Grounding Language Instructions that Refer to the Past, Present, and/or Future State of Objects for Human-Robot Interaction

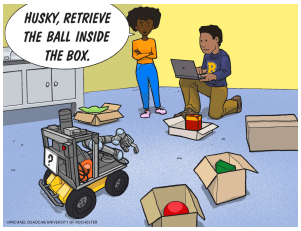
Tabib Wasit Rahman[1], Katelyn Shakir[1], Nikola Raicevic[2], and Thomas M. Howard[1]

***Abstract*— For robots to effectively collaborate with human partners, they need to be able to understand what instructions and/or statements mean in the context of their environment. To ground objects in a dynamic world, robots need an understanding of how both spatial and temporal relationships evolve as a function of time in the environment. However, it is computationally intensive to classify, track, and predict the motion of all objects. Approaches based on Language-Guided Temporally Adaptive Perception (LGTAP) utilize information embedded in the instruction to selectively classify objects to construct minimal but sufficiently detailed models of the environment for symbol grounding. Such methods, however, fail when the instruction refers to the future state of the environment as it lacks any notion of whether to and for how long a future prediction is necessary to ground the instruction. This prompts a reformulation of LGTAP that can selectively utilize information from past observations to accurately predict the future state of objects. This paper describes a novel approach for LGTAP for instructions that may refer to the past, present, and/or future state of the environment by closing the loop around symbol grounding and adaptive perception. A detailed analysis of a grounding problem that refers to the future state of the environment, a corpus-based analysis of performance, and a physical demonstration of natural language understanding is presented along with a description of this novel architecture.**

I. Introduction

As robots become more capable, aware, and safer to operate, applications are growing where they assist humans with daily tasks. A prerequisite for effective teaming is the ability to accurately and efficiently communicate with humans. Natural language understanding is one important component of this capability, which requires a robot to understand what is being said to it in the context of the environment in which the instruction is given. Grounded language communication is a subset of natural language understanding which refers to interactions that make specific references to the environment and can only be understood in the context of that environment. Because the environment is not necessarily known a priori, sensor observations must be collected, interpreted and/or processed to successfully translate the meaning of what was said to the robot.

When a human gives an instruction to a robot, it may refer to the past, the present, or the future state of the world to disambiguate objects in the environment. Three of these examples are illustrated in Figure 1. The first image in Figure 1(a) illustrates the instruction, "Husky, retrieve the ball inside the box" that refers to the present state of the world. To ground this instruction, it can attempt to process the immediate sensor data or prior model of the world to identify "ball" and "box" objects and reason about the inferred spatial relationship from the word "inside" to interpret the meaning of the instruction. The second image in Figure 1(b) shows the instruction "Hand me the wrench that I was using a few minutes ago". This instruction is different from the first because it requires knowledge about the past to ground the instruction. To accurately disambiguate which "wrench" is being referred to in an environment where many objects may satisfy that semantic category, a robot must be able to recall information about the relationships of



(a) A scenario where the current state of objects is required

(b) A scenario where the past state of objects is required

(c) A scenario where the future state of objects is required

Fig. 1. Three unique human-robot collaboration situations in non-trivial dynamic environments. The image in 1(a) shows the robot requiring the present state of the world to successfully interpret the utterance. The image in 1(b) shows the robot requiring past and present dynamics of the world to ground the object of interest. The image in 1(c) shows the robot needing to utilize past and present states of the objects to predict the future dynamics of the world to interpret and execute the instruction.

*This work was supported by National Science Foundation under grant number 2144804. [1]Tabib Wasit Rahman, Katelyn Shakir and Thomas M. Howard are with the University of Rochester, Rochester, NY, United States trahman2@u.rochester.edu. [2]Nikola Raicevic is with the University of California San Diego. Nikola contributed to this work as a student at the University of Rochester.

the person asking the question and the objects in the scene. The last image in Figure 1(c) exhibits the instruction "Grab the cup that is about to fall off the table". This instruction differs from the previous two because it requires knowledge about the future state of the objects in the scene. To ground the meaning of the noun phrase "the cup that is about to fall off the table" we must understand that it is referring to a cup that is currently on the table but will soon not be on the table. These examples illustrate three important problems in grounded language communication for human-robot interaction that can occur when a human is naturally interacting with a robot partner.

Over the past several decades, many different approaches [4], [9], [17] have been explored to translate instructions into a meaningful representation of a task in the context of the robot's environment. Present-day models frame this problem as one of inferring associations between linguistic constituents of the utterance and the symbolic representation of the entities observed in the world, their metric and semantic properties, and potential actions that the robot can execute. As elaborated further in Section II, many recent approaches only work within the context of instantaneous worlds, and while some have been able to work in dynamic environments [11], [14], their approaches focus only on past world states and fail to address language instructions referencing future world dynamics.

