Preprint

Verifiably Following Complex Robot Instructions with Foundation Models

Benedict Quartey*
benedict_quartey@brown.edu
benedictquartey.com

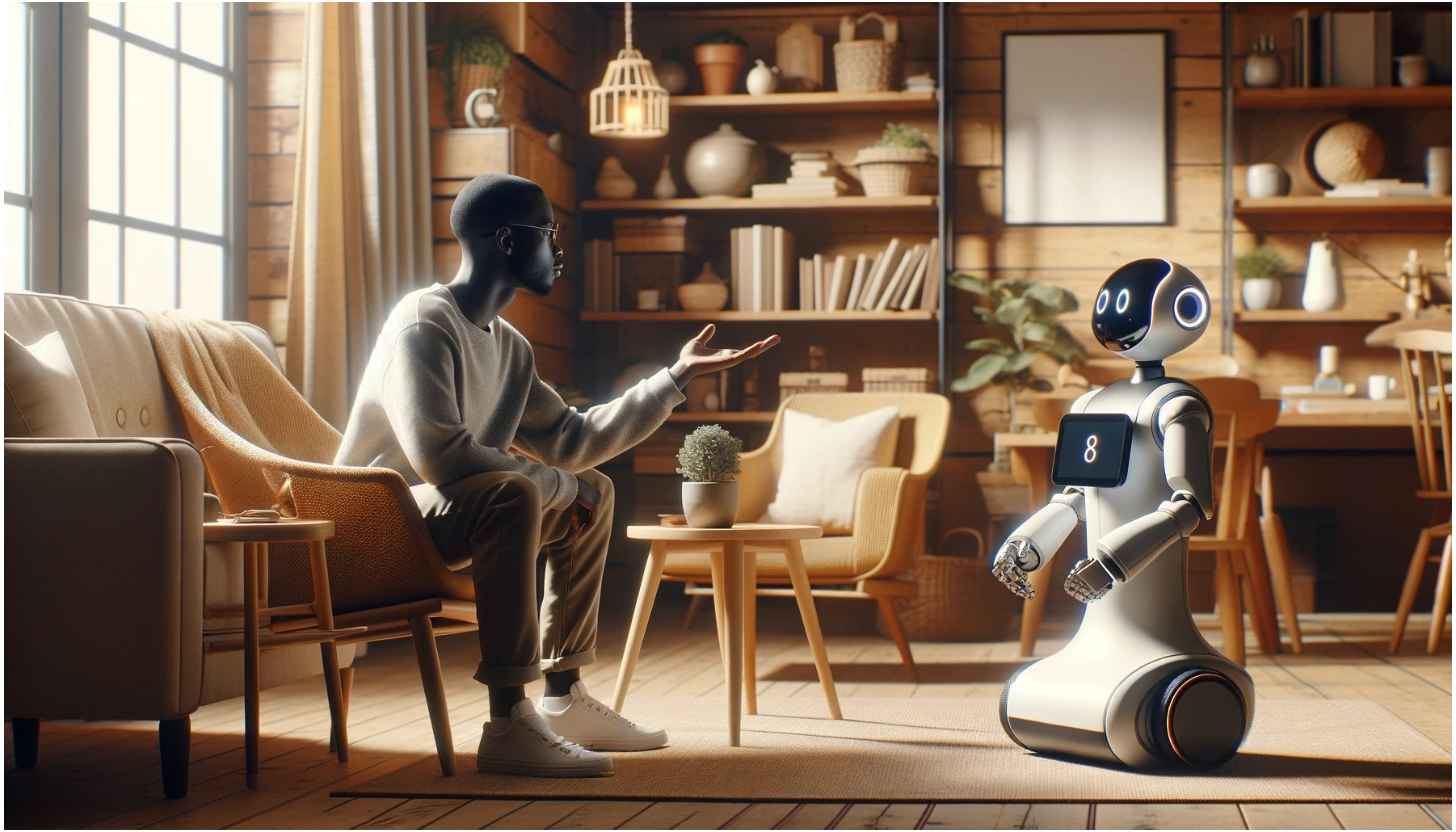
With
Eric Rosen*, Stefanie Tellex and George Konidaris

Feb 23 2024

https://robotlimp.github.io

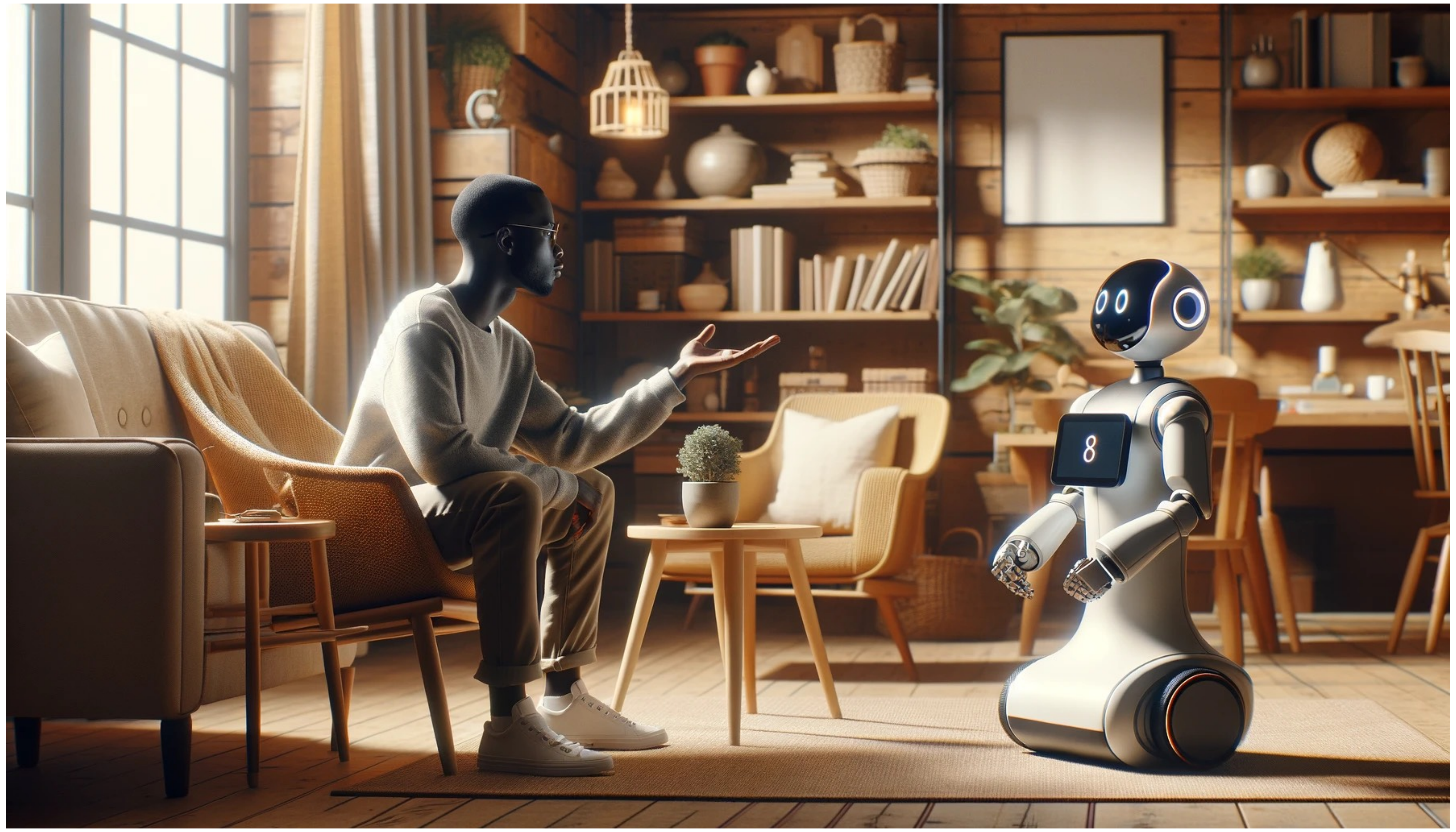
* Equal Contribution

Motivation



Go to the kitchen while avoiding the orange table and bring me the book between the microwave and sink.

