# On the Performance of Hierarchical Distributed Correspondence Graphs for Efficient Symbol Grounding of Robot Instructions

Istvan Chung*, Oron Propp*, Matthew R. Walter, and Thomas M. Howard

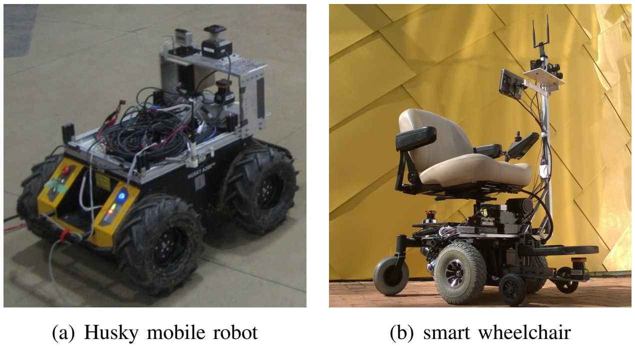
***Abstract*— Natural language interfaces are powerful tools that enables humans and robots to convey information without the need for extensive training or complex graphical interfaces. Statistical techniques that employ probabilistic graphical models have proven effective at interpreting symbols that represent commands and observations for robot direction-following and object manipulation. A limitation of these approaches is their inefficiency in dealing with larger and more complex symbolic representations. Herein, we present a model for language understanding that uses parse trees and environment models both to learn the structure of probabilistic graphical models and to perform inference over this learned structure for symbol grounding. This model, called the Hierarchical Distributed Correspondence Graph (HDCG), exploits information about symbols that are expressed in the corpus to construct minimalist graphical models that are more efficient to search. In a series of comparative experiments, we demonstrate a significant improvement in efficiency without loss in accuracy over contemporary approaches for human-robot interaction.**

## I. INTRODUCTION

Advances in natural language understanding will enable non-experts to effectively interact with robots in meaningful ways. Recent applications that include route direction-following [1, 2, 3, 4, 5], map building [6], and object manipulation [7, 8, 9] show that statistical approaches to language modeling can effectively translate intent expressed in free-form language to robots in restricted domains. The problem of command understanding is often considered as one of determining the most probable trajectory $\mathbf{x}^*(t)$ for a given natural language utterance $\Lambda$, based upon the perceived environment model $\Upsilon$ and space of trajectories $\mathbf{X}(t)$:

$$\mathbf{x}^*(t) = \underset{\mathbf{x}(t)\in\mathbf{X}(t)}{\arg\max}\; p\left(\mathbf{x}(t)\,|\Lambda, \Upsilon\right) \tag{1}$$

A limiting factor of contemporary approaches for natural language understanding of robot instructions is that they exhaustively reason over a fixed resolution of the environment representation. Consequently, real-time performance is often sacrificed for increasingly complex tasks and cluttered environments. In this paper, we present an algorithm that addresses this limitation by learning a distribution over the state space so as to exclude the constituents that are unlikely to be expressed during probabilistic inference. The Hierarchical Distributed Correspondence Graph (HDCG) increases performance by running probabilistic inference with the Distributed Correspondence Graph (DCG) [9] twice on the utterance [10]. The HDCG achieves improved performance by first learning a distribution over constituents that are relevant to the utterance. This distribution is then used to build a simplified language understanding model that can be searched more efficiently. This approach has been demonstrated to be effective for interpreting natural language instructions conveyed to different mobile robots (Figure 1) in domains that involved complex groundings composed of thousands of constituents. The HDCG has been used to translate natural language instructions in a multimodal interface to structured Tactical Behavior Sequences (TBS) [11] for outdoor navigation with an unmanned ground vehicle. Hemachandra et al. [12] utilize the HDCG on a voice-commandable wheelchair to infer annotations and behaviors in real-time for an intelligence architecture composed of a semantic map and a policy-based motion planner.



(a) Husky mobile robot (b) smart wheelchair

Fig. 1. Two mobile robots that have used the HDCG to interpret natural language instructions for autonomous navigation.

Hemachandra et al. [12] introduced the concept and presented a novel application of the HDCG. However, this work did not attempt to characterize the algorithm's performance in the context of other approaches. The main contributions of this paper include a formal description of the HDCG and a thorough evaluation of its computational complexity in comparison to that of related natural language understanding algorithms based on probabilistic graphical models.

## II. PROBABILISTIC GRAPHICAL MODELS FOR NATURAL LANGUAGE UNDERSTANDING

One class of techniques for natural language understanding that have proven effective of late are those that take a probabilistic approach to solving the symbol grounding prob-

*The first two authors contributed equally to this paper.

