Language Guided Temporally Adaptive Perception for Efficient Natural Language Grounding in Cluttered Dynamic Worlds

Siddharth Patki, Jacob Arkin, Nikola Raicevic and Thomas M. Howard

***Abstract*— As robots operate alongside humans in shared spaces, such as homes and offices, it is essential to have an effective mechanism for interacting with them. Natural language offers an intuitive interface for communicating with robots, but most of the recent approaches to grounded language understanding reason only in the context of an instantaneous state of the world. Though this allows for interpreting a variety of utterances in the current context of the world, these models fail to interpret utterances which require the knowledge of past dynamics of the world, thereby hindering effective human-robot collaboration in dynamic environments. Constructing a comprehensive model of the world that tracks the dynamics of all objects in the robot's workspace is computationally expensive and difficult to scale with increasingly complex environments. To address this challenge, we propose a learned model of language and perception that facilitates the construction of temporally compact models of dynamic worlds through closed-loop grounding and perception. Our experimental results on the task of grounding referring expressions demonstrate more accurate interpretation of robot instructions in cluttered and dynamic table-top environments without a significant increase in runtime as compared to an open-loop baseline.**

I. Introduction

Robots are transitioning from performing fixed and repetitive tasks in highly structured spaces, such as factory floors, to helping humans in daily tasks in shared workspaces, such as homes and offices. A critical challenge for enabling effective human-robot collaboration in such settings is to have efficient means of communication with these robots. Natural language is a popular modality for this purpose due to its vast domain applicability and ease of use. It provides a bidirectional interface to communicate intent and/or share knowledge about the environment with the robots. Whether it be instructions to follow, information to assimilate or questions to be asked, such interfaces must be able to associate language with a model of the environment to collaborate on tasks in a human-robot team. As such, physically grounded language understanding poses a unique challenge because of the central role of the physical world.

The ability of the robot to collaborate on a wide variety of tasks is inherently linked to the richness of its representation of the world. With few exceptions [1], most of the contemporary approaches to grounded language understanding [2], [3], [4], [5], [6] reason in the context of a rich but instantaneous state of the world. Even though this allows for interpreting a variety of utterances in the present context of the world, these models fail to interpret utterances which require the knowledge of past dynamics of the world. For example, consider a robot instructed to "retrieve the ball inside the box" as shown in Figure 1. To interpret and execute that instruction, the robot only needs to be aware of the present state of the objects in its environment. However, to interpret a command such as "hand me the wrench I was using a few minutes ago", as shown in Figure 1, the robot must have a model of the world that not only represents the current states of objects but also their past dynamics.

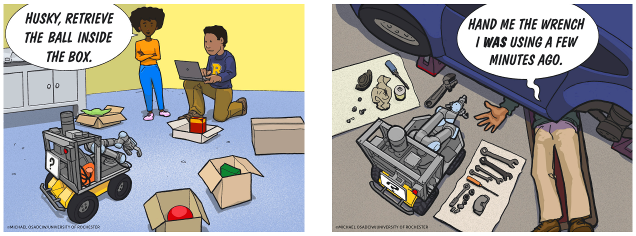

HUSKY, RETRIEVE THE BALL INSIDE THE BOX.
©MICHAEL OSADCIW/UNIVERSITY OF ROCHESTER
HAND ME THE WRENCH I WAS USING A FEW MINUTES AGO.
©MICHAEL OSADCIW/UNIVERSITY OF ROCHESTER

Fig. 1. This figure illustrates two unique human-robot interaction scenarios situated in everyday environments. Most contemporary approaches to language grounding reason in the context of a rich but static model of the world, limiting their ability to interpret language that references past events. Constructing a comprehensive model of the world that tracks states of all objects in robot's workspace is computationally expensive and limits scalability in non-trivial dynamics environments. In this work we provide a novel approach for adaptively constructing compact models of dynamic environments for efficient grounded language understanding in cluttered, dynamic table-top worlds.

Extending contemporary models to reason about utterances which require knowledge of the past dynamics of the world introduces non-trivial challenges pertaining to both state estimation and symbol grounding. Advances in sensor technology, machine perception, and natural language processing have provided access to a wealth of metric and semantic data that can be infused into the world model of a robot. However, constructing a comprehensive model of the world which tracks dynamics and semantics of all of the objects in the robot's workspace through time is computationally expensive and thus prevents effective human-robot collaboration in non-trivial dynamic environments. On the other hand, a poorly detailed model of the environment limits the diversity of the utterances that can be interpreted and executed. A fundamental research question then is, how to efficiently reason over this rich information in a manner that enables robots to efficiently execute a variety of natural language instructions in complex dynamic worlds?

