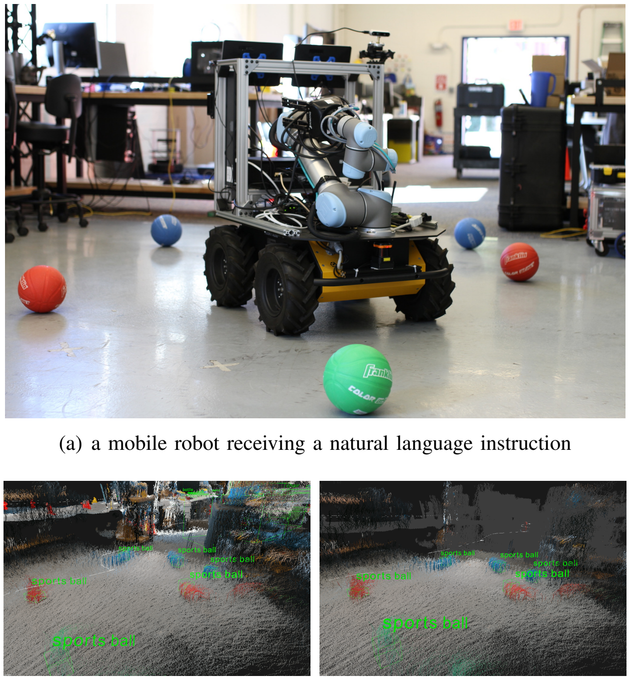
Inferring Compact Representations for Efficient Natural Language Understanding of Robot Instructions

Siddharth Patki Andrea F. Daniele Matthew R. Walter Thomas M. Howard

***Abstract*— The speed and accuracy with which robots are able to interpret natural language is fundamental to realizing effective human-robot interaction. A great deal of attention has been paid to developing models and approximate inference algorithms that improve the efficiency of language understanding. However, existing methods still attempt to reason over a representation of the environment that is flat and unnecessarily detailed, which limits scalability. An open problem is then to develop methods capable of producing the most compact environment model sufficient for accurate and efficient natural language understanding. We propose a model that leverages environment-related information encoded within instructions to identify the subset of observations and perceptual classifiers necessary to perceive a succinct, instruction-specific environment representation. The framework uses three probabilistic graphical models trained from a corpus of annotated instructions to infer salient scene semantics, perceptual classifiers, and grounded symbols. Experimental results on two robots operating in different environments demonstrate that by exploiting the content and the structure of the instructions, our method learns compact environment representations that significantly improve the efficiency of natural language symbol grounding.**

I. Introduction

The ability for robots to perform complex tasks is inherently linked to the richness of their environment models. Advances in sensor technology, machine perception, and natural language understanding provide a wealth of data that can be infused into these models. These innovations raise new questions with regards to how to assimilate, manage, and utilize this abundance of knowledge. A fundamental problem is how to reason over this rich information in a manner that enables robots to efficiently plan in diverse environments of varying scales and complexities. Consider the human-robot teaming scenario illustrated in Figure 1, in which a user instructs the mobile robot to "navigate to the nearest red ball." If we assume that the robot has access to knowledge bases (e.g., campus-level maps) and various sensor measurements (e.g., images, laser scans, audio, etc.) that it has accumulated over time, the problem becomes one of situating or "grounding" the instruction in the context of the perceived environment. With a few exceptions [1–5], contemporary methods attempt to fuse the knowledge bases and sensor measurements into a single, flat representation of the environment (i.e., the "world model") that expresses all metric [6–12] as well as semantic [4, 5, 13–16] knowledge gleaned from the observations. There are three fundamental limitations to this approach.



(a) a mobile robot receiving a natural language instruction

(b) detailed world model (c) compact world model

Fig. 1. Our framework learns to build a minimal representation of the environment sufficient to interpret a given natural language instruction. In this example, (a) a mobile robot is directed to "navigate to the nearest ball in the lab." Traditional methods interpret the instruction in the context of (b) an exhaustive world model, whereas our method maintains (c) a compact world model sufficient to ground the provided instruction.

First, a consistent, high fidelity model of the environment is expensive to maintain in terms of both compute and memory storage. Second, searching over dense models is computationally prohibitive in the context of both planning and natural language understanding [17–19], with costs as high as exponential in the size of the model [17]. More generally, it is unnecessarily detailed for most tasks. Ideally, one would reason over the most compact representation of the environment necessary to understand the instruction. However, this representation can not be inferred until after the instruction is received. Third, in situations in which concepts are taught or evolve in-situ from human demonstrations, previous interpretations of the environment may become incorrect or deficient, necessitating a means of revisiting these models as needed.

