the product of individual factors.

$$\Phi^* = \underset{\phi_{ij}\in\mathbf{\Phi}}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma^s|} p(\phi_{ij}|\gamma_{ij}, \lambda_i, \Phi_{ci}, \Upsilon) \tag{4}$$

Inferred correspondence variables Φ^* represent the expression of the most likely groundings Γ^{s*}. The factors in the equation 4 are approximated by log-linear models Ψ:

$$\Phi^* = \underset{\phi_{ij}\in\mathbf{\Phi}}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma^s|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{ci}, \Upsilon) \tag{5}$$

Model training involves learning the log-linear factors from the labeled data relating phrases with true groundings. Inference process involves searching for the set of correspondence variables that satisfy the above equation. The run-time performance of probabilistic inference with the DCG

is positively correlated with the complexity of the world model Υ. This is because the size of the symbolic representation Γ^s increases with the number of objects in the environment representation. Recognizing that some objects (and the symbols based on those objects) are inconsequential to the meaning of the instruction, we consider the optimal representation of the environment Υ^* as one which is necessary and sufficient to solve equation 5. Thus we hypothesize that the time to solve equation 6 will be less than that for the equation 5.

$$\Phi^* = \underset{\phi_{ij} \in \mathbf{\Phi}}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma^s|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{ci}, \Upsilon^*) \tag{6}$$

Typically the environment model Υ is computed by a perception module P from a set of observations $\mathbf{z}_{1:t} = \{\mathbf{z}_1 \dots \mathbf{z}_t\}$. In cluttered environments we assume that inferring an exhaustively detailed representation of the world that satisfies all possible instructions is impractical for real-time human-robot interactions. We propose using language as mean to guide the generation of these necessary and sufficient environment representations Υ^* in turn making it a task adaptive process. Thus we define Υ^* inferred from a single observation as:

$$\Upsilon^* \approx f(P, \mathbf{z}_t, \Lambda) \tag{7}$$

where P denotes the perception pipeline of the robotic intelligence architecture. We adapt DCG to model the above function by creating a novel class of symbols called as perceptual symbols Γ^P. Perceptual symbols are tied to their corresponding elements in the perception pipeline. i.e. to the vision algorithms. Since this grounding space is independent of the world model Υ, the random variable used to represent the environment is removed from equation 5. We add a subscript p to denote that we are reasoning in the perceptual grounding space.

$$\Phi^* = \underset{\phi_{ij} \in \mathbf{\Phi}}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma^P|} \Psi(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{c_i}) \tag{8}$$

Equation 8 represents the proposed model which we refer to as the language-perception model (LPM). It infers the symbols that inform the perception pipeline configurations given a natural language instruction describing the task. The space of symbols Γ^P describe all possible configurations of the perception pipeline. For example, as shown in the Figure 1, for the instruction "pick up the leftmost blue gear", we may need elements in our pipeline that can detect blue objects and gears. Detecting green objects, spherical shapes, or six-dimensional pose of the chips can object would not be necessary to generate the symbols necessary for the robot to perform the instruction.

We assume that the perception pipeline (P) is populated with a set of elements $E = \{E_1, ..., E_n\}$ such that each subset $E_i \in E$ represents a set of algorithms that are responsible for inferring a specific property of an object. e.g. a red color-detection algorithm would be a member of the color detector family responsible for inferring the semantic property "color" of the object. While a six degree-of-freedom (DOF) pose detection algorithm would be a member of the pose detector family. More generally, E can be defined as: $E = \{e_1, e_2, ..., e_m\}$. With these assumptions, we define our independent perceptual symbols as:

$$\Gamma_P^{ID} = \{\gamma_{e_i} | e_i \in E\} \tag{9}$$

We can imagine that these symbols would be useful to ground simple phrases such as "the red object" or "the ball" etc. where the phrases refer to a single property of the object. In the more complicated phrases such as "the red ball" or "the blue box" we have a joint expression of properties. i.e. we are looking for objects which maximize the joint likelihood $p(red, sphere|o)$. Since these properties are independent we can infer them separately for every object $o_k \in O$. However, we can represent the above joint likelihood expression as $p(red, sphere) = p(red)p(sphere|red)$. In this case, it allows conditioning the evaluation of sphere detection on only a subset of objects which were classified as being red by the red detector. To add this degree of freedom in the construction of the perception pipeline, we define additional set of symbols which we refer to as conditionally dependent perceptual symbols:

$$\Gamma_P^{CD} = \{\gamma_{e_i,e_j} | e_i, e_j \in E \,;\, i \neq j\} \tag{10}$$