Istvan Chung and Oron Propp are with the Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Matthew R. Walter is with the Toyota Technological Institute at Chicago, Chicago, Illinois 60637, USA

Thomas M. Howard is with the University of Rochester, Rochester, NY 14627, USA

lem [13], that of mapping (*grounding*) linguistic elements to their corresponding referents in the external world. One such algorithm is the Generalized Grounding Graph (G$^3$) [7], which addresses this problem by constructing a probabilistic graphical model that expresses the relationship between elements from the language $\lambda_i \in \Lambda$ and their corresponding groundings $\gamma_i \in \mathbf{\Gamma}$ through binary correspondence variables $\phi_i \in \Phi$ according to the grammatical structure of the parse tree. Groundings are symbols that express some physical concept, such as an object, a region, a constraint, or a trajectory. Correspondence variables $\phi_i$ associated with each symbol-word pair $(\gamma_i, \lambda_i)$ denote whether the symbol $\gamma_i$ is the referent for the word $\lambda_i$. The G$^3$ assumes the correspondences to be true ($\phi_i = \textsc{True}$) and builds a factor graph representation of the distribution $p(\Phi = \textsc{True}|\Gamma, \Lambda, \Upsilon)$, with one correspondence node (assumed observed) and one grounding (unobserved) for each word in the utterance (observed). The algorithm then incrementally searches the graph for the most likely set of groundings $\gamma_i$ in the context of the phrases $\lambda_i$, child groundings $\Gamma_{c_i}$, and world model $\Upsilon$:

$$\Gamma^* = \underset{\gamma_1 \dots \gamma_n}{\arg\max}\; p\left(\Phi = \textsc{True} \,|\, \gamma_1 \dots \gamma_n, \Lambda, \Upsilon\right) \tag{2a}$$

$$\Gamma^* = \underset{\gamma_1 \cdots \gamma_n}{\arg\max} \prod_i p\left(\phi_i = \textsc{True} \,|\, \gamma_i, \lambda_i, \Gamma_{c_i}, \Upsilon\right) \tag{2b}$$

In order for G$^3$ to infer the most likely set of groundings, it searches all symbols that can be represented by the model. Since this space may contain all possible robot motions, this space is infinitely large for non-trivial robot domains. Generating an approximation of this space that is coarse enough to search efficiently and diverse enough to represent a variety of robot behaviors is quite difficult in practice. An alternative to inferring the most likely sequence of actions or states is to instead convert the natural language instruction into an equivalent representation as a planning problem. Directly inferring the most probable constraints, objective function, and model dynamics using the G$^3$ algorithm would also be difficult, because it requires evaluating each individual problem description in aggregate. To address this, Howard et al. [9] developed an alternative probabilistic graphical model formulation for natural language understanding of robot instructions called the DCG. This model builds upon the conditional independence assumptions made by the G$^3$ by assuming that constituents of a grounding are also conditionally independent and that the space of constituents is finite and known. What is not known in the DCG is how the constituents correlate to the phrases in the natural language utterance. Therefore, the DCG searches for the most likely correspondence variables $\phi_{i_j}$ in the context of the groundings $\gamma_{i_j}$, phrases $\lambda_i$, child groundings $\Gamma_{c_i}$, child correspondences $\Phi_{c_i}$, and world model $\Upsilon$ by maximizing the product of individual factors and uses the most likely set of correspondence variables $\Phi^*$ to express the most probable set of groundings $\Gamma^*$:

$$\Phi^* = \underset{\phi_{1_1} \cdots \phi_{n_m}}{\arg\max} \prod_i \prod_j p\left(\phi_{i_j} | \gamma_{i_j}, \lambda_i, \Phi_{c_i}, \Gamma_{c_i}, \Upsilon\right) \tag{3}$$

The runtime of probabilistic inference for the DCG can be shown to increase linearly with the number of constituents within a grounding set [9], which enables real-time translation of natural language into symbols for small structured languages. For example, Duvallet et al. [14] use the DCG to convert natural language utterances into annotations (observations about the environment) and behaviors (the objective function for robot navigation), thereby allowing users to command the smart wheelchair in Figure 1 using natural language instructions. Though significantly more efficient than G$^3$, the runtime of DCG depends significantly on the number of constituents expressed by the groundings, which limits its performance for complex tasks and environments.

