Adaptive Simulation for Grounding Language Instructions that Refer to the Future State of Objects

Tabib Wasit Rahman, Katelyn Shakir, and Thomas M. Howard

***Abstract*— Accurate and efficient communication is essential for human-robot interaction. Spatiotemporal relationships are commonly used in language to resolve ambiguity about referred objects when they cannot be uniquely identified based on visual features. Grounding of instructions that involve spatiotemporal relationships remains a difficult problem in human-robot interaction because of the lack of annotated data and the difficulty of representing or encoding such information in an environment model. This paper outlines an approach that builds from previous methods that explored minimal but sufficient environment models for symbol grounding that address spatiotemporal relationships that project into the future. It specifically explores the application of an adaptive timestep that eliminates unnecessary cycles when we can predict that the outcome of inference will not change due to a small step forward in time. An evaluation based on a small corpus and a detailed example are presented.**

I. Introduction

For a human and robot to effectively collaborate on non-trivial tasks in unstructured environments, they must be able to efficiently and effectively communicate about their joint task. Spatiotemporal relationships, such as those that refer to the relative state of objects at different times, is one type of concept that people use to resolve objects that are not semantically unique or easily identifiable based on their visual appearance. Consider the example illustrated in Figure 1 where the instruction "Grab the cup that is about to fall off the table" is given to a robot. This instruction requires knowledge about the future state of the objects in the scene to accurately interpret the instruction. In order to ground the meaning of the noun phrase "the cup that is about to fall off the table", we must resolve that the cup that is referred to is a cup that is currently on the table but will not be on the table at a future time. This paper explores an architecture for grounding instructions that refer to the future state of objects based on other models that selectively interpret previous observations to resolve instructions that refer to the past state of objects. We specifically explore how an adaptive timestep for simulation of a world model can be used to improve the efficiency of symbol grounding. The contributions of this paper include a discussion of an architecture for natural language understanding of robot instructions that refer to the future state of the world, a corpus-based evaluation of a fixed and an adaptive timestep for world model simulation that enables faster symbol grounding, and a detailed analysis of the performance of symbol grounding for an instruction that refers to the future state of the world.

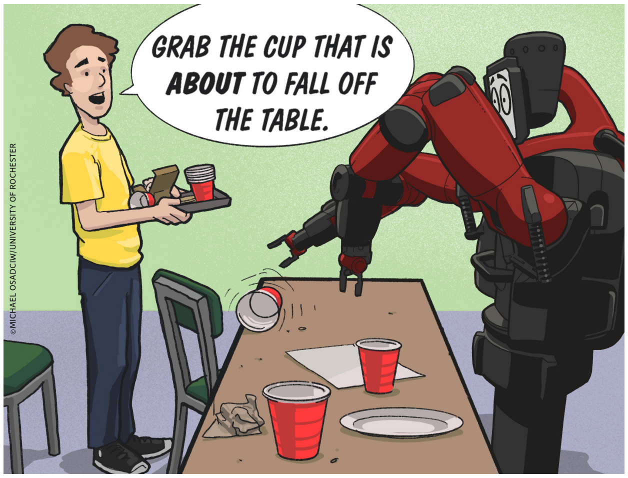

GRAB THE CUP THAT IS ABOUT TO FALL OFF THE TABLE.
©MICHAEL OSADCIW/UNIVERSITY OF ROCHESTER

Fig. 1. An illustration of an instruction where the robot needs to utilize past and present states of the objects to predict the future dynamics of the world to interpret and execute the instruction.

II. Background

Natural language understanding of robot instructions for human-robot interaction is a field that dates back several decades. Earlier approaches, such as those based on probabilistic graphical models [6], [12], [14] are now being evaluated against more recent methods based on Large Language Models [1], [7], [8] that resolve problems of scale and the lack of annotated data. Such recent models however have not demonstrated a similar proficiency for spatial relationships and spatiotemporal relationships such as those explored in this paper. This paper will explore how spatiotemporal relationship, specifically those that refer to the future state of the world, would be handled by methods based on the Distributed Correspondence Graph (DCG) [5], [6], [10]. DCGs are probabilistic models that infer the most likely set of symbols $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ from language $\Lambda_t = \{\lambda_1, \ldots, \lambda_n\}$ and world model Υ_t at time t. Typically the world model is represented as a map of objects with unique identifiers that contain information about the metric state, semantic types, and/or relationships to other objects [4]. This information is the product of a perception pipeline that models these objects from information contained in the history of sensor observations $z_{1:t}$. Although sensor observations used to construct a metric-semantic representation of the world most commonly rely on