Motivation



Navigation and manipulation

Ground arbitrary referents

Referent disambiguation

Behavior Verification

Go to the kitchen while avoiding the orange table and bring me the book between the microwave and sink.

Language Instruction grounding for Motion Planning (LIMP)

* Construct 3D representation of an environment via SLAM.
* Leverage LLMs to translate complex natural language instructions into linear temporal logic specifications with a novel composable syntax that enables referent disambiguation.
* Instruction referents are detected and grounded via VLMs and spatial reasoning
* Dynamically generate semantic maps to localize regions of interest and progressively synthesize constraint-satisfying motion plans to achieve the subgoals required to satisfy the instruction

Problem Definition

Objective

Given a natural language instruction, our goal is to synthesize navigation and manipulation skills to produce a policy that faithfully satisfies the constraints of the instruction.

Assumptions

Unlike previous works we do not assume access to a prebuilt semantic map with locations of objects or predicates prespecified.
However we assume:

* Access to a robot equipped an RGBD camera
* Access to a task agnostic visual language model
* Access to an auto-regressive large language model

Navigation

Object goal oriented path planning problems in continuous space. Generate paths to goal set while staying in feasible regions and avoiding infeasible regions.

Manipulation

Object parameterized options. Initiation set, policy and termination condition are functions of robot pose and an object parameter θ

$$o_\theta = (I_\theta, \pi_\theta, \beta_\theta)$$

Linear Temporal Logic (LTL)

LTL presents an expressive grammar for specifying temporal behavior. Formulas are composed of atomic propositions, logical connectives and temporal operators.

Logical Connectives		Temporal Operators	
Conjunction	$\wedge$	Next	$\mathcal{X}$
Negation	$\neg$	Until	$\mathbf{U}$
Disjunction	$\vee$	Globally/Always	$\mathcal{G}$
Implication	$\rightarrow$	Finally/Eventually	$\mathcal{F}$

"Go to the kitchen then the fridge"

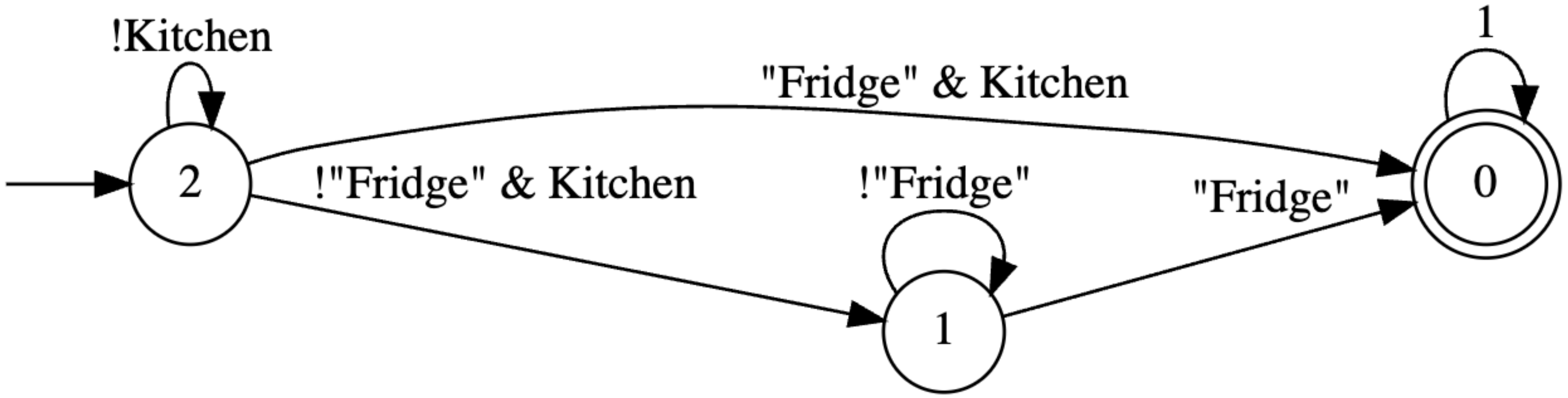
$$\mathcal{F}(\text{Kitchen} \wedge \mathcal{F}(\text{Fridge}))$$

Pnueli, Amir. "The temporal logic of programs." *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. ieee, 1977.

Linear Temporal Logic (LTL)

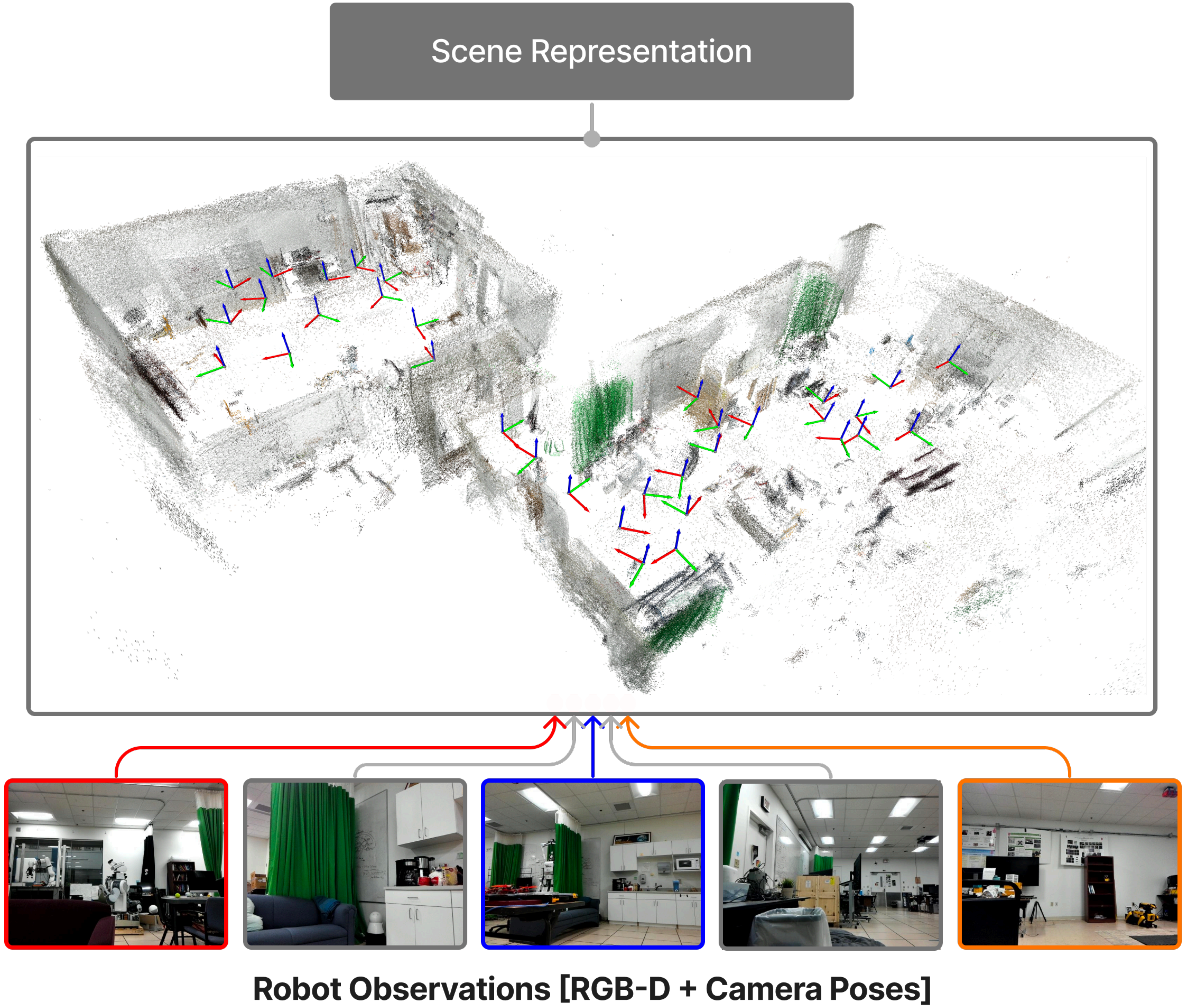
"Go to the kitchen then the fridge"