We propose a framework that explicitly reasons over

Siddharth Patki and Thomas M. Howard are with the University of Rochester, Rochester, NY USA, spatki@ur.rochester.edu, thomas.howard@rochester.edu

Andrea F. Daniele and Matthew R. Walter are with the Toyota Technological Institute at Chicago, Chicago, IL USA, {afdaniele,mwalter}@ttic.edu

the relevance of the observations and perceptual classifiers available, so as to learn a task-relevant, scalable environment representation sufficient for planning and natural language understanding. Underlying this method is a learned probabilistic model that can be readily adapted based upon the difficulty of the task and the complexity of the environment. Importantly, the method infers an efficient environment representation online by leveraging a learned model of saliency. This model extracts characteristics of the representation from free-form utterances to "lazily" reason over the small subset of available knowledge pertinent to the task. Specifically, we build upon recent work on adapting perception pipelines from natural language instructions [20] to infer subsets of observations that we use to construct instruction-specific representations of the environment. These induced representations are more efficient to search, yet still express the correct hierarchies and affordances necessary to perform the task. In scenarios where humans can interactively teach robots to classify objects in-situ, past observations of such objects could be added to the world model given utterances that reference the object.

The central contribution of this paper is a framework that exploits three probabilistic graphical models in the form of Distributed Correspondence Graphs [18] to adaptively model the environment representation in a task-specific manner. These models are trained from examples of how language maps to the relevant scene semantics, perceptual classifiers, and the symbols used to ground language-based instructions. Experimental results demonstrate that the ability to dynamically adapt perception and observation models significantly improves the computational efficiency of natural language symbol grounding.

II. Related Work

Existing language understanding methods reason over a flat, unified symbolic model of the world that expresses the spatial, semantic, and/or topologic properties of the environment through a representation that is assumed to be globally consistent. In practice, these models are typically constructed by running a state-of-the-art SLAM algorithm [6, 7, 10, 11, 21], which provides flat, globally metric models of the environment that are limited to spatial information. Semantic and topologic properties are then manually injected to realize a representation suitable for language grounding. Localization and mapping methods that attempt to jointly reason over spatial, semantic, and topologic properties of the environment have also been proposed [4, 5, 14–16, 22–24]. With few exceptions [22], however, these methods still attempt to maintain a single globally consistent environment representation, which is both unnecessarily detailed for language grounding and also resource (e.g., memory) intensive.

Given a natural language utterance, grounding methods [18, 25, 26] attempt to associate each word in the utterance with its corresponding referent in this environment model and the robot's symbolic action space. Semantic parsing-based methods [27–29] similarly map natural language to meaning representations, typically in the form of a lambda calculus. Early work in grounding [30, 31] employs manually engineered correspondences and features between words in a flat representation of the environment. Modern day methods [17–19, 26, 32] take a statistical approach to language grounding (and similarly for inverse grounding [33–35]) that employs probabilistic models that relate words to their corresponding referents according to the hierarchical structure of language, enabling the resolution of complex free-form language. These models are typically learned from annotated natural language corpora as well as through interaction with humans [29, 36, 37]. Probabilistic grounding models have been shown to be effective at interpreting cooking instructions [38], learning spatial relations in semantic maps [5, 15], and directing mobile manipulators [39], among others.

These methods perform inference over the entire set of state and action symbols, resulting in a computational complexity that is proportional to the power set of objects, regions, and constraints. This limits inference to simple tasks with a few interchangeable constraints or requires access to a set of predefined environment-specific behaviors. To improve scalability, Howard et al. [18] developed the Distributed Correspondence Graph (DCG) model that separates inference across conditionally independent constituents of the graph. In effect, this distributes inference across multiple factors in a graphical model, transforming the computational complexity from exponential to linear in the number of symbols. Chung et al. [19] propose the Hierarchical Distributed Correspondence Graph (HDCG), which improves the efficiency of inference by learning to construct a more efficient approximation of the space of relevant symbols for probabilistic language grounding. Paul et al. [40] describe a method that partitions the joint distribution into concrete and abstract factors. The algortihm performs inference in two stages per phrase. In the first stage, distributions of concrete symbols are inferred and used to inform sparse approximations of the abstract symbolic representation that are more efficient to search. In the second stage, distributions of abstract symbols are inferred and joined with the concrete symbols to represent the meaning of each phrase.