The expression of the symbol γ_{e_i,e_j} refers to running the element e_i from the perception pipeline on the subset of objects which were classified positive by the element e_j. Finally the complete perceptual symbol space is:

$$\Gamma^P = \{\Gamma_P^{ID} \cup \Gamma_P^{CD}\} \tag{11}$$

4 EXPERIMENTAL DESIGN

Herein with our experiments we demonstrate the utility of our language perception model for the task of grounded language understanding of the manipulation instructions. As shown in Figure 3 the process involves two distinct inferences: Inferring the perceptual groundings given a language instruction (eq. 8), and inferring high level motion planning constraints given the language and the generated world model (eq. 5 and eq. 6). In this section we describe our assumptions, and define the distinct symbolic representations used in our experiments for each of the above tasks. We then discuss our instruction corpus and the details of the individual experiments.

Robot and the Environment

For our experiments a Rethink Robotics Baxter Research Robot is placed behind a table. The robot is assumed to perceive the environment using a head-mounted RGB-D sensor. Robot's work space is populated using objects from the standard YCB dataset (Berk Calli, 2017), custom 3D printed ABS plastic objects, and multicolored rubber blocks. We define the world complexity in terms of the number of objects present on the table in the robot's field of view. The world complexity ranges from 15 to 20 in our experiments.

Symbolic Representation

The symbolic representation defines the space of symbols or *meanings* in which the natural language instruction will be grounded or *understood*. As mentioned before we define two distinct sets of symbols in our experiments. Γ^P defines the set of perceptual symbols which are used by the language perception model, and Γ^S defines the set of symbols which are used by the symbol grounding model.

Γ^P is a function of the elements E of the perception pipeline. The elements $e_i \in E$ in our perception pipeline are selected such that they can model the robot's environment with a spectrum of semantic and metric properties which will be necessary towards performing symbol grounding and planning for all of the instructions in our corpus. In our experiment we define E as:

$$E = \{C \cup G \cup L \cup B \cup R \cup \mathcal{P}\} \tag{12}$$

Here, C is a set of color detectors, G is a set of geometry detectors, L is a set of object label detectors, B is a set of bounding box detectors, R is a set of region detectors, and $\mathcal{P}$ is a set of pose detectors.

$$\begin{aligned} C &= \{\, cd_i \mid i \in color\} \\ G &= \{\, gd_i \mid i \in geometry\} \\ L &= \{\, ld_i \mid i \in label\} \\ B &= \{\, bd_i \mid i \in bbox\} \\ R &= \{\, rd_i \mid i \in region\} \\ \mathcal{P} &= \{\, pd_i \mid i \in pose\} \end{aligned} \tag{13}$$

where *color* = {red, green, blue, white, yellow, orange}, *geometry* = {sphere, cylinder, cuboid}, *label* = {crackers box, chips can, pudding box, master chef can, bleach cleanser, soccer ball, mustard sauce bottle, sugar packet}, *bbox* = {non-oriented, oriented }, *region* = {left, right, center}, *pose* = { 3 DOF, 6 DOF }. Given the perception elements defined in the equation 13, we define the independent perceptual groundings (Γ_P^{ID}) previously defined in equation 9 as follows:

$$\begin{aligned} \Gamma^C &= \{\gamma_{cd_i} \mid cd_i \in C\} \\ \Gamma^G &= \{\gamma_{gd_i} \mid gd_i \in B\} \\ \Gamma^L &= \{\gamma_{ld_i} \mid ld_i \in L\} \\ \Gamma^B &= \{\gamma_{bd_i} \mid bd_i \in B\} \\ \Gamma^R &= \{\gamma_{rd_i} \mid rd_i \in R\} \\ \Gamma^{\mathcal{P}} &= \{\gamma_{pd_i} \mid pd_i \in \mathcal{P}\} \end{aligned} \tag{14}$$

$$\Gamma_P^{ID} = \{\, \Gamma^C \cup \Gamma^G \cup \Gamma^L \cup \Gamma^B \cup \Gamma^R \cup \Gamma^{\mathcal{P}}\} \tag{15}$$