$$\mathcal{F}(\text{Kitchen} \wedge \mathcal{F}(\text{Fridge}))$$

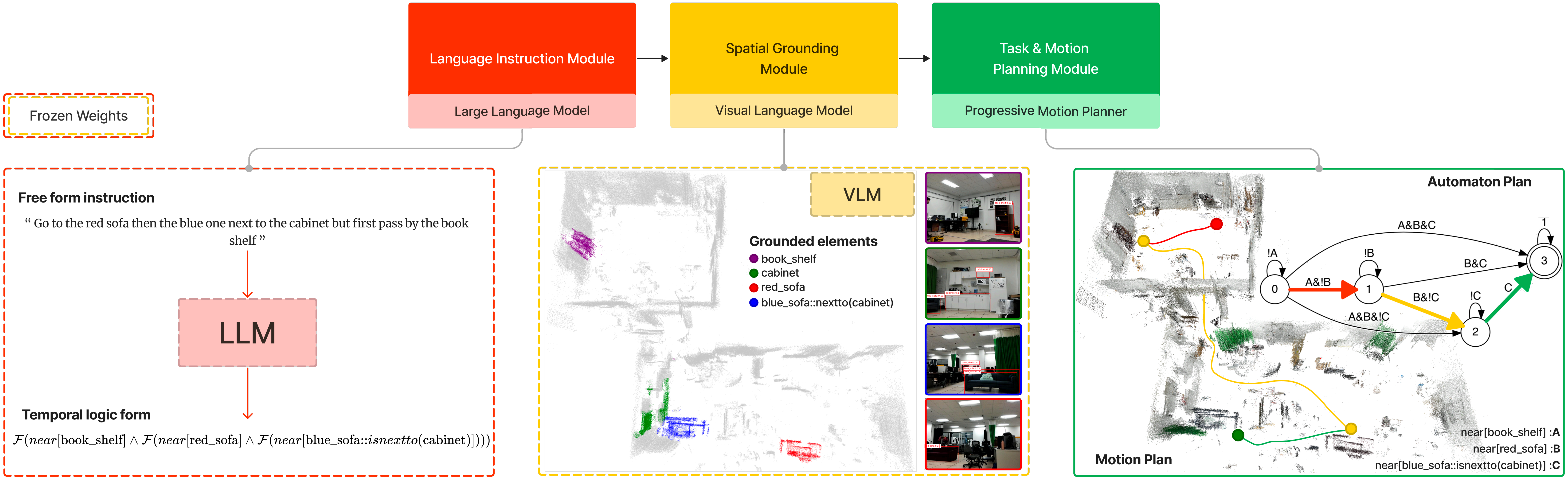


Pnueli, Amir. "The temporal logic of programs." *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. ieee, 1977.

Approach


Scene Representation
Robot Observations [RGB-D + Camera Poses]

Approach



Language Instruction Module
Large Language Model
Spatial Grounding Module
Visual Language Model
Task & Motion Planning Module
Progressive Motion Planner
Frozen Weights
Free form instruction
“ Go to the red sofa then the blue one next to the cabinet but first pass by the book shelf ”
LLM
Temporal logic form
$\mathcal{F}(near[\text{book\_shelf}] \wedge \mathcal{F}(near[\text{red\_sofa}] \wedge \mathcal{F}(near[\text{blue\_sofa}::isnextto(\text{cabinet})])))$
VLM
Grounded elements
book_shelf
cabinet
red_sofa
blue_sofa::nextto(cabinet)
Automaton Plan
A&B&C
1
!A
!B
B&C
0
A&!B
1
3
B&!C
!C
C
A&B&!C
2
near[book_shelf] :A
near[red_sofa] :B
near[blue_sofa::isnextto(cabinet)] :C
Motion Plan

Language Instruction Module

Large Language Model

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

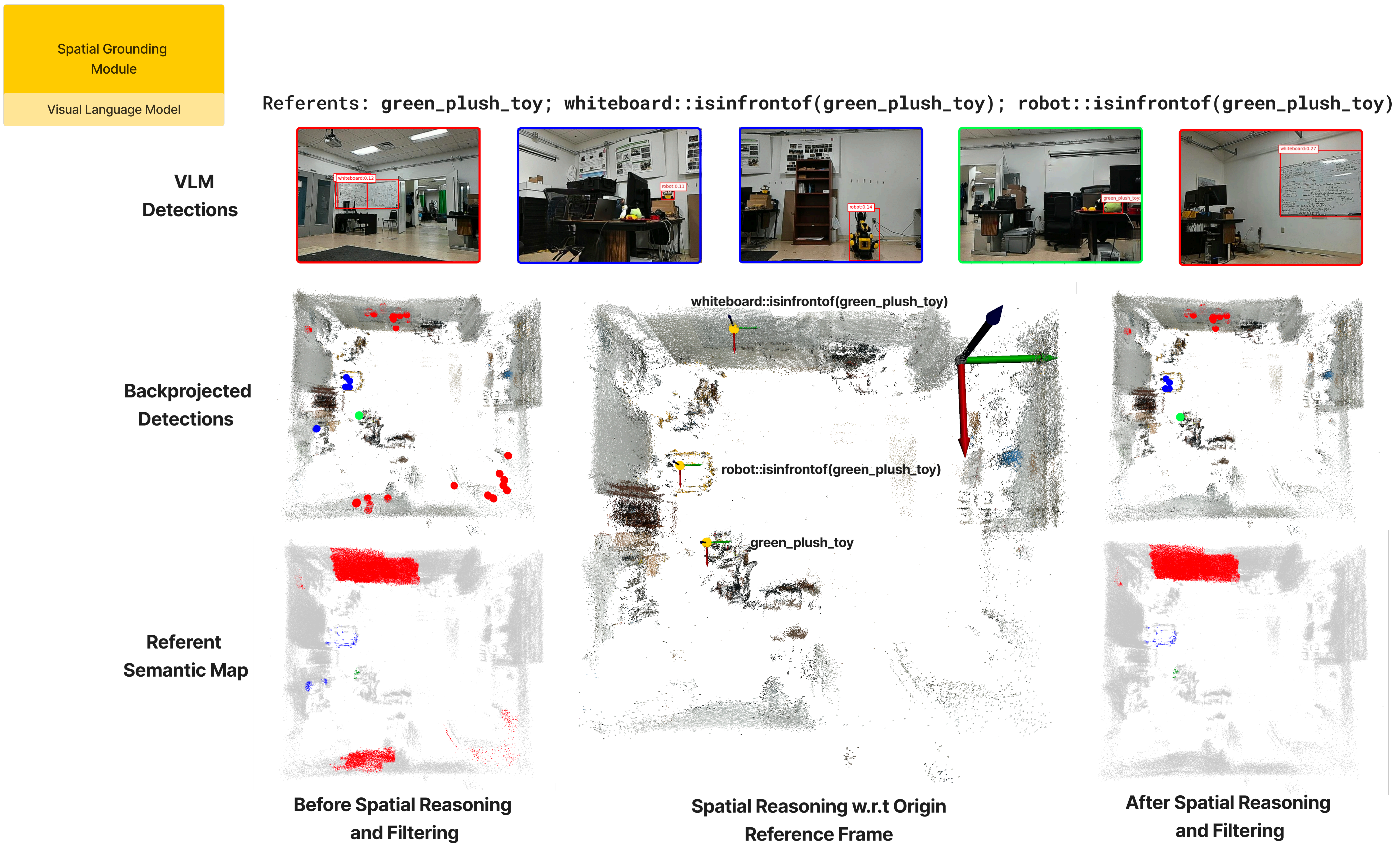
```
A: near[green_plush_toy]
B: pick[green_plush_toy]
C: near[whiteboard::isinfrontof(green_plush_toy)]
D: near[robot::isinfrontof(green_plush_toy)]
E: release[green_plush_toy, whiteboard::infrontof(green_plush_toy)]
```