III. Technical Approach

The problem of natural language understanding is commonly framed as inference over a learned distribution that associates linguistic elements with their corresponding symbolic representation of the robot's state and action spaces. More specifically, inference involves reasoning over a representation Γ_s that symbolizes objects, places, constraints, actions, trajectories, and others concepts expressed by the robot's world model. The set of symbols forms a discrete and finite space in which the instruction can be grounded. The distribution over groundings is conditioned over a parse of the utterance $\mathbf{\Lambda}$ as well as a world model $\mathbf{\Upsilon}_t$ expressing environment knowledge that may be known a priori $\mathbf{\Upsilon}_0$ or extracted from multimodal observations $\mathbf{z}_{1:t}$ using the classifiers in the robot's perception pipeline $\mathbf{P}$

$$\mathbf{\Upsilon}_t \approx f(\mathbf{z}_{1:t}, \mathbf{P}, \mathbf{\Upsilon}_0). \tag{1}$$

Natural language understanding then follows as maximum a posteriori (MAP) inference over Γ_s

$$\Gamma_s^* = \underset{\gamma_1 \ldots \gamma_n \in \boldsymbol{\Gamma_s}}{\arg\max} \; p\left(\Gamma_s | \boldsymbol{\Lambda}, \boldsymbol{\Upsilon}_t\right). \tag{2}$$

Several contemporary approaches [17, 18, 40] formulate this problem as probabilistic inference in a factor graph with a hierarchical structure dictated by the compositional nature of the utterance, symbolic representation, and environment. This enables the model to reason about the meaning of particular phrases in terms of the symbolic grounding space based upon their child phrases, and a model of the environment. The parameters of the grounding model (e.g., feature weights in a log-linear model) are learned from annotated corpora that express the meaning of each phrase in the context of the child groundings and phrases.

In practical settings, the the space of groundings Γ_s, the environment $\boldsymbol{\Upsilon}_t$ is complex, and the free-form instructions Λ may be complex and diverse, making exact inference computationally intractable. To address this, the Distributed Correspondence Graph [18] proposes an approximate factorization of the grounding distribution that affords an efficient inference

$$\Phi_s^* = \underset{\phi_{ij} \in \Phi_s}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma_s|} p(\phi_{ij} | \gamma_{ij}, \lambda_i, \Phi_{ci}, \boldsymbol{\Upsilon}_t). \tag{3}$$

Formally, DCG inference involves searching for the most likely assignment of boolean correspondence variables Φ_s^* [41] in the context of the groundings $\gamma_{ij} \in \Gamma_s$, phrases $\lambda_i \in \Lambda$, child correspondences Φ_{ci}, and the world model $\boldsymbol{\Upsilon}_t$ by maximizing the factorization in Equation 3. In such model, a correspondence variable ϕ_{ij} being true expresses the fact that the corresponding grounding γ_{ij} matches the associated phrase in the command.

The ability to ground free-form instructions is inherently linked to the richness of the robot's environment representation $\boldsymbol{\Upsilon}_t$. However, building exhaustively detailed world models using all available knowledge bases and observations $\mathbf{z}_{1:t}$ is computationally expensive, particularly in large-scale, unstructured environments. The runtime of common language understanding models such as G^3 are exponential in the cardinality of the symbol space $|\Gamma_s|$ [18]. DCG improves this complexity to being linear in the size of the world model, however the cost of inference still inhibits real-time human-robot interaction.

In practice, a large fraction of the objects and their corresponding symbols that comprise the inferred world model are typically inconsequential to the meaning of the utterance. In such cases, there exists a compact environment representation $\boldsymbol{\Upsilon}_t^*$ that is sufficient to interpret the utterance, providing a significant improvement in the computational efficiency of inference relative to the standard model (Equation 3).

We propose a probabilistic model that exploits natural language in order to guide the generation of these compact world models $\boldsymbol{\Upsilon}_t^*$. Integral to this approach is the ability to infer a small, succinct subset of perceptual classifiers $\mathbf{P}^* \in \mathbf{P}$ in a manner that dynamically adapts the robot's perceptual capabilities according to the current task

$$\mathbf{P}^* \approx f\left(\mathbf{P}, \boldsymbol{\Lambda}\right), \tag{4}$$

resulting in the compact world model

$$\boldsymbol{\Upsilon_t^*} \approx f\left(\mathbf{z}_{1:t}, \mathbf{P}^*, \boldsymbol{\Upsilon}_0\right) \tag{5}$$

We further observe that not all observations are necessary to produce this compact representation $\boldsymbol{\Upsilon}_t^*$. For instructions in which the context of the observation may be evident (e.g., "drive to the nearest red ball in the hallway"), samples outside of these semantically classified regions (i.e., hallways) can be pruned from the space of observations. As the robot drives through the environment, a real-time scene classifier produces a semantic label (i.e., a scene category) that will be associated with all of the observations (from all available sensors) and pose measurements. The ability to assign a label to the current region in real-time allows us to treat such information as an observation produced by a virtual sensor (i.e., the scene classifier).