We define the conditionally dependent perceptual groundings (Γ_P^{CD}) previously defined in equation 10 as following:

$$\begin{aligned} \Gamma^{GC} &= \{\gamma_{(gd_i,cd_j)} \mid gd_i \in G, cd_j \in C\} \\ \Gamma^{LC} &= \{\gamma_{(ld_i,cd_j)} \mid ld_i \in L, cd_j \in C\} \\ \Gamma^{\mathcal{P}C} &= \{\gamma_{(pd_i,cd_j)} \mid pd_i \in \mathcal{P}, cd_j \in C\} \\ \Gamma^{\mathcal{P}G} &= \{\gamma_{(pd_i,gd_j)} \mid pd_i \in \mathcal{P}, gd_j \in G\} \\ \Gamma^{\mathcal{P}L} &= \{\gamma_{(pd_i,ld_j)} \mid pd_i \in \mathcal{P}, ld_j \in L\} \end{aligned} \tag{16}$$

$$\Gamma_P^{CD} = \{\, \Gamma^{GC} \cup \Gamma^{LC} \cup \Gamma^{\mathcal{P}C} \cup \Gamma^{\mathcal{P}G} \cup \Gamma^{\mathcal{P}L}\} \tag{17}$$

These symbols provide us the ability to selectively infer desired properties in the world. Above presented independent and conditionally dependent symbols together cover the complete space of perceptual symbols used by the LPM:

$$\Gamma^P = \{\Gamma_P^{ID} \cup \Gamma_P^{CD}\} \tag{18}$$

Algorithmic details of the percepion elements are as follows : A single RGB point cloud is fed in as a raw sensor observation to the pipeline. A RANSAC (Fischler and Bolles, 1981) based 3D plane detection technique is used for segmenting the table-top and the objects. HSV colorspace is used for detecting colors. RANSAC based model fitting algorithms form the core of the geometry detectors. A 4 layer (256 - 128 - 64 - 32) feed forward neural network is trained to infer the semantic labels of the objects. It takes in a 32 x 32 RGB image and infers a distribution over 8 unique YCB object classes. A PCA based oriented bounding box estimation algorithm is used to approximate the 6 DOF pose for the individual objects. Algorithms are implemented using OpenCV and PCL library (Rusu and Cousins, 2011).

The space of symbols for the symbol grounding model is similar to the representation defined in (Paul et al., 2016). This space uses symbols to represent objects in the world model ($\Gamma^{\mathcal{O}}$), semantic object labels ($\Gamma^{\mathcal{L}}$), object color($\Gamma^{\mathcal{C}}$), object geometry($\Gamma^{\mathcal{G}}$) regions in the world($\Gamma^{\mathcal{R}}$), spatial relationships ($\Gamma^{\mathcal{SR}}$) and finally high level planning constraints that define the end goal ($\Gamma^{\mathcal{PC}}$). The inferred constraints forms an input to a planning algorithm that can then generate trajectories to accomplish the desired task. Thus the complete symbol space for the symbol grounding model is:

$$\Gamma^{S} = \{\, \Gamma^{\mathcal{O}} \cup \Gamma^{\mathcal{L}} \cup, \Gamma^{\mathcal{C}} \cup \Gamma^{\mathcal{G}} \cup \Gamma^{\mathcal{R}} \cup \Gamma^{\mathcal{SR}} \cup \Gamma^{\mathcal{PC}} \} \quad (19)$$

Corpus

For training and testing the performance of the system we generate an instruction corpus using the linguistic patterns similar to that described in (Paul et al., 2016). The corpus used in our experiments consists of 100 unique natural language instructions. Details of the grammar extracted from this corpus is described in the appendix. Each instruction describes a manipulation command to the robot while referring to the objects of interest using their semantic or metric properties. e.g. “pick up the green cup” or “pick up the biggest blue object”. If multiple instances of the same objects are present in the robot’s work space then the reference resolution is achieved by using spatial relationships to describe the object of interest. e.g.“the leftmost blue cube” or “rightmost red object” etc.

As shown in Figure 2, the instructions in the corpus are in the form of syntactically parsed trees.