Extending such models to construct minimal but sufficient models of the environment for grounded language communication with reference to the past, present, and/or future world dynamics requires a novel approach that uses language in several ways. First, as in [11], a language model is used to infer grounding constraints that dictate what kinds of symbols are expected to be inferred from the instruction. Second, symbols are inferred from language from a second language model to identify what minimal representation of the environment is needed for grounding and what changes to the history of observations may be required to satisfy grounding constraints. A departure from the approach described in [11] is the ability to extract observations backward from the current time, forward from the first observation, and/or forward from the current time to understand instructions that may refer to the past, present, and/or future state of objects in the world. Third, language is used to perform grounding using weighted binary features that trigger from spatiotemporal relationships in the inferred minimal but sufficient environment model. This work represents the first probabilistic approach to natural language symbol grounding that simulates the future state of the environment only when it is deemed necessary for interpreting the meaning of the statement. The contributions of this paper are as follows:

- A novel architecture for grounded language communication with robots that enables the extraction of a minimal but sufficient environment representation for interpreting instructions that may refer to the past, present, and/or future state of the environment
- A detailed analysis of the application of the proposed model for an example of language understanding of an instruction that refers to the future state of the environment
- An analysis of model performance across a corpus of instructions that refer to the past, present, and/or future state of the environment
- A demonstration of the proposed model for real-time human-robot interaction of an instruction that refers to the future state of the environment

II. Related Work

Recent approaches to symbol grounding reason in the context of a rich but instantaneous version of the world [3], [13], [16] and are effective for interpreting a variety of utterances relating to the present, but they fail to execute utterances that require knowledge of past and/or future dynamics of the world. Temporal Grounding Graphs [14] lazily infer the context implied by the utterance through a probabilistic inference on a factor graph to resolve some issues involving past utterances, but again do not extend to models that may refer to the future state of the environment. Other works such as [10], [15] use large-scale visual-language pre-trained models and embeddings to resolve underspecified instructions and partially unknown environments. However, these approaches address the problem of grounding through interactive dialogue to resolve ambiguities. Extensions to Distributed Correspondence Graphs, including our more recent approaches involving language-guided adaptive perception [12], have attempted to use language to generate minimal but sufficient models of the environment that contain only the details needed to understand the instruction. A recent paper presenting a novel approach to resolving instructions that may refer to the past or present states of objects is presented in [11]. This paper introduced the idea of grounding language in a closed loop, guided by symbols that predicted the necessity to extend the temporal horizon of the world model backwards. However, these approaches would also fail to address language instructions that refer to the future state of objects in the world.

In recent years, Large Language Models (LLMs) have shown significant improvements in language-related tasks such as engaging in conversations, question answering, and prompt-based text generation. This raises the question of whether LLM-based techniques can also perform symbol grounding effectively. Recent approaches have shown that LLM-based techniques work well to convert complex navigation commands to Linear Temporal Logic (LTL) specifications [8], as well as long-horizon manipulation tasks [1], [7] with robustness to the diversity of natural language. The framework used in [8] has shown to use pre-trained LLMs with a known static semantic map to parse navigation commands in indoor and outdoor settings and output an LTL formula for an LTL-based motion planner. The frameworks used in [1], [7] have shown to use LLMs that have been trained on low-level skills to parse long-horizon tasks in static environments and generate sequences of skills that perform the tasks. Furthermore, the framework in [7] is also capable of refining the solutions generated by the LLM-based

techniques using the present state of the robot's environment. However, many LLM-based approaches assume a single camera frame input and do not refer to actions that require the past and/or future states of the world.

In summary, recent approaches to symbol grounding only reason about natural language using a static representation of the robot's environment. Even though LLM-based techniques can capture the diversity and nuances of natural language utterances, they have not been used to reason about instructions in a dynamic environment relating to the past and/or the future states of the robot's environment. This paper builds upon the work of [11] to propose a new model that not only enables robots to efficiently interpret natural language instructions that refer to the past, but also infer instructions that reason about the future dynamics of the world.

III. Background

The language grounding models used in this paper are based on the Distributed Correspondence Graph (DCG) [13]. DCGs are probabilistic models that infer a set of most likely symbols $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ from language $\Lambda_t = \{\lambda_1, \ldots, \lambda_n\}$ and world model Υ_t at time t. The world model is extracted from the history of sensor observations $z_{1:t}$ using a set of detectors in the robot's perception pipeline $\Delta = \{\delta_1, \ldots, \delta_n\}$ capable of detecting various semantic and metric properties of the entities in the robot's workspace. Each factor in the DCG searches for the most likely associations between the linguistic elements $\lambda_i \in \Lambda_t$ and the symbolic constituents $\gamma_{ij} \in \Gamma_t$ by introducing the notion of unknown random variables called correspondence variables $\phi_{ij} \in \Phi$. The correspondence variables ϕ_{ij} associate a phrase λ_i with a symbol γ_{ij}. The model reasons about the meaning of a particular phrase conditioned on the grounded meaning of its immediate children phrases Φ_{c_i}, and then assumes conditional independence across the linguistic and symbolic constituents to propose an approximate factorization of the grounding distribution that affords an efficient inference. 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. A graphical depiction of a DCG for the expression "the ball that will hit the table second" is illustrated in Figure 2.