Spatial Grounding Module

Visual Language Model

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

VLM Detections


Spatial Grounding Module
Visual Language Model
Referents: green_plush_toy; whiteboard::isinfrontof(green_plush_toy); robot::isinfrontof(green_plush_toy)
VLM Detections
Backprojected Detections
Referent Semantic Map
whiteboard::isinfrontof(green_plush_toy)
robot::isinfrontof(green_plush_toy)
green_plush_toy
Before Spatial Reasoning and Filtering
Spatial Reasoning w.r.t Origin Reference Frame
After Spatial Reasoning and Filtering

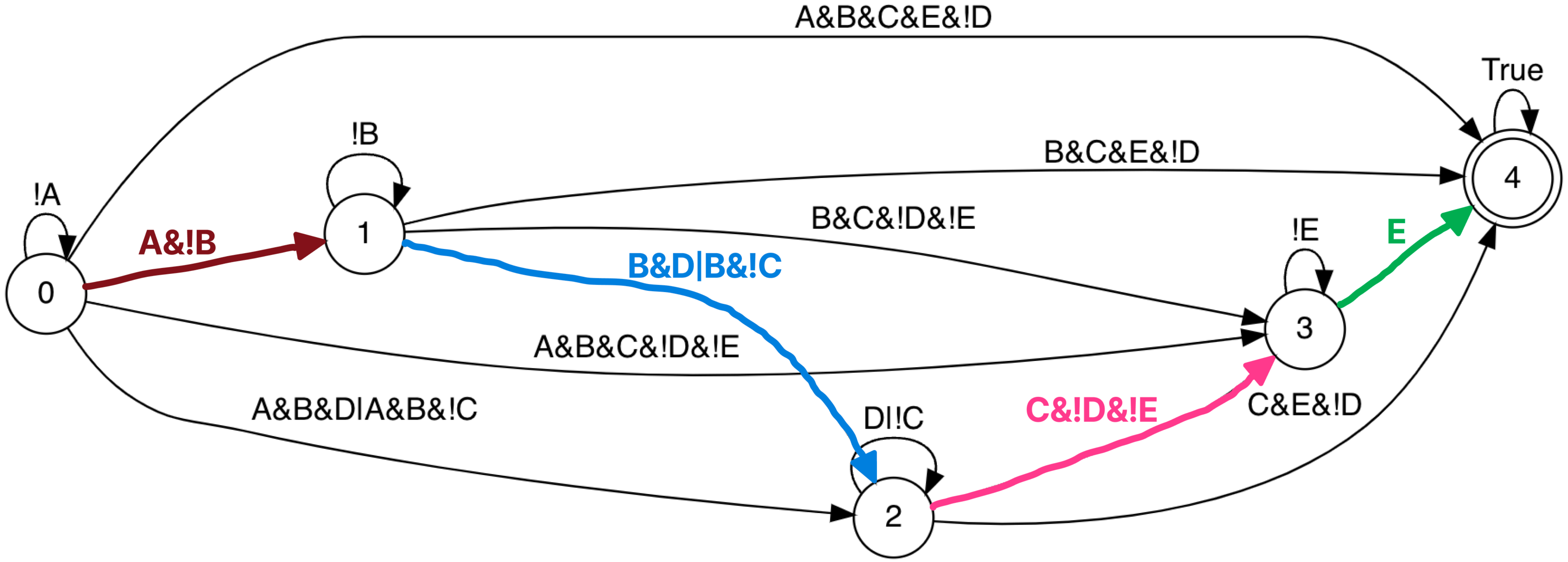
Task & Motion Planning Module

Progressive Motion Planner

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

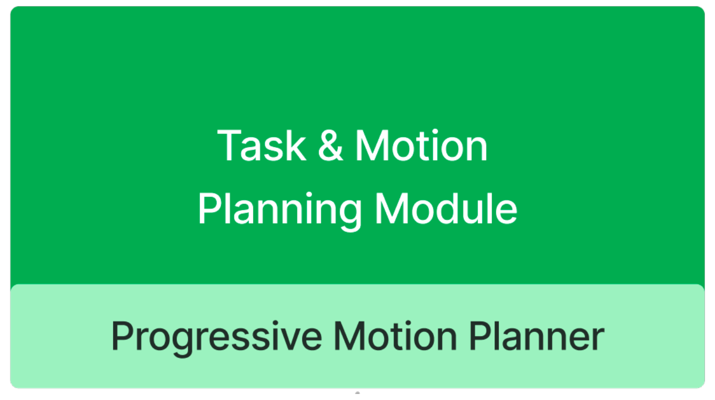
Our LTL Syntax : φ_l

$$\mathcal{F}(A \wedge \mathcal{F}(B \wedge \mathcal{F}(C \wedge \neg D \wedge \mathcal{F}E)))$$



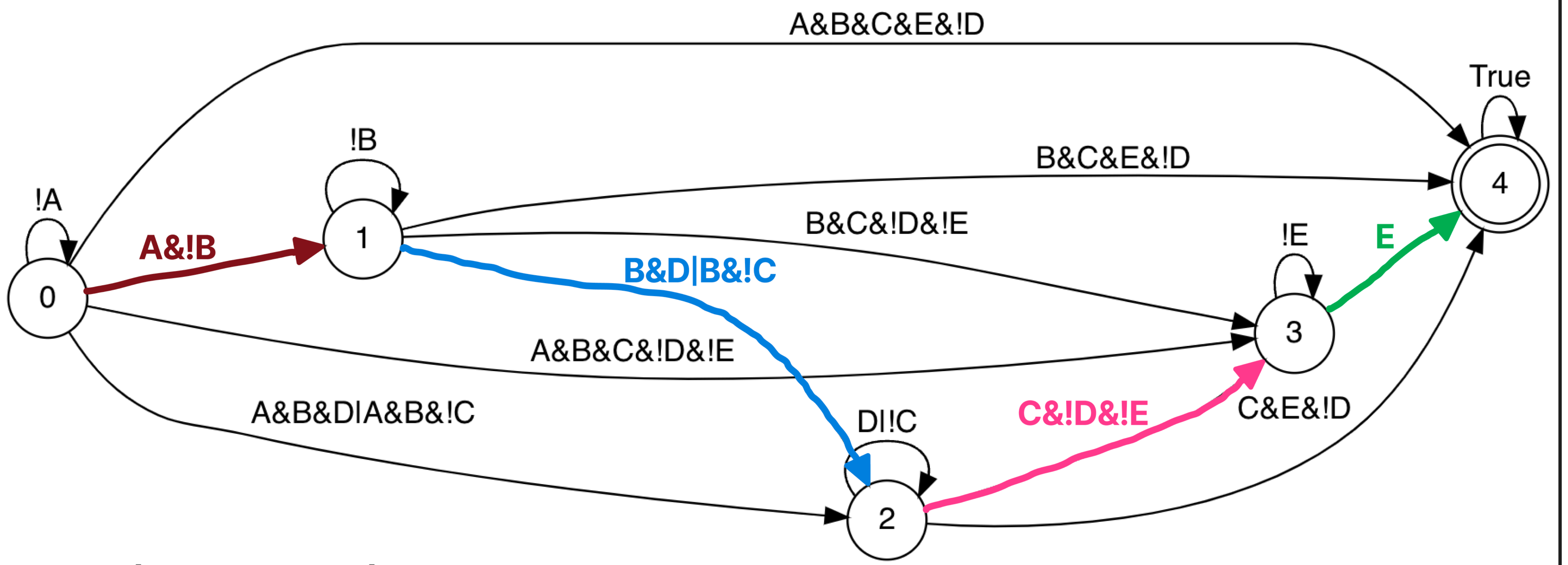

```
A: near[green_plush_toy]
B: pick[green_plush_toy]
C: near[whiteboard::isinfrontof(green_plush_toy)]
D: near[robot::isinfrontof(green_plush_toy)]
E: release[green_plush_toy, whiteboard::infrontof(green_plush_toy)]
```