This work was supported by National Science Foundation under grant number 2144804. Tabib Wasit Rahman, Katelyn Shakir and Thomas M. Howard are with the University of Rochester, Rochester, NY, United States trahman2@u.rochester.edu.

RGB-D and LIDAR data streams, [2] showed how language can be used as a complementary sensing modality to inform the robot about the state of objects that cannot be visually observed. DCG search is performed by searching all factors in a factor graph for the most likely associations between language $\lambda_i \in \Lambda_t$, symbolic constituents $\gamma_{ij} \in \Gamma_t$, correspondence variables $\phi_{ij} \in \Phi$, and the expressed symbolic constitutes of the immediate children phrases Φ_{c_i}. DCG inference involves searching over the graph for the most likely correspondence variables ϕ_{ij} associated with phrase λ_i and the j^{th} symbol for that phrase γ_{ij} by maximizing the factored distribution. Once the factor graph has found the distribution of most likely correspondence variables in the factor graph, those where the correspondence variable is most likely "true" can be used to extract the symbolic representation of the sentence. Conditional probabilities in the DCG are modeled using a log-linear model [3] with binary features. Engineered features that evaluate spatial relationships, temporal relationships, semantic types, words, metric values, etc. enable the log-linear model to learn relationships between the symbolic constitutes of a phrase and it's immediate child phrases. Weights for these features are learned by training on a corpus of labeled examples that annotate the *true* groundings (symbols) associated with the individual phrases in a constituency parse of an utterance. *false* groundings are similarly learned by associating that label with any unexpressed symbols in the annotation when the example is associated with an environment model. The next section illustrates an extension to that architecture that enables the interpretation of instructions that refer to the future state of the world.

III. Technical Approach

The approach in [9] outlined a method for natural language understanding of robot instructions that refer to the past and/or present state of objects. This architecture uses language in three ways. Consider the interpretation of the sentence "pick up the apple on the table that was to the left of the red cup". First, symbols are inferred that guide the interpretation of the language instruction. The sentence contains information that specifies several objects that need to be in the environment model and the kinds of spatial relationships that also may need to be considered. Second, symbols are inferred to determine constraints over the kinds of symbols that are required to be inferred for a correct interpretation of the sentence. The same sentence also indicates that at the root of the expression the robot should be performing an action that picks up some object in the environment. Third, an attempt at symbol grounding is performed by a natural language understanding algorithm that produces a distribution of likely symbols. If those symbols do not satisfy the constraints inferred by the grounding constraint inference module, then it is hypothesized that the root of the problem lies in an insufficiently detailed environment model. The architecture in [9] closes a loop around this cycle of perception, modeling, inference, and constraint checking to iterate through past observations until a minimal but sufficient environment representation is extracted that provides the necessary symbols and environment model to satisfy the grounding constraints of the instruction. If there are multiple apples on the table, then the meaning of "the apple on the table that *was* to the left of the red cup" cannot be resolved without revisiting past observations. If however the instruction was "pick up the apple on the table that *is* to the left of the red cup" then grounding could be performed with the current observation as the model could uniquely identify the object referred to by "the apple on the table that is to the left of the red cup". The ability for DCGs to effectively reason about spatial relationships, including ones that refer to the relative placing of objects in a group, is described in [10]. The main contribution of this paper outlines how such an approach would be used to interpret instructions that refer not to the past or present state of objects, but the future state of an object. Consider a modification of the architecture from [9] illustrated in Figure 2. If we want to understand the instruction "pick up the apple before it rolls off the table", then we need to predict the future state of the world and enable the features in the log-linear models that represent the conditional probabilities in the factor graph to utilize this information. A naive approach to simulation would advance the world by a fixed timestep, attempt natural language understanding on that world, check the inferred symbols against the grounding constraints, and then iterate if necessary. This workshop paper explored an adaptive timestep for environment model simulation that checks for differences in the ordering of spatial relationships, specifically those involving contact between pairs of objects. Knowing that the engineered features inside of the log-linear models used by the DCGs trained on the corpus of sentences described in the next section change their output based on the relative ordering of contacts between objects in the environment, we can exploit this information to know that such features will not change their expression until this relative ordering change is observed. Simulation of a semantic-metric world model continues until some spatial relationship, such as novel contact between a pair of objects, is observed. We theorize that such an approach would reduce the number of attempts at natural language understanding and thus exhibit a generally faster result without a loss in accuracy for instructions that refer to the future state of the environment.

