Verifiably Following Complex Robot Instructions with Foundation Models

Benedict Quartey* **Eric Rosen*** **Stefanie Tellex** **George Konidaris**

Department of Computer Science, Brown University

Project Website

https://rb.gy/94unhv

Abstract. Enabling mobile robots to follow complex natural language instructions is an important yet challenging problem. People want to flexibly express constraints, refer to arbitrary landmarks and verify behavior when instructing robots. Conversely, robots must disambiguate human instructions into specifications and ground instruction referents in the real world. We propose Language Instruction grounding for Motion Planning (LIMP), an approach that enables robots to verifiably follow expressive and complex open-ended instructions in real-world environments without prebuilt semantic maps. LIMP constructs a symbolic instruction representation that reveals the robot's alignment with an instructor's intended motives and affords the synthesis of robot behaviors that are correct-byconstruction. We perform a large scale evaluation and demonstrate our approach on 150 instructions in five real-world environments showing the generality of our approach and the ease of deployment in novel unstructured domains. In our experiments, LIMP performs comparably with state-of-the-art LLM task planners and LLM code-writing planners on standard open vocabulary tasks and additionally achieves 79% success rate on complex spatiotemporal instructions while LLM and Code-writing planners both achieve 38%. See supplementary materials and demo videos at our project website.

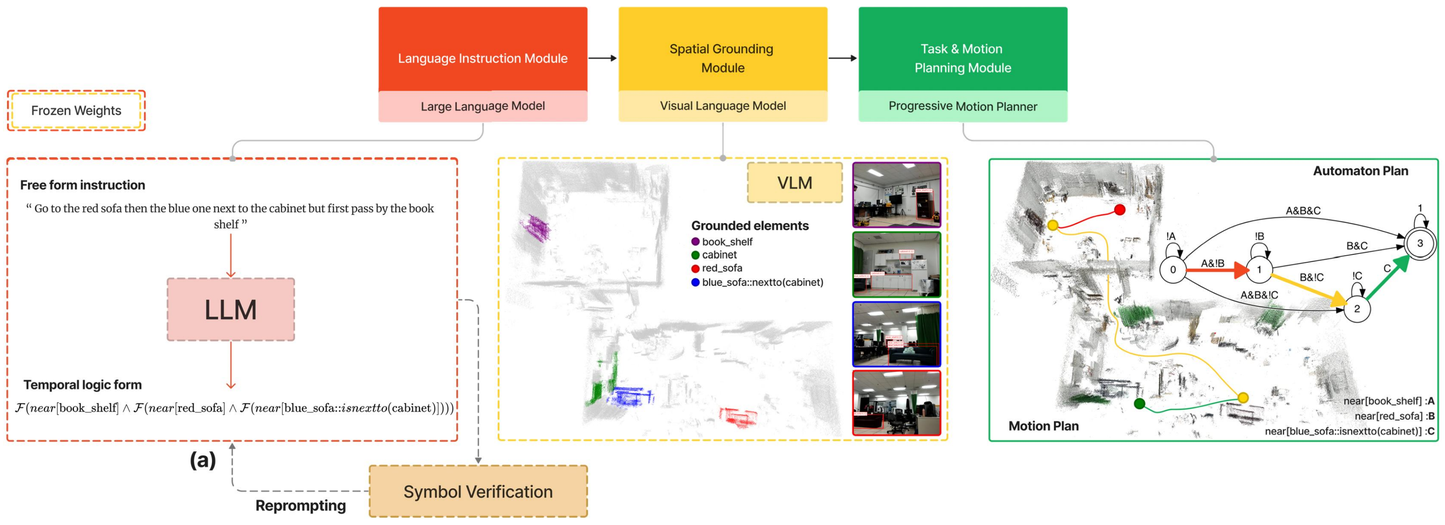
Motivation

How do we get robots to verifiably follow complex open-ended instructions without prebuilt semantic maps ?

Proposal. Combine the generality of foundation models with the verifiability and explainability of temporal logics to generate instruction conditioned semantic maps that affords constraint satisfying task and motion planning.