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



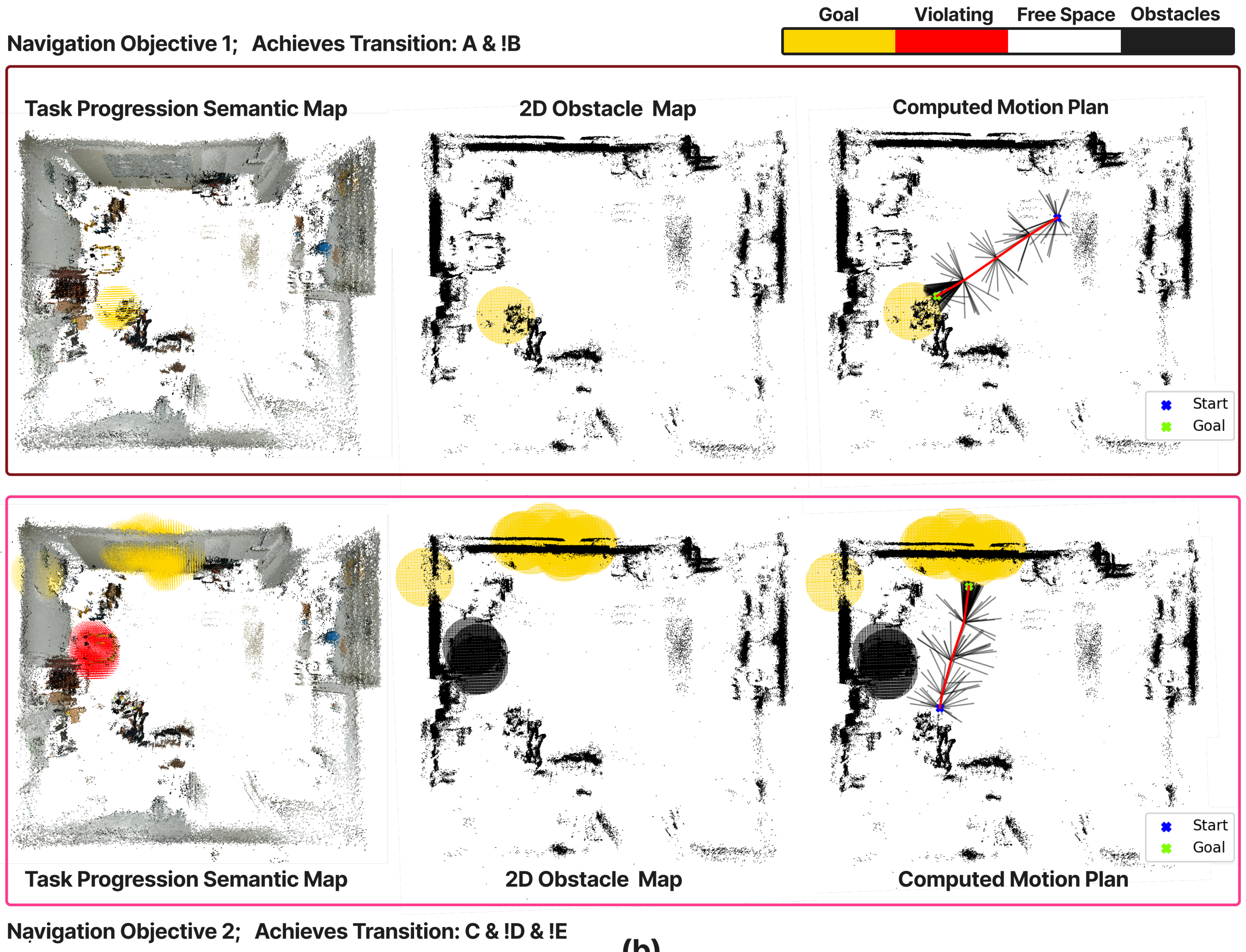
Input Instruction : " Bring the green plush toy to the whiteboard in front of it, watch out for the robot in front of the toy"

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



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



(b)

Demonstration


“Bring the toy cat between the coffee machine and the water filter to the black bag in front of the red sofa.
I don’t want you to go near the blue sofa or the fridge next to the water filter when going for the cat.”
See Demo at https://robotlimp.github.io/
CHEEZ-IT
Boston Dynamics

Free form instruction

“ Bring the toy cat between the coffee machine and the water filter to the black bag in front of the red sofa. I dont want you to go near the blue sofa or the fridge next to the water filter when going for the cat”

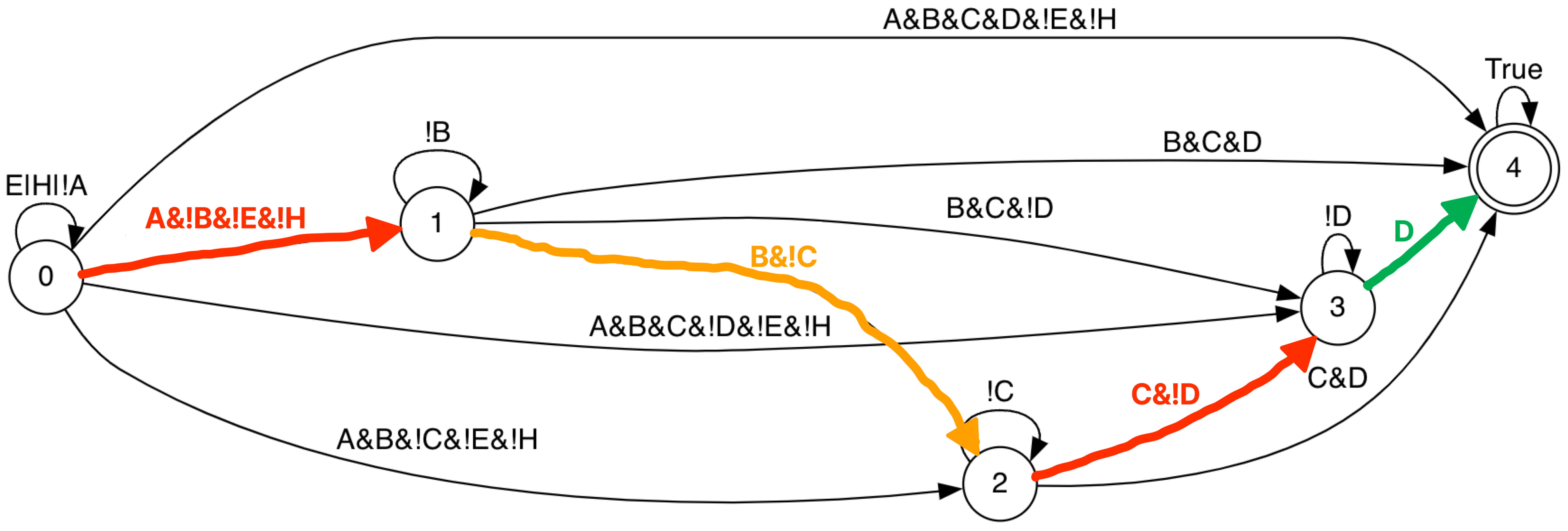
Translated LTL Formula

$$\mathcal{F}(A \wedge \neg E \wedge \neg H \wedge \mathcal{F}(B \wedge \mathcal{F}(C \wedge \mathcal{F} D)))$$

Resolved Referents

```
A: near[toy_cat::isbetween(coffee_machine,water_filter)]
B: pick[toy_cat::isbetween(coffee_machine,water_filter)]
C: near[black_bag::isinfrontof(red_sofa)]
D: release[toy_cat,black_bag::isinfrontof(red_sofa)]
E: near[blue_sofa]
H: near[fridge::isnextto(water_filter)]
```