A recent line of work [7], [8], [9] addresses a component of this problem by providing a learned language-guided

*This work was supported by National Science Foundation under grant number 2144804 and Early Career Faculty grant from NASA's Space Technology Research Grants Program. All authors are with Electrical and Computer Engineering Department, University of Rochester, USA {spatki}@ur.rochester.edu

mechanism to adapt a robot's perceptual capabilities in an online manner as and when require by the instructed task. This allows the robot to construct compact, object-centric models of the world that only represent entities relevant to the given task. These compact, object-centric world representations afford faster perception and grounding in cluttered environments. However, these approaches also consider a static and instantaneous state of the world which prevents them from interpreting utterances that require knowledge of the past dynamics of the world. We argue that even with a task-adapted set of detectors, processing the entire observation history is computationally expensive and unnecessary for accurate interpretation of the given instruction. For instructions in which the temporal context may be evident (e.g., "show me the cup that I last used"), visual observations collected before the occurrence of referred event can be filtered from the space of observations. In this work we provide a learned model of language and perception that allows a robot to perform perception and symbol grounding in a closed loop fashion, thereby enabling a lazy, backward search through the space of past observations to construct temporally compact, task-relevant environment models. These environment models are minimal but sufficient for interpreting the meaning of utterances that require knowledge of the present and/or the past states of dynamic environments. With quantitative experimental analysis on the task of grounding referring expressions in cluttered table-top worlds, we demonstrate more accurate interpretation of robot instructions without a significant increase in runtime as compared to an open-loop baseline which performs perception and symbol grounding independently.

II. Related Work

Symbol grounding [10] is the problem of providing meaning to the linguistic constructs such as words and phrases (symbols) by associating (grounding) them with physical entities or processes in the world. Early work in symbol grounding [11], [12] utilized rule-based techniques or manually engineered features that related words to the symbolic representations of entities in the world or the actions that the robot can take. These approaches demonstrated basic capabilities in very simple worlds and with constrained language. For example, the agent SHRLDU [11] operated in a simulated world consisting of cubes, balls, pyramids, and cones etc. Robot SHAKEY [12] used basic algorithms to perceive a constrained set of rectangular objects such as rooms, doors, and hallways. Consequently, these approaches were limited in the diversity of language and environments that they can handle.

Recent developments in representation learning [13] have allowed for the creation of deep neural networks that can address the symbol grounding problem in an end-to-end fashion. These networks can map low-level sensor data (RGB-D images) and language embeddings directly to robot actions and can be trained using reinforcement or imitation learning to reason over the semantics, geometry, and affordances of the environment. Trained entirely in a data-driven fashion, these models have been demonstrated to follow navigation [14] and manipulation [15], [16] instructions in complex environments. With few exceptions, most of these these methods [17], [18], [19], [20] have only been demonstrated in photorealistic simulators assuming noise-free observations, perfect robot localization and static environments. A separate class of models has attempted to combine the advantages of modular and end-to-end policies by fusing object-centric world representations generated by traditional 3D object detection pipelines [21], [22] with large language models [23]. These models have demonstrated successful spatial reference resolution in room sized 3D maps of world [2] and long horizon task following [6] in everyday environments. However, the non-adaptive perception pipelines used in these approaches produce static and flat representations of the world prevents them from interpreting instructions which require knowledge of the world dynamics.

In summary, most of the contemporary approaches to robot instruction following are limited to reasoning in the context of a static and non-adaptive representation of the world. Scaling these systems to operate in more cluttered and dynamic environments causes computational bottlenecks in achieving effective human-robot collaboration. This paper proposes a novel model and system architecture that builds upon [7], [8], [9] to enable robots to efficiently interpret natural language in cluttered dynamic environments.

III. Background

We formulate the problem of grounded language understanding as a probabilistic inference over a learned distribution that associates linguistic elements to their corresponding referents in a symbolic representation of meaning. The set of symbols $\Gamma_t = \{\gamma_1, \ldots, \gamma_n\}$ generally includes concepts derived from the robot's environment model, such as objects, specific regions, spatio-temporal relationships and viable robot behaviors, such as manipulating a specific object or navigating to the desired location, etc. The learned distribution over symbols is conditioned on the constituency parse $\Lambda_t = \{\lambda_1, \ldots, \lambda_n\}$ of the free-form utterance provided by the human collaborator and a corresponding world model $\Upsilon_{1:t}$ that represents the metric and semantic state of the robot's environment till time t. The world model is extracted by processing the history of sensor observations $z_{1:t}$ using a set of various metric and semantic detectors $\Delta = \{\delta_1, \ldots, \delta_n\}$ available in robot's perception pipeline. Framed as a symbol grounding problem [10], natural language understanding then typically follows a maximum a posteriori inference over the space of referent symbols Γ_t.

$$\Gamma_t^* = \arg\max_{\Gamma_t} p(\Gamma_t | \Lambda_t, \Upsilon_{1:t}) \qquad (1)$$

Distributed Correspondance Graphs (DCG) [4] frames this problem as a probabilistic inference over a factor graph having hierarchical structure dictated by the compositional nature of the utterance as illustrated in Figure 2. DCG formulates the problem of language understanding as one of finding the most likely associations between the linguistic