IV. Experimental Design

For preliminary testing of the proposed model described in Figure 2, an analysis of the performance of a small corpus is presented where statistics about the overall runtime, average number of model iterations, and runtimes for individual aspects of the model are reported. A corpus of self-annotated instructions using linguistic patterns similar to those found in [11] were used for these experiments. This corpus consisted of 8 unique instructions that refer to the past and/or future states of the world across three unique environments for a total of 24 instructions. Examples of such instructions include "the ball that will hit the table second" and "the first

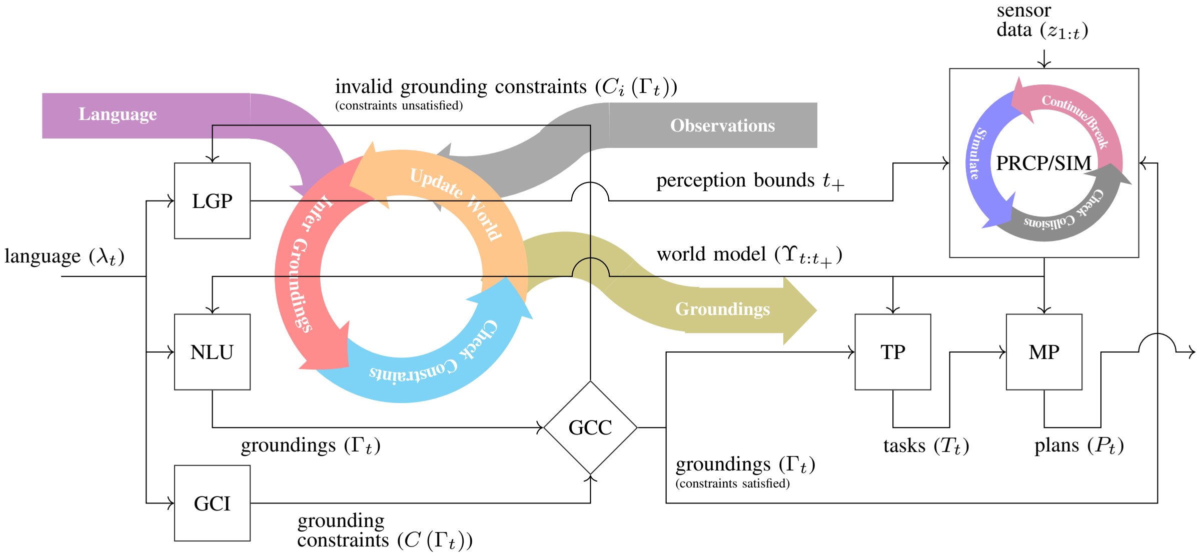
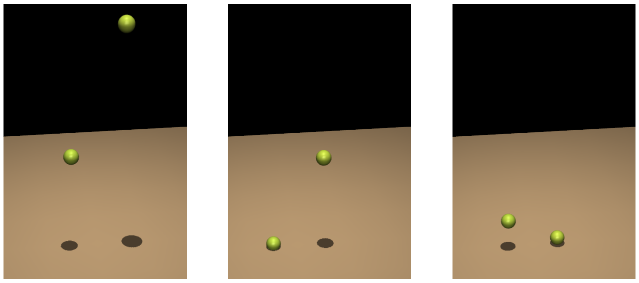


Fig. 2. The proposed intelligence architecture to infer minimal but sufficient environment models for instructions that refer to the future state of objects with simulation that tracks collision between objects to determine the gap between attempts at grounding the instruction. As in [9] this framework requires the development of model for grounding constraint inference (GCI) and grounding constraint checking (GCC), but considers instructions that refer to the future state of objects. The process of world modeling, language understanding, and symbol checking now exhibits two cycles, one cyclic behavior that iteratively refines the environment model until the grounding constraints are satisfied and another that continue simulations until an event that would alter the expression of log-linear model features is observed.