## III. HIERARCHICAL DISTRIBUTED CORRESPONDENCE GRAPHS

The core difference between HDCG, DCG, and G$^3$ is that HDCG assumes that the space of symbols $\mathbf{\Gamma}$ for natural language understanding is itself a function of the utterance in the context of the environment. Direct inference of the most probable $\mathbf{\Gamma}\left(\Lambda, \Upsilon\right) \subset \mathbf{\Gamma}$ is difficult because of the inherent diversity of symbols and the computational cost of evaluating the likelihood of individual symbols. Instead, we observe that a set of rules that govern how the space is constructed may be easier to learn and faster to evaluate.

We introduce a set of rules $\mathrm{P} \in \mathbf{P}$. Each rule $\rho$ in the rule set $\mathrm{P}$ admits a certain subset of symbols in $\mathbf{\Gamma}$. The space of symbols is then a function of rules $\mathbf{\Gamma}\left(\mathrm{P}\right)$ rather than a fixed set; it consists only of the set of symbols created from constituents allowed by the rules $\mathrm{P}$. In HDCG, we run the first DCG in the space of rules $\mathbf{P}$ to infer a distribution of rule correspondence variables $\Phi_\rho$ in the context of the natural language instruction $\Lambda$ and the world model $\Upsilon$. Just as we use inferred correspondence variables to express groundings in Equation 3, we use the distribution of rule correspondence variables to infer a distribution of rule sets $\mathrm{P}$. We then run the second DCG over the distribution of reduced spaces $\mathbf{\Gamma}(\mathrm{P})$ to determine the most probable correspondence variables $\Phi^*$ that we use to express the most probable set of symbols $\Gamma^*$. We introduce the rule inference model as a latent random variable in Equation 3 that we marginalize out as illustrated in Equations 4 and 5. In practice, we approximate this summation using a limited number of most likely samples that result from beam search and use log-linear models trained from a corpus of examples to represent the probability of the correspondence between phrases and constituents.

This concept is illustrated in Figures 2 through 7 for "go to the kitchen that is down the hall," the example natural language expression from Hemachandra et al. [12] in an environment composed of seven regions of five different types (Figure 2). Constituents in this example symbol grounding model include regions, spatial relations, and goals. If we assume seven regions in the world model, eight spatial relation types (*below*, *above*, *near*, *far*, *right*, *left*, *back*, and *front*), and a number of goals that is equal to the number of regions in the world model, there exist seventy constituents

$$\Phi^* = \underset{\phi_{1_1}\ldots\phi_{n_m}}{\arg\max} \sum_{\Phi_\rho \in \boldsymbol{\Phi}_\rho} \prod_i \prod_j p\left(\phi_{i_j} | \gamma_{i_j}(\mathrm{P}), \lambda_i, \Phi_{c_i}, \Gamma_{c_i}(\mathrm{P}), \Upsilon\right) p\left(\Phi_\rho | \mathrm{P}, \Lambda, \Phi_{\rho_c}, \mathrm{P}_c, \Upsilon\right) \tag{4}$$

$$\Phi^* = \underset{\phi_{1_1}\ldots\phi_{n_m}}{\arg\max} \sum_{\Phi_\rho \in \boldsymbol{\Phi}_\rho} \prod_i \prod_j f\left(\phi_{i_j}, \gamma_{i_j}(\mathrm{P}), \lambda_i, \Phi_{c_i}, \Gamma_{c_i}(\mathrm{P}), \Upsilon\right) \prod_k \prod_l f_\rho\left(\phi_{\rho_{k_l}}, \rho_{k_l}, \lambda_k, \Phi_{\rho_{c_k}}, \mathrm{P}_{c_k}, \Upsilon\right) \tag{5}$$

in the symbol grounding model to evaluate (seven regions ($\gamma_1, \ldots, \gamma_7$), fifty-six spatial relationships ($\gamma_8, \ldots, \gamma_{63}$), and seven goals ($\gamma_{64}, \ldots, \gamma_{70}$)). In this example, it is clear that two types of semantic regions (*kitchen* and *hallway*) and one spatial relationship (*down*) are likely to be expressed during symbol grounding and therefore need to be represented in the model for natural language understanding.