In practice [6] this learned function $\Psi(\phi_{ij}, \gamma_{ij}, \Phi_{c_i}, \lambda_i, \Upsilon_t)$ is implemented as a log-linear model [5] with binary features being evaluated for each known random variable in a factor. Weight parameters associated with each feature function are optimized by training on a corpus of examples [13] that were self-labeled by the authors.

The formulation introduced in [13] and other applications consider a static model of the world Υ_t which represents the state of the world at the time of the given utterance. For instructions that refer to the past, these models must track the state of all objects $\Upsilon_{1:t}$, which is a nontrivial computational burden for dynamic environments with many objects. A solution to this problem is explored in [11], which introduces the notion of Language Guided Temporally Adaptive Perception (LGTAP) for symbol grounding of instructions that refer to the past or present state of the world. This model performs inference in a temporally compact world model $\Upsilon^*_{t_-:t}$ with objects selectively interpreted from object detectors that are themselves inferred from the language. A DCG based symbol grounding inference when reasoning in the context of this temporally compact world model then takes the following form:

$$\Phi^* = \arg\max_{\phi_{ij} \in \Phi} \prod_{i=1}^{|\Lambda_t|} \prod_{j=1}^{|\Gamma_t|} p(\phi_{ij} | \gamma_{ij}, \Phi_{c_i}, \lambda_i, \Upsilon^*_{t_-:t}) \quad (1)$$

Experimental results presented in [11] illustrate the advantages of inferring minimal but sufficient environment models for language understanding in cluttered and dynamic environments. Such models however would not be able to ground the meaning of instructions that refer to the future state of the environment, such as the example illustrated in Figure 2. The next section illustrates an extension to that architecture that enables the construction of minimal but sufficient environment models that may refer to the past, present, and/or future state of the world.

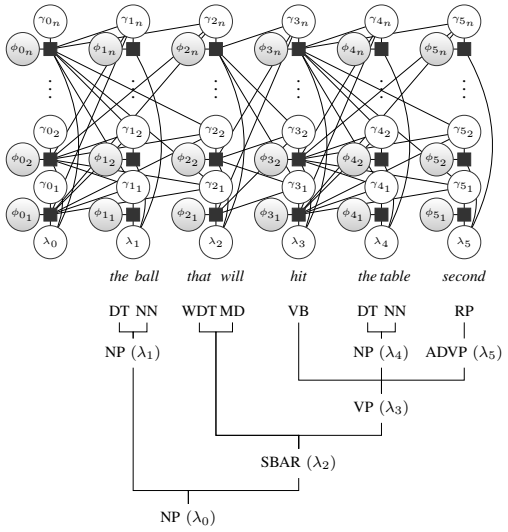


Fig. 2. The DCG for the expression "the ball that will hit the table second".

IV. Technical Approach

Inspired by the approach described in [11], we seek an approach to computationally efficient grounded language communication that constructs a minimal but sufficient environment model for language instructions that refer to the past, present, and/or future state of the world. This requires the construction of $\Upsilon_{t_-:t_+}$ with a minimum number of objects and their state histories needed to accurately interpret the meaning of the sentence. In this section, we introduce this novel architecture in Section IV-A, review a space of symbols necessary for each language model in Section IV-B, and discuss novel features necessary to ground instructions that refer to the future in Section IV-C.

A. Architecture

Figure 3 illustrates this new architecture, which differs from [11] in that it permits the extension of the perception bounds, and therefore the environment model, forward and/or backwards in time. First, language (λ_t) at time t is processed by a Language-Guided Perception (LGP) model to infer the direction of the perception bounds and the perceptual classifiers that will be needed to ground the instruction. If the symbols inferred by LGP indicate a future relationship, the Perception/Simulation (PRCP/SIM) module creates the world model and simulates forward the state of only the objects deemed relevant by LGP, e.g. ball, at time t. When previous observations are required, it seeks observations either from the start of the observation history or incorporates the most recent observations the robot has acquired, depending on the contents of the symbols extracted from the LGP module. Additionally, the language input is processed by a Grounding Constraint Inference (GCI) model to infer what kinds of symbols are to be expected from the NLU model. Because Distributed Correspondence Graphs (DCGs) are used as the model for natural language understanding in this architecture, if the model deems all symbols to be more probably *false* than *true* then no symbols will be expressed. This can occur when columns of factors of the model illustrated in Figure 2 fail to express symbols that are needed by factors higher in the hierarchy to express weighted binary features. If the model is trained with examples that cover the kinds of spatial, temporal, or spatiotemporal relationships that are represented in λ_t but fails to satisfy the grounding constraints $\mathrm{C}(\Gamma_t)$ inferred by the GCI model, then we hypothesize that it is not an issue of accuracy of the language model but an absence in detail and/or symbols provided by the environment model $\Upsilon_{t_-:t_+}$. The following discussion will elaborate on the symbols used by the NLU, LGP, and GCI models, and the binary features required to implement the architecture described in Figure 3.