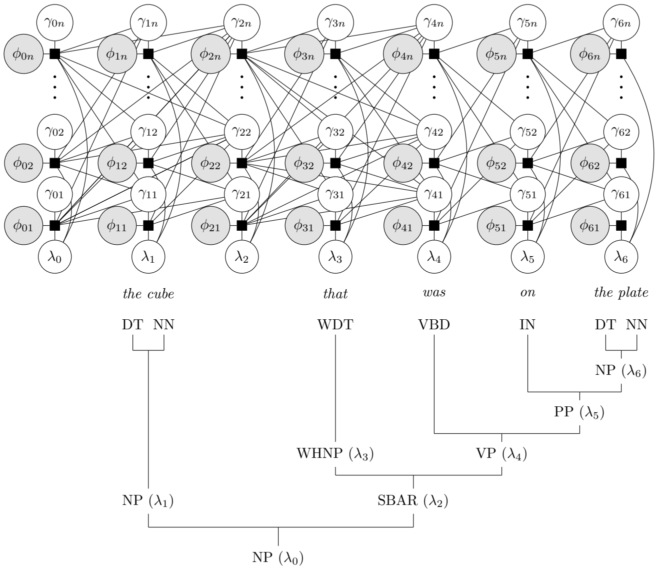


Fig. 2. Distributed Correspondance Graph linked to a constituency parse.

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 $\phi_{ij} \in \Phi$. A correspondence variable ϕ_{ij} associates a phrase λ_i with a symbol γ_{ij}. The hierarchical structure of the graph derived from the parse of an utterance enables the model to reason about the meaning of a particular phrase as conditioned on the grounded meaning of its immediate children phrases Φ_{c_i}. DCG then assumes conditional independence across the linguistic and symbolic constituents to propose a factorization of the grounding distribution.

$$\Phi_t^* = \underset{\phi_{ij} \in \Phi}{\arg\max} \prod_{i=1}^{|\Lambda_t|} \prod_{j=1}^{|\Gamma_t|} p(\phi_{ij} | \gamma_{ij}, \Phi_{c_i}, \lambda_i, \Upsilon_{1:t}) \tag{2}$$

The factorization based on the conditional independence assumption reduces the combinatorially large search over the power set $\mathcal{P}(\Gamma_t)$ to a linear bottom-up search over the individual factors in the DCG making the inference tractable. In practice, the conditional probability over the correspondence variables is approximated by a learned log-linear model [24] $\Psi(\phi_{ij}, \gamma_{ij}, \Phi_{c_i}, \lambda_i, \Upsilon_{1:t})$ consisting of weighted binary features that evaluate various properties of known random variables in a factor. Weight parameters associated with each feature function are optimized by training on a corpus of labeled examples that annotate phrases in a constituency parse with *true* groundings.

IV. Technical Approach

In practice, constructing a world model $\Upsilon_{1:t}$ that tracks the metric-semantic states of all of the entities in the robot's workspace over an extended period of time and performing grounding inference in its context is computational expensive and therefore prohibits effective human-robot collaboration in non-trivial dynamic environments. On the other hand, a poorly detailed and static model of the environment limits the diversity of the utterances that can be interpreted and executed. An interesting research question then is, how to efficiently reason over the rich history of sensory observations $z_{1:t}$ in a manner that enables robots to efficiently understand and execute a variety of natural language instructions in complex dynamic environments while maintaining a reasonable operational tempo. Towards addressing this problem, we argue that there exists a concise model of the world, denoted by $\Upsilon^*_{t\_:t}$, that is sufficient to interpret the meaning of a given natural language instruction and execute it. Specifically, we argue that only a subset of sensory observations and objects in the world are relevant for a given utterance, and the observation history can be selectively processed to build compact task relevant world models. DCG inference performed in the context of such compact world models then takes the following form.

$$\Phi_t^* = \underset{\phi_{ij} \in \Phi}{\arg\max} \prod_{i=1}^{|\Lambda_t|} \prod_{j=1}^{|\Gamma_t|} p(\phi_{ij} | \gamma_{ij}, \Phi_{c_i}, \lambda_i, \Upsilon^*_{t\_:t}) \tag{3}$$

We hypothesize that by constructing a minimal model $\Upsilon^*_{t\_:t}$ of the environment and reasoning in its context (Equation 3), we can achieve greater accuracy in responding to instructions in dynamic, cluttered settings. This should be achievable without significantly increasing the computational burden, unlike the non-adaptive perception pipeline which attempts to build a comprehensive model of the environment by processing the entire observation history $z_{1:t}$.

Central to the problem of constructing compact environment models is the determination of which classifiers from the set Δ_t and which sensor observations from the history $z_{1:t}$ are relevant to the given utterance so that the world models constructed from them are minimal and sufficient for natural language understanding, task planning and motion planning for the given utterance Λ_t. The key observation that informs the ideas explored in this paper is that the difference between the *expected* and *inferred* grounded symbols can inform the system about the adequacy of the world model. For example, consider the utterance "the two cubes that were on the plate". If the set of grounded symbols inferred for this sentence was empty or only included a single cube object, then it could mean that the world model did not contain enough detail about the dynamics of both or at least one of the cubes. In this work we propose a novel intelligence architecture, called Language Guided Temporally Adaptive Perception (LG-TAP) (Figure 3), for natural language understanding, that exploits such information by closing the loop around perception and symbol grounding with the help of a novel learned model called Grounding Constraints Inference. Specifically, this architecture allows for an efficient lazy search through the observation history $z_{1:t}$ by incorporating feedback from the Grounding Constraints Inference (GCI) and Language Guided Perception (LGP) modules. Note that there is an implicit assumption that the error observed between the expected versus inferred symbols is due missing world knowledge and not a failure of the learned grounding model. Meaning, if the world were to contain sufficient information, NLU would infer the expected groundings.