Task Automaton



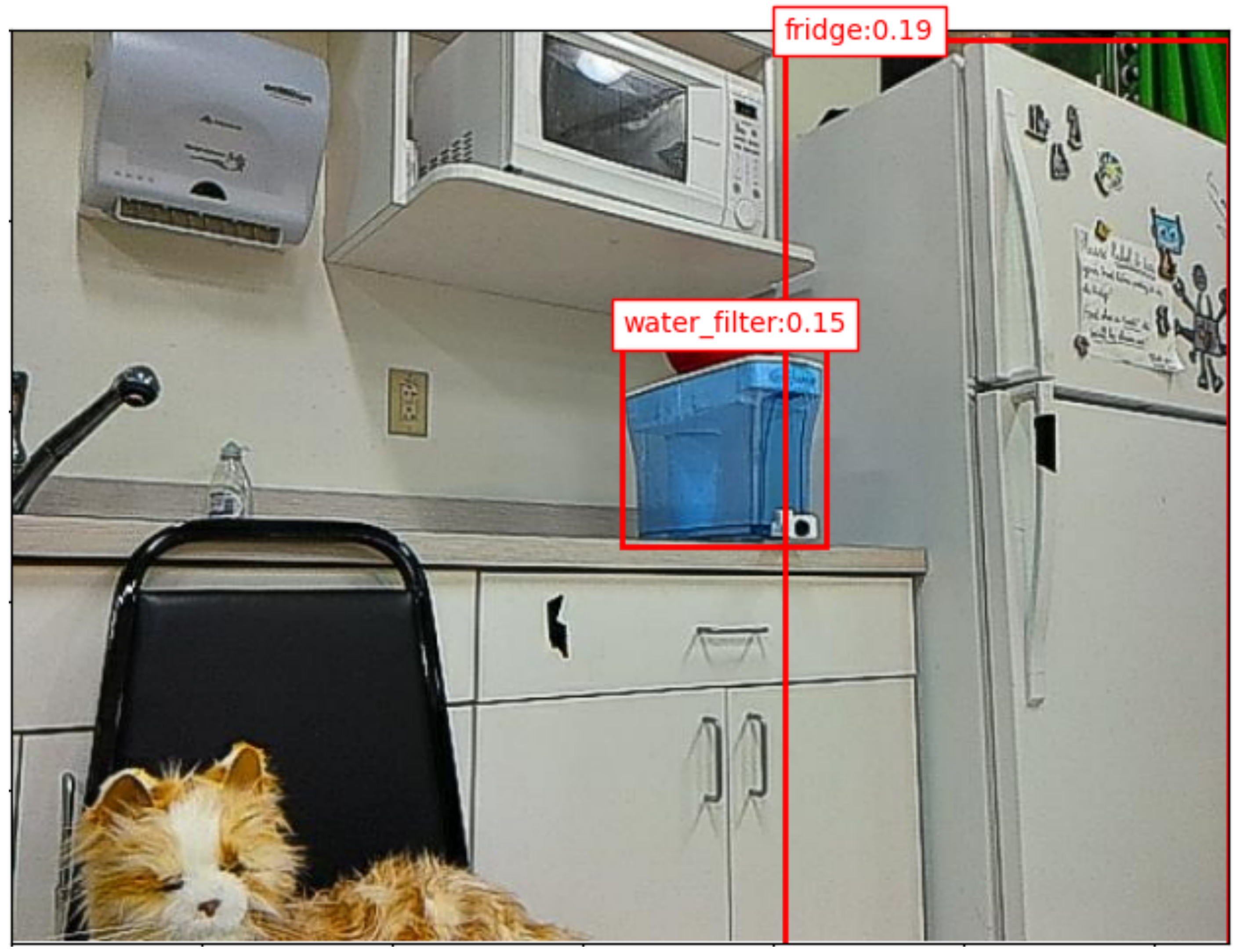
Resolved Referents

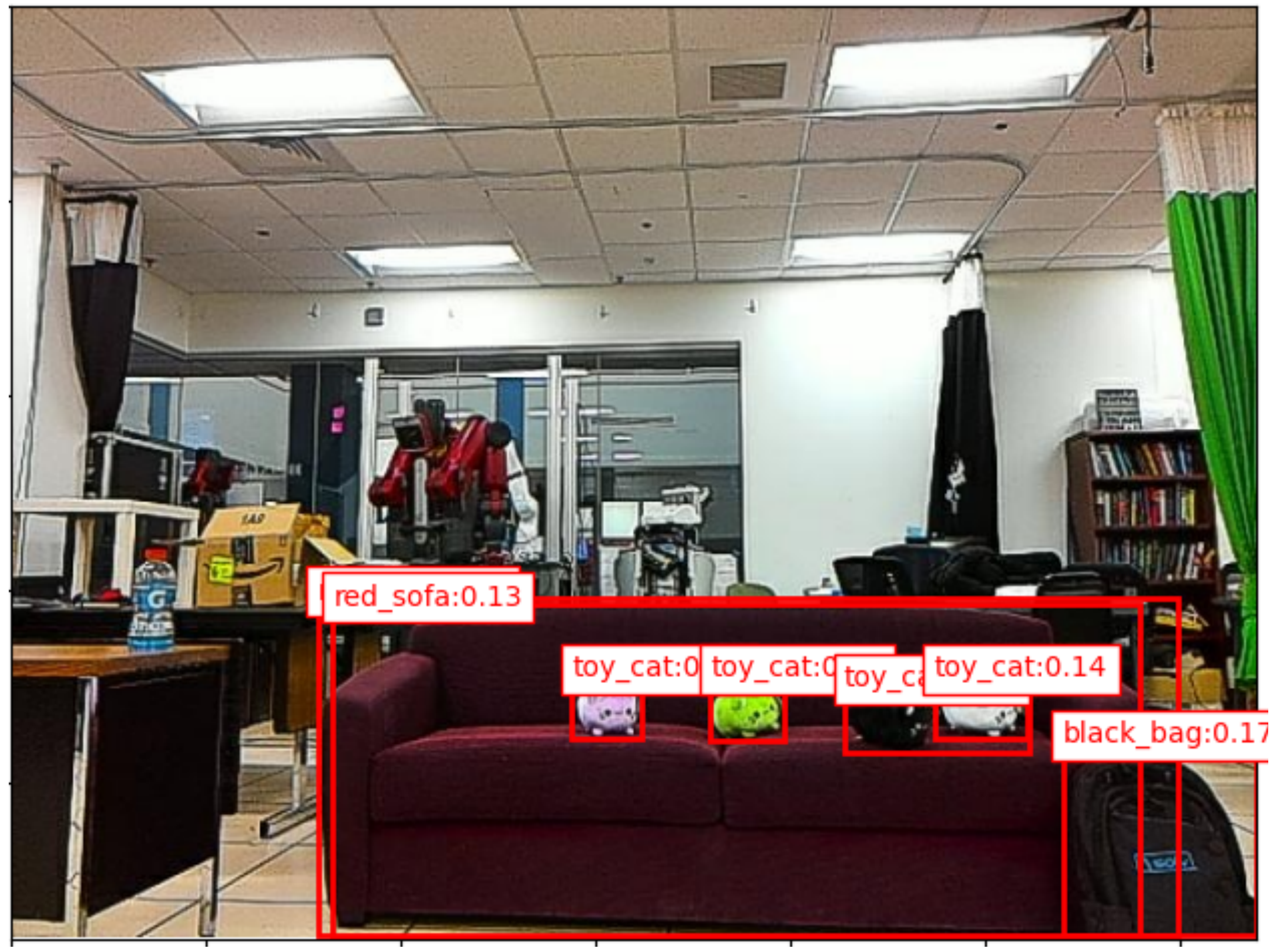
```
A: near[toy_cat::isbetween(coffee_machine,water_filter)]
B: pick[toy_cat::isbetween(coffee_machine,water_filter)]
C: near[black_bag::isinfrontof(red_sofa)]
D: release[toy_cat,black_bag::isinfrontof(red_sofa)]
E: near[blue_sofa]
H: near[fridge::isnextto(water_filter)]
```