B. Symbols

The first set of symbols reviewed are those for the NLU model. Just as in [13], many of these symbols represent the objects present in the robot's world model and their semantic properties. We assume that the robot's world model Υ_t is populated with the set of perceived objects $\mathcal{O}$ each having an associated semantic type $\mathcal{L}$ e.g. "ball", "table", "camera", "robot", etc. We define the set of symbols referring to these perceived objects as:

$$\Gamma^{\mathcal{O}} = \{\gamma_{o_i} | o_i \in \mathcal{O}\} \tag{2}$$

The set of symbols that refer to the semantic type of these objects is defined as:

$$\Gamma^{\mathcal{L}} = \{\gamma_{l_i} | l_i \in \mathcal{L}\} \tag{3}$$

To introduce the idea of each object type possessing a specific order in the context of an utterance such as "the first ball", we use $\mathcal{H}$ to represent the set of order relations such as "first", "second", "last", etc., and define the associated order and ordered object type symbols as follows:

$$\Gamma^{\mathcal{H}} = \{\gamma_{h_i} | h_i \in \mathcal{H}\} \tag{4}$$

$$\Gamma^{\mathcal{L}_{\mathcal{H}}} = \left\{\gamma_{l_i h_j} | l_i \in \mathcal{L}, h_j \in \mathcal{H}\right\} \tag{5}$$

Next, we consider the symbols representing a contact relation of an object type in phrases such as "hit the ground" or "roll off the table". We can use the same set of object types $\mathcal{L}$ to represent the association with words like "hit" and "roll" to define contact symbols as:

$$\Gamma^{\mathcal{C}} = \{\gamma_{c_i} | c_i \in \mathcal{L}\} \tag{6}$$

To handle more complex spatiotemporal relationships that may refer to the past, present, and/or future state of the environment, new symbols representing such relationships are needed. We introduce temporal relation symbols within the contact relations for phrases like "will hit the table" and "had hit the table". In $\Gamma^{\mathcal{C}^{\mathcal{T}}}$ we define the set $\mathcal{D}$ to represent the temporal direction of events such as forwards in time or backwards in time, and the set $\mathcal{R}$ to represent the temporal frame of reference such as the current time or the start time of the observation history:

$$\Gamma^{\mathcal{C}^{\mathcal{T}}} = \left\{\gamma_{c_i}^{d_j r_k} | c_i \in \mathcal{L}, d_j \in \mathcal{D}, r_k \in \mathcal{R}\right\} \tag{7}$$

Like how we augmented object types with an order relation to refer to phrases such as "first ball", we must do the same for the contact and temporal contact symbols mentioned above to interpret phrases such as "hit the table first" and "will hit the table second". So, using the set $\mathcal{H}$ as introduced above, we define the set of symbols for ordered contact $\Gamma^{\mathcal{C}_{\mathcal{H}}}$ and ordered temporal contact $\Gamma^{\mathcal{C}^{\mathcal{T}}_{\mathcal{H}}}$ symbols as follows:

$$\Gamma^{\mathcal{C}_{\mathcal{H}}} = \left\{\gamma_{c_i h_j} | c_i \in \mathcal{L}, h_j \in \mathcal{H}\right\} \tag{8}$$

$$\Gamma^{\mathcal{C}^{\mathcal{T}}_{\mathcal{H}}} = \left\{\gamma_{c_i h_l}^{d_j r_k} | c_i \in \mathcal{L}, d_j \in \mathcal{D}, r_k \in \mathcal{R}, h_l \in \mathcal{H}\right\} \tag{9}$$

The symbol space for the NLU model is the union of symbols defined in Equations 2 through 9. The symbols for the LGP model are the union of symbols defined by Equations 3 through 9. The LGP model is used to inform the construction of $\Upsilon^*_{t_-:t_+}$, and therefore cannot be dependent on the object symbols described in Equation 2. Labels for the LGP model examples derive from the human-annotated grounded language examples by inspecting the types of objects and the direction of temporal relationships labeled at each phrase.