(a) At time $t_{+0.30\text{sec}}$ no contacts have taken place

(b) At time $t_{+0.45\text{sec}}$ o_1 registers a contact with the table

(c) At time $t_{+0.50\text{sec}}$ o_2 registers a contact with the table

Fig. 3. A visualization depicting the internal state of the MuJoCo simulation during the PRCP/SIM module's extension of the environment model horizon. In this example the environment at time t uses two spherical objects to represent tennis balls o_1 and o_2 at positions $(x, y, z) = (0.0\text{m}, 0.1\text{m}, 1.0\text{m})$ and $(x, y, z) = (0.0\text{m}, -0.1\text{m}, 1.25\text{m})$ respectively above an planar objects to represent a table o_3 at position $(x, y, z) = (0.0\text{m}, 0.0\text{m}, 0.0\text{m})$ at time t with no initial velocity. The simulation ran for a duration of 0.50sec at 0.05sec intervals. 3(a) shows the state of the simulation at time $t_{+0.30\text{sec}}$ when both the spheres are still falling under gravity and yet to make contact with the table. 3(b) shows the state of the simulation between time $t_{+0.45\text{sec}}$ when the spheres on the left registers a contact with the table. 3(c) shows the state of the simulation between times $t_{+0.50\text{sec}}$ when the spheres on the right registers a contact with the table.

ball that will hit the table”. For the corpus-based experiments, three distinct training and testing sets were constructed by placing two random environment examples in the training set and the other in the test set. This eliminated inference errors that could have resulted from a lack of coverage of the language but analyzed the generalization across different environments. The NLU DCG was trained using a feature set of 2 correspondence features, 16 linguistic features, and 54 symbol features, from the fully annotated examples. The GCI and LGP DCGs were trained by extracting world-independent symbols from each fully annotated example. The feature sets of both the DCGs used 2 correspondence features and 16 linguistic features, in addition to the GCI DCG using 20 symbol features, and the LGP DCG using 50 symbol features. Probabilistic inference was performed with a beamwidth of two and a simulation timestep of 0.05 seconds for all examples. The simulation of the world model for actions that referred to the future used MuJoCo [13] to estimate the motion of objects in each example. An illustration of the progression of the environment model during this process is shown in Figure 3. Since the contribution of this paper is about the effect of a fixed timestep versus an adaptive timestep for the re-inference process as mentioned in Figure 2, we run the three corpus of examples using both the fixed and adptive simulation step algorithms.

V. Results

Table I illustrates the results of the corpus-based experiments described in Section IV. Each time the framework was trained on one of the three distinct training sets containing 16 examples and then tested on its corresponding test set of 8 examples. The number of iterations and runtimes of all the modules in the architecture were recorded for each of the test examples. At the end of the three sets, all the test example data were collected together and their results averaged. The two columns of Table I show the mean number of iterations and runtimes in milliseconds of the different modules during the inference process with a 95% confidence interval with the fixed and adaptive timesteps.

We observed that the average number of iterations of

mean $\pm 2\sigma$	fixed timestep	adaptive timestep
Iterations (#)	11.38 ± 1.66	3.75 ± 1.45
LGP (ms)	40.28 ± 13.81	36.21 ± 1.59
GCI (ms)	5.16 ± 0.71	5.23 ± 0.20
NLU (ms)	224.80 ± 31.20	96.97 ± 27.34
GCC (ms)	0.11 ± 0.03	0.05 ± 0.02
PRCP/SIM (ms)	3.46 ± 0.66	3.39 ± 0.68
Total (ms)	274.08 ± 38.50	141.98 ± 28.10

TABLE I
CORPUS-BASED EXPERIMENTAL RESULTS WITH AND WITHOUT ADAPTIVE SIMULATION STEP

the inference loop decreased from 11.38 to 3.75 when the adaptive simulation step was used. This is an expected result because the adaptive timestep only runs the re-inference process whenever a spatiotemporal relationship of the objects in the world model change. This is further supported by the result that the average time taken for the NLU module has decreased by 57%, from 224.80 milliseconds to 96.97 milliseconds with the use of the adaptive simulation step. Since the algorithm takes less number of iterations of the inference process to ground instructions refering to the future, the architecture spends less time in the NLU module.