We define a minimal set of observations $\mathbf{z}^* \in \mathbf{z}_{1:t}$ that, based on their semantic labels, are used to construct the compact representation that is sufficiently detailed to contain all symbols necessary to be expressed by the natural language symbol grounding model

$$\mathbf{z}^* \approx f\left(\mathbf{z}_{1:t}, \boldsymbol{\Lambda}\right) \tag{6a}$$

$$\boldsymbol{\Upsilon_t^*} \approx f\left(\mathbf{z}^*, \mathbf{P}^*, \boldsymbol{\Upsilon}_0\right). \tag{6b}$$

Using the subsampled set of observations to construct a compact representation for symbol grounding transforms the expression for natural language inference (Eqn. 3) to

$$\Phi_s^* = \underset{\phi_{ij} \in \Phi_s}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma_s|} p(\phi_{ij} | \gamma_{ij}, \lambda_i, \Phi_{ci}, \boldsymbol{\Upsilon_t^*}). \tag{7}$$

This inference problem requires that we learn three models (Fig. 2): an adaptive perception model, an observation filtering model, and a natural language symbol grounding model. The process for training these models begins with the natural language symbol grounding module, in which symbols that represent objects, spatial relationships, containers, constraints, actions, and other types are associated with language [18, 40]. The process of training the observation filtering and adaptive perception models requires one to fit the minimum set of semantic labels and perceptual classifiers. Such classifiers are the ones that extract the most compact environment representation for each example that will not prune out any of the annotated ground-truth symbols from the corpus of instructions. This process yields three separate corpora with common instructions, but different symbolic representations and annotations that we use to train the three distinct models.

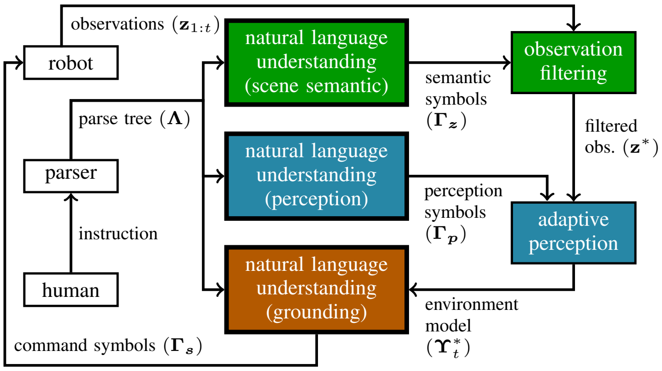

observations ($\mathbf{z}_{1:t}$)
robot
natural language understanding (scene semantic)
semantic symbols ($\mathbf{\Gamma_z}$)
observation filtering
filtered obs. ($\mathbf{z}^*$)
parse tree ($\mathbf{\Lambda}$)
parser
natural language understanding (perception)
perception symbols ($\mathbf{\Gamma_p}$)
adaptive perception
instruction
human
natural language understanding (grounding)
environment model ($\mathbf{\Upsilon}_t^*$)
command symbols ($\mathbf{\Gamma_s}$)

Fig. 2. The system architecture for language-guided observation filtering, adaptive perception, and natural language symbol grounding. The three natural language understanding models that are learned from the annotated instructions are highlighted in bold.

IV. Experimental Setup

Figure 2 illustrates the software architecture that we implemented for experimental evaluation of the proposed algorithm. In this architecture, the robot stores the sensors measurements in the observation filtering module. When the human provides a textual instruction, we convert the text into a parse tree $\mathbf{\Lambda}$ that is provided to the three natural language understanding modules. The *scene semantics* natural language understanding module extracts the salient scene semantics $\mathbf{\Gamma_z}$ pertaining to the instruction. The observations filtering module then extracts a subset of observations $\mathbf{z}^*$ (Eqn. 6a) based on the inferred scene semantic label(s). The *perception* natural language understanding module extracts the symbols representing the classifiers (Eqn. 4) that are necessary to detect the objects that are relevant to the natural language instruction. This information is then passed to the adaptive perception node that extracts an approximation of the environment model $\mathbf{\Upsilon}_t^*$ (Eqn. 6b) from $\mathbf{z}^*$ using the sub-sampled classifiers $\mathbf{P}^*$. The *symbol grounding* natural language understanding module uses the parse tree and the world model approximation to extract a distribution of symbols that represents the robot behavior $\mathbf{\Gamma_s}$ (Eqn. 7).

All of the natural language understanding modules are implemented as Distributed Correspondence Graphs [18] with symbolic representations and features adapted for each of the scene semantics, perception, and grounding domains.