Like the LGP model symbols, the symbols for the GCI model are also independent of $\Upsilon^*_{t_-:t_+}$. The GCI model's symbol space is solely based on grounding constraints, which represent the type and number of symbols of that type that are expected to be expressed for a given instruction. For example, if Γ is the set of grounded symbols for the instruction "the last ball that had hit the table", the symbols inferred by the root of the DCG must satisfy the constraints $|\Gamma| = 1$ and $\Gamma \in \Gamma^{\mathcal{O}}$ because that statement refers to a single object. Symbols for grounding constraints are parameterized by $\mathcal{N}$, the number of symbols and $\mathcal{S}$, the symbol types

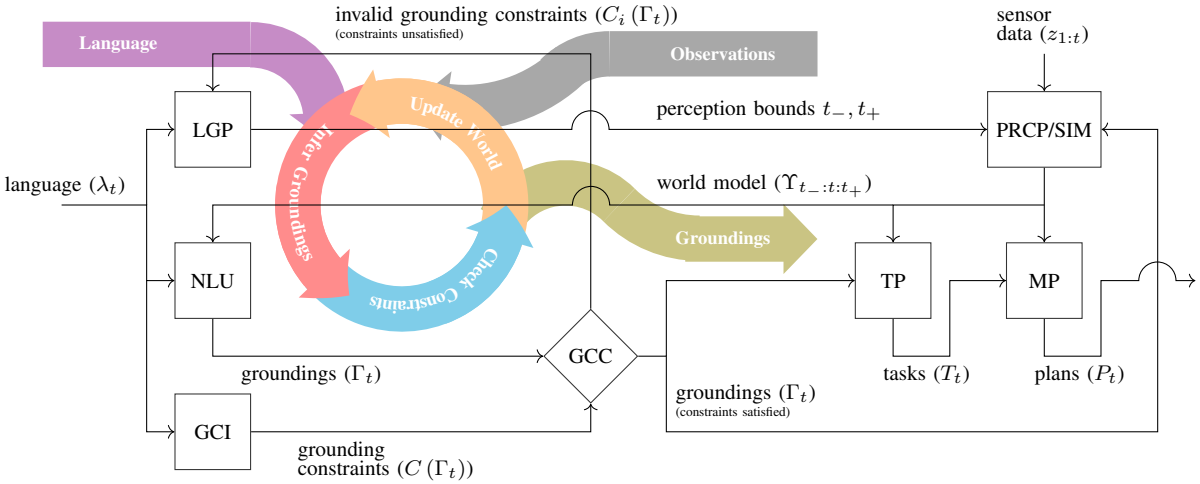


Fig. 3. The proposed intelligence architecture to infer minimal but sufficient environment models for instructions that refer to the past, present, and/or future state of objects. This framework requires the development of a new model for grounding constraint inference (GCI) and grounding constraint checking (GCC). The process of world modeling, language understanding, and symbol checking now exhibits a cyclic behavior that iteratively refines the environment model until the grounding constraints are satisfied.

respectively:

$$\Gamma^{\text{GCI}} = \left\{ \gamma_{s_i}^{n_i} | n_i \in \mathcal{N}, s_j \in \mathcal{S} \right\} \tag{10}$$

Like the LGP model labels, the labels for the grounding constraints are derived from the human-annotated grounded language examples.

C. Features

The NLU, LGP, and GCI models utilize similar structures for the set of binary features used in training of their log-linear models. Each feature set is composed as a cartesian product of vectors of correspondence features ξ^{ϕ}, linguistic features ξ^{λ}, and symbol features ξ^{γ}:

$$f(\phi_{ij}, \gamma_j, \Phi_{c_i}, \lambda_i, \Upsilon) = \left\{ \xi^{\phi_{ij}} \times \xi^{\lambda_i} \times \xi^{\gamma_j} \right\} \tag{11}$$

Correspondence features encode the correspondence variable $\phi_{ij} \in \Phi$ associated between each phrase $\lambda_i \in \Lambda$ and symbol $\gamma_j \in \Gamma$. Linguistic features encode facts about each phrase $\lambda_i \in \Lambda$, such as the phrase type, the number of child phrases, and each word present in the phrase. Symbol features encode information about each symbol $\gamma_{ij} \in \Gamma$. Some symbols are independent of the environment and are only functions of the symbol's type and properties. Others are dependent on the state of the world, such as those that check the relative position of the object to other objects, or the alignment to various reference frames. Such features are required to resolve spatial relationships such as "the leftmost block" or "the moving vehicle". Other features are purely concerned with the relationship between the current symbol and symbols expressed by the child phrases. These are important for generalizing linguistic patterns like conjunctions, in phrases such as "the red block and the blue block". Lastly, there are complex features that are important for resolving ambiguities based on patterns of symbols expressed across multiple child phrases. The motivating example of "the ball that will hit the table second" is a good example of this, because it must infer which ball is being referred to among multiple types of those objects in the world based on an inferred spatiotemporal relationship from "that will hit the table second". The models used in this paper use implementations of new binary features that consider such spatiotemporal relationships, which were needed to accurately infer the meaning of such instructions. The advantage of partitioning the symbol feature vector as such allows us to efficiently update the solutions inferred by the DCG using the Efficient Graph Update (EGU) technique described in [2] when there are only updates to the world model. As demonstrated in [11], EGU can have a significant impact on runtime in sufficiently cluttered environments.