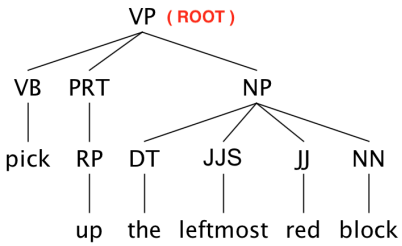

Figure 2: Syntactically parsed tree for the instruction ”pick up the leftmost red block”.

Each instruction is generated in the context of a specific table-top object arrangement. Thus each instruction is associated with a pair of RGB-D image. A total of 10 unique table-top arrangements are used to generate the set of 100 instructions.

One copy of the corpora is annotated for training LPM using (Γ^{P}) while another for training the symbol grounding model using (Γ^{S}). The annotations for LPM corpus are selected such that that the perception pipelines configured using the annotated groundings would generate the optimal world representations that are necessary and sufficient to support grounding and planning for the given tasks.

We have instructions with varying complexity in our corpus. The instruction complexity from the perception point of view is quantified in terms of the total number of perceptual groundings expressed at the root level. e.g. “pick up the ball” is relatively a simple instruction with only single grounding expressed at the root level, while “pick up the blue cube and put the blue cube near the crackers box” is a more complicated instruction having seven groundings expressed at the root level. This number was found to vary in the range of one to seven in our corpus.

Experiments and Metrics

We structure our experiments to validate two claims. The first claim is that adaptively inferring the task optimal representations reduce the perception run-time by avoiding exhaustively detailed uniform modeling of the world. The second claim is that reasoning in the context of these optimal representations also reduces the inference run-time of the symbol grounding model. An outline of our experiments is illustrated in Figure 3. In the first experiment, we study the root-level inference accuracy of LPM (groundings expressed at the root level of the phrase) as a function of the gradual increase in the training fraction. For each

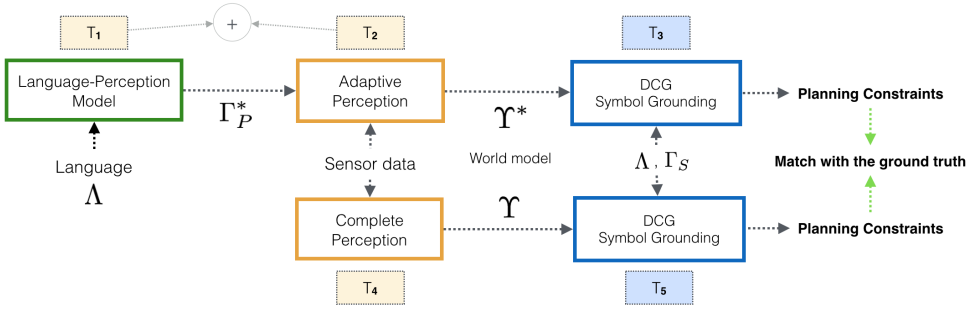

Figure 3: Comparative Experiments: Boxes in the bottom half denote the baseline framework whereas the boxes in the top half represent the proposed framework. Filled boxes enclose the variables that are compared in the experiments.

value of training fraction in the range [0.2 , 0.9] increasing with a step of 0.1, we perform 15 validation experiments. The training data is sampled randomly for every individual experiment. Additionally, we perform a leave-one-out cross validation experiment. We use the inferences generated by the leave-one-out cross validation experiments as inputs to drive the adaptive perception for each instruction.

In the second experiment, we compare the cumulative run-time of LPM inference (eq. 8) and adaptive perception (T_1+T_2) against the run-time for complete perception (T_4) - our baseline, for increasingly complex worlds.

In the third experiment, we compare the inference time of the symbol grounding model reasoning in the context of the adaptively generated optimal world models (T_3, eq. 6) against the inference time of the same model but when reasoning in the context of the complete world models (T_5, eq. 5). We also check whether the planning constraints inferred in both cases match the ground truth or not. Experiments are performed on a system running a 2.2 GHz Intel Core i7 CPU with 16 GB RAM.

5 RESULTS

This section presents the results obtained for the above mentioned three experiments. Specifically, the learning characteristics of LPM, the impact of LPM on the perception run-time, and the impact the adaptive representations on the symbol grounding run-time.

Leftmost graph in the Figure 4 shows the results of the first experiment. We can see that the inference accuracy grows as a function of a gradual increase in the training data. A growing trend is an indicator of the language diversity in the corpus. Mean inference accuracy starts at 39.25%±5 for k = 0.2 and it reaches 84% for leave-one-out cross validation experiment (k = 0.99).