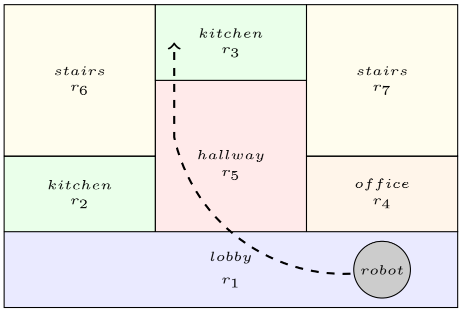
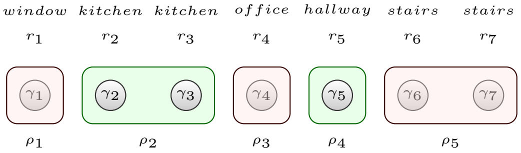


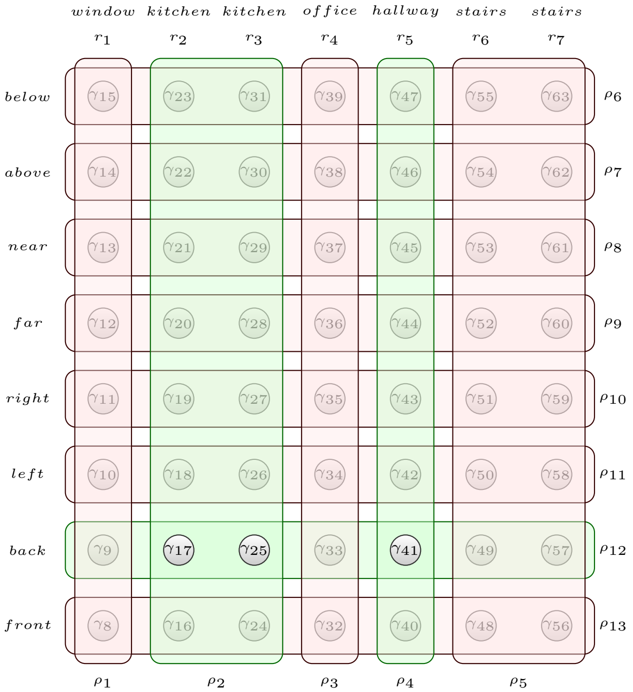
Fig. 2. An illustration of a world composed of seven regions. There exists one *lobby* region ($r_1$), two *kitchen* regions ($r_2$ and $r_3$), one office region ($r_4$), one hallway region ($r_5$), and two stair regions ($r_6$ and $r_7$). A robot is initially located in $r_1$ and the desired motion from the natural language instruction "go to the kitchen down the hall" is illustrated as a dashed line.

For this example, we assume one rule for each of the five region types ($\rho_1, \ldots, \rho_5$) and one rule for each of the eight spatial relations ($\rho_6, \ldots, \rho_{13}$). If a rule for a particular region type is active, all regions, goals, and spatial relationships with those regions are included in the symbol grounding model. Similarly, if a rule for a particular spatial relation type is active, all spatial relationships of that type are also included in the model for symbol grounding. Figures 3 through 5 illustrate partitioning of the space of constituents when three rules that correspond to the *kitchen* and *hallway* region types ($\rho_2$, $\rho_4$) and the *back* spatial relationship type ($\rho_{12}$) are active. Note that the expression of active and inactive rules is illustrated as translucent green and red boxes respectively.



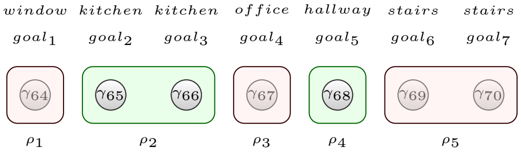
*window* $r_1$
*kitchen* $r_2$
*kitchen* $r_3$
*office* $r_4$
*hallway* $r_5$
*stairs* $r_6$
*stairs* $r_7$
$\gamma_1$ $\gamma_2$ $\gamma_3$ $\gamma_4$ $\gamma_5$ $\gamma_6$ $\gamma_7$
$\rho_1$ $\rho_2$ $\rho_3$ $\rho_4$ $\rho_5$

Fig. 3. An illustration of how rules filter the space of regions. The space of regions is formed by partitioning the space of all regions with rules that evaluate their semantic tag. In this example, two region rules ($\rho_2$, $\rho_4$) reduce the space of symbols from seven to three ($\gamma_2$, $\gamma_3$, and $\gamma_5$).