V. Experimental Design

Three experiments are presented that will analyze different aspects of the proposed model's performance. First, a detailed example of the performance of the proposed model for a language instruction that fails to ground at Υ_t is examined. Overall runtime of probabilistic inference, as well as runtimes for individual aspects of the model are reported for this detailed example. Simulation of the world model for actions that referred to the future used MuJoCo [18] to estimate the motion of objects in each example. Second, a corpus-based analysis 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-labeled instructions using linguistic patterns like those found in [13] were used for these experiments. This corpus consisted of 18 unique instructions that refer to the past and/or future states of the world across three unique environments for a total of 54 instructions. Examples of such instructions include "the ball that had hit the table second", "the first ball that had hit the table", "the fourth ball that will hit the table", and "the ball that will hit

the table third". 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 binary correspondence features, 16 linguistic features, and 54 symbol features. The GCI and LGP DCGs were trained by extracting world-independent symbols from each fully labeled example because these models are trained on the uttered instructions alone. The feature sets of both the DCGs used 2 binary 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 for all examples. To analyze the performance of the model in physically realistic setting, we used the Rethink Robotics Sawyer Collaborative Robot shown in Figure 5 for a manipulation task where the instruction refers to the future state of the environment. A custom-made gripper is mounted on the robot's arm and a net is placed inside the gripper to attempt catching the ball. A human operator releases tennis balls inside the robot's workspace as another operator sends the instruction to be grounded by the robot. The robot then initiates the proposed architecture above and moves its arm towards the ball that the instruction refers to. The perception pipeline used to detect and track the state of objects used RGB and depth images from an Intel RealSense D455 camera. The OpenCV library was used to create a color segmentation algorithm to detect the tennis balls and generate bounding boxes of the tennis balls in the RGB image. Using the pointclouds generated by the depth images and a distance-based object tracking algorithm, the 3D positions of the tennis balls at each time frame were stored in the perception pipeline.

VI. Results

A. Detailed Example

For the detailed example of the process illustrated in Figure 3, the sentence "the ball that will hit the table second" will be explored. The corresponding DCG for this instruction is illustrated in Figure 2. The proposed architecture receives the parsed sentence and uses it to generate three DCGs for LGP, GCI, and NLU. Because no environment model is required for LGP or GCI, inference is first performed on those to understand the temporal relationships contained in the sentence and what kinds of symbols are expected. The GCI model infers grounding constraints for the GCC module. For this sentence, the most likely grounding constraints inferred at the root of the DCG illustrated in Figure 2 is

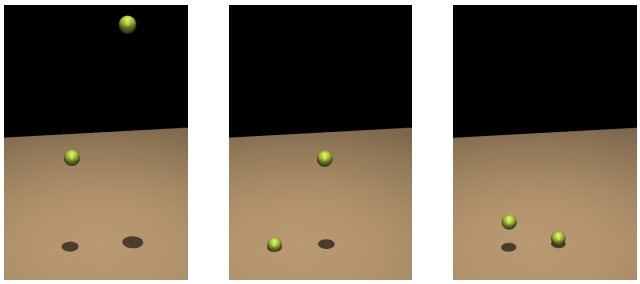
$$C(\Gamma_t) = \{\Gamma_t \in \Gamma^{\mathcal{O}} \wedge |\Gamma_t| = 1\}$$

This means that the grounded symbol from the NLU must be of an object type as defined in Equation 2, and there must be only one grounded symbol. GCI constructed a DCG with 8 symbols per factor and took 7.65 milliseconds to perform inference. For this sentence the LGP model infers a forward direction of temporal horizon for the PRCP/SIM module and the kinds of object classifiers that would be needed to perform inference. LGP constructed a DCG with 65 symbols per factor and took 52.85 milliseconds to perform inference. For this instruction, we see object classifiers for $\gamma_{o_{\text{ball}}}$ and $\gamma_{o_{\text{table}}}$ are expressed at the root of the model in addition to temporal relationships that describe contact between a pair of objects at a future state. At first NLU uses a model of the environment consisting of objects corresponding to those classified objects at time t to perform symbol grounding. NLU constructed a DCG with 68 symbols per factor and took 61.62 milliseconds to perform inference.