Middle graph in the Figure 4 shows the result of the second experiment. We can clearly see that the run-time for complete perception grows with the world complexity while the run-time of adaptive perception stays nearly flat and is significantly lower in all cases. Since the adaptive perception run-time varies according to the task, we see bigger error bars. The drop in the complete perception run-time for world complexity of 20 is justifiable as the run-time of our geometry detection algorithm was proportional to the size of the individual objects, and all of the objects for that example world were smaller than other examples.

World Complexity	T_4 (sec) baseline	$T_1 + T_2$ (sec) proposed
15	4.40 ± 0.05	0.96 ± 0.07
16	4.99 ± 0.02	1.33 ± 0.34
17	5.40 ± 0.06	1.11 ± 0.11
18	5.82 ± 0.18	1.51 ± 0.26
20	4.17 ± 0.05	1.11 ± 0.25
Mean	$\mathbf{5.03 \pm 0.07}$	$\mathbf{1.20 \pm 0.21}$

Table 1: Adaptive perception run-time compared against complete perception run-time. Deviation measures are 95% confidence interval values.

Rightmost graph in the Figure 4 shows the result of the third experiment. It shows that the symbol grounding run-time when reasoning in the context of detailed world models(Υ) grows as a function of the world complexity. However, it is significantly lower when reasoning in the context of adaptively generated world models (Υ^*) and is independent of the world complexity.

The achieved run-time gains are meaningful

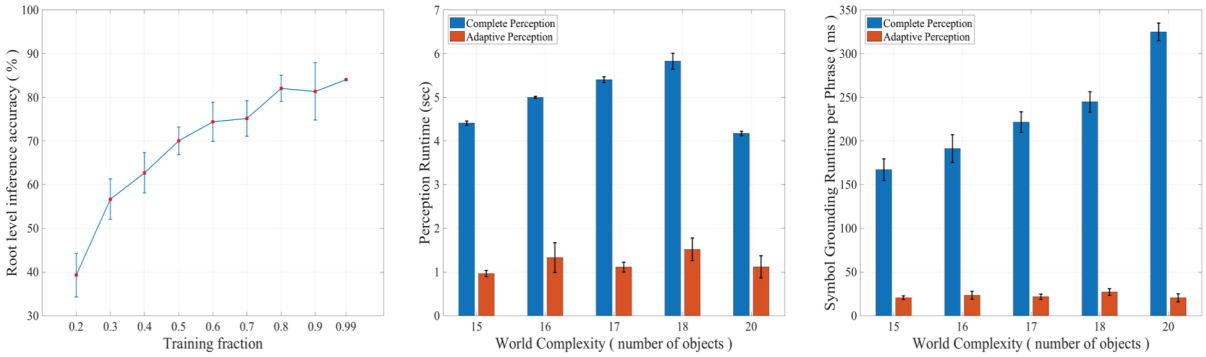

Figure 4: Graph on the left shows the LPM inference accuracy as a function of gradual increase in the training fraction. In the middle is the bar chart comparing the run-time for complete perception (T_4) against the cumulative run-time of LPM inference and adaptive perception ($T_1 + T_2$). Finally, on the right is a bar chart comparing the run-time of symbol grounding when reasoning in the context of the adaptively generated optimal representations (T_3) against when reasoning in the context of exhaustively detailed world models (T_5). The error bars indicate 95% confidence intervals.

World complexity	T_5 (ms) baseline	T_3 (ms) proposed
15	167 ± 12	21 ± 2
16	191 ± 16	23 ± 5
17	222 ± 12	22 ± 3
18	245 ± 12	27 ± 4
20	325 ± 10	20 ± 5
Mean	$\mathbf{214 \pm 12}$	$\mathbf{23 \pm 4}$

Table 2: Per phrase symbol grounding run-time in ms (rounded to the nearest integer) using adaptive representations compared against the same when using complete representations. Deviation measures are 95% confidence interval values.

only if we do not incur a loss in the symbol grounding accuracy. Table 3 shows the impact of LPM on SG accuracy and summarizes the gains.