*window* $r_1$
*kitchen* $r_2$
*kitchen* $r_3$
*office* $r_4$
*hallway* $r_5$
*stairs* $r_6$
*stairs* $r_7$
*below* $\gamma_{15}$ $\gamma_{23}$ $\gamma_{31}$ $\gamma_{39}$ $\gamma_{47}$ $\gamma_{55}$ $\gamma_{63}$ $\rho_6$
*above* $\gamma_{14}$ $\gamma_{22}$ $\gamma_{30}$ $\gamma_{38}$ $\gamma_{46}$ $\gamma_{54}$ $\gamma_{62}$ $\rho_7$
*near* $\gamma_{13}$ $\gamma_{21}$ $\gamma_{29}$ $\gamma_{37}$ $\gamma_{45}$ $\gamma_{53}$ $\gamma_{61}$ $\rho_8$
*far* $\gamma_{12}$ $\gamma_{20}$ $\gamma_{28}$ $\gamma_{36}$ $\gamma_{44}$ $\gamma_{52}$ $\gamma_{60}$ $\rho_9$
*right* $\gamma_{11}$ $\gamma_{19}$ $\gamma_{27}$ $\gamma_{35}$ $\gamma_{43}$ $\gamma_{51}$ $\gamma_{59}$ $\rho_{10}$
*left* $\gamma_{10}$ $\gamma_{18}$ $\gamma_{26}$ $\gamma_{34}$ $\gamma_{42}$ $\gamma_{50}$ $\gamma_{58}$ $\rho_{11}$
*back* $\gamma_9$ $\gamma_{17}$ $\gamma_{25}$ $\gamma_{33}$ $\gamma_{41}$ $\gamma_{49}$ $\gamma_{57}$ $\rho_{12}$
*front* $\gamma_8$ $\gamma_{16}$ $\gamma_{24}$ $\gamma_{32}$ $\gamma_{40}$ $\gamma_{48}$ $\gamma_{56}$ $\rho_{13}$
$\rho_1$ $\rho_2$ $\rho_3$ $\rho_4$ $\rho_5$

Fig. 4. An illustration of how rules filter the space of spatial relations. The space of spatial relations is formed by the intersection of rules that permit particular regions and spatial relation types. In this example, the intersection of two region rules ($\rho_2$, $\rho_4$) and one spatial relation rule ($\rho_7$) reduce the space of symbols from fifty-six to three ($\gamma_{17}$, $\gamma_{25}$, and $\gamma_{41}$).



*window* $goal_1$
*kitchen* $goal_2$
*kitchen* $goal_3$
*office* $goal_4$
*hallway* $goal_5$
*stairs* $goal_6$
*stairs* $goal_7$
$\gamma_{64}$ $\gamma_{65}$ $\gamma_{66}$ $\gamma_{67}$ $\gamma_{68}$ $\gamma_{69}$ $\gamma_{70}$
$\rho_1$ $\rho_2$ $\rho_3$ $\rho_4$ $\rho_5$

Fig. 5. An illustration of how rules filter the space of navigation goals. The space of goals is formed by partitioning the space of all goals with rules that evaluate their semantic tag. In this example, two region rules ($\rho_2$, $\rho_4$) reduce the space of symbols from seven to three ($\gamma_{65}$, $\gamma_{67}$, and $\gamma_{68}$).

These three rules reduce the space of constituents for symbol grounding from seventy to nine ($\gamma_2$, $\gamma_3$, $\gamma_5$, $\gamma_{17}$, $\gamma_{25}$, $\gamma_{41}$, $\gamma_{65}$, $\gamma_{66}$, and $\gamma_{68}$). Figure 7 illustrates the constituents that would be actively expressed in a DCG originating from the parse tree in Figure 6. Of particular interest is how the ambiguity of the phrase "the kitchen" would be resolved by incorporating the information from the spatial relationship inferred from "that is down the hall." Since all of the actively expressed constituents are expressed by the simplified model, we would expect that the DCG would find the same solution as the full model in less time.

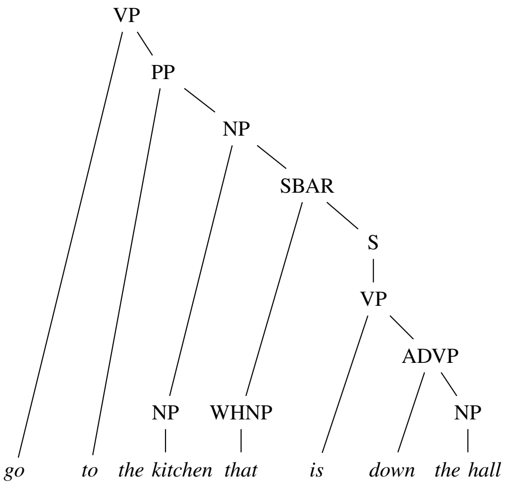


Fig. 6. A parse tree for the natural language instruction “go to the kitchen that is down the hall” using tags from the Penn Treebank [15].