We trained the natural language understanding modules with a synthetic corpus of annotated examples consistent with example robot instructions, such as "navigate to the nearest cone in the parking lot" or "navigate to the farthest blue ball." Approximately 500 instructions were annotated for the scene semantic, perception, and grounding models in accordance with their symbolic representation. The software was integrated onto two Clearpath Robotics Husky A200 Unmanned Ground Vehicles (Fig. 1) and used for dataset collection at two distinct sites. Visual observations were collected using the RealSense D435 RGB-D sensor. Robot localization was performed using laser-scan matching with a planar LIDAR sensor.

In these experiments, we use eight semantic labels such as "kitchen," "laboratory," "parking lot," etc., which are associated with sensor observations. To detect the semantics of the scene, we use a YOLO object detector [42] trained on the COCO dataset [43]. Object detections are passed to a scene classifier. The scene classifier then assigns a semantic label to each observation based on an object co-occurrence model that relates objects and scene classes. Objects that are not characteristic of any particular scene (e.g., person, cat, or horse) are ignored. The perception pipeline within in the adaptive perception node contains multiple elements including a YOLO-based object detector, a noise removal filter that refines the segmented object clusters, a 3d bounding box detector, an LUV color space-based color detector, and a 3-DOF pose detector. We limit the sensing range to 3.5 m to avoid processing noisy point cloud data.

The experiments were designed to explore the impact of observation filtering and adaptive perception on the task of mobile robot instruction following. We quantify the performance of the system using metrics of computational efficiency of perception for symbol grounding under the assumption of lazy evaluation of the observations.

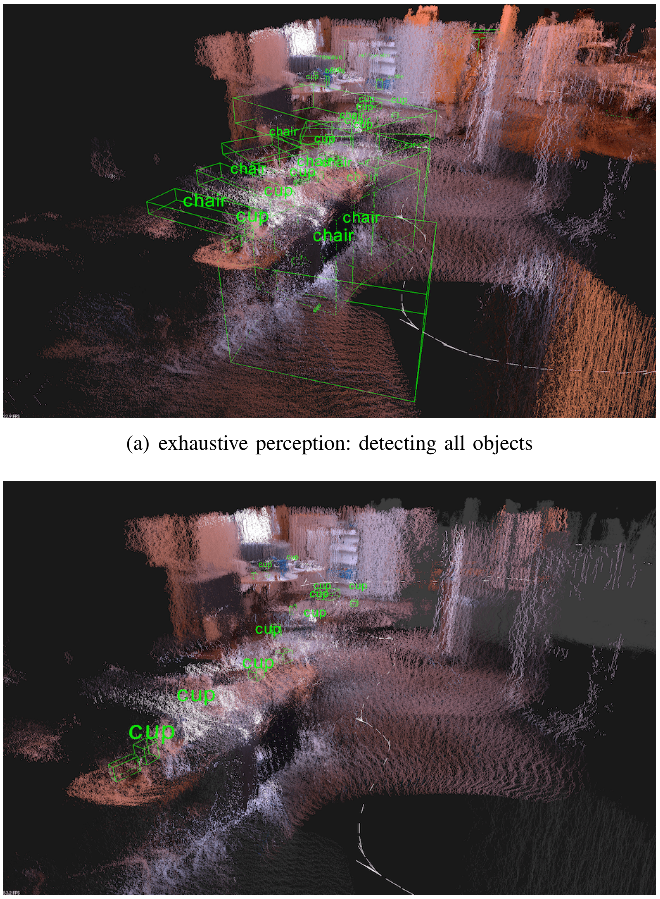
V. Results

This section presents results highlighting the performance of different aspects of the learned models in our proposed architecture. First, we highlight the computational efficiency of adaptive perception applied in the navigation domain. Second, we demonstrate how observation filtering reduces the number of observations we need to reason over in order to extract task-relevant objects. Later, we demonstrate the efficiency gains achieved by combining these two strategies in order to generate compact world representations.

A. Adaptive Perception

In previous experiments [20], we observed that language grounding was faster in environments inferred by adaptive perception than non-adaptive perception. Also the adaptive perception was found to be faster than its counterpart. To verify the predicted behavior of the adaptive perception pipeline, we analyzed its impact on the runtime of perception by evaluating it on the datasets collected at two diffrent sites for six different instructions. Table I presents the results demonstrating the impact of adaptive perception (AP) on the perception runtime against the standard baseline (B) that corresponds to the standard approach of invoking all classifiers and observations. Table II shows the impact of adaptive perception on the compactness of the approximated world representations. Consistent with previous evaluations [20], reducing the cardinality of the world model improves the runtime of language grounding.