Perception Type	Avg. T_P (sec)	Avg. T_{SG} (ms)	SG Acc.
Complete	5.03 ± 0.07	214 ± 12	**63%**
Adaptive	1.20 ± 0.21	23 ± 4	**66%**
Ratio	**4.19**	**9.30**	

Table 3: Impact of LPM on average perception run-time per instruction (T_P), average symbol grounding run-time per instruction (T_{SG}), and the symbol grounding accuracy.

6 CONCLUSIONS

Real-time human-robot interaction is critical for the utility of the collaborative robotic manipulators in shared tasks. In scenarios where inferring exhaustively detailed models of all the objects is prohibitive, perception represents a bottleneck that inhibits real-time interactions with collaborative robots. Language provides an intuitive and multi-resolution interface to interact with these robots. While recent probabilistic frameworks have advanced our ability to interpret the meaning of complex instructions in cluttered environments, the problem of how language can channel the interpretation of the raw observations to construct world models which are necessary and sufficient for the symbol grounding task is not extensively studied. Our proposed DCG based Language Perception Model, demonstrates that we can guide perception using language to construct world models which are suitable for efficiently interpreting the instruction. This provides run-time gains in terms of both perception and symbol grounding, thereby improving the speed with which collaborative robots can understand and act upon human instructions. In ongoing and future work we are exploring how language can aid efficient construction of global maps for robot navigation and manipulation by intelligently sampling relevant observations from a set of observations.

7 ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant IIS-1637813 and the New York State Center of Excellence in Data Science at the University of Rochester.

References

James Bruce Aaron Walsman Kurt Konolige Siddhartha Srinivasa Pieter Abbeel Aaron M Dollar Berk Calli, Arjun Singh. 2017. Yale-cmu-berkeley dataset for robotic manipulation research. volume 36, page 261 268.

Abdeslam Boularias, Felix Duvallet, Jean Hyaejin Oh, and Anthony (Tony) Stentz. 2015. Grounding spatial relations for outdoor robot navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*.

Andrea F. Daniele, Thomas M. Howard, and Matthew R. Walter. 2017. A multiview approach to learning articulated motion models. In *Proceedings of the International Symposium of Robotics Research (ISRR)*.

F. Duvallet, M.R. Walter, T.M. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz. 2014. A probabilistic framework for inferring maps and behaviors from natural language. In *Proceedings of the 14th International Symposium on Experimental Robotics*.

A Elfes. 1987. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3).

Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Roberto Martín-Martín, Arne Sieverling, Vincent Wall, and Oliver Brock. 2016. Lessons from the amazon picking challenge: Four aspects of building robotic systems. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan.

Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, Toby Schneider, Hongkai Dai, Claudia Prez D'Arpino, Robin Deits, Matt DiCicco, Dehann Fourie, Twan Koolen, Pat Marion, Michael Posa, Andrs Valenzuela, KuanTing Yu, Julie Shah, Karl Iagnemma, Russ Tedrake, and Seth Teller. 2014. An architecture for online affordancebased perception and wholebody planning. *Journal of Field Robotics*, 32(2):229–254.

Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez. 2005. Multi-hierarchical semantic maps for mobile robotics. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2278–2283.

Stevan Harnad. 1990. The symbol grounding problem. In *Physica D: Nonlinear Phenomena*, volume 42, pages 335–346.

S. Hemachandra, F. Duvallet, T.M. Howard, N. Roy, A. Stentz, and M.R. Walter. 2015. Learning models for following natural language directions in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE.

A. Hornung, K. M. Wurm, and M. Bennewitz. 2010. Humanoid robot localization in complex indoor environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1690–1695.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. 2013. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.

T.M Howard, S. Tellex, and N. Roy. 2014. A natural language planner interface for mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6652–6659. IEEE.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. 2016. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer.

N. Hudson, T.M. Howard, J. Ma, A. Jain, M. Bajracharya, S. Myint, L. Matthies, P. Backes, P Hebert, T. Fuchs, and J. Burdick. 2012. End-to-end dexterous manipulation with deliberative interactive estimation. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*.

K. Huebner and D. Kragic. 2008. Selection of robot pre-grasps using box-based shape approximation. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1765–1770.

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2013. *Learning to Parse Natural Language Commands to a Robot Control System*. Springer International Publishing, Heidelberg.

Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. 2003. Automatic grasp planning using shape primitives. In *ICRA*.

R. Paul, J. Arkin, N. Roy, and T.M. Howard. 2016. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Proceedings of the 2016 Robotics: Science and Systems Conference*.