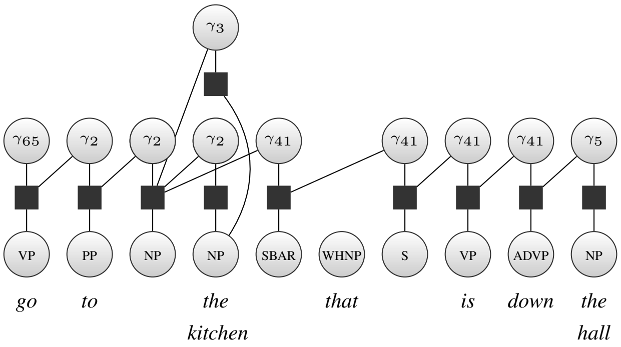


Fig. 7. The actively expressed constituents in the symbol inference model for the natural language instruction “go to the kitchen that is down the hall” in the context of a map with seven objects.

Figure 8 illustrates how the DCG applies to the problem of inferring a set of rules to construct the simplified model in Figure 7. The constituents at the root of the sentence contain the rules that describe all actively expressed constituents in the symbol inference model for that particular expression. We infer a likelihood of individual models because labeling of natural language expressions can vary between annotators.

## IV. EXPERIMENTS

We evaluate the performance of the G$^3$, DCG, and HDCG models through comparative experiments that measure both the runtime and number of factor computations required of each model using a corpus of navigation instructions. Each test consisted of training each model on a corpus of examples and then measuring the CPU time and number of factor evaluations executed by the model for each example. To vary the complexity of the phrases, we performed tests over corpus subsets of varying size. This was accomplished by randomly sampling a subset of the corpus examples and then constructing a space of only those groundings comprised of the symbols (region and spatial relation types) present in the subset. By changing the size of the subset of the corpus, we were able to adjust the size of the grounding space used for each test, thus enabling us to measure the performance of each model in situations of varying complexity.

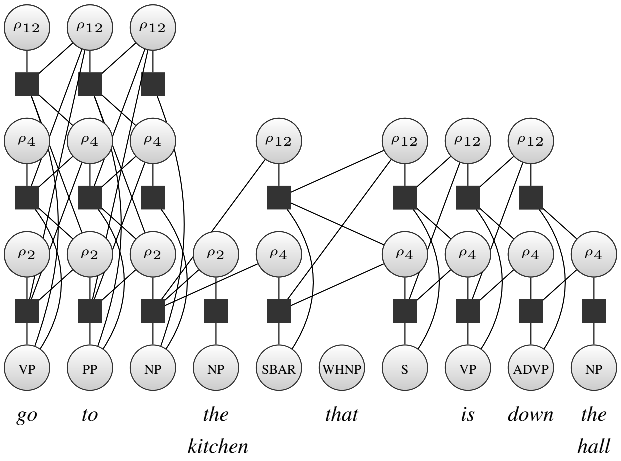


Fig. 8. The actively expressed constituents in the rule inference model for the natural language instruction “go to the kitchen that is down the hall” in the context of a map with seven objects.

Even with a limited number of corpus examples, the exponentially large number of possible sets of groundings made it infeasible to test the G$^3$ in the same way as DCG and HDCG. In order to create grounding spaces small enough to run the G$^3$, we constructed grounding spaces consisting of only the individual groundings present in the corpus subset. This is an even smaller subset of the grounding space than the subset of all groundings comprised of symbols present in the corpus subset (the space used for DCG and HDCG). The full grounding space was then constructed as the powerset of these individual groundings and the G$^3$ was run on these smaller spaces.

Experiments were performed using a Python implementation of G$^3$, DCG, and HDCG on the open-source H2SL[1] corpus of 32 examples, referencing 10 region types and eight different spatial relations. All tests were run on a quad-core Intel Core i7 processor with 16 GB of RAM running Mac OS X 10.10 (x86_64 architecture).

Figures 9 and 10 present log-log plots of the runtimes and factor evaluations for the DCG, HDCG and G$^3$ models, respectively. In these plots, we normalize the runtimes and factor evaluations over the number of phrases in the instruction, since the model is run over each of these phrases individually. Although this normalization does not account for the fact that different phrase types have grounding spaces of different sizes, it effectively reduces the possible confounding effect of complex sentence structure. Also note that the independent variable is the number of individual

[1] https://github.com/tmhoward/h2sl

groundings used to construct the grounding space and not the size of the grounding space itself, which for the G$^3$ is constructed as the powerset of these individual groundings. Finally, note that in these plots and in all others throughout this paper, the error bars denote standard error.