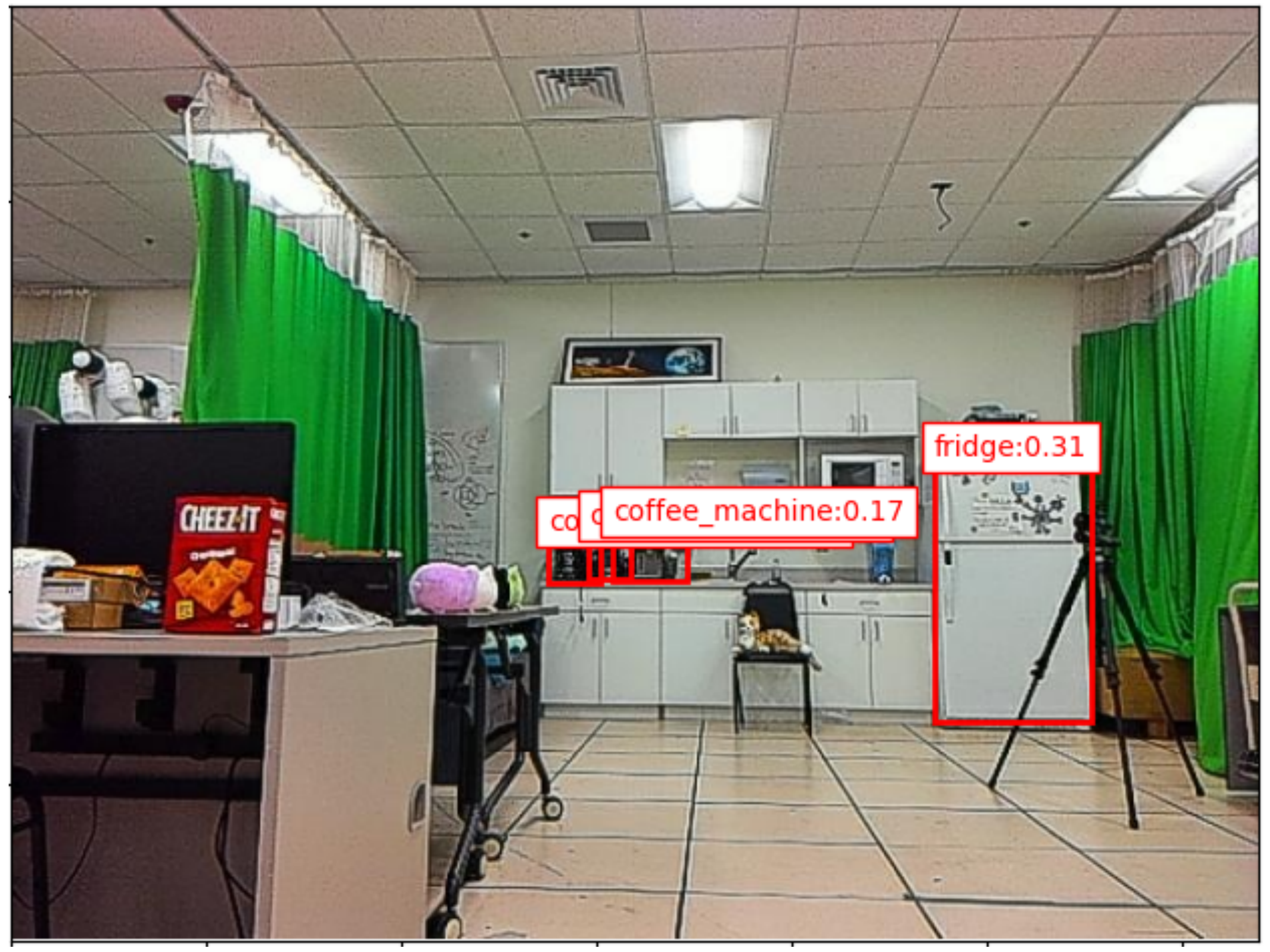
Chosen Automaton Path Key

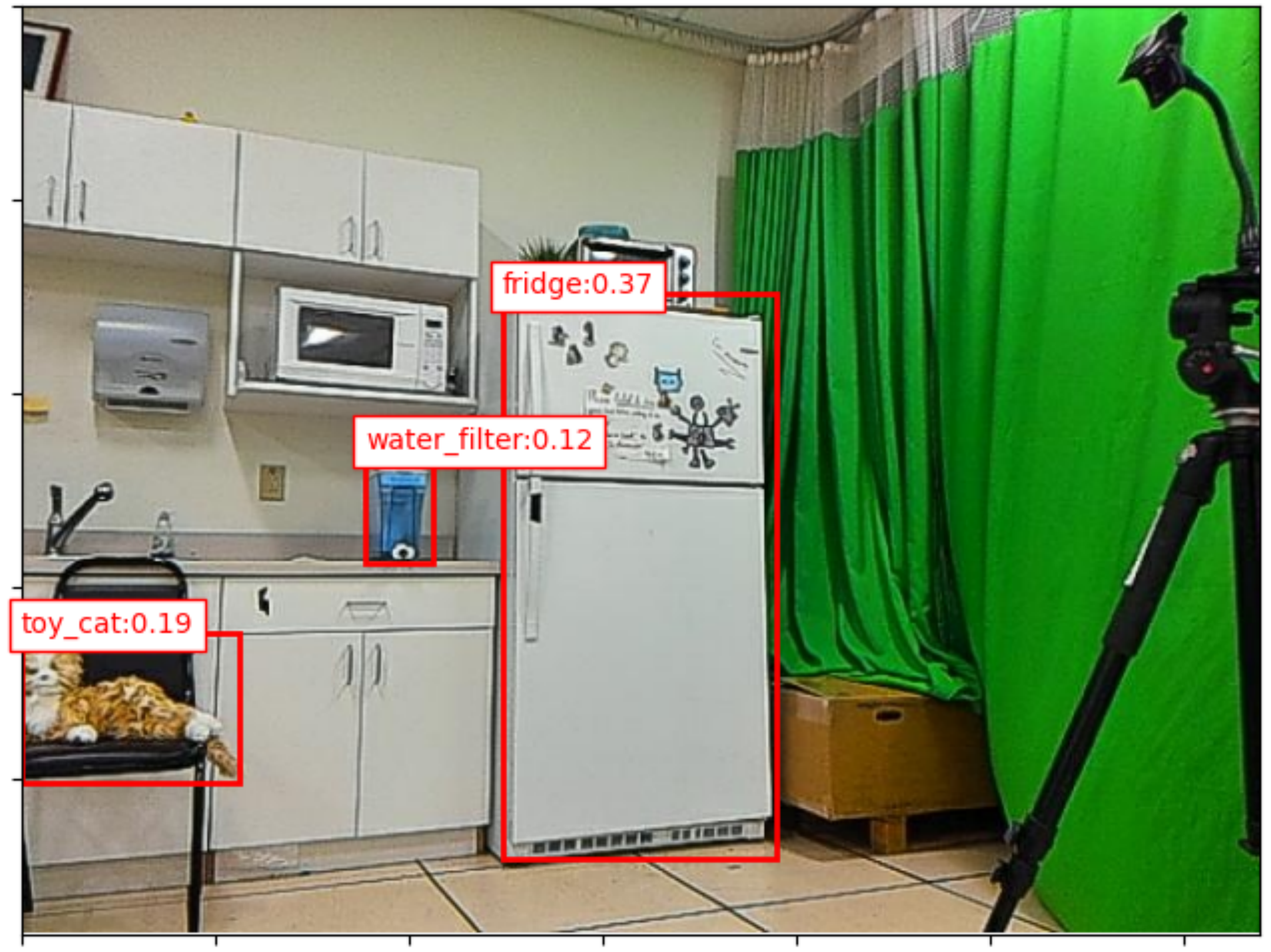
Red: Navigation Skill Objective
Orange: Pick Skill Objective
Green: Release Skill Objective