Figure 3 demonstrates the impact of adaptive perception for the example instruction "drive to the nearest cup in the kitchen." In this particular example, the model is able to independently evaluate which object detectors should be engaged to construct an instruction-specific world model. By using the information contained within the instructions, our method



(a) exhaustive perception: detecting all objects

(b) adaptive perception: detecting only cups

Fig. 3. Impact of adaptive perception for the command "drive to the farthest cup in the kitchen." A standard approach requires generating and reasoning over (a) an exhaustive map generated using all of the available object detectors, resulting in a map with 37 objects and a runtime of 408 s. In contrast, our adaptive method generates (b) a more compact map only using detectors relevant to the command, resulting in a map with 11 objects and a runtime of 225 s. .

TABLE I

IMPROVEMENT IN THE PERCEPTION RUNTIME AT SITES 1 & 2

		(runtime in seconds)			
Instruction	Site	B	OF	AP	OF+AP
"go to the farthest umbrella in the hallway"	1	401	60	242	55
"go to the nearest suitcase in the parking lot"	2	306	136	220	99
"go to the farthest cup in the kitchen"	1	401	146	225	75
"go to the nearest keyboard in the office"	2	306	74	222	46
"go to the nearest ball in the hallway"	1	401	59	217	38
"go to the farthest ball in the lab"	2	306	67	206	48

TABLE II

IMPROVEMENT IN THE REPRESENTATION COMPACTNESS AT SITES 1 & 2

		(# of detected objects)			
Instruction	Site	B	OF	AP	OF+AP
"go to the farthest umbrella in the hallway"	1	37	4	2	2
"go to the nearest suitcase in the parking lot"	2	36	3	3	2
"go to the farthest cup in the kitchen"	1	37	29	11	9
"go to the nearest keyboard in the office"	2	36	29	3	3
"go to the nearest ball in the hallway"	1	37	4	1	1
"go to the farthest ball in the lab"	2	36	7	7	7

results in a 36% reduction in the time required to build an environment representation for inferring the instruction "go to the nearest cup in the kitchen." This demonstrates how inferring the classifiers useful for generating task-relevant compact representations can reduce the runtime requirements of robot perception. As we have seen [20], the reduction in runtime is proportional to the sparsity of classifiers necessary to extract a sufficient detailed environment model that is suitable for the grounding of specific instructions.

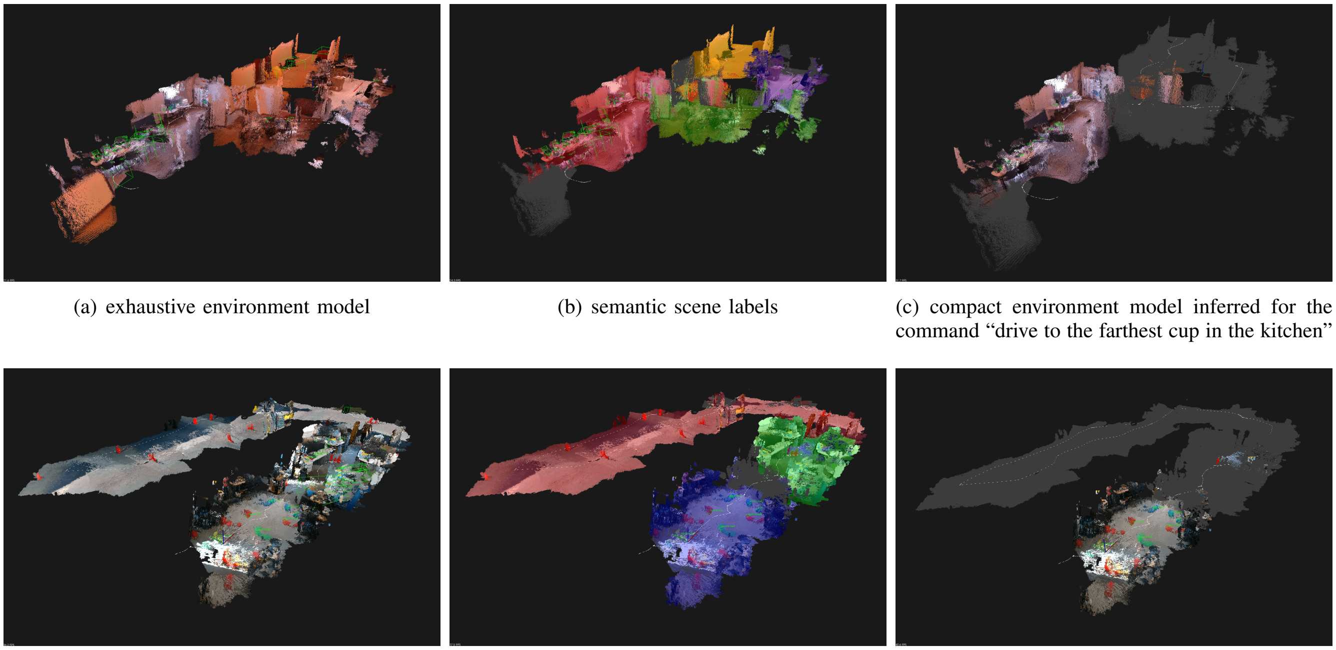
As more complex detectors are considered (e.g., ICP-based point cloud matching), we expect to find that these differences will become increasingly significant. For example, an operator performing service on a truck may require a robot to "turn the top-left screw on the back panel by forty five degrees" at one point during an activity, while it may also ask the same robot to "unload the truck of all of the pallets" at a later time. The computational requirements of the multitude of classifiers necessary to generate a consistent interpretation of the environment that is sophisticated enough to perform both of these tasks may be too burdensome for an robot to extract in real-time. We hypothesize that as the interactions approach such diversity and complexity, a model that extracts the salient information from the command and constructs a representation suitable for natural language symbol grounding will outperform non-adaptive representations of the environment.