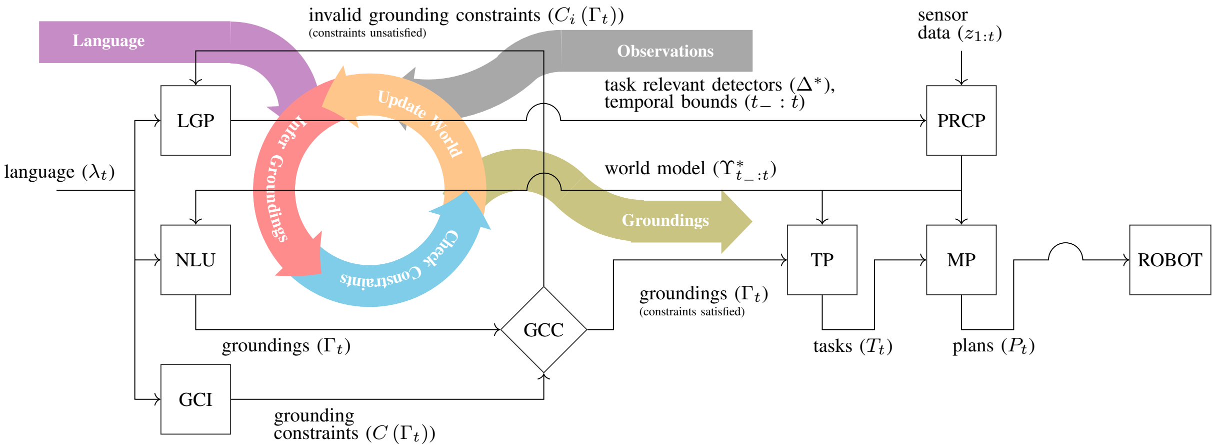


Fig. 3. Language Guided Temporally Adaptive Perception (LG-TAP) - the proposed intelligence architecture that constructs temporally compact world models for instructions that refer to the past and/or present state of the world. This framework leverages the novel Grounding Constraints Inference (GCI) and Groundings Constraints Checker (GCC) modules to perform a lazy, backwards search through space of past observations to construct temporally compact world models. The process of perception and grounded language understanding now exhibits a cyclic behavior that iteratively refines the world model until the grounding constraints are satisfied.

In this architecture, language is used in three ways. First, the utterance Λ_t is processed by the LGP module to infer a set of task relevant detectors Δ^* which will only model the entities in the robot's workspace which are relevant to the given utterance. Initially the perception pipeline assumes access to only a brief history of observations $z_{t_-:t}$ collected around the utterance time t. An environment model between these temporal bounds $\Upsilon^*_{t_-:t}$, is constructed by the Perception Module (PRCP) by engaging the task relevant detectors Δ^* and is consumed by a DCG-based Natural Language Understanding (NLU) module to infer a set of groundings (symbols representing meaning) Γ_t associated with language. The novel component called Grounding Constraints Inference also consumes the language Λ_t to infer a set of grounding constraints $C\left(\Gamma_t\right)$ that represents rules regarding the expected groundings Γ_t from the NLU module for the same utterance. For example, the grounding constraints inferred for the sentence “hand me the wrench I was using sometime ago” will represent a rule $C\left(\Gamma_t\right) = \{\ \exists!\ \gamma_{a_i}^{(o_i,o_j)} \in \Gamma_t \mid \mathit{type}(o_i) = \mathit{robot}, \mathit{type}(o_j) = \mathit{wrench}, \mathit{type}(a_i) = \mathit{pick} \wedge \exists!\ \gamma_{a_i}^{(o_i,o_j)} \in \Gamma_t \mid \mathit{type}(o_i) = \mathit{wrench}, \mathit{type}(o_j) = \mathit{person}, \mathit{type}(a_i) = \mathit{pass} \wedge |\Gamma_t| = 2\ \}$. Similarly, grounding constraints inferred for the sentence “the cube that was on your right” will represent a rule $C\left(\Gamma_t\right) = \{\ \exists!\ \gamma_{o_i} \in \Gamma_t \mid \mathit{type}(o_i) = \mathit{cube}\ |\Gamma_t| = 1\ \}$. Here o_i, o_j denote objects in the world, and a_i represents the action type expected to be inferred by NLU. It is important to note that the constraints are inferred independently of the world model and solely represent ungrounded semantics of the utterance.