Desirable Properties. 1) General purpose open vocabulary instruction following, 2) Explainable instruction representation, 3) Verifiably correct behavior synthesis

Approach



Running Example

Language Instruction Module

Input Instruction : " Bring the green plush toy to the whiteboard in front of it, watch out for the robot in front of the toy"

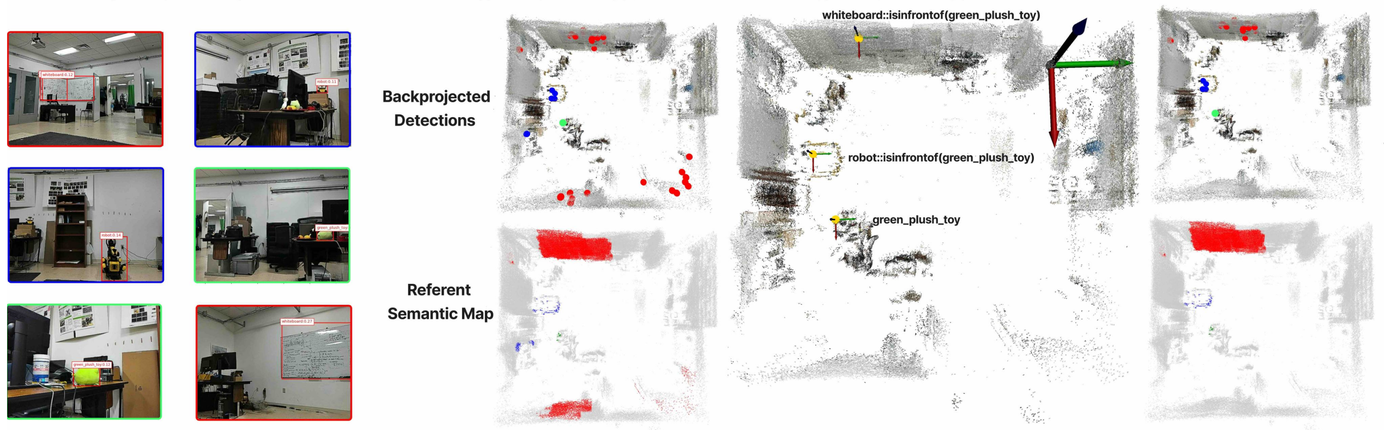
Conventional LTL : $\mathcal{F}(\text{green\_plush\_toy} \wedge \mathcal{F}(\text{whiteboard} \wedge \neg\text{robot}))$

Our LTL Syntax : $\mathcal{F}(A \wedge \mathcal{F}(B \wedge \mathcal{F}(C \wedge \neg D \wedge \mathcal{F}E)))$

```
A: near[green_plush_toy]
B: pick[green_plush_toy]
C: near[whiteboard::isinfrontof(green_plush_toy)]
D: near[robot::isinfrontof(green_plush_toy)]
E: release[green_plush_toy, whiteboard::infrontof(green_plush_toy)]
```

Spatial Grounding Module

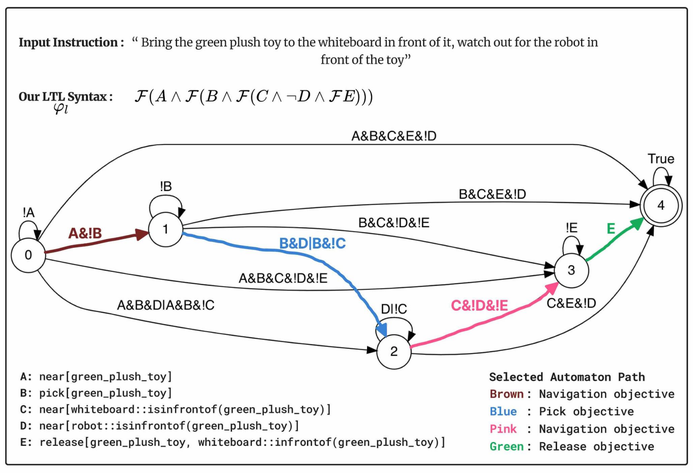
```
Referents: green_plush_toy; whiteboard::isinfrontof(green_plush_toy); robot::isinfrontof(green_plush_toy)
```



Task and Motion Planning Module

Input Instruction : " Bring the green plush toy to the whiteboard in front of it, watch out for the robot in front of the toy"