Sample VLM Detections







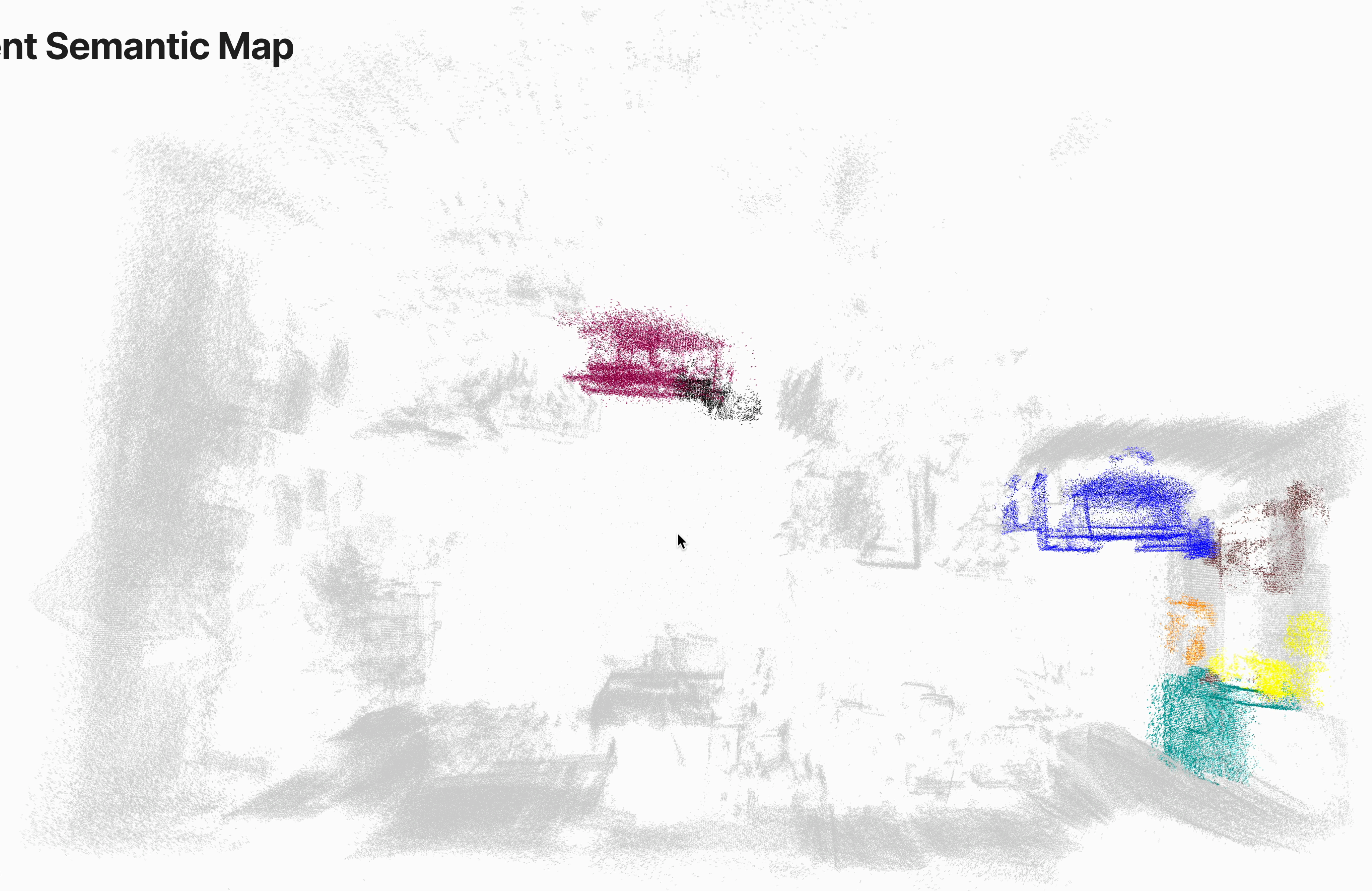


Grounded Detections after Spatial Reasoning

Key

Red: red sofa || **Blue**: blue sofa || **Orange**: coffee machine || **Brown**: water filter
Black: black bag in front of red sofa || **Violet**: toy cat between coffee machine and water filter
Cyan: fridge next to water filter

Referent Semantic Map



Task Progression Semantic Map

(First Navigation Objective)

Task relevant regions of interest

Red: avoidance region

Green: allowable region

Gold: goal region

Achieves Transition

(A&!B&!E&!H)

Computed Motion Plan

(First Navigation Objective)

Achieves Transition

(A&!B&!E&!H)

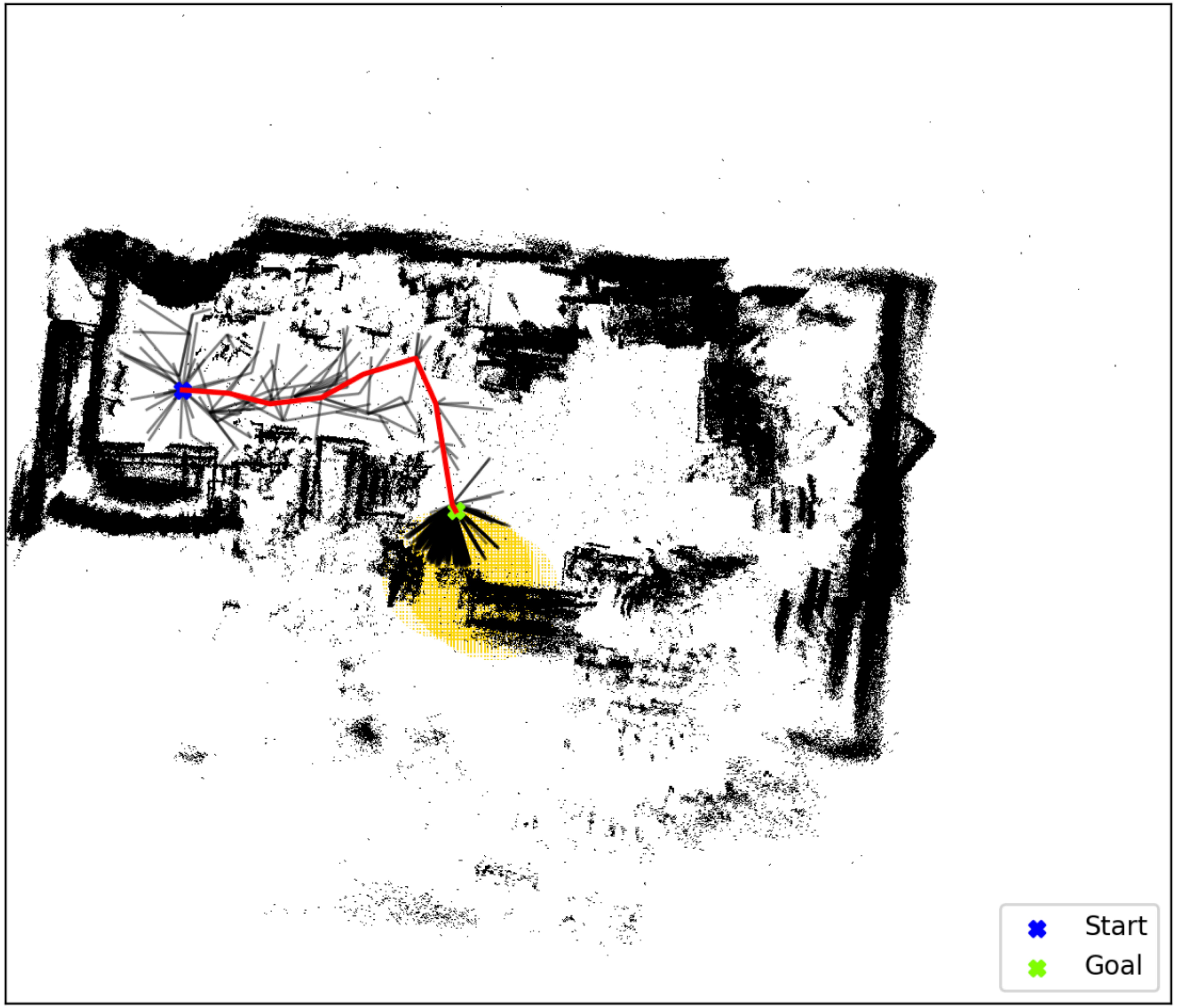
Task Progression Semantic Map
(Second Navigation Objective)

Task relevant regions of interest
Red: avoidance region
Green: allowable region
Gold: goal region

Achieves Transition
(C&!D)

Computed Motion Plan

(Second Navigation Objective)



Achieves Transition

(C&!D)

Concluding Remarks

Question

How do we get robots to verifiably follow complex open-ended instructions?

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

* General purpose instruction following
* Explainable instruction representation
* Verifiably correct behavior synthesis

The End!

Benedict Quartey*
benedict_quartey@brown.edu
benedictquartey.com

With
Eric Rosen*, Stefanie Tellex and George Konidaris

Kindly check out our Preprint and website

https://arxiv.org/abs/2402.11498
https://robotlimp.github.io

Feb 23 2024

* Equal Contribution