Using the environment model at the time of the language input Υ_t the NLU model fails to infer any groundings at the root of the sentence that satisfies the inferred grounding constraints because the current environment detected no contacts between "ball" or "table" objects. The architecture now feeds back this failed inference into the PRCP/SIM module, which uses the symbols from the LGP model to move the temporal horizon in the forward direction. To move the temporal horizon forward, the PRCP/SIM module instantiates a MuJoCo simulation environment with three objects that represent the state of two tennis balls and the table contained in the world model Υ_t. The model simulation inside MuJoCo was set to run at 500Hz. Updates with an increment of 0.1sec were used between each Υ update so that the world model is not unnecessarily simulated beyond the horizon required to perform inference. Contacts between objects are recorded in the updated world model, which is used by symbol features in the NLU's DCG to interpret words such as "on", "hit", or "touch". The updated world model is loaded into the NLU's DCG model and inference is performed to update the robot's interpretation of the sentence in the context of the environment. This is where the efficient graph updates become useful; since only the world model has changed and the uttered sentence has not, we do not need to generate a new DCG for the NLU module for re-inference and re-evaluate all the binary weighted features but can only re-evaluate the world-dependent features. This process continues until the NLU produces symbols that satisfy the grounding constraints inferred by the GCI model. In this example, 7 environment updates were needed, which took a total of 309.00 milliseconds to go from the receipt of the sentence to accurately inferred symbols with a simulated world $\Upsilon_{t:t+0.7\text{sec}}$. Incremental updates to the NLU took an average of 26.11 milliseconds. An illustration of the progression of the environment model during this process is shown in Figure 4.

B. Corpus Based Experiments

Table I illustrates the results of the corpus-based experiments of the novel architecture without the use of EGU and with the use of EGU, as described in Section V. Each time the framework was trained on one of the three distinct training sets containing 36 examples and then tested on



(a) At time $t_{+0.4\text{sec}}$ no contacts have taken place

(b) At time $t_{+0.6\text{sec}}$ o_1 registers a contact with the table

(c) At time $t_{+0.7\text{sec}}$ o_2 registers a contact with the table

Fig. 4. 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 $(x, y, z) = (0.0\text{m}, 0.1\text{m}, 1.5\text{m})$ and $(x, y, z) = (0.0\text{m}, -0.1\text{m}, 2.0\text{m})$ 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 0.7sec at 0.1sec intervals. 4(a) shows the state of the simulation at time $t_{+0.4\text{sec}}$ when both the spheres are still falling under gravity and yet to make contact with the table. 4(b) shows the state of the simulation at time $t_{+0.6\text{sec}}$ when the sphere on the left registers a contact with the table. 4(c) shows the state of the simulation at time $t_{+0.7\text{sec}}$ when the sphere on the right registers a contact with the table.

its corresponding test set of 18 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 experiments, all the test example data were collected together, and their results averaged. Column 1 of Table I shows the mean number of iterations and runtimes in milliseconds of the different modules during the inference process with a 95% confidence interval without the use of Efficient Graph Updates (EGU). For comparison purposes, the three experiments were repeated with EGU enabled, and the results are shown in column 2 of Table I. The number of iterations required for the inference process did not change at all when EGU was enabled. This is expected because the EGU algorithm is a mechanism that allows to efficiently re-evaluate the binary features used in the inference process and does not contribute to the number of iterations, which is dependent on the compactness of the robot's world model. Similarly, it can be seen that the runtimes of the GCC and PRCP/SIM modules are also unaffected by the introduction of EGU, because the processes taking place inside the modules are completely independent of the EGU algorithm.

The large spread in the PRCP/SIM module average runtime can be explained as follows: the distribution of PRCP/SIM runtimes is bimodal, i.e. it depends on if the uttered instruction refers to the past or the future. In the case that the instruction refers to the past, the SIM part of the module is not initiated at all, and it is a matter of refining the world model using past observations inside the PRCP part of the module. This is a relatively very fast process compared to when the uttered instruction refers to the future, in which case the SIM part of the module is initialized and the world model is simulated forwards in time, resulting in a more computationally expensive and time-consuming process. Hence, a single confidence interval does not really provide a useful insight into the spread of the average runtime of the PRCP/SIM module.

mean $\pm 2\sigma$	without EGU	EGU
Iterations (#)	11.06 $\pm$ 5.47	11.06 $\pm$ 5.47
LGP (ms)	37.63 $\pm$ 11.11	42.17 $\pm$ 24.04
GCI (ms)	5.38 $\pm$ 1.52	5.76 $\pm$ 2.14
NLU (ms)	261.26 $\pm$ 144.76	236.83 $\pm$ 129.84
GCC (ms)	0.13 $\pm$ 0.07	0.13 $\pm$ 0.08
PRCP/SIM (ms)	2.35 $\pm$ 2.96	2.35 $\pm$ 2.93
Total (ms)	307.06 $\pm$ 151.58	287.55 $\pm$ 138.81