B. *Observation Filtering*

To explore the impact of observation filtering, we evaluated the runtime performance of perception on the same six instructions explored for the adaptive perception experiment. Table I presents the results that reveal the impact of observation filtering (OF) against the standard baseline (B). This result demonstrates how removing observations inferred to be unnecessary to extract the meaning of the natural language instruction can improve the runtime performance of robot perception. The results demonstrate a 55% reduction in runtime for the instruction "go to the nearest suitcase in the parking lot" over the baseline. The improvement is a function of the diversity of scene labels across all observations. Table II shows the impact of observations filtering on the compactness of the approximated world representation. In this case the improvement is a function of the distribution of objects across different regions in the world.

C. *Observation Filtering with Adaptive Perception*

The last model that we considered combines observation filtering with adaptive perception. The results in Table I show the improvement of observation filtering with adaptive perception (OF+AP) against the standard baseline (B). As expected, combining both of these approaches reduces the time required to extract a suitable world model for natural language symbol grounding in all six scenarios. An example is depicted in Figure 4. In the best case, we observed a 90% improvement in runtime performance for the instruction "go



(a) exhaustive environment model

(b) semantic scene labels

(c) compact environment model inferred for the command “drive to the farthest cup in the kitchen”

(d) exhaustive environment model

(e) semantic scene labels

(f) compact environment model inferred for the command “drive to the nearest ball in the lab”

Fig. 4. A visualization of environment representations for Site 1 (top) and Site 2 (bottom). The renderings in (b) and (e) depict the scene labels. The standard approach of employing all observations and object classifiers results in (a), (d) an exhaustive representation of the environment. In contrast, inferring the set of observations and detectors relevant to the command yields (c), (f) compact environment models that afford more efficient grounding.

to the nearest ball in the hallway.” Table II lists the number of objects extracted by the perception pipeline. Reducing the number of objects significantly improves the runtime of symbol grounding, which is at best linear [18, 19] and at worst exponential [26] in the size of the world model.

VI. Conclusions

In this paper, we presented a novel framework that improves the efficiency of natural language understanding by generating and reasoning over a compact, instruction-specific world model. Underlying the framework are three primary methods that exploit the structure of language to facilitate inference. First, we use language reduce the set of all observations available to the robot by extracting semantic labels for the context in which the salient observations occur. Second, language is used to infer a subset of perceptual classifiers that extract a compact but sufficiently complex environment model that is suitable for interpreting the meaning of the instruction. Third, language is used in the context of the compact environment representation to infer the symbolic meaning of the instruction. Experimental results demonstrate how adaptive perception and observation filtering improve the computational efficiency of inference without affecting the accuracy of language grounding. In ongoing work, we are exploring methods to improve the robustness of semantic label classification for observations, including per-pixel semantic classification approaches.

This work also presents a number of interesting areas of future research. In the examples considered here, we did not exploit prior knowledge about the environment. However, one can easily extrapolate how using past compact representations to seed future models might mitigate the need to re-classify all objects for every instruction. A model that does not discard the information, but incrementally builds a rich spatial-semantic environment model over time is likely to be highly effective and efficient for human-robot interaction in complex environments with diverse tasks. Training and evaluating the performance of language models that use corpora collected from studies involving human-robot interaction and more complex tasks, robots, and environments that exploit differences in scale remain as future work. Such additional experiments would further characterize the performance of the proposed model and enrich our understanding of how to best construct efficient, hierarchical representations of environments for multi-modal human-robot interaction.

VII. Acknowledgements

This work was supported in part by the National Science Foundation under grants IIS-1638072 and IIS-1637813, by the Robotics Consortium of the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program Cooperative Agreement W911NF-10-2-0016, and by ARO grants W911NF-15-1-0402 and W911NF-17-1-0188.