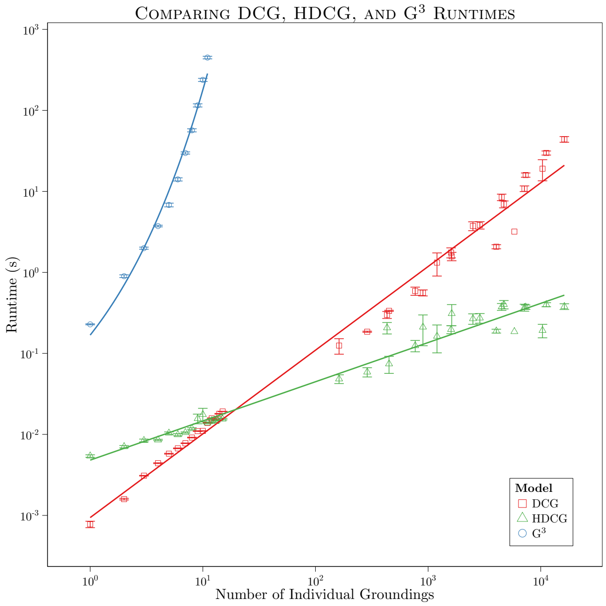


Fig. 9. Log-log plot of models' runtime for inferring groundings vs. number of individual groundings, normalized over number of phrases.

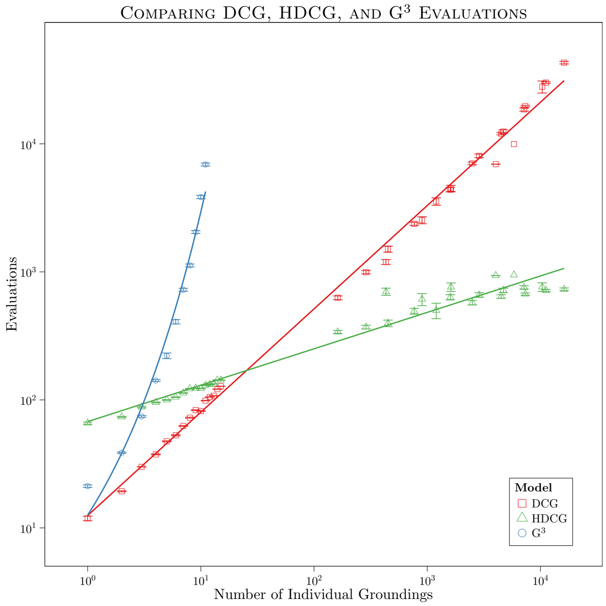


Fig. 10. Log-log plot of models' factor evaluations vs. number of individual groundings, normalized over number of phrases.

In order to check whether the improved performance of the HDCG is not at the expense of correctness, we performed an additional series of tests to evaluate accuracy. We trained the DCG and HDCG models on a randomly sampled subset of the corpus, then tested each on the full corpus. We varied the percentage of the corpus sampled from 0% to 100% over intervals of 10%, and recorded whether the models inferred the correct grounding. Figure 11 displays a plot comparing the accuracy DCG and HDCG.

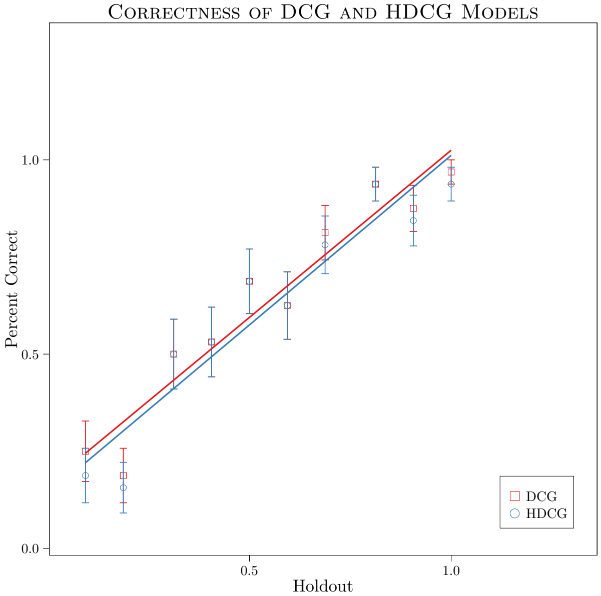


Fig. 11. Percentage of examples inferred correctly vs. percentage of examples used for training for the DCG and HDCG models.