TABLE I

CORPUS-BASED EXPERIMENTAL RESULTS WITH AND WITHOUT EFFICIENT GRAPH UPDATES

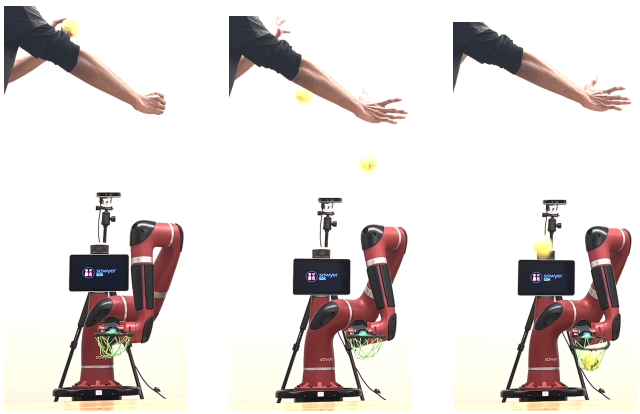
Comparing the average runtimes of the LGP and GCI modules with their EGU counterparts, we see that using EGU resulted in slightly higher runtimes of about 4.54 and 0.38 milliseconds respectively. Since we know that the inference process for the LGP and GCI modules takes place in a single inference step, using EGU should not affect the LGP and GCI average runtimes significantly because the benefit of efficient re-evaluation of features show up at successive re-inference steps. We argue that since the spread of the average runtimes for both the LGP and GCI modules increased by almost two-folds using the EGU algorithm, we can contribute the apparent slight increase in average runtimes of the modules to random noise. Interesting to note that the ratio of average runtimes between the LGP and GCI modules in both cases are approximately equal to the ratio of the number of symbols in the symbol space of their respective DCGs. In other words, there are approximately 8 times more symbols used in inference inside the LGP module compared to the GCI module, hence the average runtime for the LGP module was approximately 8 times longer that the GCI module.

Another useful result from Table I is the average runtimes of the NLU module with and without EGU. Using EGU makes the NLU module 24.42 milliseconds faster overall than without EGU, which is an approximately 10% reduction in execution time. We argue that this is an acceptable reduction in execution time because the world and child phrase dependent features that are being evaluated inside the NLU take up a large portion of the runtime because of their need to iterate over the history of observed states or the simulated states many times over all objects present in the robot's environment. Hence the world and child phrase independent features that EGU does not re-evaluate already contributes to a very small chunk of the average runtime.

C. Physical Robot Experiments

Figure 5 shows three snapshots from a demonstration of the framework described in Section IV. The demonstration shows a human operator releasing two tennis balls on the

table placed in front of the Sawyer Collaborative Robot. The balls were held approximately 1.0m and 1.25m above the table and were about 20cm apart. The robot then receives the text-based instruction "the ball that will hit the table first". This required the robot to run a Kalman filter to estimate the present velocities of the tennis balls and simulate forwards into the future states of the world model. The simulator was set to update the world model at 0.05s intervals, and it took the framework 10 iterations of the NLU to correctly ground the object of interest. Upon inferring that the ball on its left will hit the table first, the robot arm moves underneath the ball to catch it with the net that is attached to its custom-made gripper attachment.



(a) (b) (c)

Fig. 5. A demonstration of the Sawyer Collaborative Robot responding to the statement "the ball that will hit the table first". Upon identifying the ball that is being referred to by the statement, the arm moves to grab the ball with its net attachment.

VII. Conclusion

This paper presents a novel approach to grounding language instructions that refer to the past, present, and/or future state of objects in the world. Building off techniques where previous observations were selectively processed to produce a minimal but sufficient representation of the world for symbol grounding, this approach extends this idea to be able to additionally handle language that refers to the future state of the environment. The efficiency of the DCG models and the EGU technique for providing iterative updates to solutions enables the model to produce solutions fast enough for human-robot communication, which to the authors' knowledge, no other published approach has been able to demonstrate. Experimental results illustrated an example in detail, analyzed performance of the model on a small corpus designed to explore language of interest, and demonstrated this idea for a language-guided task on a physical robot manipulator. 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. 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, S. Patki, J. Rosser, and T.M. Howard. An efficient algorithm for visualization and interpretation of grounded language models. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication*, pages 266–273. IEEE, 2022.

[3] V. Blukis, C. Paxton, D. Fox, A. Garg, and Y. Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *5th Annual Conference on Robot Learning*, 2021.

[4] A. Boularias, F. Duvallet, J. Oh, and A. Stentz. Grounding spatial relations for outdoor robot navigation. In *Proceedings of (ICRA) International Conference on Robotics and Automation*, pages 1976 – 1982, May 2015.

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

[6] 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.

[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] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. 2013.

[10] Y. Mo, H. Zhang, and T. Kong. Towards open-world interactive disambiguation for robotic grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8061–8067. IEEE, 2023.

[11] 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.

[12] 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.

[13] 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.

[14] R. Paul, A. Barbu, S. Felshin, B. Katz, and N. Roy. Temporal grounding graphs for language understanding with accrued visual-linguistic context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4506–4514, 2017.

[15] P. Pramanick, C. Sarkar, S. Paul, R. dev Roychoudhury, and B. Bhowmick. Doro: Disambiguation of referred object for embodied agents. *IEEE Robotics and Automation Letters*, 7(4):10826–10833, 2022.

[16] J. Roh, K. Desingh, A. Farhadi, and D. Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *5th Annual Conference on Robot Learning*, 2021.

[17] 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.

[18] 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.