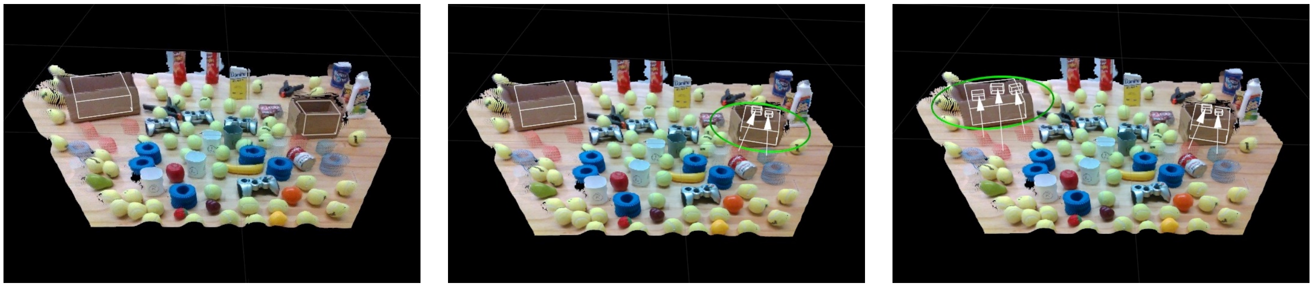
These constraints are then validated by the Grounding Constraints Checker (GCC) to determine if the inferred groundings Γ_t align with the model's expectation for the given the utterance. If the grounding constraints $C\left(\Gamma_t\right)$ are satisfied, the groundings are passed onto the task planning (TP) component of the architecture to execute the instructed task. However, if the groundings don't satisfy the grounding constraints $C\left(\Gamma_t\right)$ then the unsatisfied grounding constraints are passed back to the LGP module to expand the temporal bounds $(t_- : t)$ to generate an updated environment model $\Upsilon^*_{t_-:t}$. This updated environment model is then processed by the NLU module to infer an updated set of groundings that may satisfy the grounding constraints. As shown in Figure 4, this loop continues until either the constraints are satisfied or the robot runs out of the observation history. The LGP, GCI and NLU modules are implemented using DCGs, each trained separately with specialized symbolic representations. Performing DCG inference repeatedly during the backwards search is computationally expensive for any non-trivial world. We leverage the efficient graph updates (EGU) technique [25] for performing symbol grounding inference with high runtime efficiency during the backwards search in our architecture. This technique selectively recomputes only the world-dependent features at the factor level in DCG, and reuses the prior computational effort to speed-up the runtime of factor graph evaluation and inference. We hypothesize that the proposed intelligence architecture will perform an efficient backwards search through the space of past observations by closing the loop between perception and symbol grounding. This effectively will enable collaborative robots to efficiently interpret and execute diverse manipulation instructions in heavily cluttered dynamic environments.

V. Experimental Design

To understand the impact of the proposed closed loop architecture, we performed a quantitative evaluation comparing its runtime performance and accuracy to an open loop baseline on the task of grounding referring expressions in dynamic table-top worlds.

A. Symbolic Representations and Model Training

All of the language models (NLU, LGP and GCI) used in the proposed architecture were implemented using Dis-



(a) World Model $\Upsilon^*_{t-0:t}$ (b) World Model $\Upsilon^*_{t-50:t}$ (c) World Model $\Upsilon^*_{t-80:t}$

Fig. 4. In a scenario of human-robot collaboration, the human participant selects three cubes from the tabletop and transfers them to one of the two boxes available on the tabletop. Subsequently, the human selects two more cubes and puts them into the other box, and instructs the robot with “point to the box that contains two cubes” and “show me the box that has three rubber cubes”. Figure 4(a) illustrates the world model at the time t which is insufficient to comprehend the intended meaning. LG-TAP constructs temporally compact world models for the two instructions as illustrated in Figures 4(b) and 4(c) respectively. The arrows indicate the displacement vectors for the recognized cubes, and the green circles indicate the referenced boxes in the instructions. As the proposed method executes perception and grounding in a closed loop, it generates a more concise world model for the task of ”point to the box with two cubes” than for ”show me the box with three rubber cubes” because of the chronological order of those two events.

tributed Correspondence Graphs [4] with symbolic representations and features adapted according to their individual functionalities. The symbolic representation used by the NLU model consisted of symbols that represented actions, objects and various spatial-temporal relations needed to represent the meaning of linguistic phrases in a given utterance. In contrast, the representation used by the LGP model consisted of symbols that represented various object detectors in robot’s perception pipeline (PRCP). The symbolic representation for the GCI model consisted of grounding constraints as described in Section IV. The three language models were trained using three different corpora consisting of about 1000 utterances each. Each of these training sets contained same utterances but were annotated differently according to the relevant symbolic representation. This training data was generated entirely in the context of simulated table-top worlds using the Gazebo [26] simulator and was self-annotated by following patterns in [4]. We encourage the reader to refer to [4], [7], [25] if a more general understanding of the symbolic representation, feature design and model optimization for DCGs is desired.

B. Workspace Setup and Model Testing

For testing, the workspace of the robot was populated with different objects, such as cups, hammers, wrenches, balls etc. taken from either the YCB dataset [27] or found otherwise in everyday spaces. In order to assess how the proposed approach scales with an increase in the number of objects in the world, we generated 8 unique test worlds by systematically incrementing the number of objects on the table from 10 to 80 with a step size of 10. To introduce dynamics, the objects in each of these 8 worlds were moved around on the table over a period of 3 minutes by a human collaborator. A total of 96 referring expressions (12 per world) were generated by describing the objects with their present or past metric-semantic states. For example, a cube that was moved from the left side of the robot to right was referred to as, “the cube that was moved” or “the cube that was on your left”. The referring expressions included references to single objects or groups of objects such as “the two wrenches that were used” etc. This set of utterances was used as a test set to evaluate the performance of the proposed approach on the task of grounding referring expressions.