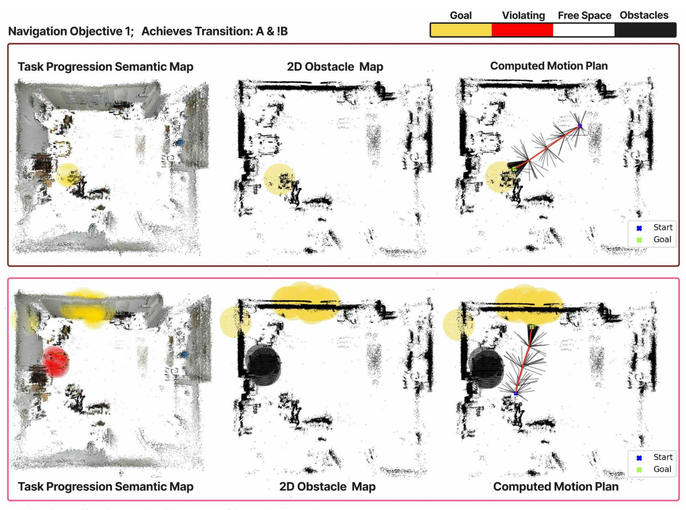
Our LTL Syntax : $\mathcal{F}(A \wedge \mathcal{F}(B \wedge \mathcal{F}(C \wedge \neg D \wedge \mathcal{F}E)))$




```
A: near[green_plush_toy]
B: pick[green_plush_toy]
C: near[whiteboard::isinfrontof(green_plush_toy)]
D: near[robot::isinfrontof(green_plush_toy)]
E: release[green_plush_toy, whiteboard::infrontof(green_plush_toy)]
```

Selected Automaton Path
Brown : Navigation objective
Blue : Pick objective
Pink : Navigation objective
Green : Release objective

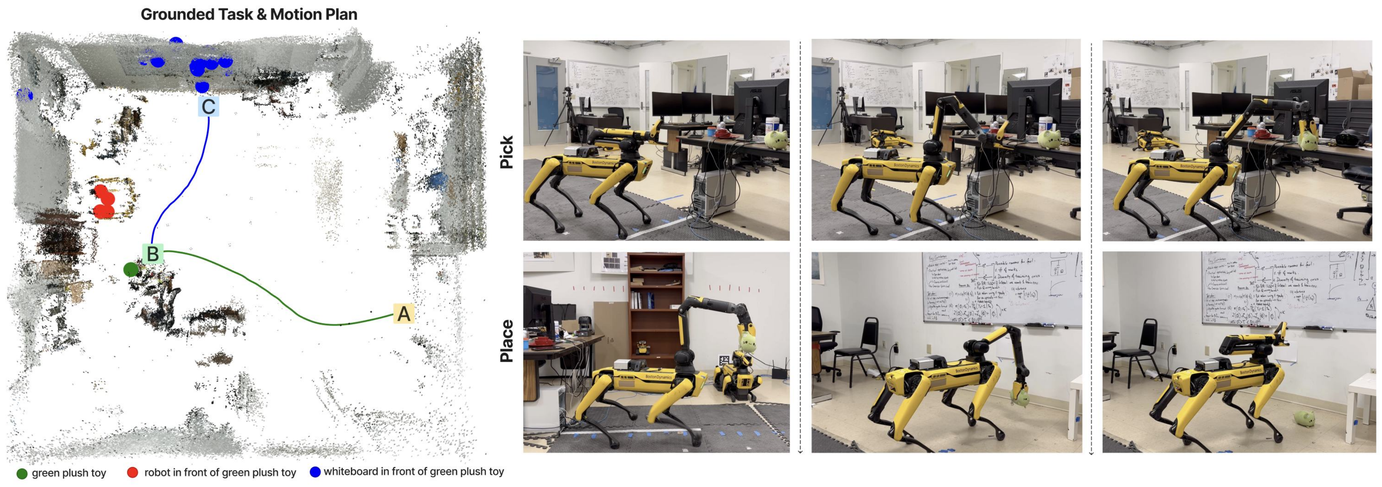
(a)



(b)