## V. Discussion

The runtime and factor evaluation costs of the HDCG and DCG models were both observed to be linear with respect to the number of individual groundings on the log-log scale, corresponding to a monomial relationship. The HDCG model's log-log line had a lower slope but greater intercept than DCG, corresponding to a smaller power but larger coefficient. This reflects the conclusion that the cost of inferring groundings is dominant for large grounding spaces, while the cost of rule inference dominates for small spaces.

The G$^3$ exhibits trends in runtime cost and number of factor evaluations that are exponential with respect to the number of individual groundings. This trend is due to the exponentially increasing number of possible sets of groundings the G$^3$ is required to evaluate. In addition, each individual G$^3$ factor evaluation requires a greater number of feature evaluations than an individual DCG factor evaluation. This is due to the presence of additional features for grounding sets, which caused the G$^3$ factor evaluations to have a greater runtime cost than the corresponding DCG factor evaluations. This explains why the G$^3$ has a greater intercept than the DCG models in the runtime plot, but not in the plot for factor evaluations. The HDCG consistently achieves correctness rates nearly equivalent to that of the DCG, showing that accuracy is not compromised for the superior performance obtained by the hierarchical model.

We found that for small grounding spaces, it was more efficient to use the DCG directly. However, as the size of the grounding space increases, the HDCG requires significantly less time and factor evaluations than the DCG, while still retaining a similar level of accuracy. The HDCG thus better supports the ability for robots to understand a more complex set of instructions in real-time.

## VI. CONCLUSIONS

A key limitation of contemporary approaches to understanding natural language robot instructions involves scaling to diverse tasks and complex environments. In this paper, we presented the Hierarchical Distributed Correspondence Graph, a novel approach to natural language understanding that achieves significant gains in efficiency by explicitly building a compact graphical model that expresses the symbol grounding problem. This HDCG addresses the linear growth in computational complexity that encumbers the efficiency of the previous state-of-the-art by inferring a distribution over simplified models for grounding symbols. We provided a detailed analysis of the computational complexity of the HDCG that offers further insights into the performance characteristics of the HDCG observed on real-time, real-world deployments [12]. Of particular note is the identification of two regions of performance, one in which the number of groundings is small and the computational cost of inferring the structure of the simplified model outweighs the benefits of more efficient symbol grounding, and the other in which the larger number of grounding constituents makes the hierarchical approach more efficient.

In future work, we will explore the application of hierarchical probabilistic graphical models to improve the computational performance of related models like $G^3$ that do not assume the conditional independence of constituents. We also seek to eliminate the dependence on training models using fully annotated corpora by exploring semi-supervised learning techniques. Additionally, we are actively exploring applications that extend beyond route directions to tasks like cooperative robot manipulation that require inference over a larger space of groundings. Lastly, we are investigating the extension of the model to additional human-robot communication mechanisms that include joint gesture, tone, speech, and facial expression.

## VII. ACKNOWLEDGMENTS

Oron Propp and Istvan Chung were supported by Massachusetts Institute of Technology's Program for Research in Mathematics, Engineering and Science for High School Students (PRIMES) and Undergraduate Research Opportunities Program (UROP).

## REFERENCES

[1] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," in *Proc. Nat'l Conf. on Artificial Intelligence*, 2006.

[2] 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*, Osaka, Japan, 2010, pp. 259–266.

[3] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proc. Nat'l Conf. on Artificial Intelligence*, August 2011, pp. 859–865.

[4] 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*, 2012.

[5] J. Fasola and M. Mataric, "Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, November 2013, pp. 143 – 150.

[6] 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*, 2013.

[7] 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*, San Francisco, CA, 2011.

[8] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from unscripted deictic gesture and language for human-robot interactions," in *Proc. AAAI Conf. on Artificial Intelligence*, 2014.

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

[10] T. Howard, I. Chung, O. Propp, M. Walter, and N. Roy, "Efficient natural language interfaces for assistive robots," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS) Work. on Rehabilitation and Assistive Robotics*, 2014.

[11] J. Oh, A. Suppe, F. Duvallet, A. Boularias, J. Vinokurov, L. Navarro-Serment, O. Romero, R. Dean, C. Lebiere, M. Hebert, and A. Stentz, "Toward mobile robots reasoning like humans," in *Proc. of the Twenty-Ninth AAAI Conf. on Artificial Intelligence*. AAAI, January 2015.

[12] S. Hemachandra, F. Duvallet, T. Howard, N. Roy, A. Stentz, and M. Walter, "Learning models for following natural language directions in unknown environments," in *Proc. of the 2015 Int. Con. on Robotics and Automation*, 2015.

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

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

[15] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.