C. Perception Pipeline and World Model

The object centric world model used in our system was capable of representing each object in the robot’s workspace, including the robot itself, with a unique ID, a semantic type, a color, and temporal state information such as its 3D position, bounding box, and linear velocity tracked over time. This model was constructed by processing RGB-D images captured by an Intel RealSense D435 sensor mounted on Baxter’s torso. A YOLOv4 [28] object detector was run on the visual stream, generating 2D object detections for each RGB-D frame. This detector was trained on 30 custom object categories using a dataset of 150K annotated images. The open-loop baseline approach utilized a non-adaptive perception pipeline that converted all 2D object detections inferred by YOLOv4 to 3D point clouds and tracked 3D poses and bounding boxes of these objects over time in an online manner. In contrast, our proposed lazy approach used an adaptive perception pipeline that could adjust its detectors, as indicated by the Language Guided Perception model in the architecture, to selectively convert only the task relevant 2D detections into 3D point clouds and track their metric states after having received an instruction. Object tracking was performed by using the Hungarian algorithm [29], and a Kalman filter was used to reliably update the state information for each tracked object over time.

D. Performance Metrics

To evaluate the performance of the proposed approach (LG-TAP), we conducted a quantitative assessment comparing its runtime and accuracy to an open loop (Exhaustive) system on the task of referring expression grounding in dynamic table-top worlds. The task of referring expression grounding involves identifying a particular object or set of objects referred to in a given utterance. Ability to quickly and accurately interpret such expressions in non-trivial dynamic worlds is critical for achieving effective human-robot collaboration. In a system that uses object centric world representations, such as ours, the computational challenges involved in perception and symbol grounding are inherently linked to the number of objects in robot's world. Therefore, to understand how the runtime and accuracy of LG-TAP scales with increase in the number of objects in robot's world, we quantified following relationships. We also quantified the compactness of the world models generated by LG-TAP in terms of their size and time spans.

- Response Time Vs. World Object Count.
- Response Accuracy Vs. World Object Count.
- Perception Cycle Frequency Vs. World Object Count.
- World Model Object Count Vs. World Object Count.
- World Model Time Span Vs. World Object Count.

First, we measured how long both approaches take to ground the referring expression. We defined this measure as the response time of the architecture and we quantified how it scales with an increase in the number of objects in the world. Second, we measured how accurately do both of the approaches ground the referring expression. We defined this measure as the response accuracy of the architecture and we quantified how it changes with increase in the number of objects in the world. Third, we measured how the rate of perception (FPS) changes with an increase in the number of objects on the table. A higher perception cycle frequency would indicate that the system can perceive changes in the world more quickly, and thus be more robust to object tracking failures. Lastly, we evaluated the degree of compactness of the world models produced by LG-TAP. This was measured by quantifying the size of the world model in terms of the number of objects it represents, and the time-span of it based on the average duration of state histories of objects modeled in it. We expect to observe a high degree of compactness in the world models produced by LG-TAP, as they contain only the necessary information to interpret and execute a given natural language instruction.

The response time of LG-TAP, which performs a backward search through past observations, depends on the depth of the search required to find the necessary event information. To investigate the impact of event timing on LG-TAP's response time, we divided the test set containing 96 referring expressions into 4 different groups based on the event lag (EL). The EL categories were defined as $EL = 0\ min, EL = 1\ min, EL = 2\ min$ and $EL = 3\ min$ and corresponded to events occurring at different times before the utterance was received. For instance, a referring expression belonging to the $EL = 1\ min$ group referred to an event that occurred roughly one minute before the utterance. By analyzing LG-TAP's response times across these EL categories, we were able to gain insights into the relationship between event timing and system performance.

E. Ablation Study

The process of perception and grounded language understanding in LG-TAP exhibits a cyclic behavior that iteratively refines the world model until the grounding constraints are satisfied. To efficiently execute this iterative process, LG-TAP leverages Language Guided Perception (LGP) and Efficient Graph Updates (EGU) [25] techniques as described in the technical approach section of this paper. To quantify the individual contributions of these two techniques towards runtime improvements, we ran an ablation study.