Robot Demonstration

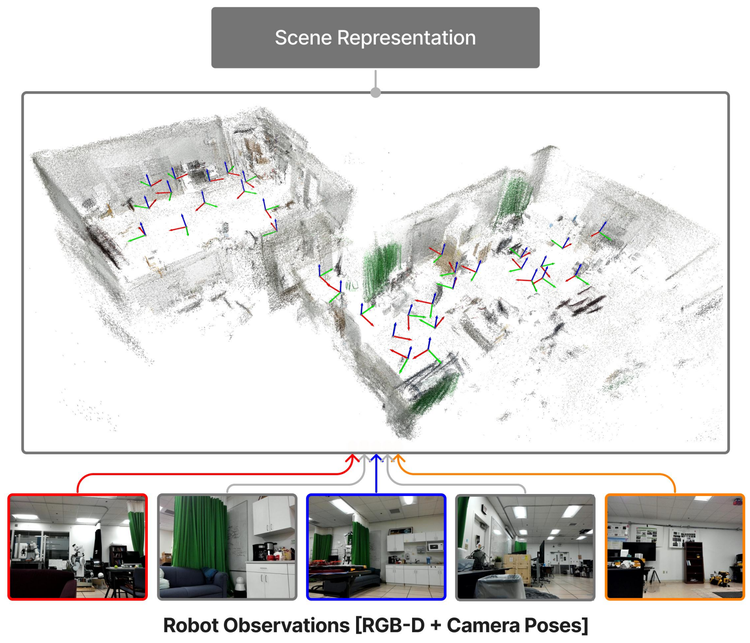
Grounded Task & Motion Plan



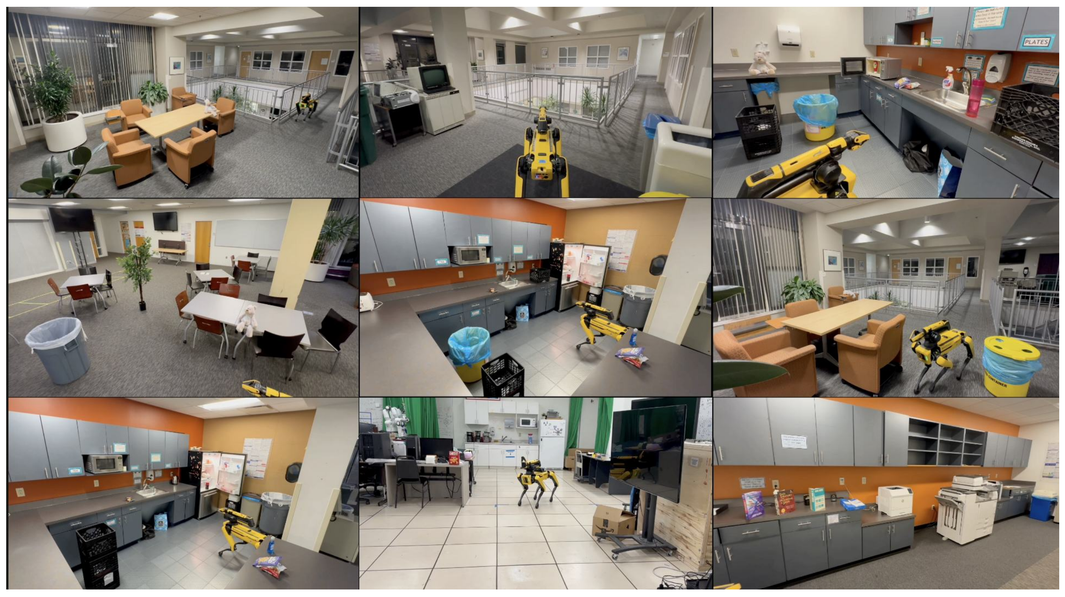
Language Instruction Grounding for Motion Planning (LIMP)

LIMP. Leverages foundation models and temporal logics to dynamically generate instruction-conditioned semantic maps that enable robots to construct verifiable controllers for following navigation and mobile manipulation instructions with open vocabulary referents and complex spatiotemporal constraints. Process:

- Construct 3D representation of an environment via SLAM (< 10 minutes)
- Leverage LLMs to translate arbitrary natural language instructions into linear temporal logic (LTL) specifications with a novel composable syntax that enables referent disambiguation.
- Instruction referents are detected and grounded via VLMs and spatial reasoning
- Dynamically generate novel Task Progression Semantic Maps to localize regions of interest and progressively synthesize constraint-satisfying motion plans.



Real-world evaluation on 150 instructions in multiple environments



Sample Qualitative Results