A. Pronobis and P. Jensfelt. 2012. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *2012 IEEE International Conference on Robotics and Automation*, pages 3515–3522.

O. Propp, I. Chung, M.R. Walter, and T.M. Howard. 2015. On the performance of hierarchical distributed correspondence graphs for efficiency symbol grounding of robot instructions. In *Proceedings*

of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Radu Bogdan Rusu and Steve Cousins. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation.

A Grammar and Lexicon of the Corpus

We list the grammar rules and the lexicon for our corpus to demonstrate the diversity of the instructions. Following table lists the words scraped from the instructions in our corpus. We have a total of 56 unique words.

$$
\begin{aligned}
\text{VB} &\rightarrow \{\text{pick}|\text{put}\} \\
\text{RP} &\rightarrow \{\text{up}\} \\
\text{DT} &\rightarrow \{\text{the}|\text{all}\} \\
\text{CC} &\rightarrow \{\text{and}\} \\
\text{VBZ} &\rightarrow \{\text{is}\} \\
\text{WDT} &\rightarrow \{\text{that}\} \\
\text{VB} &\rightarrow \{\text{near}|\text{in}|\text{on}\} \\
\text{PRP} &\rightarrow \{\text{your}\} \\
\text{NN} &\rightarrow \left\{\begin{array}{l} \text{cup}|\text{pudding}|\text{box}|\text{cube}| \\ \text{object}|\text{ball}|\text{master}| \\ \text{chef}|\text{can}|\text{soccer}| \\ \text{gear}|\text{mustard}|\text{sauce}|\text{bottle}| \\ \text{sugar}|\text{packet}|\text{block}| \\ \text{cleanser}|\text{middle}|\text{left}| \\ \text{right}|\text{crackers}|\text{cheezit}| \\ \text{cleanser}|\text{packet}|\text{block} \end{array}\right\} \\
\text{NNS} &\rightarrow \left\{\begin{array}{l} \text{cups}|\text{chips}|\text{cubes}| \\ \text{objects}|\text{balls} \end{array}\right\} \\
\text{JJ} &\rightarrow \left\{\begin{array}{l} \text{blue}|\text{green}|\text{yellow}| \\ \text{red}|\text{white} \end{array}\right\} \\
\text{JJS} &\rightarrow \left\{\begin{array}{l} \text{nearest}|\text{rightmost}|\text{leftmost}| \\ \text{farthest}|\text{biggest}|\text{smallest}| \\ \text{largest}|\text{closest} \end{array}\right\}
\end{aligned}
$$

Table 4: The words scraped from the corpus of annotated examples

Following table lists the grammar rules scraped from the instructions in our corpus. We have a total of 23 unique grammar rules.

$$
\begin{aligned}
\text{SBAR} &\rightarrow \text{WHNP S} \\
\text{S} &\rightarrow \text{VP} \\
\text{VP} &\rightarrow \text{VB PRT NP} \\
\text{VP} &\rightarrow \text{CC VP VP} \\
\text{VP} &\rightarrow \text{VB NP PP} \\
\text{VP} &\rightarrow \text{VBZ PP} \\
\text{WHNP} &\rightarrow \text{WDT} \\
\text{PRT} &\rightarrow \textit{RP} \\
\text{PP} &\rightarrow \text{IN NP} \\
\text{NP} &\rightarrow \text{DT JJ NN} \\
\text{NP} &\rightarrow \text{DT NN NN} \\
\text{NP} &\rightarrow \text{DT JJS JJ NN} \\
\text{NP} &\rightarrow \text{NP PP} \\
\text{NP} &\rightarrow \text{DT} \\
\text{NP} &\rightarrow \text{DT JJ NNS} \\
\text{NP} &\rightarrow \text{DT NN} \\
\text{NP} &\rightarrow \text{DT NN NN NN} \\
\text{NP} &\rightarrow \text{DT JJ NN NN} \\
\text{NP} &\rightarrow \text{DT JJS NN} \\
\text{NP} &\rightarrow \text{DT NNS NN} \\
\text{NP} &\rightarrow \text{PRP NN} \\
\text{NP} &\rightarrow \text{NP SBAR} \\
\text{NP} &\rightarrow \text{DT NNS}
\end{aligned}
$$

Table 5: The grammar rules scraped from the corpus of annotated examples