VI. Results and Discussion

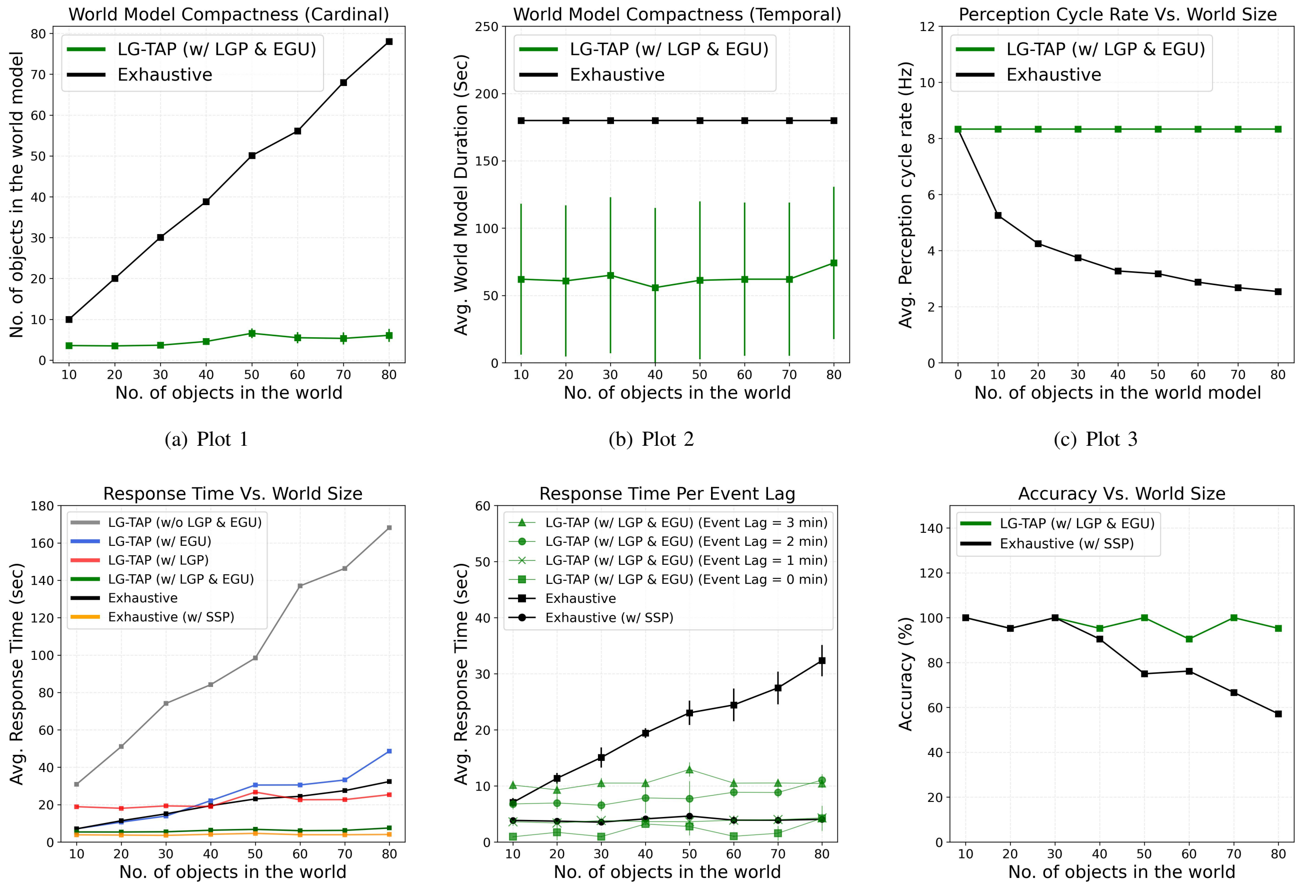
This section presents experimental findings, comparing the performance of proposed closed-loop (LG-TAP) approach to an open-loop (Exhaustive) baseline.

A. Impact on Compactness

The results depicted in Figure 5(a) and Figure 5(b) clearly illustrate that LG-TAP constructs considerably more compact world models as compared to the Exhaustive baseline. First, the compactness in the size of the world model (number of objects) can be attributed to the Language Guided Perception (LGP) model in the architecture. This is because, LGP adapts the detectors in robot's perception pipeline by leveraging the information in the utterance to model only the task relevant entities in robot's world. In contrast, the Exhaustive baseline attempts to detect all of the objects and thus shows nearly perfect correlation between the number of objects reflected in world model versus present in the actual world. Another factor contributing to the compactness of the world models generated by LG-TAP is the iterative nature of the approach, facilitated by the Grounding Constraints Inference (GCI) and Grounding Constraints Checker (GCC) modules. LG-TAP is designed to construct temporally compact world models that contain only the necessary information about the world dynamics to interpret a given instruction. As such, the high variance in the time-span of world models generated by LG-TAP is a desired and expected outcome. This ensures that the models contain only the minimal temporal information required for accurate interpretation of the instruction.

B. Impact on Runtime

Figure 5(d) presents the results of an ablation study that evaluates the response time of our proposed approach. In this study, response time is defined as the total elapsed time between receiving an utterance and successfully inferring its meaning. For efficient perception and grounding, LG-TAP leverages LGP and EGU [25] techniques as described in the technical approach section of this paper. The results show that LG-TAP with LGP and EGU (indicated in green) outperforms the Exhaustive baseline (indicated in black) in terms of response time. In contrast, when both LGP and EGU are omitted from the architecture (indicated in gray), the response time of LG-TAP significantly increases, highlighting the importance of these techniques. The plots drawn in red and blue offer insights into the individual contributions of LGP and EGU in the architecture. Notably, LGP enables the response time to be independent of the number of objects in the world. This is because LGP adapts perception to model only the task-relevant entities, making



(a) Plot 1

(b) Plot 2

(c) Plot 3

(d) Plot 4

(e) Plot 5

(f) Plot 6

Fig. 5. Plots demonstrating the effective scaling of proposed closed-loop approach (indicated in green) when compared against an open-loop baseline (indicated in black) on the task of referring expression grounding. The data points indicate mean values, and the error bars indicate 95% confidence intervals. We can observe that the response time of the proposed approach is closely comparable to the best case baseline (exhaustive with SSP), whereas it is significantly more accurate in worlds with more number of objects.

it nearly independent of the total number of objects in the world.