References

[1] B. Kuipers, J. Modayil, P. Beeson, and M. MacMahon, “Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2004.

[2] J. Modayil, P. Beeson, and B. Kuipers, “Using the topological skeleton for scalable global metrical map-building,” in

Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS), 2004.

[3] P. Beeson, J. Modayil, and B. Kuipers, "Factoring the mapping problem: Moble robot map-building in the Hybrid Spatial Semantic Hierarchy," *Int'l J. of Robotics Research*, vol. 29, no. 4, 2010.

[4] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2012.

[5] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "Learning semantic maps from natural language descriptions," in *Proc. Robotics: Science and Systems (RSS)*, 2013.

[6] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2005.

[7] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2006.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[9] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[10] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int'l J. of Robotics Research*, vol. 26, no. 4, 2007.

[11] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *Trans. on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[12] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM – FAB-MAP 2.0," in *Proc. Robotics: Science and Systems (RSS)*, 2009.

[13] O. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, 2007.

[14] H. Zender, O. Martínez Mozos, P. Jensfelt, G. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, 2008.

[15] S. Hemachandra, M. R. Walter, S. Tellex, and S. Teller, "Learning spatial-semantic representations from natural language descriptions and scene classifications," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2014.

[16] S. Hemachandra, F. Duvallet, T. M. Howard, N. Roy, A. Stentz, and M. R. Walter, "Learning models for following natural language directions in unknown environments," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2015.

[17] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Approaching the symbol grounding problem with probabilistic graphical models," *AI Magazine*, vol. 32, no. 4, pp. 64–76, 2011.

[18] T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2014.

[19] I. Chung, O. Propp, M. Walter, and T. Howard, "On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[20] S. Patki and T. M. Howard, "Language-guided adaptive perception for efficient grounded communication with robotic manipulators in cluttered environments," in *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2018.

[21] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, 2009.

[22] B. Kuipers, "The spatial semantic hierarchy," *Artificial Intelligence*, vol. 119, no. 1, 2000.

[23] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, 2008.

[24] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz, "Inferring maps and behaviors from natural language instructions," in *Proc. Int'l. Symp. on Experimental Robotics (ISER)*, 2014.

[25] S. Harnad, "The symbol grounding problem," *Physica D*, vol. 42, pp. 335–346, 1990.

[26] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proc. Nat'l Conf. on Artificial Intelligence (AAAI)*, 2011.

[27] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *Proc. ACM/IEEE Int'l. Conf. on Human-Robot Interaction (HRI)*, 2010.

[28] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Proc. Int'l. Symp. on Experimental Robotics (ISER)*, 2012.

[29] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, "Learning to interpret natural language commands through human-robot dialog," in *Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 2015.

[30] T. Winograd, "Procedures as a representation for data in a computer program for understanding natural language," Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

[31] D. Roy, K.-Y. Hsiao, and N. Mavridis, "Conversational robots: Building blocks for grounding word meaning," in *Proc. HLT-NAACL Workshop on Learning Word Meaning from Non-Linguistic Data*, 2003.

[32] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proc. ACM/IEEE Int'l. Conf. on Human-Robot Interaction (HRI)*, 2010.

[33] S. Tellex, P. Thaker, R. Deits, T. Kollar, and N. Roy, "Toward information theoretic human-robot dialog," in *Proc. Robotics: Science and Systems (RSS)*, 2012.

[34] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Proc. Robotics: Science and Systems (RSS)*, 2014.

[35] Z. Gong and Y. Zhang, "Temporal spatial inverse semantics for robots communicating with humans," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2018.

[36] M. Spranger and L. Steels, "Co-acquisition of syntax and semantics-an investigation in spatial language," in *Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 2015.

[37] L. She and J. Chai, "Interactive learning of grounded verb semantics towards human-robot communication," in *Proc. Association for Computational Linguistics (ACL)*, 2017.

[38] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and executing recipes with a cooking robot," in *Proc. Int'l. Symp. on Experimental Robotics (ISER)*, 2010.

[39] M. Walter, M. Antone, E. Chuangsuwanich, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, Y. Friedman, J. Glass, J. How, J. Jeon, S. Karaman, B. Luders, N. Roy, S. Tellex, and S. Teller, "A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments," *J. of Field Robotics*, 2014.

[40] R. Paul, J. Arkin, N. Roy, and T. M. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," in *Proc. Robotics: Sci-*

ence and Systems (RSS), 2016.

[41] R. Paul, J. Arkin, D. Aksaray, N. Roy, and T. M. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms," *Int'l J. of Robotics Research*, 2018.

[42] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014.