Comparing the average times of the LGP, GCI, and PRCP/SIM modules, it can be noticed that the adaptive timestep does not significantly affect their running times. This is true for the LGP and the GCI modules because these modules execute the inference process only once per instruction. As for the PRCP/SIM module, it is expected that that the average runtime be unchanged because even though there are less number of re-inference steps taking place in the NLU, the simulator still simulates and updates the world model the same number of times.

Taking the example of "the ball that will hit the table second" as in Figure 3, simulating the state of the world at 0.05 seconds and running the inference process at every step results in eleven iterations of the NLU. However, using the adaptive simulation step still simulates the world every 0.05 seconds but only runs the re-inference process whenever a change in the contacts between objects take place; this results in only three iterations of the NLU module. Furthermore, the NLU modules takes a total of 230.12 milliseconds using a fixed timestep whereas using an adaptive timestep takes 87.02 milliseconds. For both versions, the times taken by the other modules were almost the same, with the total runtime of the fixed timestep version being 275.94 milliseconds and the adaptive timestep version taking 132.26 milliseconds.

VI. CONCLUSION

This paper presents an architecture for grounding natural language instructions that refer to the future state of objects. This method iteratively simulates the world based on the observations at the time of the instruction until a satisfactory solution is found. Additionally, this paper outlines a method for adaptive simulation between symbol grounding attempts using information about what changes in the environment will change the expression of weighted features that are used to estimate the conditional probabilities inside factors of the DCG. Experimental results analyzed the performance of the model on a small corpus designed to explore the language of interest and illustrated the benefits of adaptive timesteps. Future work will investigate the performance of these models on a larger corpus of instructions and further explore the details of the architecture's behavior for instructions that refer to the past, present, and/or future state of the world.

REFERENCES

[1] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R.J. Ruano, K. Jeffrey, S. Jesmonth, N.J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[2] J. Arkin, Daehyung Park, Subhro Roy, M.R. Walter, N. Roy, T.M. Howard, and R. Paul. Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions. *The International Journal of Robotics Research*, 39(10-11):1279–1304, 2020.

[3] M. Collins. Log-linear models. *Self-Published Tutorial*, 2005.

[4] E. Fahnestock, S. Patki, and T.M. Howard. Language-guided adaptive perception with hierarchical symbolic representations for mobile manipulators, 2019.

[5] T. M. Howard, E. Stump, J. Fink, J. Arkin, R. Paul, D. Park, S. Roy, D. Barber, R. Bendell, K. Schmeckpeper, J. Tian, J. Oh, M. Wigness, L. Quang, B. Rothrock, J. Nash, M. R. Walter, F. Jentsch, and N. Roy. An intelligence architecture for grounded language communication with field robots. *Field Robotics*, 2021.

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

[7] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

[8] J.X. Liu, Z. Yang, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and A. Shah. Grounding complex natural language commands for temporal tasks in unseen environments. In *Conference on Robot Learning*, pages 1084–1110. PMLR, 2023.

[9] S. Patki, J. Arkin, N. Raicevic, and T.M.. Howard. Language guided temporally adaptive perception for efficient natural language grounding in cluttered dynamic worlds. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7854–7861, 2023.

[10] 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. *International Journal of Robotics Research*, 37(10):1269–1299, June 2018.

[11] R. Paul, J. Arkin, N. Roy, and Thomas M Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. 2016.

[12] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, Ashis G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, pages 1507–1514, 2011.

[13] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[14] M.R. Walter, M. Antone, E. Chuangsuwanich, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, Y. Friedman, J. Glass, Jon P. How, Jeong H. 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. 32(4):590–628, 2015.