On the other hand, the Exhaustive approach generates a detailed world model that includes all objects in robot's workspace and performs grounding inference in its context. Since the runtime of the DCG (Equation 2) is linear with respect to the number of objects in the robot's world model, we observe a consistent increase in response time for the Exhaustive baseline. A more efficient version of the Exhaustive baseline (indicated in orange), employs Search Space Pruning (SSP) to reduce the response time of baseline significantly. SSP and LGP have similar effects on the size of the search space; however, LGP reduces it by filtering objects during perception, whereas SSP does so by filtering objects after they have been perceived. Although this method considerably improves the runtime of the Exhaustive baseline, its accuracy declines similarly to that of the exhaustive approach, as explained in the following section.

Since the proposed approach employs a backward, lazy search through past observations to recursively estimate a world model and grounded symbols, its runtime is dependent on the depth it must traverse to resolve the uttered reference. This implies that it takes longer to resolve references that occurred in the distant past and shorter time to resolve references that are more recent. Figure 5(e) illustrates this effect. As seen in plots, the runtime for LG-TAP increases approximately linearly with increase in the delay between the utterance and event time. For example, the response time to the utterance “the cube that was moved” would differ, depending on when the cube was moved in past.

C. Impact on Accuracy

The results presented in Figure 5(f) indicate that the proposed approach achieves higher accuracy than the baseline in grounding referring expressions in sufficiently cluttered dynamic environments. To better understand this performance gain, we also evaluated how the frequency of perception in both the baseline and proposed approaches change as the number of objects in the environment are increased. As depicted in Figure 5(c), the frequency of perception cycles in the baseline approach drops with increasing number of objects, whereas our proposed approach maintains a constant frequency. The baseline approach utilizes a non-adaptive perception pipeline that converts all 2D object detections into 3D object point clouds and tracks their 3D pose and bounding box state over time in an online manner. However, the runtime of this pipeline increases with the number of objects in the environment, which lowers the rate at which it can successfully process RGB-D frames streaming from the sensor. As a result, the baseline suffers from poor object tracking performance in highly cluttered dynamic environments, which leads to noisy world models and inaccurate symbol grounding. In contrast, LG-TAP does not process visual observations until after an utterance is received. This allows it to store the observation history at a consistent frame rate. Later, it employs an adaptive perception pipeline that can adjust its detectors to selectively convert only task-relevant 2D detections into 3D point clouds and track their metric states opportunistically after receiving an instruction. This approach enables LG-TAP to perform 3D object state estimation at a constant rate, regardless of the number of objects in the environment. This results in maintaining high accuracy in sufficiently cluttered dynamic environments. Overall, the proposed approach outperforms the baseline in accurately grounding referring expressions in dynamic settings while maintaining a comparable runtime.

VII. Conclusion

In this work, we proposed a solution for improving the efficiency of robot instruction understanding in dynamic spaces. Our approach leverages closed-loop grounding and perception to construct temporally compact models of dynamic worlds. Experimental results on the task of grounding referring expressions demonstrate that our approach leads to more accurate interpretation of robot instructions in cluttered and dynamic table-top environments without a significant increase in runtime compared to an open-loop baseline. In conclusion, our work provides a promising approach for improving human-robot teaming in shared, dynamic spaces.

References

[1] 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*, 2017, pp. 4506–4514.

[2] 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.

[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] 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*, vol. 37, no. 10, pp. 1269–1299, June 2018.

[5] V. Blukis, Y. Terme, E. Niklasson, R. A. Knepper, and Y. Artzi, "Learning to map natural language instructions to physical quadcopter control using simulated flight," in *Conference on Robot Learning*. PMLR, 2020, pp. 1415–1438.

[6] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *6th Annual Conference on Robot Learning*, 2022.

[7] 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.

[8] S. Patki, A. Daniele, M. Walter, and T. Howard, "Inferring compact representations for efficient natural language understanding of robot instructions," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2019.

[9] S. Patki, E. Fahnestock, T. M. Howard, and M. R. Walter, "Language-guided semantic mapping and mobile manipulation in partially observable environments," in *Conference on Robot Learning*. PMLR, 2020, pp. 1201–1210.

[10] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

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

[12] N. J. Nilsson, "Shakey the robot," SRI INTERNATIONAL MENLO PARK CA, Tech. Rep., 1984.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[14] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Conference on Robot Learning (CoRL)*, 2020.

[15] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[16] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[17] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.

[18] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[19] X. Wang, W. Xiong, H. Wang, and W. Y. Wang, "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–53.

[20] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," *arXiv preprint arXiv:1904.04195*, 2019.

[21] Q. C. Ruizhongtai, S. Hao, M. Kaichun, and G. L. J, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[22] Q. C. Ruizhongtai, Y. Li, S. Hao, and G. L. J, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

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

[25] J. Arkin, S. Patki, J. D. Rosser, and T. M. Howard, "Efficient graph updates for natural language understanding in dynamic environments," in *31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022.

[26] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[27] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[29] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.